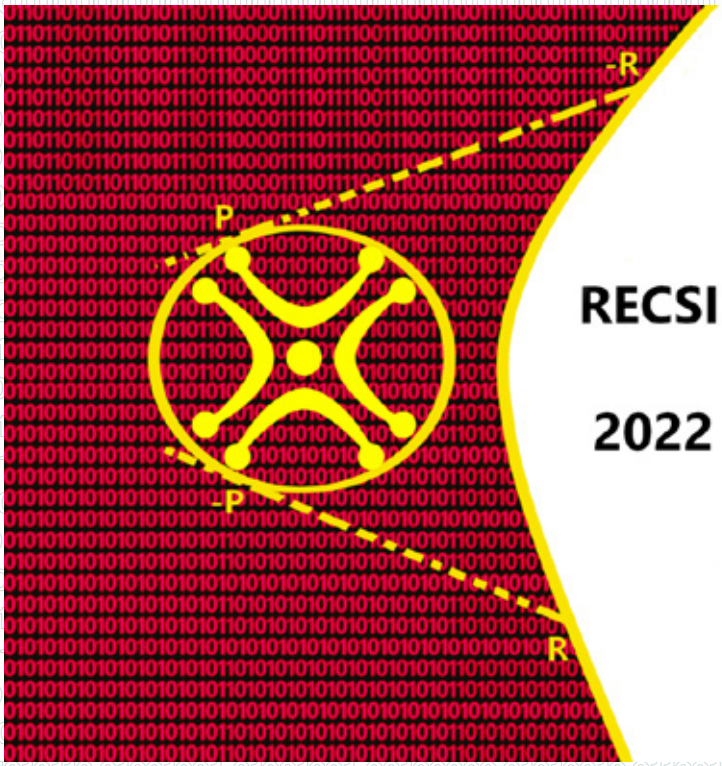
# XVII REUNIÓN ESPAÑOLA SOBRE CRIPTOLOGÍA Y SEGURIDAD DE LA INFORMACIÓN




*Santander, 19-21·octubre·2022*

**Daniel Sadornil Renedo**
**(editor)**

Ediciones Universidad Cantabria

# Organización XVII RECSI

### Comité Científico

- Florina Almenares, UC3M
- Maria Bras, URV
- Pino Caballero, ULL
- Jordi Castellà, URV
- Joan Josep Climent, UA
- Vanesa Daza, UPF
- Antonio Fco. Díaz, UGR
- Josep Domingo, URV
- Raül Durán, UAH
- Juan Manuel Estévez, UC3M
- José Luis Ferrer, UIB
- Amparo Fuster, CSIC
- Pedro García, UGR
- Víctor Gayoso, CSIC
- María Isabel González, URJC
- Jaime Gutiérrez, UC
- Luis Hernández, CSIC
- Candelaria Hernández, ULL
- Javier Herranz, UPC
- Jordi Herrera, UAB
- Llorenç Huguet, UIB
- Eduardo Jacob, UPV/EHU
- Vicente Jara, UPM
- Francisco Javier Lobillo, UGR
- Javier López, NICS-Lab
- Ángel Martín, USAL
- Agustín Martín, CSIC
- Consuelo Martínez, UniOvi
- David Megías, UOC
- Josep M. Miret, UdL
- M. Paz Morillo, UPC
- Alfonso Muñoz, Criptored-UPM
- Carles Padro, UPC
- M. Magdalena Payeras, UIB
- Alberto Peinado, UMA
- Jorge Ramió, UPM-CriptoRed
- Daniel Sadornil, UC
- Germán Sáez, UPC
- José Luis Salazar, UniZar
- Carmen Sánchez, UPM
- Luis Enrique Sánchez, UCLM
- Francesc Sebé, UdL
- Miguel Soriano, UPC
- Juan Tena, UVa
- Victor Villagrá, UPM
- Antonio Zamora, UA
- Urko Zurutuza, MU

### Comité Organizador

- Ana Isabel Gómez, URJC
- Domingo Gómez, UC
- Jaime Gutiérrez, UC
- Daniel Sadornil, UC
- David Sevilla, UEX

# XVII REUNIÓN ESPAÑOLA SOBRE CRIPTOLOGÍA Y SEGURIDAD DE LA INFORMACIÓN

## RECSI 2022

Colección DIFUNDE # 265

# XVII REUNIÓN ESPAÑOLA SOBRE CRIPTOLOGÍA Y SEGURIDAD DE LA INFORMACIÓN

# RECSI 2022

**Daniel Sadornil Renedo**
**(editor)**

Ediciones
Universidad
Cantabria

Reunión Española sobre Criptología y Seguridad de la Información (17ª : 2022 : Santander), autor.

XVII Reunión Española sobre Criptología y Seguridad de la Información : RECSI 2022 / Daniel Sadornil Renedo (editor). – Santander : Editorial de la Universidad de Cantabria, 2022.

239 páginas : ilustraciones. – (Difunde ; 265)

Comunicaciones presentadas en la XVII Reunión Española sobre Criptología y Seguridad de la Información (RECSI), organizada en Santander del 19 al 21 de octubre de 2022, por el grupo de investigación Algorithmic Mathematics and Cryptography de la Universidad de Cantabria.

ISBN 978-84-19024-14-5

1. Criptografía-Congresos. 2. Protección de la información (Informática)-Congresos. 3. Informática-Seguridad-Congresos. I. Sadornil Renedo, Daniel, editor de compilación. II. RECSI 2022.

004.056(063)

THEMA: GPJ, PDM, URY, UBH, UNKD



Digitalización: Manuel Ángel Ortiz Velasco [emeaov]



Avda. de los Castros, 52 - 39005 Santander. Cantabria (España)
www.editorial.unican.es
ISNI: 0000 0005 0686 0180

ISBN: 978-84-19024-14-5 (PDF)

DOI: https://doi.org/10.22429/Euc2022.028

Hecho en España-*Made in Spain*

Santander, 2022

## Prefacio

Este libro recoge las comunicaciones presentadas en la XVII Reunión Española de Criptología y Seguridad de la Información (RECSI), organizada en Santander, del 19 al 21 de octubre de 2022, por el grupo de investigación Algorithmic Mathematics and Cryptography de la Universidad de Cantabria.

La RECSI es el congreso científico referente español en el tema de la Seguridad en las Tecnologías de la Información y Comunicación (TIC), donde se dan cita de forma bienal los principales científicos españoles en el tema, así como invitados extranjeros de reconocido prestigio. En estos encuentros se muestran los avances de los grupos de investigación que presentan comunicaciones y fomentan la participación de los jóvenes investigadores. Las anteriores ediciones tuvieron lugar en Palma de Mallorca (1991), Madrid (1992), Barcelona (1994), Valladolid (1996), Torremolinos (1998), Santa Cruz de Tenerife (2000), Oviedo (2002), Leganés (2004), Barcelona (2006), Salamanca (2008), Tarragona (2010), San Sebastián (2012), Alicante(2014), Maó (2016), Granada (2018) y Lleida (2020).

El programa de esta edición RECSI 2022 consta de tres charlas plenarias impartidas por Gerhard Frey (Universität Duisburg-Essen), Pino Caballero-Gil (Universidad de la Laguna) y Dario Fiore (IMDEA Software Institute); 42 comunicaciones y la sesión ”Seguridad de la Información en instituciones y empresas”, donde se pretende mostrar una panorámica actual del impacto y el desarrollo de los temas del congreso en la sociedad. Este programa da testimonio de la buena salud de la criptología y seguridad de la información en España y el gran interés en esta comunidad científica por contribuir y difundir sistemas que permitan alcanzar un desarrollo de las TIC con las protecciones pertinentes, para las múltiples y conocidas facetas de nuestra vida que quedan afectadas por esas tecnologías.

Queremos agradecer a todas las instituciones que se han involucrado y nos han ayudado: Universidad de Cantabria, E.T.S. Ing. Industriales y de Telecomunicación, Centro Criptológico Nacional y Ayuntamiento de Santander. Y, por supuesto, agradecemos también a los conferenciantes invitados y a los autores participantes la elaboración y la presentación de sus interesantes aportaciones. Igualmente, expresamos nuestra gratitud al comité científico por la revisión de los trabajos, aspecto clave para garantizar la calidad científica del evento.

Santander, septiembre 2022.

Jaime Gutierrez

# Contenidos

# Diseño e implementación de un esquema criptobiométrico post-cuántico de protección de patrones. Aplicación en reconocimiento biométrico mediante mano

Diego José Abengózar Vilar
Grupo de Biometría, Bioseñales,
Seguridad y Smart Mobility
Dpto. de Matemática Aplicada a las TICs - ETSIT
Universidad Politécnica de Madrid
Avda. Complutense 30, 28040 Madrid
diegojose.abengozar.vilar@alumnos.upm.es

Carmen Sánchez Ávila
Grupo de Biometría, Bioseñales,
Seguridad y Smart Mobility
Dpto. de Matemática Aplicada a las TICs - ETSIT
Universidad Politécnica de Madrid
Avda. Complutense 30, 28040 Madrid
carmen.sanchez.avila@upm.es

***Abstract*—La evolución de la Biometría en las últimas décadas ha permitido su uso en sistemas de autenticación para mejorar el paradigma tradicional de identificación basado en *algo que sabes* y *algo que tienes*, añadiendo ahora el *algo que eres*. Esta nueva visión permite mejorar la usabilidad y seguridad de los sistemas de autenticación.**

**No obstante, los sistemas de identificación biométrica también presentan algunos inconvenientes relacionados la privacidad de los usuarios. En particular, en los últimos años, la posible irrupción de la computación cúantica ha hecho que surja la necesidad de desarrollar sistemas post-cuánticos.**

**En el presente artículo proponemos un esquema de protección de patrón biométrico basado en el criptosistema post-cuántico de McEliece.**

***Index Terms*—Biometrics, Cryptography, McEliece, Post-quantum Cryptography, Information Security, Biometric Template Protection**

## I. Introduction

En los últimos años se ha producido una gran expansión de la tecnología en todos los sectores. El mundo digital ha permeado muchos ámbitos de la vida, permitiendo una mejora en el bienestar humano.

Uno de los aspectos que facilitan el desarrollo de este mundo digital es la **Biometría**. La Biometría es la caracterización estadística de señales biológicas. En los últimos años, ésta ha tenido gran influencia en los sistemas de identificación, los cuales han sufrido grandes cambios.

Tradicionalmente, los **sistemas de autenticación** estaban basados en establecer la identidad mediante *algo que sabes* y *algo que tienes*. Sin embargo, este modelo tiene varios problemas, por ejemplo, el uso de contraseñas (*algo que sabes*) obliga al usuario a recordar claves complicadas. Además, se pueden robar o perder con relativa facilidad. De la misma forma, el uso de *tokens* de identificación, como puede ser un teléfono móvil o un USB específico (*algo que tienes*), también es vulnerable a robos.

Actualmente, la introducción de la Biometría permite incorporar un tercer elemento a los sistemas de autenticación: *algo que eres*. Este nuevo paradigma permite una mejor usabilidad para el usuario, que no tiene que recordar nada y no puede perder nada. Asímismo, la seguridad se ve incrementada ya que las características biométricas identifican de manera más fiel a un usuario.

Hasta ahora, la seguridad tanto de los sistemas biométricos como de la mayoría de sistemas digitales y criptográficos está medida en función de la capacidad computacional de los ordenadores actuales. Sin embargo, el auge de la **computación cuántica** pondría en peligro buena parte de estos sistemas. Es por ello que, criptosistemas post-cuánticos (resistentes a ataques de ordenadores cuánticos) a los que hasta ahora no se les había prestado mucha atención, cobran gran importancia.

El **criptosistema de McEliece**, basado en el uso de códigos lineales (concretamente utiliza la familia de códigos Goppa), fue presentado en 1978 por Robert J. McEliece [1]. Es resistente a ataques cuánticos [2] y, de hecho, una de sus variantes es candidato en la fase 3 del concurso de estandarización de criptosistemas post-cuánticos del NIST [3].

Dentro de este contexto, y observando que hay un cierto vacío en este área de estudio, vemos que es de gran interés desarrollar un esquema de protección de patrones biométricos basado en el criptosistema de McEliece, para estar preparados frente a la irrupción de la computación cuántica.

## II. Criptosistema de McEliece con códigos Goppa

### A. Códigos Goppa

El criptosistema de McEliece hace uso de los códigos Goppa [4], introducidos por Valery Denisovich Goppa en 1970. Se trata de una familia de códigos lineales correctores de errores. Son la principal elección a la hora de trabajar con el criptosistema de McEliece debido a que existe un algoritmo rápido de decodificación en tiempo polinómico y que son "fáciles de generar pero difíciles de encontrar": dado un polinomio sobre un cuerpo finito se puede crear fácilmente un código Goppa, pero la matriz generadora de un código Goppa es prácticamente aleatoria, por lo que no se puede realizar la operación contraria en un tiempo razonable.

Sea el polinomio

$$g(z) = g_0 + g_1 z + g_2 z^2 + \ldots + g_t z^t \in \mathbb{F}_{q^m}[z] \qquad (1)$$

y sea el conjunto

$$L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq \mathbb{F}_{q^m} : g(\alpha_i) \neq 0 \ \forall \alpha_i \in L \quad (2)$$

El código definido por

$$\left\{ c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n : \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} \equiv 0 \mod g(z) \right\}$$

es un **código de Goppa**, y se denota por $\Gamma(L, g(z))$.

### *B. Criptosistema de McEliece*

El primer criptosistema de clave pública basado en códigos fue presentado por Robert J. McEliece en 1978 [1]. Se fundamenta en codificar un mensaje y añadirle una serie de errores aleatorios para obtener un mensaje cifrado. Posteriormente, el código corrector se encarga de subsanar dichos errores y realizar la decodificación para recuperar el texto en claro original. Aunque parece relativamente antiguo, y en un primer momento no tuvo mucha difusión debido a que era menos eficiente que otros criptosistemas como el RSA o aquellos basados en el problema del logaritmo discreto, es de gran interés ahora ya que es un criptosistema resistente a ataques de ordenadores cuánticos [2], [5].

La version original del criptosistema de McEliece basada en códigos Goppa binarios se describe a continuación:

Bob quiere enviar un mensaje a Alice:

1) Alice genera sus claves pública y privada: (i)
   a) Alice elige un [n,k]-código Goppa binario capaz de corregir $t$ errores, y una matriz generadora $G$ de dimensión $k \times n$.
   b) Elige una matriz invertible $S$ de tamaño $k \times k$, y una matriz de permutación (una matriz con exactamente un 1 en cada fila y columa y el resto 0) $P$ de tamaño $n \times n$.
   c) Calcula la matriz $\hat{G} = S \cdot G \cdot P$.
   d) La clave pública de Alice es la pareja $(\hat{G}, t)$ y la clave privada es $(S, G, P)$
2) Bob le envía un mensaje cifrado a Alice: (i)
   a) Bob tiene un mensaje $\vec{m}$ en claro, de longitud $k$.
   b) Genera un vector aleatorio $\vec{e}$ de longitud $n$ con peso Hamming $t$. Este vector sirve para introducir errores tras codificar el mensaje.
   c) Codifica el mensaje y añade los errores: $\vec{c} = \vec{m} \cdot \hat{G} + \vec{e}$. Envía $\vec{c}$ a Alice.
3) Alice recibe $\vec{c}$ y lo descifra. (i)
   a) Calcula $P^{-1}$ con su clave privada.
   b) Calcula $\vec{c} \cdot P^{-1} = \vec{m} \cdot S \cdot G \cdot P \cdot P^{-1} + \vec{e} \cdot P^{-1} = \vec{m} \cdot S \cdot G + \vec{e} \cdot P^{-1}$. Dado que $P$ es una matriz de permutación, el vector de errores sigue teniendo peso $t$.
   c) Alice utiliza un algoritmo de decodificación de códigos Goppa para eliminar los errores y así obtener $\vec{m}S$. Con su clave privada calcula $S^{-1}$ y recupera el mensaje original $\vec{m}$.

## III. Esquema de protección de patrón

### *A. Introducción a la biometría*

Un sistema biométrico es un sistema automático de reconocimiento de patrones que utiliza características biométricas, es decir, características físicas o de comportamiento para reconocer a individuos. Existen distintas técnicas en función del rasgo que se quiera capturar.

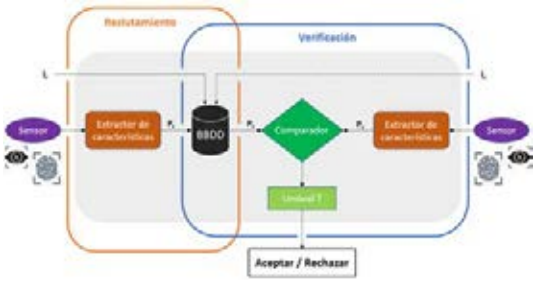



Figure 1. Esquema de un sistema de reconocimiento biométrico

Para realizar la identificación biométrica se lleva a cabo primero un **proceso de inscripción o reclutamiento** de los datos biométricos, en donde la representación digital de las características biométricas (**patrones**) son almacenadas en el sistema. Una vez que los usuarios están registrados, y un usuario desea autenticarse en el sistema, se lleva a cabo el **proceso de extracción** de sus características propias y su posterior **comparación con el patrón almacenado**. Este último se puede realizar en dos modalidades:

- **Identificación**: consiste en determinar a qué identidad corresponde el patrón introducido por un usuario. Para ello hay que realizar una comparación con todos los patrones biométricos recogidos $(1 - N)$.
- **Verificación**: consisten en determinar si una persona es quien dice ser. Sólo hay que comparar sus características con el patrón biométrico correspondiente $(1 - 1)$.

Además, hay que tener en cuenta que, dada la variabilidad de las señales biométricas, el patrón extraído y el patrón almacenado en la base de datos suelen no coincidir exactamente. Por ello, en la modalidad de verificación y en ciertos modelos de identificación, hay que determinar un **umbral** a partir del cual se considera afirmativa o negativa la comparación. Esta elección tendrá influencia directa en la cantidad de falsos negativos y falsos positivos que se produzcan.

En la figura 1 vemos el proceso de un sistema biométrico. Primero, un sensor captura el rasgo biométrico. Los datos capturados pasan al extractor de características que, en la fase de reclutamiento, genera un patrón digital $(P_r)$ y lo almacena en la base de datos, junto con alguna información adicional $(I_r)$ del usuario (ej. nombre o identificador). En la fase de verificación se extraen las características biométricas del usuario (patrón de consulta, $P_c$) y se comparan con el patrón de referencia correspondiente a la información adicional introducida $(I_c)$. En caso de utilizar el modo de identificación no se introduciría información adicional.

Centrándonos en la arquitectura de verificación vamos a definir ciertas tasas de error, que sirven para poder realizar una evaluación de los errores de un esquema de patrón biométrico. Debido a los problemas de variabilidad de las características biométricas no puede haber exactitud en las comparaciones de patrones. Si el umbral elegido es demasiado exigente habrá patrones legítimos que no serán dados por válidos y si el umbral es muy laxo un impostor tendrá más fácil "colarse" en el sistema. Para cuantificar la exactitud del módulo de decisión

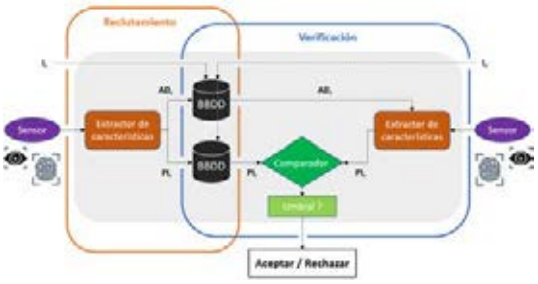



Figure 2. Esquema seguro de un sistema de reconocimiento biométrico

de los sistemas biométricos se definen las siguientes tasas de error:

- **Tasa de falsos rechazos** (FRR, *False Reject Rate*): proporción de usuarios que siendo legítimos son rechazados por el sistema. Cuanto mayor es el umbral, más aumenta la tasa de falsos rechazos.
- **Tasa de falsas aceptaciones** (FAR, *False Accept Rate*): proporción de usuarios que siendo impostores son aceptados por el sistema. Cuanto menor es el umbral, más aumenta la tasa de falsas aceptaciones.
- **Tasa de igual error** (EER, *Equal Error Rate*): momento en el que la tasa FAR es igual a la tasa FRR.
- **Cero FAR**: menor valor de la tasa FRR para el cual FAR = 0.
- **Cero FRR**: menor valor de la tasa FAR para el cual FRR = 0.

### *B. Esquemas de protección de patrón biométrico*

Existen varias preocupaciones en cuanto a la privacidad de las características biométricas almacenadas, tales como la posibilidad de obtener la información biométrica original a partir del patrón almacenado, la capacidad de relacionar los patrones de distintas bases de datos para rastrear las actividades de un usuario, o el problema que supone el robo de un patrón biométrico.

El estándar ISO 24745:2022 [6] establece ciertos requisitos que deben cumplir los esquemas de protección de patrón para mitigar riesgos de seguridad y garantizar la privacidad de los datos personales:

- **Irreversibilidad**: el patrón biométrico almacenado no debe revelar información que permita reconstruir la señal biométrica original.
- **Desvinculabilidad**: dados dos o más patrones extraídos de la misma característica biométrica no debe poder determinarse si pertenecen al mismo individuo.
- **Revocabilidad**: un patrón biométrico que se haya visto comprometido debe poder ser reemplazado por uno nuevo (y distinto) generado a partir de la misma característica biométrica del usuario afectado.

Además de estos requisitos, podemos considerar parámetros como el rendimiento y la precisión a la hora de evaluar los distintos esquemas de protección de patrones biométricos.

Debido a los problemas de privacidad y seguridad expuestos, dicho estándar también propone un nueva arquitectura de referencia para la protección de patrones biométricos:

Vemos que en el caso de la figura 2, el extractor de características se encarga de descomponer el patrón de referencia en un **identificador pseudónimo** ($PI$), que equivale a un patrón biométrico protegido, y **datos auxiliares** ($AD$), almacenados en bases de datos diferentes. Durante la fase de verificación, se toma el patrón de consulta $P_c$, y los datos auxiliares $AD$ correspondientes al identificador $I_c$ introducido para calcular el identificador pseudónimo de consulta ($PI_c$) y compararlo con el de referencia de la base de datos.

### *C. Clasificación*

El diseño de métodos de protección de patrón que cumplan todos los requisitos anteriores es un reto. Los distintos esquemas presentan ventajas e inconvenientes. Podemos clasificarlos en tres grandes tipos.

- **Biometría cancelable**: consiste en distorsionar las señales biométricas a través de diversas transformaciones (repetibles) y realizar la comparación con el patrón en el dominio protegido. Hay dos tipos principales:
  - Transformaciones irreversibles.
  - *Salting*, en donde se utilizan datos auxiliares para mezclarlos con la señal biométrica y así obtener la versión distorsionada.
- **Sistemas criptobiométricos**: combinan claves criptográficas con transformaciones de los datos biométricos originales para obtener patrones seguros. Según el tipo de datos auxiliares que utilicen se pueden clasificar en dos tipos:
  - Esquemas de vinculación de clave: los datos auxiliares se obtienen combinando la clave criptográfica con el patrón biométrico. En la fase de verificación, se recupera la clave a través de los datos auxiliares y el patrón.
  - Esquemas de generación de clave: tanto los datos auxiliares como la clave se generan a partir del patrón biométrico.
- **Biometría en el dominio cifrado**: consiste en cifrar los datos biométricos mediante cifrado homomórfico y realizar la comparación con el patrón en el dominio cifrado. El cifrado homomórfico permite realizar operaciones sobre los textos cifrados y obtener los mismos resultados que si se estuviera trabajando con los textos en claro. Dado que la implementación de sistemas con cifrado homomórfico no es sencilla, se han propuesto también esquemas que utilizan cifrados semi-homomórficos, es decir, que sólo permiten realizar ciertas operaciones en el dominio cifrado.

El esquema de protección de patrón biométrico que se desarrolla en el presente trabajo se encuadra dentro de la categoría de biometría en el dominio cifrado.

## IV. Esquema propuesto

A continuación se propone un esquema de protección de patrón apoyado en el criptosistema de McEliece. En la fase de inscripción, los patrones (binarios) obtenidos se cifran con el criptosistema de McEliece, utilizando una clave de un tamaño determinado en función del nivel de seguridad que se quiera obtener, y se almacenan en una base de datos. En la fase de identificación o verificación el patrón de consulta introducido por el usuario se cifra con la misma clave pública

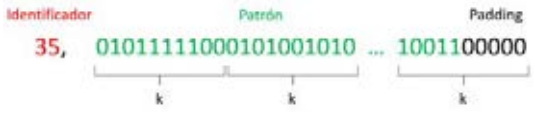



Figure 3. Formato del patrón en la fase de inscripción

Figure 4. Bases de datos generadas en la fase de inscripción

y se compara con los patrones almacenados en la base de datos.

Debido a que el criptosistema de McEliece no es completamente homomórfico [7], vamos a realizar una modificación en la fase de cifrado: realizaremos las operaciones en $\mathbb{Z}$ en lugar de en $\mathbb{Z}_2$. Al trabajar con números binarios, una pequeña perturbación en el patrón de consulta produce grandes diferencias en el dominio cifrado, por lo que sería difícil llevar a cabo las comparaciones. Sin embargo, trabajando en $\mathbb{Z}$ podemos discernir en mayor medida la similitud de distintos patrones en el dominio cifrado.

### A. *Fase de inscripción*

1) Se genera un par de claves, pública y privada, del criptosistema de McEliece.
2) Se separa cada patrón en bloques de la longitud correspondiente a las claves elegidas. En caso de que el último bloque no tenga la longitud suficiente, se añade un padding.
3) Cada bloque se cifra con la clave pública (matriz $SGP$) y se almacena en la base de datos del sistema biométrico. Recordemos que en la fase de cifrado se añade un vector de errores aleatorio, por lo que, además de la clave pública, se almacena también cada vector de errores utilizado.
4) En una base de datos auxiliar se almacena el identificador de usuario correspondiente a cada patrón.

### B. *Fase de verificación*

Se recibe un patrón de consulta junto con un identificador de usuario.

1) En la base de datos auxiliar se busca el identificador de usuario para determinar con qué patrón de referencia hay que comparar.
2) Se separa el patrón de consulta en bloques de la longitud correspondiente.
3) Se recupera la clave pública y el vector de errores correspondiente para cada bloque, y se cifra el patrón bloque a bloque.
4) Se realiza la comparación en el dominio cifrado de ambos patrones.
5) Si la similitud de ambos patrones supera un cierto umbral, se acepta como válida la verificación.

### C. *Fase de identificación*

Se recibe un patrón de consulta.

1) Se separa el patrón de consulta en bloques de la longitud correspondiente.
2) Se recupera la clave pública. Para cada patrón de referencia, se recuperan los vectores de errores correspondientes, y se cifra el patrón de consulta bloque a bloque.
3) Para cada patrón de referencia, se realiza la comparación en el dominio cifrado con el patrón de consulta.
4) Si en alguna comparación la similitud de ambos patrones supera un cierto umbral, se acepta como válida la identificación.

## V. Análisis de resultados

### A. *Base de datos gb2s_ID*

Para nuestro primer escenario práctico utilizaremos la base de datos de patrones de mano *gb2s_ID* [8].

Cada imagen fue capturada mediante el dispositivo móvil HTC Desire S, con una resolución de $2592\times1552$ píxeles. El proceso se llevó a cabo con luz natural y sin grandes restricciones en cuanto a la distancia, posición, condiciones lumínicas o objetos que llevara el usuario (anillo, reloj, etc.). La base de datos cuenta con 96 usuarios, y 10 patrones por cada usuario. Hay patrones de hombres y mujeres en similar propoción, así como patrones de diverso origen étnico [9].

Por todo ello, *gb2s_ID* presenta una gran variabilidad entre las distintas muestras que recoge.

Las imágenes tomadas se han procesado con el filtro de Gabor de extracción de características principales [10], y para nuestro caso práctico trabajaremos con todas las direcciones de dicho filtro.

*1) Evaluación de errores:* La metodología seguida en este trabajo consiste en utilizar uno de los diez patrones de cada usuario para la fase de inscripción, y posteriomente realizar la comparación con todos los patrones, de manera que determinamos las tasas de falsas aceptaciones y falsos rechazos. Se utilizará la distancia Euclídea en el módulo comparador. Compararemos las tasas de error del esquema desarrollado con las que obtendríamos si se compararan los patrones directamente, sin cifrar, para ver la eficacia del esquema que se propone en este trabajo.

Los rangos posibles del umbral se han determinado de forma experimental, y se han normalizado al intervalo $[0, 1]$, siendo 0 el valor del umbral más laxo posible y 1 el más estricto, para facilitar la comparativa.

En la figura 5 vemos las Tasas de Falsa Aceptación (FAR) y Falso Rechazo (FRR) superpuestas, con el umbral en el eje de abcisas y el porcentaje en el eje de ordenadas.

La intersección de ambas es la Tasa de Igual Error (ERR), que en este caso toma un valor de 36,1% (con un umbral de 0,79). Vemos también el valor de Cero FAR (valor de FRR para el que la tasa FAR es 0%), que se produce con un umbral de 0,9 y el valor Cero FRR, con un umbral de 0,34. Estas tasas indican el compromiso que hay entre FAR y FRR. Por ejemplo, si movemos el umbral hacia valores más altos aumenta el FRR y disminuye el FAR. Si quisieramos tener 0% de falsos rechazos tendríamos que aceptar un 100% de falsas aceptaciones. Si, por el contrario, quisieramos un 0% de falsas aceptaciones, tendríamos que soportar un 83% de falsos rechazos.

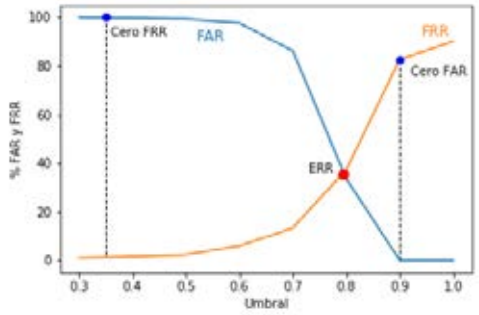



Figure 5. Tasa de igual error en función del umbral (*gb2s_ID*)

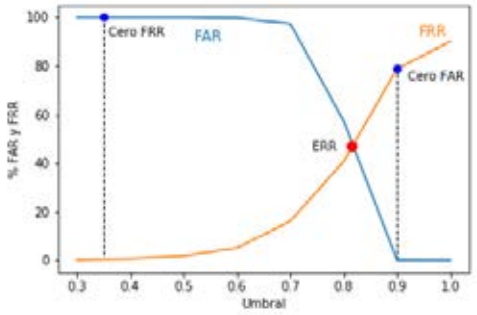



Figure 7. Tasa de igual error en función del umbral (Hong Kong)

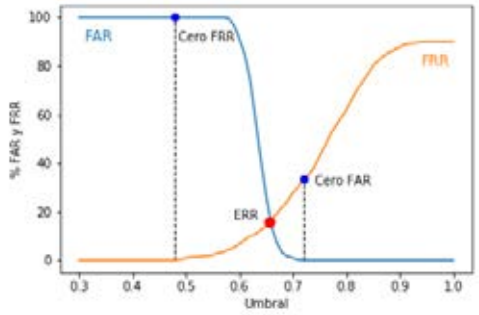



Figure 6. Tasa de igual error en función del umbral (*gb2s_ID*, dominio sin cifrar)

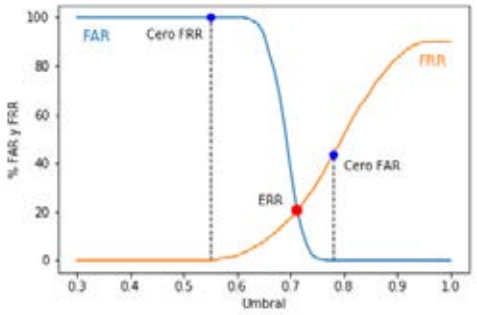



Figure 8. Tasa de igual error en función del umbral (Hong Kong, dominio sin cifrar)

Como referencia, vemos en la figura 6 las tasas FAR y FRR realizando las comparaciones en el dominio no cifrado, donde no hay seguridad alguna en la protección del patrón.

Vemos que la Tasa de Igual Error toma un valor del 16%, con un umbral de 0,66. En este caso, el valor de Cero FAR es 37% FRR, y se produce con un umbral de 0,72. El valor Cero FRR es 100% FAR, con un umbral de 0,48.

### *B. Base de datos Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database*

En este segundo escenario utilizaremos la base de datos de patrones de mano tomados por la Universidad Politécnica de Hong Kong [11]. Para nuestro caso, utilizaremos las imágenes 2D de dicha base de datos.

Las imágenes, de resolución 640×480, han sido tomadas con el dispositivo 3D Minolta VIVID 910. Contiene imágenes de 177 voluntarios (10 patrones por cada uno) con edades entre 18 y 50 años, de distintos orígenes étnicos. Las capturas fueron realizadas en interiores, sin restricciones en cuanto a la iluminación, la posición de la mano ni objetos que llevara el usuario. Se utilizó un fondo negro para mejorar el contraste [9].

Nuevamente, las imágenes han sido procesadas con el filtro de Gabor de extracción de características principales.

*1) Evaluación de errores:* Vemos de nuevo las Tasas de Falsa Aceptación (FAR) y Falso Rechazo (FRR) superpuestas, con el umbral en el eje de abcisas y el porcentaje en el eje de ordenadas (figura 7).

La intersección de ambas es la Tasa de Igual Error (ERR). Se produce con un umbral de 0,81 y que toma un valor de 47,3%. Vemos que el valor de Cero FAR es 79%, y se produce con un umbral de 0,9 y el valor Cero FRR es 100%, con un umbral de 0,35.

En la figura 8 vemos las tasas FAR y FRR realizando las comparaciones en el dominio no cifrado.

La Tasa de Igual Error toma un valor de 20%, con un umbral de 0,71. En este caso, el valor de Cero FAR es 43,5% FRR, y se produce con un umbral de 0,78. El valor Cero FRR es 100% FAR, con un umbral de 0,55.

### *C. Evaluación de seguridad*

Las posibles brechas de seguridad del esquema propuesto consisten en atacar el criptosistema subyacente para obtener los patrones en claro. El ataque más conocido y eficaz contra el criptosistema de McEliece es el basado en *Information Set Decoding* [1], [12]. En resumen, este consiste en "adivinar" las posiciones del vector original a las que ha afectado el vector de errores aleatorio introducido en la fase de cifrado.

Por tanto, hay dos posibles vectores de ataque a nuestro esquema de protección de patrones.

Por un lado, los errores almacenados en la base de datos han de estar cifrados y transmitirse de manera segura, ya que en caso de ser descubiertos se podría recuperar el patrón original. Una solución puede consistir en utilizar un cifrado simétrico como AES128 o AES256 (resistente a la computación cuántica [13]) para cifrar los errores previamente a almacenarlos en la base de datos. La clave de cifrado empleada estaría guardada únicamente en el módulo donde se realiza el cifrado y transmisión de los vectores de errores.

Por otro lado, sin conocer las posiciones de los errores, al atacante no le queda más remedio que intentar descubrirlas a base de fuerza bruta, método que ya se ha demostrado

poco eficaz [12]. Además, cabe remarcar que la modificación introducida en el esquema propuesto en la fase de cifrado no facilitaría este proceso al atacante.

### D. Evaluación de rendimiento

Para ver la utilidad del esquema propuesto en entornos reales es necesario también tener en cuenta la usabilidad para los usuarios. Un factor clave para determinar la usabilidad es el tiempo de procesamiento de las características biométricas introducidas por el usuario. Experimentalmente se han observado los siguientes tiempos de procesado:

- Identificación: con la base de datos *gb2s_ID* se ha observado un tiempo medio de 90 segundos cuando el patrón de consulta es rechazado, y un tiempo de 1,4 segundos cuando el patrón sí es aceptado. Realizando las mismas pruebas con la base de datos de Hong Kong, se han obtenido unos tiempos de 162 y 1,8 segundos respectivamente.
- Verificación: con la base de datos *gb2s_ID* se obtiene un tiempo de 1,4 segundos (tanto si el patrón es aceptado como si no) mientras que con la base de datos de Hong Kong obtenemos un tiempo de 1,8 segundos.

Vemos que la modalidad de verificación presenta un tiempo muy reducido, por lo que podría ser utilizada en gran variedad de entornos reales.

Cabe remarcar que a la hora de realizar una implementación concreta de este esquema se utilizaría hardware específico que mejoraría las prestaciones, reduciendo así los tiempos.

## VI. Conclusiones y líneas futuras

En este trabajo se ha diseñado un esquema de protección de patrón biométrico que trata de resolver uno de los problemas que pueden surgir en un futuro cercano: la irrupción de la computación cuántica. Es un trabajo de gran interés y dificultad, además de novedoso, debido a que no hay nada desarollado en este área de estudio en la literatura presente. Para ello se ha estudiado e implementado el criptosistema desarrollado por Robert J. McEliece en 1978, no muy utilizado hasta ahora, pero que cobra interés al ser un criptosistema post-cuántico.

Se ha probado el esquema propuesto con dos bases de datos de patrones de mano: *gb2s_ID* y *Hong Kong Polytechnic University Contact-free 3D/2D Hand Images Database*, realizando una evaluación de la eficacia y el rendimiento.

Hemos calculado las tasas de fallos de falsas aceptaciones y falsos rechazos. Para poder juzgar los resultados obtenidos los hemos comparado con las tasas que se obtendrían trabajando en el dominio sin cifrar, que mostrarían el mejor rendimiento posible aunque lógicamente no presentan seguridad alguna a la hora de proteger los patrones biométricos. Las tasas de fallos obtenidas con el esquema propuesto son relativamente buenas comparándolas con éstas últimas. Además, podríamos exigir que las muestras biométricas se tomaran con un sensor de mayor calidad para mejorar las tasas de fallos.

En otro orden de ideas, la evaluación del rendimiento muestra que, al utilizar la modalidad de verificación, se podría implantar este sistema en una gran variedad de entornos debido a su tiempo muy reducido (del orden de 1 segundo). Al utilizar la modalidad de identificación es más costoso en cuanto a tiempo, por lo que tendría que utilizarse en entornos más particulares. No obstante, al realizar una implantación concreta de este esquema, se utilizaría hardware específico que mejoraría las prestaciones.

Queda claro, pues, que nuestro esquema de protección de patrón muestra la viabilidad del desarrollo de esquemas de protección de patrón post-cuántico. Se trata de una buena primera aproximación, que da pie a pensar que puede mejorarse en futuros análisis.

Varias líneas de investigación que pueden surgir a partir del presente trabajo son:

- Buscar un mecanismo distinto para solventar los problemas derivados de que el criptosistema de McEliece no es completamente homomórfico.
- Debido a la modificación introducida en la fase de cifrado, y al no estar ya restringidos al uso de filtros binarios de extracción de características, se podrían probar otros algoritmos de extracción de características distintos al de Gabor.
- Investigar otras posibles modificaciones del criptosistema para evitar tener que almacenar los vectores aleatorios de la fase de cifrado.
- Utilización de otras técnicas de comparación tales como *Support Vector Machines* [14].

## Agradecimientos

Este trabajo es parte del proyecto de I+D+i PID2019-107274RB-I00, financiado por MCIN/AEI/10.13039/5011000 11033/.

## References

[1] Robert J. McEliece: “A public-key cryptosystem based on algebraic coding theory” *DSN Progress Report*, vol. 42, pp. 114-116, 1978.

[2] H. Dinh, C. Moore, A. Russell: “McEliece and Niederreiter Cryptosystems That Resist Quantum Fourier Sampling Attacks” *Advances in Cryptology - CRYPTO 2011*, pp. 761-779, 2011.

[3] Martin R. Albrecht, Daniel J. Bernstein et al.: “Classic McEliece: conservative code-based cryptography” *NIST PQC Competition*, 2020.

[4] Valery D. Goppa: “A new class of linear error-correcting codes” *Probl. Peresach. Inform.*, vol. 6, n. 3, pp. 24-30, 1970.

[5] P. W. Shor: “Algorithms for quantum computation: discrete logarithms and factoring” *Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124-134, 1994.*

*[6] ISO Central Secretary: “Information security, cybersecurity and privacy protection - Biometric information protection”*, https://www.iso.org/standard/75302.html, 2022.

[7] C. Zhao, Y. Ya-Tao, L. Zi-Chen: “The Homomorphic Properties of McEliece Public-Key Cryptosystem” *Proceedings of the 2012 Fourth International Conference on Multimedia Information Networking and Security*, pp. 39-42, 2012.

[8] A. de Santos-Sierra, J. Guerra-Casanova, C. Sánchez Ávila, V. Jara-Vera: “Silhoutte-based Hand Recognition on Mobile Devices” *43rd Annual 2009 International Carnahan Conference on Security Technology*, pp. 160-166, 2009.

[9] Belén Ríos Sánchez: “Analysis of Image Processing, Machine Learning and Information Fusion techniques for Contact-less Hand Biometrics” *Universidad Politécnica de Madrid (UPM)*, 2017.

[10] J. Zhang, T. Tan, L. Ma: “Invariante Texture Segmentation Via Circula Gabor Filters” *Proceesings of 16th International Conference on Pattern Recognition*, vol. 2, pp. 901-904, 2002.

[11] V. Kanhangad, A. Kumar, D. Zhang: “Contactless and pose invariant biometric identification using hand surface” *IEEE Trans. Image Processing*, vol. 6, pp. 1014-1027, 2011.

[12] D. J. Bernstein, T. Lange, C. Peters: “Attacking and Defending the McEliece Cryptosystem” *PQCrypto ’08: Proceedings of the 2nd International Workshop on Post-Quantum Cryptography*, pp. 31-46, 2008.

[13] X. Bonnetain, M. Naya-Plasencia, A. Schrottenloher: “Quantum Security Analysis of AES” *IACR Transactions on Symmetric Cryptology*, pp. 55-93, 2019.

[14] C. Cortes, V. Vapnik: “Support-vector networks”, *Mach Learn*, vol. 20, pp. 273-297, 1995.

# Anonymity and unlinkability in ring signature-based discussion boards

Oriol Alàs
Dept. of Mathematics. U. de Lleida
C. Jaume II, 69,
E-25001 Lleida (Spain)
oriol.alas@udl.cat

Francesc Sebé
Dept. of Mathematics. U. de Lleida
C. Jaume II, 69,
E-25001 Lleida (Spain)
francesc.sebe@udl.cat

Sergi Simón
Dept. of Mathematics. U. de Lleida
C.Jaume II, 69,
E-25001 Lleida (Spain)
sergi.simon@udl.cat

***Abstract*—An on-line forum is a platform which enables its participants to post messages which become available to any person with access to it. We focus our research on forums in which messages are both anonymous and authenticated. That is, the author of a message cannot be identified, but it can be proven that they belong to a set of authorized authors. In this paper we analyze the feasibility of ring signatures as a cryptographic scheme able to provide both authentication and privacy to on-line forums. The attained privacy has been measured throughout simulations.**

***Index Terms*—cryptography, discussion board, privacy, ring signature**

## I. Introduction

With the advent of Web 2.0, Internet moved from static content to dynamic web pages allowing the contribution from navigators. This fact gave rise to the appearance of wikis, blogs and discussion boards. A *wiki* is a document centered website that provides collaborative modification of its content directly from the web browser. A *blog* is an author centered website in which an author regularly publishes diary-style entries while readers can add comments to author's posts. A *discussion board* is a topic centered website in which discussions are organized into topics and all the participants have the same voice.

In this paper we focus on an online platform in which members of a well defined group can post messages anonymously. A platform in which a town council publishes a preliminary text of ordinances so that citizens can comment on them, or a debate board in a university in which only lecturers and/or students can participate, would fit in with the scope of our research. All these examples correspond to the discussion board concept. From now on, we will refer to that platform as a *forum*.

Such a system requires some cryptographic tool providing *authentication* (only members of the group can post messages), *anonymity* (the identity of a message author is not revealed), and *unlinkability* (it is not possible to determine whether two anonymous comments were authored by the same person or not). *Group signatures* [1] provide the required security properties. Unfortunately, group signature schemes include a trusted group manager who is able to revoke the anonymity of a signed message when required. In this paper we analyze the feasibility of implementing such a system employing a similar tool, namely *ring signatures*. A ring signature scheme [2] does not require its users to trust any entity with the exception of certificate authorities issuing public key certificates.

## II. Preliminaries

This section briefly presents the basic cryptographic primitive our analysis is based on, namely, *ring signatures*. For the sake of completeness, *group signatures* are also introduced.

### A. Group signatures

A *group signature* scheme, as first formalized in [1], enables a member of a group to anonymously sign a message on behalf of the group so that: (i) only members of the group can sign, (ii) the receiver can verify a group signature, but cannot discover which group member made it, (iii) there exists a party who can "open" signatures, that is, it can determine the identity of the message signer. The *group manager* is a trusted entity which sets up the system and is in charge of managing group membership and revoking the anonymity of a signature when required. Some proposals distribute this manager into a *membership manager* and a *revocation manager*. Efficent group signature schemes exist [3] in which the length of the group public key and group signatures do not depend on the size of the group.

### B. Ring signatures

A *ring signature* scheme, as formalized in [2], is a concept similar to a group signature scheme, but it has no trusted managers, no setup and no revocation procedures. A ring signature is computed by means of a procedure that takes as input the message to be signed, the private key of the signer together with the public keys of a group of people chosen by the signer. This group of people is known as the *ring* and it always includes the signer. The resulting ring signature is validated by means of a procedure that takes as input the signed message, its ring signature, and the public key of each member of the ring. Validation of a ring signature provides no information about the member of the ring who actually computed the signature. The size of a ring signature must grow linearly with the size of the ring, since it must list the ring members. The proposal [2] is RSA-based while [4] is based on ElGamal cryptosystem. There exist ring signature schemes which are linkable [5], [6], *i.e.*, it is possible to determine whether the signatures on two messages were computed by the same entity or not.

In [7] a construction of a constant-size[1] ring signature scheme given an accumulator with one-way domain is proposed. An efficient implementation is presented employing an

[1]Constant-size can only be achieved when the ring members are implicitly known or can be described in constant size.

accumulator that requires secrecy about the factorization of a shared RSA modulus. In such a system, concerns about the way in which that modulus was created would arise. Since our objective is to avoid the need for any user to trust any party, we discard the applicability of the mentioned system.

To sum up, a ring signature scheme is defined by three procedures:

- $\mathrm{KeyCreation}(1^n)$, which returns a private-public key pair given a specified security level.
- $\mathrm{RingSign}(M, PubKey_1, \ldots, PubKey_r, s, PrivKey_s)$, which produces a ring signature $\sigma_M$ for message $M$, given the public keys $PubKey_1, \ldots, PubKey_r$ of the ring members and the private key of its $s$-th member, $PrivKey_s$.
- $\mathrm{RingVerify}(M, \sigma_M, PubKey_1, \ldots, PubKey_r)$, which verifies the signature $\sigma_M$ for message $M$ under public keys $PubKey_1, \ldots, PubKey_r$ and outputs a boolean indicating whether the signature is valid of not.

## III. System description

A forum with $M$ registered members in which messages are authenticated employing group signatures attains $M$-anonymity [8] concerning message authorship. In effect, given a posted message, any of the $M$ forum members could have authored it. An attacker aiming to determine the author of a message cannot do better than randomly guessing among the forum members. Regarding message linkability, there is no way to determine whether two messages were authored by the same forum member or not.

Unfortunately, group signatures require its users to trust the revocation manager. This entity is in possession of cryptographic keys enabling it to lift the anonymity of any group signature. If corrupted, this entity can learn the identity behind each published message without being detected.

Use of ring signatures with rings including all the forum members provides an equivalent privacy without the need to trust any party. Unfortunately, the size, generation, and validation costs of a ring signature grow linearly with the ring size. This would cause the system to be extremely inefficient specially if the forum has a large membership.

In this paper we analyze the feasibility of an anonymous and unlinkable forum employing ring signatures in which the security and unlinkability requirements can be relaxed. That is, message anonymity can be relaxed to be $K$-anonymous with $K < M$, and message linkability can be relaxed according to a parameter $\mathcal{K}$ in the sense that no more than $1/\mathcal{K}$-th of messages attributable to a given author were really authored by them. Summarizing, we eliminate the need to have a trusted party at the expense of relaxing the provided anonymity and unlinkability, and increasing the system cost.

### A. Forum joining procedure

When some person $\mathcal{P}_i$ wishes to join a forum, they have to request for it. Upon being accepted, they receive a text $Join_{\mathcal{P}_i}$ with the terms and conditions and an statement in which $\mathcal{P}_i$ indicates that they want to become a member of the forum at the current date. Then, $\mathcal{P}_i$ signs $Join_{\mathcal{P}_i}$, and sends the resulting signature $Sign_{\mathcal{P}_i}(Join_{\mathcal{P}_i})$ to the forum together with their public key certificate $Cert_{CA}(PubKey_{\mathcal{P}_i})$ issued by some trusted certificate authority (CA).

Finally, $Sign_{\mathcal{P}_i}(Join_{\mathcal{P}_i})$ and $Cert_{CA}(PubKey_{\mathcal{P}_i})$ are added to the community members list $\mathcal{L}$. This procedure guarantees the other forum members that $\mathcal{P}_i$ really wishes to sign up for the forum.

### B. Private message posting

Let $\mathcal{L}$ be the forum members list, and let $K$ denote the size of the ring of each signed message. When a community member $\mathcal{P}_i$ is to post a message, they proceed as follows:

1) Write message $M$.
2) Choose at random $K-1$ forum members from $\mathcal{L}$ and take their public keys.
3) Generate a list $Ring_M$ which contains the public key of $\mathcal{P}_i$ together with the public keys of the other $K-1$ ring members.
4) Compute a ring signature on $M$ under the public keys in $Ring_M$ and the private key of $\mathcal{P}_i$.
5) Post $M$ together with its ring signature and the identity of the chosen ring members.

### C. Message authentication

So as to check whether a message $M$ published on the forum has really been authored by an authorized member, its ring signature has to be verified. A successful validation guarantees that it has been authored by one of the members listed in $Ring_M$.

## IV. Privacy analysis

In this section, the privacy of the proposal is analyzed both in terms of anonymity and unlinkability.

### A. Anonymity

The first issue regarding privacy addresses the uncertainty about the authorship of a posted message. Let $K$ be the size of the rings.

*Lemma.* The authorship of a posted message is $K$-anonymous.

*Proof.* A posted message carries attached a ring signature whose ring size is $K$. Hence, any of the $K$ persons in the ring may be the actual author of the message.

### B. Unlinkability

In our forum, only the messages whose ring includes $\mathcal{P}_i$ could have been authored by $\mathcal{P}_i$. Let $N_i$ be the quantity of such candidate messages, while $n_i$ is the number of messages really authored by $\mathcal{P}_i$. Obviously, $N_i \geq n_i$. The ratio

$$K_i = \frac{N_i}{n_i}, \tag{1}$$

tells about the confusion an attacker gets about the messages really authored by $\mathcal{P}_i$. The larger the value for $K_i$, the higher the confusion is.

Assuming each community member posted one message at least, each value $K_i \geq 1$ is a random variable whose distribution is to be analyzed.

### C. Uniformly random choice of ring members

In this section, we analyze $K_i$ assuming the members of the signature ring (step 2 of the procedure given in Section III-B) are chosen uniformly at random.

Let $N_i = n_i + r_i$ with $n_i$ being the quantity of messages authored by $\mathcal{P}_i$, and $r_i$ being the times $\mathcal{P}_i$ is part of the ring of a message not authored by them. Let $\hat{N}$ be the number of messages in the forum not authored by $\mathcal{P}_i$ and let $\hat{M}$ be the quantity of forum members excluding $\mathcal{P}_i$.

It follows that $r_i$ is a random variable with a binomial distribution $r_i \approx \text{Bin}\left(\hat{N}; \frac{K-1}{\hat{M}}\right)$ with mean $E[r_i] = \hat{N}\frac{(K-1)}{\hat{M}}$.

For a given $n_i$, we have $K_i = \frac{N_i}{n_i} = \frac{n_i+r_i}{n_i}$, so that, the probability of getting a privacy level not above $\mathcal{K}$ is

$$p\left[K_i \leq \mathcal{K}\right] = p\left[\frac{n_i + r_i}{n_i} \leq \mathcal{K}\right] = p\left[r_i \leq (\mathcal{K}-1)n_i\right].$$

Our objective is to take an appropriate $K$ which allows to upperbound the probability of such privacy loss. The Chernoff bound states that for a binomial random variable $r_i$ with mean $\mu$ (in our case, $\mu = \hat{N}\frac{(K-1)}{\hat{M}}$ ), then

$$p\left[r_i \leq (1-\delta)\mu\right] \leq e^{-\frac{\mu\delta^2}{2}}.$$

In our scenario, we choose a security parameter $\epsilon$ and then compute the minimum value for $K$ so that

$$p\left[K_i \leq \mathcal{K}\right] \leq e^{-\epsilon}.$$

The value for $K$ is computed from the following system of equations

$$\begin{cases} (1-\delta)\mu = (\mathcal{K}-1)n_i, \\ \frac{\mu\delta^2}{2} = \epsilon. \end{cases}$$

which solves to $\mu = (\mathcal{K}-1)n_i + \epsilon + \sqrt{(2(\mathcal{K}-1)n_i + \epsilon)\epsilon}$. Since $K = \mu\frac{\hat{M}}{\hat{N}} + 1$ we get the value for $K$ to be,

$$K = \left\lceil 1 + \frac{(\mathcal{K}-1)n_i + \epsilon + \sqrt{(2(\mathcal{K}-1)n_i + \epsilon)\epsilon}}{\frac{\hat{N}}{\hat{M}}} \right\rceil.$$

We conclude that, $K = O\left(\frac{\mathcal{K}n_i+\epsilon}{\frac{\hat{N}}{\hat{M}}}\right)$.

As it can be seen, the value $K$ has to be chosen taking into account the relation between the number of messages posted by a forum member ($n_i$) and the average number of messages sent by the rest ($\hat{N}/\hat{M}$). So as for all the members to be protected, the value $K$ has to be taken considering the most active member. If the expectations at setup time do not hold, then the most active members are likely to get a privacy level below $\mathcal{K}$. Also, the parameter $K$ computed by considering the most active member causes an overprotection for less active ones.

### D. Preferential attachment choice of ring members

So as for all forum members to get a minimum privacy level, the ratio $K_i = \frac{n_i+r_i}{n_i}$ should not fall below a stablished minimum value. This requires the number of times a participant $\mathcal{P}_i$ belongs to the ring of a message not authored by them, $r_i$, to grow with the quantity of messages really authored by them, namely $n_i$.

This is not a trivial matter because the authorship of messages is anonymous, so the real value $n_i$ of each participant is secret.

So as to address this issue, we have tested a preferential attachment strategy. More precisely, the probability each participant has of being included in a ring grows with the number of times they appear in the ring of already posted messages. The rationale behind this strategy is that highly active participants, those with a large $n_i$, tend to appear in more rings because they are always, at least, members of the rings of messages really authored by them. In this way, they will have a greater chance of being selected to be part of rings of messages not authored by them so that their $r_i$ value will increase.

Our first simulations showed that this strategy caused a snowball effect in which the participants selected to be members of a ring during the first iterations had a tendency to monopolize the inclusion in rings in the future in such a way that many other participants stood almost no chance of being selected.

So as to avoid this undesired effect, the probability of being chosen has been set according to a method in which each participant is assigned an initial number of points, $w_i$. In addition to these initial points, a participant receives an additional number of points, namely $w_m$, for each ring it has been member of. Then, when a new ring is to be formed, the probability of being selected is proportional to the quantity of points each member is in possession of. This system enabled us to balance between a uniform choice among forum participants, with $w_m = 0$, and a purely preferential attachment strategy, with $w_i = 0$.

### E. Experimental results

The performances of both uniform and preferential attachment strategies have been tested by means of a simulator implemented in Python. The simulated forum is composed of 400 participants with a post frequency distributed following a Zipf's law (with parameter $s = 1.3$) with each participant posting between one and fifteen messages. Zipf's distribution has been reported as a suitable distribution for representing the so-called 90–9–1 rule [9] which states that the majority of content in an Internet community is produced by only 1% of the participants ('superusers'), a minority of content is produced by a further 9% of participants ('contributors') while 90% of people just consume content posted by others ('lurkers'). In our simulations, an overall number of 1427 messages are posted.

Each simulation has focused on four specific participants contributing 1, 5, 10 and 15 messages, respectively. At the end of the simulation, we have measured the privacy attained by each of them as the ratio, $K_i = N_i/n_i$, (see Eq. 1) with $N_i$ being the number of times the identity of $\mathcal{P}_i$ appears in the ring of a mesage and $n_i$ being the quantity of messages actually posted by them. Each value for $K_i$ depicted in Table I and Table II has been computed by averaging the results from 200 simulations. We include results for ring sizes $K = 8, 12, 16$.

As to the uniform strategy for choosing ring members, as expected from the analysis in Section IV-C, Table I shows that all the members of the community have been included

a similar number of times in rings of messages not authored by them. For instance, in the $K = 8$ experiments, we get that $r_i$ is approximately 25 for all the participants. This value has been computed from the data on the table and the expressions $K_i = N_i/n_i$ and $N_i = n_i + r_i$. A similar $r_i$ value for all the participants implies that the privacy achieved by participants decreases with the quantity of messages they post.

TABLE I
AVERAGE PRIVACY RESULTS MEASURED AS THE $K_i$ RATIO UNDER A UNIFORM CHOICE STRATEGY.

| $K$ | $User_1$ | $User_5$ | $User_{10}$ | $User_{15}$ |
|---|---|---|---|---|
| 8 | 26.38 | 6.00 | 3.47 | 2.64 |
| 12 | 40.24 | 8.78 | 4.89 | 3.57 |
| 16 | 54.13 | 11.62 | 6.34 | 4.51 |

TABLE II
AVERAGE PRIVACY RESULTS MEASURED AS THE $K_i$ RATIO UNDER A PREFERENTIAL ATTACHMENT STRATEGY.

| $K$ | $User_1$ | $User_5$ | $User_{10}$ | $User_{15}$ |
|---|---|---|---|---|
| 8 | 20.66 | 6.35 | 4.83 | 4.37 |
| 12 | 33.27 | 9.65 | 6.64 | 6.02 |
| 16 | 50.12 | 13.60 | 8.70 | 6.54 |

Regarding the simulations about the preferential attachment strategy, in order to avoid the inclusion of an excessive amount of data, we present the results under the parameter choice that has provided better results, namely, $w_i = 3$ and $w_m = 10$. The first conclusion we can take from Table II is that the preferential attachment strategy effectively increases the number of times that more active members are included in a ring. For the particular case, $K = 8$, a user who sent just one message has been part of about 20 rings of messages not authored by them, while one sending 15 messages was part of about 50, on average.

Nevertheless, Table II also shows that the degree of privacy decreases as members become more active. Ideally, the privacy level should not depend on the activity of participants, but this has not been the case. We can get more active members to be included in more rings by increasing the $w_m$ value, but this decision causes the undesired snowball effect mentioned in Section IV-D to happen again.

The results on Table II lead us to the conclusion that, under our experiments settings, so as to protect the most active members, the actual size of the rings has to be chosen to be between two and two and a half times the desired privacy level.

## V. CONCLUSION

This paper has analyzed the feasibility of ring signatures as a cryptographic tool for providing both anonymity and unlinkability to messages published on an on-line forum in which only authorized parties can post. The main advantage of ring signatures over group signatures is the absence of a revocation manager who, if corrupted, could lift the anonymity of all the messages.

We have analyzed two strategies for choosing the members to be part of the ring of a published message. The results show that the preferential attachment strategy leads to better results regarding the obtained unlinkability.

## ACKNOWLEDGMENTS

This work was funded in part by the Spanish Ministry of Science, Innovation and Universities (project number MTM2017-83271-R).

## REFERENCES

[1] D. Chaum and E. van Heyst, "Group signatures", in *Advances in Cryptology – EUROCRYPT'91*, Lecture Notes in Computer Science, vol. 547, pp. 257–256, 1991.

[2] R.L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret", in *Intl. Conf. on the Theory and Application of Cryptology and Information Security*, Lecture Notes in Computer Science, vol. 2248, pp. 552–565, 2001.

[3] J. Camenisch and M. Michels: "A group signature scheme based on an RSA-variant", in *BRICS Report Series*, RS-98-27, 1998.

[4] J. Ren and L. Harn, "Ring signature based on ElGamal signature", in *Intl. Conf. on Wireless Algorithms, Systems, and Applications*, Lecture Notes in Computer Science, vol. 4138, pp. 445-456, 2006.

[5] P.P. Tsang and V.K. Wei, "Short linkable ring signatures for e-voting, e-cash and attestation", in *Intl. Conf. on Information Security Practice and Experience*, Lecture Notes in Computer Science, vol. 3439, pp. 48–60, 2005.

[6] J.K. Liu, M.H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity", in *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, n. 1, pp. 157–165, 2014.

[7] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous identification in ad hoc groups", in *Intl. Conf. on the Theory and Applications of Cryptographic Techniques*, Springer, pp. 609–626, 2004.

[8] L. Sweeney, "K-anonymity: a model for protecting privacy", in *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, n. 5, pp. 557–570, 2002.

[9] B. Carron-Arthur, J.A. Cunningham, and K.M. Griffiths, "Describing the distribution of engagement in an Internet support group by post frequency: A comparison of the 90-9-1 Principle and Zipf's Law", in *Internet Interventions*, vol. 1, n. 4, pp. 165–168, 2014.

# Aplicación basada en Blockchain para una Lotería en línea con el uso de Tokens ERC-20 y ERC-721

Joan Amengual Mesquida
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
jamengual150899@gmail.com

M. Magdalena Payeras Capellà
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
mpayeras@uib.cat

Macià Mut Puigserver
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
macia.mut@uib.cat

***Resumen*—La lotería es una actividad financiera que año tras año atrae a más usuarios con la esperanza de ganar premios millonarios. La Sociedad Estatal Loterías y Apuestas del Estado (SELAE) estimó que cada español llegó a tener un gasto medio de 66,60 euros en sus décimos en la lotería nacional de navidad de 2021 [1], acumulando una cantidad destacable de 2.408 millones de euros en premios [2]. A pesar de ser la atención de muchos usuarios y acumulando altísimos premios, los sistemas de lotería siguen conservando sus características tradicionales y con ello siguen sin aportar la confianza requerida por los usuarios, que pueden poner en duda los resultados ganadores. El objetivo de este proyecto es proporcionar un protocolo para el funcionamiento de una lotería, mediante el cual se aporte transparencia y seguridad a sus usuarios. El protocolo resultante se basa en el uso de la tecnología blockchain que proporciona una estructura de datos pública, descentralizada, abierta e inmutable.**

***Index Terms*—Blockchain, Smart Contracts**

## I. Introducción

La lotería es una actividad financiera en la que los usuarios pagan dinero por sus apuestas, en la mayoría de los casos, por determinadas secuencias de números, teniendo derecho a ganar premios [3].

El sistema actual de lotería funciona de la siguiente manera:

1. Una empresa de lotería pone en marcha una lotería y dispone de un conjunto de números para ser comprados por los usuarios.
2. Las personas interesadas hacen apuestas comprando los números disponibles y pagan dinero para obtenerlos.
3. La empresa de lotería genera algunos números ganadores al azar.
4. Los usuarios que dispongan de los números ganadores cobran sus premios.

Estas empresas son los centros de estas actividades y cuentan con la confianza de los participantes. Por ello establecen normas y reglamentos que deben ser iguales para todos los participantes. Los eventos de lotería suelen ser organizados por empresas de lotería o gobiernos, y todo el proceso de lotería dura un período de tiempo determinado, como puede ser un día entero. Cuando se organizan eventos de lotería tradicional se invierte un largo tiempo en trámites de gestión de los boletos de lotería, lo que hace que el proceso de la lotería tradicional consuma mucho tiempo y resulte incómodo para las personas que organizan eventos de lotería.

En 1999, David Leason y Scott L. Sulliv desarrollaron un tipo de sistema de lotería en línea con una función centralizada. El diseño del sistema fue una de las mayores contribuciones al sistema de lotería tradicional, y satisfizo el deseo de la gente de celebrar eventos de lotería de forma instantánea y relativamente conveniente. Sin embargo, el sistema de lotería en línea de a menudo no garantiza a sus usuarios la necesidad fundamental de seguridad, tanto en lo que respecta a la seguridad de la información personal como a la seguridad de la propiedad para garantizar sus beneficios legales.

Es importante destacar que las loterías en línea tienen un coste mucho menor que las loterías tradicionales que funcionan fuera de la red. Debido a que las loterías en línea no requerien de personal para llevar a cabo el proceso de lotería, ya que todo esta automatizado. Aun así, se enfrentan a problemas únicos que socavan la confianza del consumidor, como por ejemplo:

- Seguridad.
- Velocidad.
- Falta de liquidez de los jugadores.

En los últimos años, los gobiernos han prohibido a muchas empresas de lotería, y el sistema de lotería en línea se ha convertido en una cuestión muy delicada. Los sistemas de lotería actuales, ya sean tradicionales o en línea, son sistemas centralizados que presentan varios problemas potenciales como los siguientes:

- El procedimiento de lotería tradicional lleva un periodo relativamente largo.
- Si el proceso no es totalmente justo al producir los ganadores, los beneficios de los jugadores se ven totalmente perjudicados.
- Los jugadores no tienen la seguridad sobre la cantidad de dinero recogida, que debería formar parte del premio, y por ello es posible que se pague a los ganadores una cantidad de dinero menor a la debida.

Existen riesgos financieros para la lotería y su gestión. Las plataformas de servidores centralizados pueden presentar vulnerabilidades. Por ejemplo, los juegos de números están legalmente obligados a utilizar la generación de números aleatorios (RNG) para mantener la imparcialidad del azar. Los casinos y las loterías deben declarar que tienen un generador de números aleatorios (RNG) preciso. No obstante, las máquinas físicas son fáciles de manipular y las versiones

digitales son fáciles de piratear.

Un caso es el de la Asociación de Loterías del Estado, que inició una investigación especial en el New York Times. Una persona cambió la funcionalidad del RNG, permitiendo que usuarios involucrados ganaran millones de dolares [4].

En los sistemas actuales los riesgos se pueden reducir aumentando la seguridad, transparencia y protección, características fundamentales de la tecnología blockchain. El retiro de ganancias es un proceso bastante lento. Puede incluir multitud de trámites, tarifas de administración, y transferencias bancarias, creando una experiencia de usuario deficiente.

Como se puede observar en lo descrito, las loterías tradicionales, ya sean físicas o en línea no favorecen un entorno dinámico y seguro, donde los usuarios disponen de una plena confianza sobre la lotería. Por ello, se ha considerado incorporar el uso de la tecnología Blockchain para mejorar estos aspectos mediante la utilización de Smart Contracts para la regulación de estos proyectos.

Véamos en detalle como estos conceptos pueden ayudar a mejorar características determinantes en el desarrollo de una lotería.

### *I-A. Blockchain*

La tecnología Blockchain o también conocida como Cadena de Bloques es una tecnología de red de pares (Peer-to-Peer, P2P) que permite almacenar datos y realizar operaciones de forma segura [5].

Mediante la tecnología Blockchain se permite la realización de transacciones económicas. Al comienzo de una transacción, el receptor de la transacción envía su clave pública al emisor mediante una firma digital basada en funciones de hash. La novedad en el proceso de transacción se puede realizar sin revelar la identidad, por lo que la seguridad de la información personal puede estar mejor protegida que en los sistemas de comercio en línea y offline originales.

La tecnología Blockchain es ideal para un proyecto de lotería, ya que nos permite mejorar el proyecto en varias fases de este, como por ejemplo:

- Los usuarios pueden conectarse al sistema y gestionar de forma rápida y dinámica sus compras, sin realizar horas de esperas.
- Los usuarios disponen de la confianza de un sistema descentralizado, sabiendo que los datos almacenados en el sistema son inmutables y una vez definidas las reglas del juego no se pueden cambiar.
- En todo momento se puede visualizar el premio de la lotería, evitando así la modificación de las cantidades a emitir en los premios.
- La tecnología Blockchain es capaz de certificar a los jugadores y a los boletos de lotería reduciendo las pérdidas, destrucción o manipulación de información.
- Un pago en los premios más rápido (instantáneo) junto con un rastro y registro completo, a través de tecnologías de libro mayor distribuido.

### *I-B. Smart Contracts*

Un Smart Contract o Contrato Inteligente se puede definir como un protocolo de transacciones informático que ejecuta las reglas definidas en un contrato.

Una vez desplegado, el Smart Contract es inmutable, es decir no se puede modificar. Mientras se cumplan los requisitos, se ejecutarán las operaciones correspondientes. Esto significa que el propio Smart Contract puede ser revisado por todos los nodos de la Blockchain y puede ser objeto de sus operaciones y de la participación de sus usuarios. Gracias a estas características, muchas actividades sociales pueden programarse en Smart Contracts para que sean más seguras y automatizadas [6].

Los Smart Contracts nos permiten definir un conjunto de reglas a cumplir para llevar a cabo las operaciones de compra y venta de boletos de lotería y la generación de un ganador de forma aleatoria, y esto resulta beneficioso para aumentar la seguridad de un proyecto de lotería, dando así confianza a los usuarios de la aleatoriedad de los resultados.

## II. Contribución del proyecto

En este proyecto se pretende dar una solución a las carencias actuales de las loterías tradicionales. Con el objetivo de disponer de un nuevo sistema descentralizado, veloz, sin intermediarios, seguro y con la privacidad requerida para los usuarios.

Las mejoras que intenta conseguir este proyecto con los sistemas actuales son las siguientes:

- Disponer del bote recolectado para el premio de la lotería de forma inmutable y público, con la finalidad de ofrecer una información transparente a los usuarios.
- Aumentar la confianza de los usuarios sobre los resultados de los números ganadores.
- Disponer de un sistema dinámico y automatizado.
- Ofrecer privacidad sobre las identidades de los usuarios.
- Evitar intermediarios en el sistema, con la finalidad de generar ganadores de forma transparente, alejando el sistema de la intervención humana.

## III. Protocolo diseñado

El protocolo de lotería en línea con el uso de la tecnología Blockchain implementado ha hecho uso de Smart Contracts, con el objetivo claro de definir las reglas requeridas de la lotería, como por ejemplo la aleatoriedad en la elección de un ganador. Además en este proyecto se ha adoptado el uso de Tokens ERC-20 y ERC-721, para llevar a un nivel más allá la experiencia del usuario.

El proyecto de lotería ha incorporado el uso de tokens ERC-20 para la realización de compras de boletos (números) de lotería. Se ha hecho uso de este estándar debido a sus características, como por ejemplo:

- Incorporan un nombre o identificador y un símbolo asociado. Por medio de estos dos valores, es posible

identificar y diferenciar los distintos tipos de tokens dentro la blockchain.
- Posee una interfaz para controlar y revisar los balances de las direcciones de sus dueños. Por este medio, el token indica el balance total de fondos contenidos en una dirección específica.
- El token tiene funciones para realizar la aprobación de fondos a terceros. Por ejemplo, un usuario puede prestar poderes de operación sobre la gestión de un determinado número de tokens a otro usuario [7].

Para hacer único este proyecto, los usuarios deben tener la propiedad absoluta sobre sus boletos, y con ello disponer de la confianza en que estos boletos son únicos y no existen dos iguales. Por este motivo se ha adaptado el proyecto para usar el token ERC-721, también conocido como Non-Fungible Token (NFT).

- Cada token ERC-721 posee un nombre. Este campo se utiliza para indicar a los contratos y aplicaciones externas la denominación del token.
- Contienen un campo que indica el balance de tokens dentro de una dirección.
- Cada token ERC-721 lleva definido un campo de funciones del propietario, usado para definir la propiedad del token y como se puede transferir la misma.
- Llevan definido un campo llamado Propietario y un campo identificador el cual permite garantizar la no fungibilidad del token [8].

El despliegue de los Smart Contracts de los tokens ERC-20 y ERC-721 es realizado de forma interna una vez el Smart Contract principal de la lotería es desplegado por el propietario (owner). Véase a continuación un diagrama de este procedimiento.

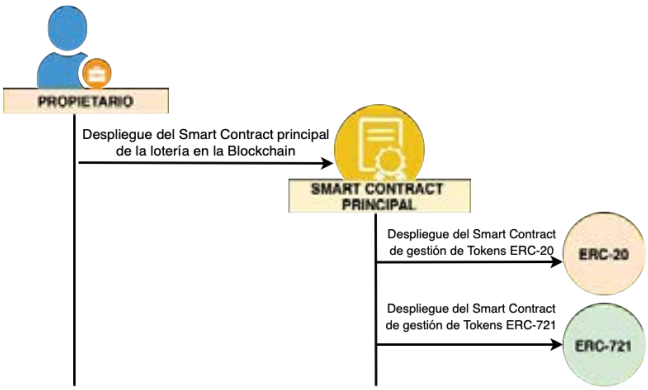



Figura 1. Diagrama del proceso de despliegue de los Smart Contracts

El protocolo de lotería se compone de tres fases bien diferenciadas.

1. Gestión de tokens ERC-20
2. Gestión de boletos de lotería
3. Emisión de los premios

A continuación se van a detallar cada una de las fases del proyecto.

### *III-A. Gestión de tokens ERC-20*

La primera fase del proyecto de lotería se enfoca en la gestión de los tokens ERC-20. Estos tokens son necesarios para que los usuarios realicen la compra de boletos (números) en la lotería, así que deben realizar la compra de estos tokens en un primer instante para proceder a la segunda fase de obtención de boletos de lotería.

El Smart Contract de la lotería dispone de las llamadas al Smart Contract del token ERC-20 para realizar el acceso a todas sus funciones.

El usuario en esta fase del proyecto requiere comprar una cantidad de tokens ERC-20. Cada token ERC-20 recibe un precio en *ethers* designado por el Smart Contract. Una vez el usuario inicia el proceso de compra y realiza el pago por los tokens de forma adecuada el Smart Contract de la lotería gestiona la emisión de estos tokens al comprador.

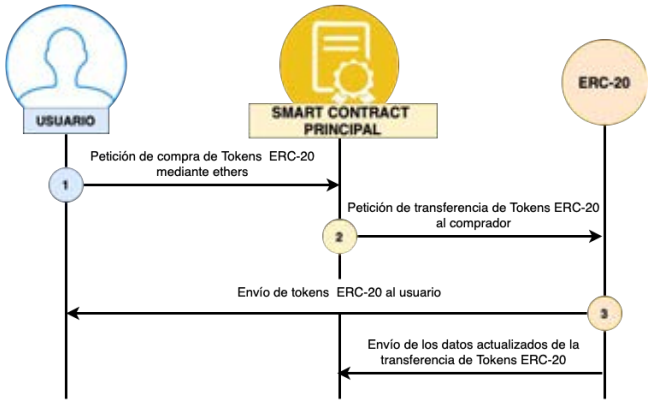



Figura 2. Diagrama del proceso de compra de tokens ERC-20.

### *III-B. Gestión de boletos de lotería*

La segunda fase del proyecto recoge las funcionalidades para la compra de los boletos y la gestión de estos. Los boletos de lotería son números generados de forma aleatoria en un rango preestablecido en el código del Smart Contract, de 10.000.000 de números disponibles. Cada número de la lotería se corresponde a un token NFT, asignando así la propiedad única de ese número al comprador.

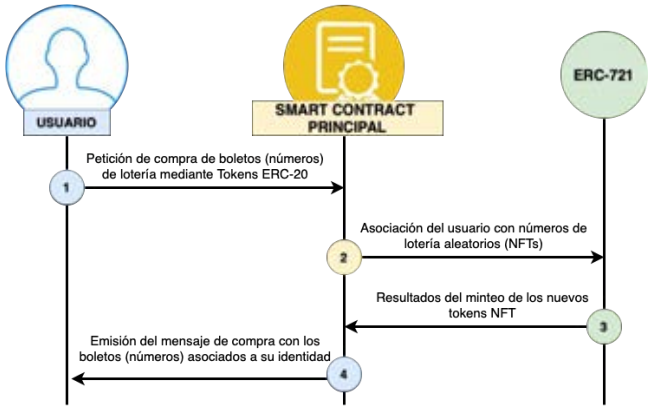



Figura 3. Diagrama del proceso de compra de boletos de lotería.

### *III-C. Emisión de los premios*

La tercera y última fase del proyecto se basa en la emisión de los premios a los ganadores. En esta fase del proyecto se va a elegir de forma aleatoria un número de boleto de lotería comprado por un usuario para designar así el ganador. La funcionalidad de la generación del ganador de forma aleatoria es donde radica todo el peso de la confianza del proyecto. Esta función depende de funciones de hash SHA256 que hacen posible salidas de datos aleatorias mediante un conjunto de entradas de datos. Donde una de las entradas es la marca de tiempo (*timestamp*) del último bloque de la Blockchain, buscando así tener más aleatoriedad en la salida de la función de hash.

Una vez seleccionado de forma aleatoria un ganador, se procede a la emisión de los premios. Para beneficiar de alguna forma a la empresa o individuo que ha gestionado todo el proyecto de lotería se hace un reparto de beneficios de la siguiente forma:

- Premio del ganador: 95 % del bote recogido.
- Recompensa al propietario del proyecto de la lotería: 5 % del bote recogido.

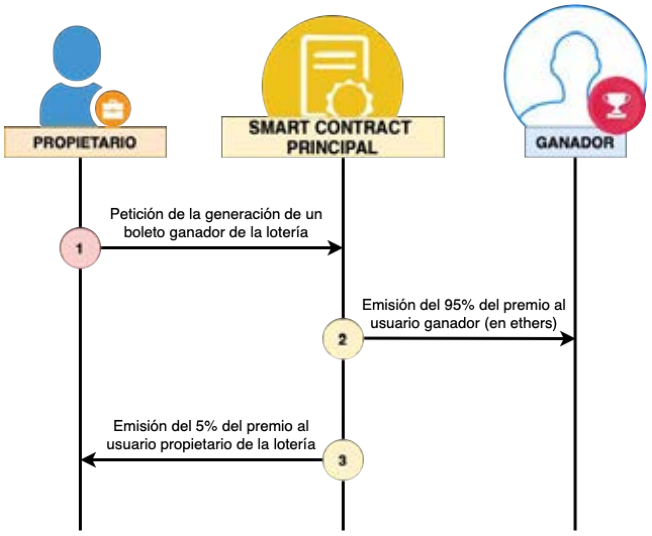



Figura 4. Diagrama del proceso de generación de un ganador de la lotería.

## IV. Análisis de costes

Este apartado se presentan los resultados de la implementación realizada del protocolo propuesto. Con el análisis de sus resultados se pretende determinar el coste de gas para cada una de las funciones del Smart Contract.

Se ha considerado realizar el despliegue de los Smart Contracts de lotería en varias blockchain para determinar la más rentable en términos económicos. Para este caso no se ha considerado el uso de la blockchain principal de Ethereum debido a su baja escalabilidad y altos precios pudiendo realizar únicamente 15 transacciones por segundo. Aún así somos capaces de utilizar otras blockchain que permiten el despliegue de Smart Contracts reduciendo los costes y aumentando la escalabilidad, como por ejemplo las siguientes:

- Polygon es una solución de escalado de *segunda capa* o *sidechain* que se ejecuta junto a la blockchain de Ethereum, lo que permite transacciones rápidas y comisiones bajas. MATIC es el token nativo de la red, que se utiliza para las comisiones, staking y mucho más. Polygon permite realizar hasta 200,000 transacciones por segundo.
- Binance Smart Chain (BSC) es una solución innovadora para introducir la interoperabilidad y la programabilidad en Binance Chain. Utiliza un sistema de 21 validadores que aprovechan el consenso Proof of Staked Authority o PoSA, lo que permite reducir las comisiones y los tiempos de los bloques.

Tabla I
Costes de ejecución de las funciones del Smart Contract.

| Funciones del Smart Contract | Blockchain | | | |
|---|---|---|---|---|
| | Polygon | | Binance Smart Chain | |
| | MATIC | USD | BNB | USD |
| Creación del Smart Contract | 0.0142 | 0.00839 | 0.0570263 | 16.98 |
| Compra de tokens ERC-20 | 0.001605 | 0.000948 | 0.006326 | 1.884 |
| Devolución de tokens | 0.000165 | 0.000097 | 0.000705 | 0.21 |
| Compra de boletos | 0.000895 | 0.00052 | 0.003354 | 0.999 |
| Generación del ganador | 0.0002652 | 0.000156 | 0.001002 | 0.2984 |

Las medidas mostradas en la Tabla I corresponden a los costes fijos de gas de cada uno de los métodos del contrato. Las medidas recogidas para la red de Polygon se han establecido en su token nativo MATIC y, analogamente, en Binance Smart Chain (BSC) se ha utilizado su propio token BNB. Adicionalmente, se ha añadido a modo orientativo el precio en dólares americanos de cada una de las funcionalidades teniendo en cuenta el valor al cambio entre monedas a 5 de junio de 2022. Tal y como se puede observar en la tabla la solución más económica es Polygon con unos costes realmente bajos y características de velocidad muy altas.

## V. Conclusiones

Ha quedado demostrado que es posible realizar la implementación de un protocolo de lotería mediante el uso de la tecnología blockchain permitiendo ofrecer unas características únicas a los usuarios como:

- Transparencia
- Mayor seguridad en la información
- Velocidad de operación

Disponemos de un proyecto de lotería con información descentralizada y almacenada de forma segura mediante las operaciones reguladas por Smart Contracts.

## Referencias

[1] El Español, https://www.elespanol.com/loterias/loteria-navidad/loteria-navidad-2021-cada-espanol-gasta-66-60-euros-probar-suerte-gordo-premio/634936655_0.html, 2021.
[2] AS, https://as.com/diarioas/2021/12/22/actualidad/1640152508_116262.html, 2021.
[3] Thomas Barker and Marjie Britz: "Jokers wild:Legalized gambling in the twenty-first century". Greenwood Publishing Group, 2000.
[4] Dan Goodin, "Neutered random number generator let man rig million dollar lotteries", https://arstechnica.com/information-technology/2016/04/neutered-random-number-generator-let-man-rig-million-dollar-lotteries/, 2022
[5] Y.Chen, S.Chen, and I.Lin: "Blockchain based smart contract for bidding system", 2018. https://doi.org/10.1109/ICASI.2018.8394569
[6] B.K.Mohanta, S.S.Panda, and D.Jena: .An Overview of Smart Contract and Use Casesïn Blockchain Technology, 2018. https://doi.org/10.1109/ICCCNT.2018.8494045
[7] Bi2me, "¿Qué es un token ERC-20?", https://academy.bit2me.com/que-es-erc-20-token/, 2022.
[8] Bi2me, "¿Qué es un token ERC 721?"https://academy.bit2me.com/que-es-token-erc-721/, 2022.

# Seguridad y Privacidad en un Sistema de Control de Acceso Distribuido para Zonas de Bajas Emisiones

Carles Anglés-Tafalla
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
UNESCO Chair in Data Privacy
Av. Països Catalans 26, E-43007 Tarragona, Spain
carles.angles@urv.cat

Jordi Castellà-Roca
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
UNESCO Chair in Data Privacy
Av. Països Catalans 26, E-43007 Tarragona, Spain
jordi.castella@urv.cat

Alexandre Viejo
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
UNESCO Chair in Data Privacy
Av. Països Catalans 26, E-43007 Tarragona, Spain
alexandre.viejo@urv.cat

***Resumen*—Las Zonas de Bajas Emisiones (LEZ) son áreas donde se aplican restricciones al acceso de vehículos contaminantes. Estas soluciones se han convertido en los mecanismos principales para hacer frente a la contaminación ambiental y congestión vehicular en grandes zonas urbanas. Los sistemas desplegados actualmente para controlar dichas restricciones hacen uso de redes de cámaras para identificar los vehículos que circulan por la zona restringida, suponiendo un riesgo para la privacidad de sus usuarios. Aunque en la literatura han aparecido propuestas más respetuosas con la privacidad que hacen un uso mínimo de las cámaras, todas ellas presentan un problema estructural común: dependen de entidades centralizadas para desplegar la infraestructura, gestionar los accesos de los vehículos y cobrar las correspondientes tarifas. Estas entidades suponen un *single point of failure* en el sistema que pone en riesgo su seguridad y disponibilidad. Para hacer frente a esta problemática, en este artículo proponemos un nuevo sistema totalmente distribuido y autónomo para gestionar los accesos a LEZs, que hace uso de los sistemas avanzados de detección y comunicación de los vehículos de nueva generación para evitar el despliegue de infraestructuras de control. Esto, combinado con un sistema de gestión descentralizado basado en smart contracts que procesa los accesos como transacciones de Blockchain, elimina cualquier dependencia hacia *third parties* centralizadas con ánimo de lucro en el proceso de control de acceso a las LEZs.**

***Index Terms*—Low emission zones, Smart cities, Privacy, Security, Smart Contracts, Blockchain.**

## I. Introducción

En los últimos años, las Zonas de Bajas Emisiones (*LEZ*) y de Cargo por Congestión (*CCZ*), es decir, áreas definidas donde se aplican restricciones a los vehículos contaminantes, se han convertido en mecanismos imprescindibles para hacer frente a la congestión del tráfico urbano y el impacto que esta ejerce sobre la contaminación ambiental. Prueba de ello es la gran proliferación de estas medidas en los países Europeos[1] o su intención de adoptarlas en los próximos años, como en el caso de España[2].

Ante este escenario, en las grandes áreas urbanas se han desplegado controles de acceso automatizados que permitan cumplir las restricciones que imponen estas zonas. El funcionamiento de dichos sistemas se basa en redes de cámaras, e.g. Barcelona, Estocolomo[3] o Londres [1], cuyo propósito consiste en fotografiar indiscriminadamente la matrícula de los vehículos que circulan por la zona restringida. Entonces, a partir de un sistema de reconocimiento de matrículas (ANPR), se identifica a los vehículos y se determina la tarifa que se debe abonar de acuerdo con la normativa o restricciones aplicables a la zona.

Tal como se infiere de los ejemplos citados, el uso de estas estrategias resulta muy invasivo, ya que todos los vehículos son identificados cada vez se aproximan a una infraestructura de control. Esta situación supone un riesgo para la privacidad de los usuarios y revela la necesidad de sistemas de control alternativos que gestionen el tránsito de las LEZ o CCZ de un modo más respetuoso con la privacidad de aquellos circulan por ellas.

Además de la amenaza en lo que a privacidad se refiere, se han detectado aspectos estructurales a mejorar en los actuales controles de acceso. Más concretamente, los actuales sistemas presentan una clara dependencia hacia las entidades centralizadas que despliegan las infraestructuras, calculan las tarifas y detectan a usuarios fraudulentos. Estas entidades se convierten en un punto crítico de la arquitectura al constituir un *single point of failure* que pone en riesgo la seguridad y disponibilidad del sistema.

Recientemente, han entrado en escena nuevos paradigmas que permiten acordar de manera descentralizada transacciones de recursos arbitrarios, tales como interacciones entre vehículos e infraestructuras, a través de un *public open ledger* verificable llamado Blockchain, sin necesidad de recurrir

[1]Urban Access Regulations, http://urbanaccessregulations.eu/userhome/map

[2]Ley de Cambio Climático y Transición Energética, https://www.miteco.gob.es/es/cambio-climatico/participacion-publica/marco-estrategico-energia-y-clima.aspx

[3]Stockholm charging scheme, http://www.stockholm.se/trangselskatt

a entidades centralizadas o de confianza. Esta tecnología, combinada con los sistemas de detección, radar, cámaras y comunicación que incorporan los vehículos de nueva generación, abre la posibilidad a la concepción de soluciones distribuidas totalmente autónomas que no dependen de *third parties* centralizadas ni de la necesidad desplegar equipamiento adicional, eliminando los costes de implementación y mantenimiento que estos lleva asociados.

### *I-A. Antecedentes*

En la última década, se ha publicado un gran número de artículos como respuesta a los problemas de privacidad en controles de acceso a zonas restringidas en entornos vehiculares, tales como Zonas de Bajas Emisiones (LEZ), Peaje por Congestión (CCZ) y Peaje Electrónico (ERP).

Inicialmente estas propuestas [2], [3], [4], [5], [6], [7] estaban basadas en la unidad de a bordo de los vehículos (OBU), recopilando datos relevantes para el cómputo de tarifas; una *third party* centralizada, i.e. Proveedor de Servicios (SP), que verifica todo el proceso y cobra las tarifas convenidas; y una red de puntos de control basada en cámaras que registra las matrículas de los vehículos para evitar el fraude. No obstante, tal como se destaca en [5], este planteamiento solo es viable si la ubicación de los puntos de control es secreto, volviéndose especialmente invasivo y poco respetuoso con la privacidad de los usuarios si el número de puntos de control es elevado para prevenir que los vehículos los eviten intencionadamente.

Desde que se identificó dicha problemática, se ha ido consolidado un nuevo paradigma desde que fue presentado por primera vez en [8] y posteriormente adoptado por [9], [10], [11], [12], el cual promueve que siempre se preserve la privacidad de los usuarios a no ser que estos intenten cometer fraude. Este planteamiento propone un proceso de autenticación cada vez que el vehículo se encuentra con alguna de las infraestructuras del sistema, fotografiando la matrícula del vehículo únicamente en caso de no completar o omitir dicho proceso. Partiendo de este planteamiento, estas propuestas protegen la privacidad de los usuarios durante el proceso de autenticación por medio de distintos mecanismos, tales como esquemas de firmas de grupo [8], [9], seudónimos renovables [10], [11] o pruebas de conocimiento nulo [12]. No obstante, la dependencia de dichas propuestas en *third parties* centralizadas a cargo de validar las pruebas de acceso, computar las subsecuentes tarifas asociadas y cobrar dichas cantidades, supone un aspecto estructural común a mejorar, revelando un "single point of failure" que hace a estos sistemas más vulnerables a fallos y ataques, y compromete su seguridad y disponibilidad.

Con el fin de eliminar esta figura centralizada, en [13], [14] proponen una mejora descentralizada sobre [10] basada en el uso de Smart Contracts para gestionar los accesos a la LEZ como transacciones en el Blockchain, permitiendo determinar y cobrar el precio de los accesos sin intervención de *third parties*. Aunque la propuesta aborda con éxito los problemas de centralización y ofrece mecanismos para mitigar en cierto punto el monopolio en la gestión de las infraestructuras de acceso, el sistema sigue dependiendo de *third parties* con ánimo de lucro que desplieguen, gestionen y mantengan las infraestructuras de control de acceso y sus correspondientes gastos asociados.

### *I-B. Contribuciones y plan del artículo*

Recientemente, la tecnología blockchain se ha convertido en la piedra angular a la hora de implementar ecosistemas de intercambio distribuidos de un modo fiable en un gran numero de dominios. Atendiendo a la dirección de la literatura actual, los escenarios de sistemas de transporte inteligente tales como LEZs o CCZs también dan muestras de adoptar esta tendencia.

Siguiendo esta dinámica descentralizada, y considerando los puntos de mejora identificados en los sistemas descentralizados en la literatura, en este artículo proponemos un nuevo sistema distribuido y totalmente autónomo para gestionar los accesos a LEZs, que hace uso de los sistemas de detección y comunicación avanzados de que disponen los vehículos de nueva generación, a cambio de incentivos, para evitar el despliegue y mantenimiento de infraestructuras de control. Además, el sistema, mediante el uso de smart contracts, permite procesar los accesos LEZ como transacciones de Blockchain y saldarlos descentralizadamente, eliminando de esta forma cualquier intervención de *third parties* centralizadas en el control, tarificación y cobro de los movimientos de los vehículos en la zona restringida.

En resumen, la propuesta ofrece los siguientes beneficios:

- *Anonimato revocable:* el sistema preserva la privacidad de los conductores que se comportan con honestidad pero es capaz de identificar a los usuarios deshonestos y sancionarlos.
- *Sistema descentralizado:* el sistema descentraliza las entidades responsables de los procesos generación, verificación, tarificación y cobro de los accesos de los vehículos a la LEZ; sustituyendo a las entidades responsables de dichos procesos por una red distribuida basada en Blockchain, combinada con el soporte de los sistemas de detección y comunicación avanzados de los vehículos de nueva generación que circulen por la LEZ.
- *Sin infraestructura:* el sistema no requiere el despliegue de infraestructuras de control de acceso gracias al uso de los sensores, cámaras y sistemas de comunicación de los vehículos, evitando los costes de implementación y mantenimiento y la dependencia hacia las entidades centralizadas que las gestionan.
- *Pago anónimo:* el sistema protege el anonimato de los usuarios durante el proceso de pago de tarifas por medio de la tecnología Blockchain y las medidas de preservación de privacidad aplicadas.

El resto del artículo está organizado de la siguiente manera. La sección II introduce la nueva propuesta. La sección III formaliza los protocolos que sustentan el sistema propuesto. Finalmente, la Sección IV recoge las conclusiones.

## II. Modelo del Sistema

### *II-A. Actores*

Nuestro sistema involucra a los siguientes actores: i) Administrador de la LEZ (*LA*); ii) Conductores (*D*); iii) Vehículo de control de Acceso (*ACV*); iv) Smart Contract de la LEZ (*SC*); y v) Servicio de Mixing de criptomonedas (*M*).

- Administrador de la LEZ (*LA*): es el encargado de dirigir la LEZ y establecer las restricciones que se aplican a los vehículos. Entre sus tareas destacan la emisión de certificados digitales para el resto de entidades, el

despliegue del Smart Contract de la LEZ y la gestión de categorías de los vehículos.

- Conductores (*D*): son los usuarios potenciales, quienes, a través de las Unidades de a Bordo (*OBU*) de sus vehículos, interactúan con las infraestructuras del sistema. Las OBU*s* son dispositivos capaces de realizar operaciones criptográficas, equipados con tecnología GPS, 4G, sistema de comunicación de corto alcance (e.g. Bluetooth, Zigbee, IEEE 802.11p/DSRC, etc.) y un Secure Element (*SE*) en el cual una autoridad de confianza ha almacenado la matrícula del vehículo.
- Vehículo de control de Acceso (*ACV*): son vehículos que generan y verifican las pruebas de acceso de los vehículos que circulan por la zona restringida. Para desempeñar esta función, un vehículo ha de contar, además de las prestaciones de $D$, con un sistema avanzado de asistencia de conducción (*ADAS*), que incluya sensores de proximidad y cámaras en la sección trasera o delantera.
- Smart Contract (*SC*): es un protocolo de transacción especialmente programado para incluir los detalles de acceso a la LEZ en el blockchain, permitiendo verificar, tarificar y pagar dichos accesos por medio de criptomonedas.
- Servicio de Mixing de Criptomonedas (*M*): es una entidad con ánimo de lucro capaz de ofuscar, a cambio de la pertinente tasa, la transferencia de fondos entre dos carteras digitales[4] [5].

### II-B. *Visión general del sistema*

La figura 1 muestra el esquema general del sistema de control de acceso propuesto. En este escenario, al acceder a la LEZ, los vehículos con las capacidades para actuar como $ACV$s generan y verifican pruebas de acceso a los vehículos de su entorno cercano. Para ello, en el momento en que el sistema de sensores de un $ACV$ detecta otro vehículo, este intenta establecer conexión con el $D$ detectado mediante un sistema de comunicación inalámbrica de corto alcance. Dicho proceso de detección puede limitarse a las partes del vehículo que disponen de cámaras (trasera y/o delantera). Una vez ambos vehículos se han autenticado, $ACV$ solicita, verifica y almacena la prueba de acceso de $D$, o la genera en caso que $D$ no disponga de una. Si en algún momento, $D$ intenta omitir o alterar este protocolo de alguna forma, $ACV$ hace una foto de la matrícula del vehículo, pudiendo así reportar su comportamiento ante el $LA$.

Una vez $D$ ha abandonado la LEZ, puede iniciar el proceso de pago invocando al método de pago del $SC$ de la LEZ, usando su recibo de acceso como parámetro. La lógica del $SC$ utiliza los datos contenidos en dicho recibo para verificar su validez, calcular el importe de la tarifa y transferir las criptomonedas correspondientes desde cartera digital de $D$ a las carteras de $ACV$ y $LA$ en la proporción pertinente.

Pasado un tiempo, fijado por la entidad que despliega el $SC$, los $ACV$s que hayan obtenido la prueba de acceso verifican si los datos de ese acceso han sido publicados en el Blockchain y dicha transacción aparece como pagada. En caso de encontrar alguna irregularidad, cualquiera de los

[4]Tornado Cash, https://defirate.com/tornado-cash/
[5]ETH-Mixer, https://eth-mixer.com/

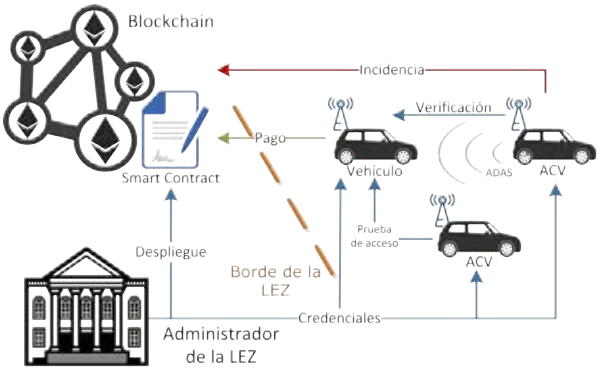



Figura 1. Esquema general del sistema

$ACV$ involucrados puede abrir una incidencia, publicando su copia almacenada de la prueba de acceso. Haciendo pública esta información, firmada por $D$, el $LA$ dispone de todo lo necesario para resolver la incidencia, identificar al $D$ fraudulento y recompensar al $ACV$ si así se requiere.

## III. Descripción del protocolo

Esta sección formaliza los protocolos que componen el sistema propuesto dando los detalles suficientes para su implementación. Estos protocolos son: *Configuración de la OBU*, *Adquisición de Tokens para la cartera*, *Generación prueba de acceso*, *Control del acceso*, *Pago*, *Verificación de Pago*, *Anti-Fraude* y *Renovación de seudónimos*.

### III-A. *Configuración de la OBU*

Este protocolo inicial consiste en configurar la OBU de $D$ para obtener las credenciales necesarias para interactuar con el resto de entidades del sistema. Con este fin, la OBU establece un canal seguro, vía TLS, con $LA$ y proporciona la información de $D$ (matrícula, marca, modelo, etc.). Se asume que la OBU no puede ser manipulada para proporcionar información falsa y que $LA$, como entidad gubernamental, puede verificar y obtener los datos de $D$. Entonces, LA realiza el proceso siguiente:

a) $LA$ verifica los datos del vehículo y obtiene los datos del propietario (nombre, residencia, etc.).
b) $LA$ genera un código de vehículo pseudoaleatorio $\beta$ y lo vincula con $D$.
c) $LA$ envía $\beta$ como *One Time Password (OTS)* a $D$ a través de un canal alternativo.

Una vez $D$ recibe $\beta$:

a) $D$ genera un par de claves $(sk^D, pk^D)$.
b) $D$ prepara una solicitud de certificado $CSR(pk^D)$ para la clave pública generada.
c) $D$ envía $\beta$ y $CSR(pk^D)$ a $LA$.

Cuando $LA$ recibe un $CSR(pk^D)$ válido:

a) $LA$ verifica la validez del código $\beta$.
b) $LA$ recupera los datos de $D$ vinculados a $\beta$.
c) $LA$ emite el certificado $\Gamma^D$, incluyendo $\beta$ en el campo *CommonName* y la categoría de emisiones del vehículo $cat$ como extensión (X509 v3).
d) $LA$ envía el certificado generado $\Gamma^D$ a $D$.

Finalmente, $D$ realiza las siguientes operaciones:

a) $D$ verifica el certificado $\Gamma^D$ a partir de $\Gamma^{LA}$. Si la verificación falla, se aborta el proceso; de lo contrario, sigue adelante.
b) $D$ almacena de forma segura $(sk^D, pk^D)$ y $\Gamma^D$.

Una vez obtenidas sus credenciales, $D$ genera una cartera digital $W_D$ compatible con Ethereum Virtual Machine (EVM) con el fin de poder interactuar con el $SC$ que gestiona la LEZ. Con este fin, D genera una clave privada de 256-bits $sk_D^W$, una clave pública de 512 bits $pk_D^W$ y su correspondiente *address*, de acuerdo con las especificaciones de la red compatible con EVM donde se despliegue el SC.

### *III-B. Adquisición de Tokens para la cartera*

Con el fin de pagar su uso de la LEZ e interactuar con el $SC$ de la LEZ, $D$ debe a adquirir criptomonedas (i.e. elementos que actúan como divisa en nuestro sistema). Con este propósito, se espera que $LA$ hospede un sitio web donde los usuarios puedan adquirir criptomonedas, que se transfieren directamente a sus carteras digitales[6], por medio de sistemas de pago clásicos (e.g. tarjeta de crédito, transferencia bancaria, etc.). Como este proceso podría permitir vincular la cartera digital de $D$ a su cuenta bancaria, poniendo en riesgo su privacidad, D realiza los pasos siguientes para evitarlo:

a) $D$ crea una cartera temporal $W_D^T$.
b) $D$ al sitio web de $LA$ y adquiere criptomonedas, que se transfieren a $W_D^T$.
c) $D$ solicita al servicio de mixing $M$, que transfiera las monedas de $W_D^T$ a la cartera real $W_D$ de $D$.
d) $M$, mediante el mixing, ofusca el vínculo entre las carteras de origen y destino al transferir los fondos, evitando que $LA$ pueda identificar las transacciones de $D$ en el Blockchain.
e) $D$ descarta la cartera temporal un vez las monedas han sido transferidas.

### *III-C. Generación prueba de acceso*

Durante el tránsito por la zona restringida, un vehículo actuando como controlador de acceso ($ACV$) se ofrece a generar pruebas de acceso para otros $D$s. Para ello, cuando los sensores delanteros o traseros del sistema ADAS detectan un vehículo $D$ cercano, $ACV$ intenta establecer conexión segura, involucrando autenticación bilateral, con $D$ usando un sistema de comunicación de corto alcance (e.g.). Este proceso supone intercambio de certificados entre $D$ y $ACV$.

a) $D$ solicita una prueba de acceso para la zona donde transita $ar$ = ($N_D$, $loc$, $t_a$, $cat$), siendo $N_D$ un nonce, $loc$ la ubicación, $t_a$ una marca temporal y $cat$ la categoría de emisiones del vehículo.
b) $ACV$ verifica los datos de $ar$ y genera el código de acceso $\delta = N_D \| N_{ACP}$.
c) $ACV$ genera un recibo $\rho$ = ($\delta$, $zona_{id}$, $t_a$, $cat$) y su firma $\rho'^W_{ACV}$, generada con clave privada de su cartera digital para que el $SC$ lo pueda verificar on-chain.
d) $ACV$ prepara la prueba de acceso $ap$ = ($\beta$, $\rho$, $\rho'^W_{ACV}$) y la firma digitalmente $ap'_{ACV}$. $\beta$ se obtiene del certificado $\Gamma^D$.
e) $D$ recibe $ap$ y $ap'_{ACV}$. Verifica la firma $ap'_{ACV}$, comprueba que los datos contenidos en $ap$ son correctos y la firma $\rho'^W_{ACV}$ es válida.
f) $D$ almacena la prueba de acceso $ap$, $ap'_{ACV}$ y el certificado $\Gamma^{ACV}$.
g) $D$ genera la respuesta $rap$ = ($t_b$, $\beta$, $\rho$, $\rho'^W_{ACV}$, $ap'_{ACV}$) y su firma digital $rap'_D$, dónde $t_b$ es el timestamp de la respuesta.
h) $ACV$ recibe y almacena $rap$ y $rap'_D$, una vez ha verificado la firma $rap'_D$ y ha contrastado los datos contenidos en $rap$ con los enviados $ap$.

Si en algún instante se interrumpe el proceso de generación, $ACP$ inicia la fase anti-fraude para obtener evidencias de un posible $D$ deshonesto.

### *III-D. Control del acceso*

En el instante que un vehículo actuando como $ACV$ detecta a otro $D$ con los sensores ADAS, $ACV$ puede iniciar un proceso de control. Para ello $ACV$ establece conexión segura con autenticación bilateral con $D$ e realiza los siguientes pasos:

a) $D$ recupera su prueba de acceso $ap$, genera una respuesta $rap$ = ($ap$, $ap'_{ACV}$, $t_c$) y la firma digitalmente $rap'_D$.
b) $D$ envia $rap$ y $rap'_D$ y el certificado del $ACV$ generador de $ap$.
c) $ACV$ verifica las firmas $rap'_D$ y $ap'_{ACV}$, y comprueba que la prueba tiene validez en el marco espacial y temporal actual.
d) $ACV$ almacena $rap$ y $rap'_D$ como prueba. Si alguna verificación falla, $ACV$ inicia la fase anti-fraude.

### *III-E. Pago*

Obtenida la prueba de acceso, $D$ puede abonar el coste de su acceso publicando el recibo $\rho$ en el blockchain haciendo uso de la lógica de los smart contracts, y evitando así la intervención de entidades centralizadas. Con ese fin, $D$ invoca al método *registrar_acceso* del $SC$ usando los datos de acceso contenidos en $\rho$ y su firma $\rho'^W_{ACV}$ como parámetros. $SC$ realiza las siguientes operaciones on-chain:

a) $SC$ verifica la firma $\rho'^W_{ACV}$ y obtiene la "address" de la cartera de $ACV$.
b) $SC$ calcula la coste de acceso a partir de la $zona_{id}$ el $t_a$ y $cat$, de acuerdo con las tasas publicadas en el blockchain por $LA$.
c) $SC$ transfiere la cantidad de criptomonedas equivalentes al coste desde la cartera de $D$ a las carteras de $ACV$ y $LA$ en la proporción pertinente.
d) $SC$ actualiza el estado del acceso $\delta$ a "pagado". Si alguno de los pasos anteriores no se puede completar, se aborta el proceso y se revierten los cambios en el $SC$.

### *III-F. Verificación de Pago*

$LA$, como responsable de la zona restringida y dueño del $SC$, fija el tiempo de que disponen los $D$s para realizar sus pagos. Transcurrido ese tiempo los $ACV$ pueden verificar y reportar los accesos impagados de los que hayan recogido pruebas:

a) $ACV$ obtiene la información de la transacción $\delta$ publicada en el blockchain.

[6]My Ether Wallet - https://ccswap.myetherwallet.com/

b) $ACV$ verifica si el acceso está publicado, su estado es "pagado" y los datos coinciden con los de su copia $rap$.
c) Si no se cumplen estas condiciones, $ACV$ invoca el método del Smart Contract *incidencia_pago* para publicar una incidencia, enviando $rap$ y $rap'_D$ como parámetros.
d) La lógica del $SC$ comprueba si se cumplen las condiciones temporales para abrir una incidencia, verifica que el acceso $\delta$ no está pagado y, en caso contrario, comprueba que los datos en $rap$ difieren de los publicados.
e) $SC$ publica $rap$ y $rap'_D$ si se cumplen las anteriores condiciones. Con esta información publicada en el blockchain, $LA$ dispone de las pruebas necesarias para identificar y sancionar la entidad fraudulenta y recompensar al vencedor de la disputa.

### *III-G. Anti-Fraude*

Este mecanismo se ejecuta en paralelo en un $ACV$ durante las fases de generación de prueba de acceso (Sección III-C) y control de acceso (Sección III-D), haciendo uso del sistema ADAS, i.e. la cámara/s del vehículo y los sensores de proximidad, para obtener información de los vehículos fraudulentos. Con el fin de simplificar este escenario, el proceso se puede restringir a los vehículos traseros y delanteros debido a la mayor sensorización presente en dichas partes del vehículo. La idea general de este sistema fue inicialmente propuesta en [11] para sistemas ERP.

Durante las mencionadas fases del protocolo, los sensores del vehículo obtienen información precisa de la localización de otros $D$s. Simultáneamente, $ACV$ hace uso de un sistema indicador de intensidad de la señal recibida (e.g. RSSI [15]) durante la comunicación con $D$, para estimar la distancia entre ambos vehículos. Esta información, junto con la geolocalización enviada por $D$ durante las fases de control, se combina para determinar si el $D$ con el que se está comunicando es el detectado por los sensores, o se trata de un vehículo fraudulento con la OBU desconectada. De forma similar, la no respuesta por parte de un vehículo detectado por los sensores de $ACV$ también indica la presencia de un $D$ fraudulento. Ante esta situación la cámara del $ACV$ hace una foto de la matrícula del infractor.

Una vez obtenida la evidencia, $ACV$ envía la prueba, junto con una marca temporal y una geolocalización, firmada digitalmente a $LA$. $LA$ recolecta las evidencias enviadas por los $ACV$s y, una vez dispone de una cierto número de pruebas emitidas por distintos $ACV$s, procede a identificar y sancionar al $D$ en cuestión. Si la sanción prospera, i.e. $D$ no aporta su prueba de acceso $ap$ consecuentemente publicada en el Blockchain, los $ACV$s que hayan contribuido son debidamente remunerados.

### *III-H. Renovación de seudónimos*

Aunque la identidad real de $D$ está oculto tras el código de su vehículo $\beta$ y la *address* de su cartera digital $W_D$, dichos elementos, si se usan en diversas interacciones, pueden llevar a la vinculación e incluso a la identificación de $D$.

Para evitarlo, un usuario $D$ puede solicitar un nuevo código $\beta^*$ para prevenir que se puedan vincular sus accesos. El cambio del código del vehículo $\beta$ implica la generación de nuevas claves criptográficas $(sk^D, pk^D)$ y certificado $\Gamma^D$, ya que incluye $\beta$ en el campo *CommonName*, en un proceso similar al descrito en la sección III-A.

En lo referente a la *address* de $W_D$, $D$ debe generar una nueva cartera digital $W_D^*$, descrito en III-A, con la que dispondrá de una nueva dirección para sus transacciones en el blockchain. En caso de no haber agotado todos los fondos de su vieja cartera, $D$ puede transferirlos a la nueva por medio de los servicios ofuscación de $M$, evitando que ambas *address* puedan ser vinculadas.

## IV. Conclusiones

Las actuales propuestas de control de acceso orientadas a la privacidad en escenarios LEZ, CCZ y ERP, presentan una dependencia clara hacia entidades centralizadas en sus principales cometidos, es decir, desplegar y mantener las infraestructuras de control, gestionar los accesos, determinar sus respectivas tarifas y promover sus cobros.

Aunque se han presentado propuestas para acabar con la estructura centralizada de algunas de estas operaciones, mediante el uso de la tecnología de los *smart contracts* y el subyacente paradigma descentralizado del Blockchain, dichas soluciones siguen dependiendo de *third parties* con ánimo de lucro que implementen, gestionen y mantengan las infraestructuras que controlan los accesos de los vehículos a la zona restringida. Siguiendo este planteamiento, en este artículo, hemos propuesto un control de acceso totalmente distribuido para escenarios LEZ que preserva privacidad de los usuarios honestos. De este modo, haciendo uso de los sistemas de detección, cámaras y comunicación avanzados que integran los vehículos de nueva generación, nuestra propuesta evita el despliegue de infraestructuras para gestionar el acceso de vehículos, evitando los subyacentes costes de despliegue y mantenimiento, y eliminando de la ecuación toda dependencia hacia entidades centralizadas.

Como trabajo futuro, se prevé implementar el sistema de control propuesto con el fin de verificar su viabilidad, así como llevar a cabo estudios basados en simulaciones que permitan determinar el umbral de ACVs honestos en circulación que nuestro sistema necesita para garantizar su correcto funcionamiento.

## Agradecimientos

This research is supported by the European Union Regional Development Fund within the framework of the ERDF Operational Program of Catalonia 2014-2020 with a grant of 50 % of the total cost eligible, under the "FEM-IOT" project [001-P-001682]; by the EU's European Regional Development Fund (ERDF), through the "ERDF Catalonia Operational Programme 2014-2020, investment priority for the creation of jobs and sustainable growth", under the Territorial Specialisation and Competitiveness Project (PECT) "Cuidem el que ens uneix - Sensòrica" project [PR15-020174]; and by Grants RTI2018-095094-B-C21 "Consent" and PID2021-125962OB-C32 "SECURING/DATA" funded by MCIN/AEI/10.13039/501100011033 FEDER, UE.

## Referencias

[1] G. Santos, "Urban congestion charging: a comparison between london and singapore," *Transport Reviews*, vol. 25, no. 5, pp. 511–534, 2005.

[2] R. A. Popa, H. Balakrishnan, and A. J. Blumberg, "Vpriv: Protecting privacy in location-based vehicular services," in *18th USENIX Security Symposium*. USENIX Association, 2009.
[3] X. Chen, G. Lenzini, S. Mauw, and J. Pang, "A group signature based electronic toll pricing system," in *2012 Seventh International Conference on Availability, Reliability and Security*. IEEE, 2012, pp. 85–93.
[4] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, "Pretp: Privacy-preserving electronic toll pricing." in *USENIX Security Symposium*, vol. 10, 2010, pp. 63–78.
[5] S. Meiklejohn, K. Mowery, S. Checkoway, and H. Shacham, "The phantom tollbooth: Privacy-preserving electronic toll collection in the presence of driver collusion." in *USENIX security symposium*, vol. 201, 2011, pp. 1–16.
[6] J. Day, Y. Huang, E. Knapp, and I. Goldberg, "Spectre: spot-checked private ecash tolling at roadside," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011, pp. 61–68.
[7] F. D. Garcia, E. R. Verheul, and B. Jacobs, "Cell-based privacy-friendly roadpricing," *Computers & Mathematics with Applications*, vol. 65, no. 5, pp. 774–785, 2013.
[8] R. Jardí-Cedó, M. Mut-Puigserver, M. M. Payeras, J. Castella-Roca, and A. Viejo, "Time-based low emission zones preserving drivers' privacy," *Future Generation Computer Systems*, vol. 80, pp. 558–571, 2018.
[9] R. Jardí-Cedó, J. Castellà, and A. Viejo, "Privacy-preserving electronic road pricing system for low emission zones with dynamic pricing," *Security and Communication Networks*, vol. 9, pp. 3197–3218, 2016.
[10] C. Anglès-Tafalla, J. Castellà-Roca, M. Mut-Puigserver, M. M. Payeras-Capellà, and A. Viejo, "Secure and privacy-preserving lightweight access control system for low emission zones," *Computer Networks*, vol. 145, pp. 13–26, 2018.
[11] S. Bouchelaghem and M. Omar, "Reliable and secure distributed smart road pricing system for smart cities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1592–1603, 2018.
[12] V. Fetzer, M. Hoffmann, M. Nagel, A. Rupp, and R. Schwerdt, "P4tc—provably-secure yet practical privacy-preserving toll collection," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 62–152, 2020.
[13] C. Anglès-Tafalla, S. Ricci, P. Dzurenda, J. Hajny, J. Castellà-Roca, and A. Viejo, "Decentralized privacy-preserving access for low emission zones," in *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT,*, INSTICC. SciTePress, 2019, pp. 485–491.
[14] C. Anglès-Tafalla, J. Castellà-Roca, and A. Viejo, "Privacy-preserving and secure decentralized access control system for low emission zones," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019.
[15] R. S. Yokoyama, B. Y. Kimura, L. A. Villas, and E. D. Moreira, "Measuring distances with rssi from vehicular short-range communications," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 2015, pp. 100–107.

# Generalized partially bent functions and cocyclic Butson matrices

José Andrés Armario
Depto de Matemática Aplicada I,
Universidad de Sevilla, Spain
armario@us.es

Ronan Egan
School of Mathematical Sciences,
Dublin City University, Ireland
ronan.egan@dcu.ie

Dane Flannery
School of Mathematical and Statistical Sciences,
University of Galway, Ireland
dane.flannery@nuigalway.ie

***Abstract*—A bent function is a Boolean function which is as far from being linear as possible. Highly non-linear functions are extremely useful for symmetric encryption, offering a robust defence against linear cryptanalysis.**

**Equivalences between generalized bent functions, group invariant Butson Hadamard matrices, and abelian splitting relative difference sets are known. Here, we establish a broader network of equivalences, including various extra objects, by considering Butson matrices that are cocyclic rather than strictly group invariant.**

***Index Terms*—Generalized partially bent functions, perfect arrays, cocyclic Butson matrices**

## I. Introduction

The theory of Boolean functions is an important area of research in discrete mathematics, with applications to cryptography and coding theory. The properties of confusion and diffusion, due to Claude Shannon, are fundamental concepts for achieving security in cryptosystems. Diffusion is related to the degree to which the influence of a single input plaintext bit is spread throughout the resulting ciphertext. Confusion deals with the complexity of the relationship between the secret key and ciphertext.

Boolean functions with high nonlinearity can be used to provide confusion in block encryption algorithms [15]. Nonlinearity is the minimum number of bits which must change in the truth table of a Boolean function for it to become affine.

The Walsh transform is the most important mathematical tool for analyzing cryptographic properties of Boolean functions. An understanding of the Walsh transform of a Boolean function uniquely determines the function; therefore, working solely with the Walsh transform is possible.

An extreme situation is reached when the Walsh transform takes only one value up to sign. In this case, the Boolean function is called *bent*. If the Walsh transform takes only one non-zero absolute value, and possibly the value 0, then it is *plateaued*. From a cryptographic point of view, bent functions $f\colon \mathbb{Z}_2^m \to \mathbb{Z}_2$ have two main sources of interest: firstly, their derivatives are balanced, i.e., take values 0 and 1 equally often; thus, any addition of a non-zero vector to the input for $f$ produces $2^{m-1}$ changes among the $2^m$ outputs. This is connected to the differential attack on block ciphers. Secondly, the Hamming distance between $f$ and the set of affine Boolean functions attains the maximum value $2^{m-1}-2^{\frac{m}{2}-1}$ ($n$ even); this has a direct relationship with the fast correlation attack on stream ciphers and the linear attack on block ciphers. However, bent functions are not balanced (another desirable cryptographic property). On the other hand, plateaued functions can be balanced and some have large nonlinearity, which provides protection against fast correlation attacks when they are used as combiners or filters in stream ciphers, and protection against linear cryptanalysis. They can also possess other desirable cryptographic characteristics such as resiliency, low additive autocorrelation, and satisfaction of propagation criteria. Yet they cannot have high algebraic degree. This makes them weak against fast algebraic attacks on stream ciphers. Trade-offs between all the cryptographic criteria must be found in order to get secure ciphers (see [5] and the references therein).

The notion of bentness admits various generalizations. We use the one in Schmidt's survey [13], where equivalences with other objects, such as group invariant Butson matrices, are described. It is well-known that group invariant matrices constitute the base case of cocyclic matrices. By considering Butson matrices that are cocyclic rather than strictly group invariant, we establish a broader network of equivalences. For certain parameters, these equivalences have those in [13] as special cases.

Many of the results of this paper were presented at the *6th International Workshop on Boolean Functions and their Applications* held in Rosendal, Norway / Hybrid, September 2021 (https://boolean.w.uib.no/bfa-2021/). We refer to [1] for proofs of the results in this paper.

## II. Preliminaries

We adopt the following definition from [13]. For positive integers $q, m, h$, and $\zeta_k$ the complex $k^{\text{th}}$ root of unity $\exp(2\pi\sqrt{-1}/k)$, a map $f\colon \mathbb{Z}_q^m \to \mathbb{Z}_h$ is a *generalized bent function (GBF)* if

$$\Big|\sum_{x\in\mathbb{Z}_q^m} \zeta_h^{f(x)}\zeta_q^{-w\cdot x}\Big|^2 = q^m \quad \forall\, w\in\mathbb{Z}_q^m,$$

where $|z|$ as usual denotes the modulus of $z\in\mathbb{C}$ and $\mathbb{Z}_t$ is the cyclic group of order $t$. Thus, a GBF for $q=h=2$ and even $m$ is a bent function. For $h=q$, Kumar, Scholtz, and Welch [10] prove that GBFs exist if $m$ is even or $q\not\equiv 2 \mod 4$. However, no GBF with $h=q$, $m$ odd, and $q\equiv 2 \mod 4$ is known [11, p. 2]. A further generalization is relevant to this paper. If the values of

$$\Big|\sum_{x\in\mathbb{Z}_q^m} \zeta_h^{f(x)}\zeta_q^{-w\cdot x}\Big|^2$$

as $w$ ranges over $\mathbb{Z}_q^m$ lie in $\{0,\alpha\}$ for a single non-zero $\alpha$, then $f$ is called a *generalized plateaued function*. Mesnager,

Tang, and Qi [12] discuss such functions under the conditions that $q$ is prime and $h = q^k$ for some positive integer $k$. They say that $f$ is an *$s$-generalized plateaued function* when $\alpha$ has the form $q^{m+s}$.

We examine the role of GBFs and generalized plateaued functions within cocyclic design theory [4], [7]. Some requisite definitions follow. Let $G$ and $U$ be finite groups, with $U$ abelian. A map $\psi\colon G \times G \to U$ such that

$$\psi(a,b)\psi(ab,c) = \psi(a,bc)\psi(b,c) \quad \forall\, a,b,c \in G$$

is a *cocycle* (*over $G$, with coefficients in $U$*). We assume that $\psi$ is *normalized*, i.e., $\psi(1,1) = 1$. For any (normalized) map $\phi\colon G \to U$, the cocycle $\partial\phi$ defined by $\partial\phi(a,b) = \phi(a)^{-1}\phi(b)^{-1}\phi(ab)$ is a *coboundary*. The set of cocycles $\psi\colon G \times G \to U$ equipped with pointwise multiplication is an abelian group, $Z^2(G,U)$. Factoring out $Z^2(G,U)$ by the subgroup $B^2(G,U)$ of coboundaries gives the *second cohomology group* $H^2(G,U)$. The elements of $H^2(G,U)$, namely cosets of $B^2(G,U)$, are *cohomology classes*. Each cocycle $\psi \in Z^2(G,U)$ is displayed as a *cocyclic matrix* $M_\psi$. That is, under some indexing of rows and columns by the elements of $G$, $M_\psi$ has entry $\psi(a,b)$ in position $(a,b)$. The main case treated in this paper is $G = \mathbb{Z}_{s_1} \times \cdots \times \mathbb{Z}_{s_m}$ and $U = \langle\zeta_h\rangle \cong \mathbb{Z}_h$, where $\langle\zeta_h\rangle$ is the (multiplicative) group $\{\zeta_h^i \mid 0 \le i \le h-1\}$ generated by $\zeta_h$.

Denote the set of $n \times n$ matrices with entries in a set $S$ by $\mathcal{M}_n(S)$. A matrix $M \in \mathcal{M}_n(\langle\zeta_k\rangle)$ is a *Butson (Hadamard) matrix* if $MM^* = nI_n$ where $I_n$ is the $n \times n$ identity matrix and $M^*$ is the complex conjugate transpose of $M$. We write $\mathrm{BH}(n,k)$ for the (possibly empty) set of all Butson matrices in $\mathcal{M}_n(\langle\zeta_k\rangle)$. For example, at every order $n$ we have the Fourier matrix $\left[\zeta_n^{(i-1)(j-1)}\right]_{i,j=1}^{n} \in \mathrm{BH}(n,n)$. Hadamard matrices of order $n$ are the elements of $\mathrm{BH}(n,2)$.

We say that $H, H' \in \mathcal{M}_n(\langle\zeta_k\rangle)$ are *equivalent* if $PHQ^* = H'$ for monomial matrices $P, Q \in \mathcal{M}_n(\langle\zeta_k\rangle \cup \{0\})$. This equivalence relation induces a partition of $\mathrm{BH}(n,k)$.

Our interest is in cocyclic Butson matrices. Let $G$ be a group of order $n$. A cocycle $\psi \in Z^2(G, \langle\zeta_k\rangle)$ such that $M_\psi \in \mathrm{BH}(n,k)$ is *orthogonal*. In particular, group invariant Butson matrices are cocyclic. Here the orthogonal cocycles involved are coboundaries, as we now explain. A matrix $X \in \mathcal{M}_n(U)$ is *group invariant*, over $G$, if $X = [x_{a,b}]_{a,b\in G}$ and $x_{ac,bc} = x_{a,b}$ for all $a,b,c \in G$. Such a group invariant matrix $X$ is equivalent to a *group-developed matrix* $[\chi(ab)]_{a,b\in G}$ for some map $\chi\colon G \to U$; in turn $[\chi(ab)]$ is equivalent to $M_{\partial\chi}$.

Cocyclic designs give rise to relative difference sets, and vice versa [4, Sections 10.4, 15.4]. Let $E$ be a group with a normal subgroup $N$, where $|N| = n$ and $|E : N| = v$. A *$(v,n,k,\lambda)$-relative difference set in $E$ relative to $N$* (the *forbidden subgroup*) is a $k$-subset $R$ of a transversal for $N$ in $E$ such that $|R \cap xR| = \lambda$ for all $x \in E \setminus N$. We call $R$ *abelian* if $E$ is abelian, and *splitting* if $N$ is a direct factor of $E$.

The final piece of background concerns arrays. Let $\mathbf{s} = (s_1, \ldots, s_m)$ be an $m$-tuple of integers $s_i > 1$, and let $G = \mathbb{Z}_{s_1} \times \cdots \times \mathbb{Z}_{s_m}$. A *$h$-ary $\mathbf{s}$-array* is just a set map $\phi\colon G \to \mathbb{Z}_h$. When $h = 2$, the array $\phi$ is *binary*. For $w \in G$, we define the *periodic autocorrelation of $\phi$ at shift $w$*, denoted $AC_\phi(w)$, by

$$AC_\phi(w) = \sum_{g\in G} \zeta_h^{\phi(g)}\zeta_h^{-\phi(g+w)}.$$

If $AC_\phi(w) = 0$ for all $w \neq 0$, then $\phi$ is *perfect*.

Now we can state the fundamental motivating result, extracted mostly from [13].

*Theorem 1:* Let $f\colon \mathbb{Z}_q^m \to \mathbb{Z}_h$ be a map. The following are equivalent:

(1) $f$ is a GBF;
(2) $\left[\zeta_h^{f(x-y)}\right]_{x,y\in\mathbb{Z}_q^m} \in \mathrm{BH}(q^m,h)$ is equivalent to a coboundary matrix $M_{\partial f}$;
(3) $f$ is a perfect $h$-ary $(q,\ldots,q)$-array.

If additionally $h$ is prime and divides $q^m$, then (1)–(3) are equivalent to

(4) $\{(f(x),x) \mid x \in \mathbb{Z}_q^m\}$ is a splitting $(q^m, h, q^m, q^m/h)$-relative difference set in $\mathbb{Z}_h \times \mathbb{Z}_q^m$.

We investigate the effect on the equivalences of Theorem 1 when non-coboundary cocyclic Butson matrices, generalized perfect arrays, and non-splitting abelian relative difference sets are considered in (2), (3), (4), respectively. Toward this end, we need some more specialized material, to be presented in the next section (which contains our main result).

## III. Equivalences between arrays, bent functions, and associated combinatorial objects

For $h = 2$, the equivalence between non-splitting abelian relative difference sets and $h$-ary arrays was discovered in [9]; an underlying orthogonal cocycle was identified in [8], thereby providing a bridge to the theory of cocyclic Hadamard matrices. The notion of a generalized perfect binary array (GPBA) was the main tool used. We extend this notion for arbitrary $h$, and thereafter study its connection to bent functions.

*Definition 1: Let $\phi\colon G \to \mathbb{Z}_h$ be an $\mathbf{s}$-array, where $\mathbf{s} = (s_1,\ldots,s_m)$ and $G = \mathbb{Z}_{s_1} \times \cdots \times \mathbb{Z}_{s_m}$. Let $\mathbf{z} = (z_1,\ldots,z_m) \in \{0,1\}^m$. The* expansion of $\phi$ of type $\mathbf{z}$ *is the map $\phi'$ from $E := \mathbb{Z}_{(z_1(h-1)+1)s_1} \times \cdots \times \mathbb{Z}_{(z_m(h-1)+1)s_m}$ to $\mathbb{Z}_h$ defined by*

$$\phi'\colon (g_1,\ldots,g_m) \mapsto \phi(a) + b \mod h,$$

*where $b = \sum_{i=1}^m \lfloor \frac{g_i}{s_i} \rfloor$ and $a \equiv g \bmod \mathbf{s} = (g_1 \bmod s_1, \ldots, g_m \bmod s_m)$.*

We isolate the following subgroups of the extension group $E$ in Definition 1:

$$L = \{(g_1,\ldots,g_m) \in E \mid g_i = y_i s_i \text{ with } 0 \le y_i < h \text{ if } z_i = 1, \text{ and } y_i = 0 \text{ if } z_i = 0\},$$
$$K = \{(g_1,\ldots,g_m) \in L \mid \textstyle\sum_i (g_i/s_i) \equiv 0 \bmod h\}.$$

Note that

- $L \cong \mathbb{Z}_h^n$ where $n = \mathrm{wt}(\mathbf{z}) = \sum_i z_i$.
- If $\mathbf{z} \neq \mathbf{0}$ then $L/K \cong \mathbb{Z}_h$; e.g., if $z_1 = 1$ then $L/K = \langle (s_1, 0, \ldots, 0) + K\rangle$.

*Lemma 1:* Let $\phi$ be a $h$-ary $(s_1,\ldots,s_m)$-array with expansion $\phi'\colon E \to \mathbb{Z}_h$. For $L \le E$ as above, if $e \in E$ and $g = (g_1,\ldots,g_m) \in L$, then $\phi'(e+g) \equiv \phi'(e) + b \mod h$ where $b = \sum_i g_i/s_i$.

*Corollary 1:* Assume the hypotheses and notation of Lemma *1*. Then

$$AC_{\phi'}(g) = \zeta_h^{-b}\, AC_{\phi'}(0) = \zeta_h^{-b} \prod_{i=1}^{m} (z_i(h-1)+1)s_i \quad \forall\, g \in L.$$

*Definition 2: A $h$-ary* $\mathbf{s}$*-array $\phi$ with expansion $\phi'\colon E \to \mathbb{Z}_h$ of type $\mathbf{z}$ is* generalized perfect *if $AC_{\phi'}(g) = 0$ for all $g \in E \setminus L$; in short, we say that $\phi$ is a* $\mathrm{GPhA}(\mathbf{s})$ *of type $\mathbf{z}$.*

*Remark 1: A GPhA*$(\mathbf{s})$ *of type $\mathbf{0}$ is exactly a perfect $h$-ary* $\mathbf{s}$*-array.*

*Remark 2: The definitions above reduce to the ones in [9] when $h = 2$.*

*Definition 3 (Cf. [16, Definition 2.2]): A* generalized partially bent function *(GPBF) is a map $f\colon \mathbb{Z}_q^m \to \mathbb{Z}_h$ such that $|AC_f(x)| = 0$ or $q^m$ for all $x \in \mathbb{Z}_q^m$.*

Obviously, the expansion of a $\mathrm{GPhA}(q^m)$ of type $\mathbf{1} := (1, 1, \ldots, 1)$ is a GPBF. However, the converse is not true in general.

*Example 1: Let $\phi\colon \mathbb{Z}_2^2 \to \mathbb{Z}_2$ be the map defined by $\phi(0,1) = 1$ and $\phi(0,0) = \phi(1,0) = \phi(1,1) = 0$. Its expansion of type $\mathbf{1}$ is a GPBF, but $\phi$ is not a* $\mathrm{GP2A}(2,2)$ *of type $\mathbf{1}$.*

Define $\mu_{\mathbf{z}} \in Z^2(G, \langle\zeta_h\rangle)$ by

$$\mu_{\mathbf{z}}(x, y) = \prod_{z_i=1} \gamma_{s_i}(x_i, y_i)$$

with $\gamma_m(j,k) = \zeta_h^{\lfloor (j+k)/m \rfloor}$, evaluating the exponent as an ordinary integer.

Now we can state our main result.

*Theorem 2:* Let $h$ be a prime divisor of $q$, and let $\phi$ be an array $\mathbb{Z}_q^m \to \mathbb{Z}_h$, with expansion $\phi'$ of type $\mathbf{z} \neq \mathbf{0}$.

(a) The following are equivalent:
  (i) $\mu_{\mathbf{z}}\partial\phi$ is orthogonal, i.e., $M_{\mu_{\mathbf{z}}\partial\phi} \in \mathrm{BH}(q^m, h)$;
  (ii) $\phi$ is a $GPhA(q^m)$ of type $\mathbf{z}$;
  (iii) $\{g + K \in E/K \mid \phi'(g) = 0\}$ is a non-splitting $(q^m, h, q^m, q^m/h)$-relative difference set in $E/K$ with forbidden subgroup $L/K$.

(b) If $\mathbf{z} = \mathbf{1}$ then (ii) is equivalent to
  (iv) $\phi'$ is a generalized plateaued function, i.e.,

$$\Big|\sum_{x \in \mathbb{Z}_{hq}^m} \zeta_h^{\phi'(x)} \zeta_{hq}^{-v\cdot x}\Big|^2 = \begin{cases} (h^2q)^m & v \in \mathcal{F} \\ 0 & \text{otherwise,} \end{cases}$$

  where $\mathcal{F} = \{v \in \mathbb{Z}_{hq}^m \mid v \equiv \mathbf{1} \mod h\}$.

(c) Let $h = q$ and $\mathbf{z} = \mathbf{1}$. Suppose that, for all $y \in \mathbb{Z}_h^m \setminus \{\mathbf{0}\}$ with $\sum y_i \equiv 0 \mod h$, there exists $x \in \mathbb{Z}_h^m$ satisfying

$$\phi(x+y) + \sum \left\lfloor \frac{x_i + y_i}{h} \right\rfloor \neq \phi(x) + \phi(y).$$

  Then (iv) is equivalent to
  (v) $\phi'$ is a GPBF.

*Remark 3:* In (c) of Theorem 2, a sufficient condition for equivalence between (iv) and (v) is provided; this condition is satisfied in the examples of the next section.

*Remark 4:* Theorem 2 gives us a procedure to compute GPBFs by means of cocyclic Butson matrices $\mathrm{BH}(q^m, q)$. The hard part of this procedure is finding a coboundary $\partial\phi$ such that $\mu_{\mathbf{z}}\partial\phi$ is orthogonal.

## IV. Examples

*Example 2: Put $A_0 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ and $A_1 = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$. Let $\phi$ be the map on $\mathbb{Z}_2^3$ with layers $A_0$ and $A_1$; here $A_i$ denotes the layer on $\{i\} \times \mathbb{Z}_2 \times \mathbb{Z}_2$ and $\phi(i,j,k) = A_i(j,k)$. Then $\phi$ is a GPBA$(2,2,2)$ of type $\mathbf{1}$. Its orthogonal cocycle is $\mu_{\mathbf{z}}\partial_2\partial_3\partial_4\partial_6$, where $\partial_i$ is the coboundary for the multiplicative Kronecker delta $\phi_i$ of $\alpha_i$, with $\alpha_0 = (0,0,0)$, $\alpha_1 = (0,0,1)$, and so on. Labeling rows and columns with the elements of $\mathbb{Z}_2^3 = \{\alpha_0, \ldots, \alpha_7\}$ in this ordering, we display the cocyclic Hadamard matrix $M_{\mu_{\mathbf{z}}\partial\phi}$ as a Hadamard (i.e., componentwise) product $M_{\mu_{\mathbf{z}}} \circ M_{\partial\phi}$ in logarithmic form:*

$$\begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&1&0&0&1&1&0 \\ 0&0&0&0&1&1&1&1 \\ 0&1&0&1&1&0&1&0 \\ 0&0&1&1&1&1&0&0 \\ 0&1&1&0&1&0&0&1 \end{bmatrix} \circ \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&1&0&1&0 \\ 0&1&1&0&1&0&0&1 \\ 0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&1 \\ 0&1&0&1&1&0&1&0 \end{bmatrix}$$

$$= \begin{bmatrix} 0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&0 \\ 0&1&1&0&1&0&0&1 \\ 0&0&0&0&1&1&1&1 \\ 0&0&1&1&1&1&0&0 \\ 0&1&0&1&1&0&1&0 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1 \end{bmatrix}.$$

*The expansion $\phi'\colon \mathbb{Z}_4^3 \to \mathbb{Z}_2$ is defined by the layers on $\{i\} \times \mathbb{Z}_4 \times \mathbb{Z}_4$, $0 \le i \le 3$, by*

$$B_i = \begin{cases} \begin{bmatrix} A_0 & A_0 \oplus J \\ A_0 \oplus J & A_0 \end{bmatrix} & i = 0, 2 \\ \\ \begin{bmatrix} A_1 \oplus J & A_1 \\ A_1 & A_1 \oplus J \end{bmatrix} & i = 1, 3, \end{cases}$$

*where $J$ is the all* 1*s matrix. That is,*

$$B_0 = B_2 = \begin{bmatrix} 0&1&1&0 \\ 1&1&0&0 \\ 1&0&0&1 \\ 0&0&1&1 \end{bmatrix},$$

$$B_1 = B_3 = \begin{bmatrix} 1&0&0&1 \\ 1&1&0&0 \\ 0&1&1&0 \\ 0&0&1&1 \end{bmatrix}.$$

*We have $L = \{(0,0,0), (0,0,2), (0,2,0), (0,2,2), (2,0,0), (2,0,2), (2,2,0), (2,2,2)\}$,*

$$AC_{\phi'}(v) = \begin{cases} (-1)^{\mathrm{wt}(v)}\, 64 & v \in L \\ 0 & v \notin L, \end{cases}$$

*$\mathcal{F} = \{(1,1,1), (1,1,3), (1,3,1), (1,3,3), (3,1,1), (3,1,3), (3,3,1), (3,3,3)\}$, and*

$$\Big|\sum_{x \in \mathbb{Z}_4^3} \zeta_2^{\phi'(x)} \zeta_4^{-v\cdot x}\Big|^2 = \begin{cases} 512 & v \in \mathcal{F} \\ 0 & v \notin \mathcal{F}. \end{cases}$$

*Therefore $\phi'$ is a GPBF.*

*Example 3: The map* $\phi = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$ *on* $\mathbb{Z}_3^2$ *is a GP3A*$(3,3)$ *of type* **1**. *Its orthogonal cocycle is* $\mu_{\mathbf{z}}\partial\phi$. *Labeling the rows and columns with the elements of* $\mathbb{Z}_3^2 = \{\alpha_0 = (0,0), \alpha_1 = (0,1), \alpha_2 = (0,2), \ldots, \alpha_8 = (2,2)\}$ *in this ordering, we display the cocyclic Butson matrix* $M_{\mu_{\mathbf{z}}\partial\phi}$:

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&1&0&0&1 \\ 0&1&1&0&1&1&0&1&1 \\ 0&0&0&0&0&0&1&1&1 \\ 0&0&1&0&0&1&1&1&2 \\ 0&1&1&0&1&1&1&2&2 \\ 0&0&0&1&1&1&1&1&1 \\ 0&0&1&1&1&2&1&1&2 \\ 0&1&1&1&2&2&1&2&2 \end{bmatrix} \circ \begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&2&0&0&2&1 \\ 0&0&0&0&2&1&2&0&1 \\ 0&1&0&2&1&1&1&1&2 \\ 0&2&2&1&2&1&0&0&1 \\ 0&0&1&1&1&2&1&1&2 \\ 0&0&2&1&0&1&2&0&0 \\ 0&2&0&1&2&1&0&2&0 \\ 0&1&1&2&1&2&0&0&2 \end{bmatrix}$$

$$= \begin{bmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&2&1&0&2&2 \\ 0&1&1&0&0&2&2&1&2 \\ 0&1&0&2&1&1&2&2&0 \\ 0&2&0&1&2&2&1&1&0 \\ 0&1&2&1&2&0&2&0&1 \\ 0&0&2&2&1&2&0&1&1 \\ 0&2&1&2&1&0&1&0&2 \\ 0&2&2&0&0&1&1&2&1 \end{bmatrix}.$$

*The expansion* $\phi' : \mathbb{Z}_9^2 \to \mathbb{Z}_3$ *is defined by*

$$\begin{bmatrix} 0&0&0&1&1&1&2&2&2 \\ 0&1&0&1&2&1&2&0&2 \\ 2&2&1&0&0&2&1&1&0 \\ 1&1&1&2&2&2&0&0&0 \\ 1&2&1&2&0&2&0&1&0 \\ 0&0&2&1&1&0&2&2&1 \\ 2&2&2&0&0&0&1&1&1 \\ 2&0&2&0&1&0&1&2&1 \\ 1&1&0&2&2&1&0&0&2 \end{bmatrix},$$

*with* $L = \{(0,0),(0,3),(0,6),(3,0),(3,3),(3,6),(6,0),(6,3),(6,6)\}$,

$$AC_{\phi'}((v_1,v_2)) = \begin{cases} 81\,\zeta_3^{-(v_1+v_2)/3} & (v_1,v_2) \in L \\ 0 & (v_1,v_2) \notin L, \end{cases}$$

$\mathcal{F} = \{(1,1),(1,4),(1,7),(4,1),(4,4),(4,7),(7,1),(7,4),(7,7)\}$, *and*

$$\Big|\sum_{x\in\mathbb{Z}_9^2} \zeta_3^{\phi'(x)}\zeta_9^{-v\cdot x}\Big|^2 = \begin{cases} 729 & v \in \mathcal{F} \\ 0 & v \notin \mathcal{F}. \end{cases}$$

*Therefore* $\phi'$ *is a generalized partially bent function. Also* $\phi'$ *is a* 4*-generalized plateaued function.*

## V. Conclusions

Bent functions and plateaued functions have important applications in cryptography and coding theory. Their generalizations have attracted attention recently in the literature [12], [14]. Knowledge of bent and plateaued functions in general contexts has subsequently increased.

The theory of orthogonal cocycles and their associated combinatorial objects has been explored during the past few decades (see, e.g., [4], [7]). The aim of our project is to enhance understanding of these objects using approaches that impinge on the cryptographic applications. As an example of progress along these lines, we mention new results concerning optimal sequences, reported in [2], [3], [6].

In the current paper, we have examined the role of GBFs and generalized plateaued functions within cocyclic design theory. Our main result is Theorem 2: this shows the effect on the established equivalences of Theorem 1 when non-coboundary cocyclic Butson matrices, generalized perfect arrays, and non-splitting abelian relative difference sets are considered in (2), (3), (4) of Theorem 1, respectively.

## Acknowledgements

This research was partially supported by project FQM-016 funded by JJAA (Spain).

## References

[1] J. A. Armario, R. Egan, and D. L. Flannery: "Generalized partially bent functions, generalized perfect arrays and cocyclic Butson matrices". https://doi.org/10.48550/arXiv.2207.05735

[2] J. A. Armario and D. L. Flannery: "Almost supplementary difference sets and quaternary sequences with optimal autocorrelation", *Cryptogr. Commun.*, vol. 12, pp. 757–768, 2020.

[3] J. A. Armario and D. L. Flannery: "Quasi-orthogonal cocycles, optimal sequences and a conjecture of Littlewood", *J. Algebraic Combin.*, vol. 55, pp. 15–25, 2022.

[4] W. de Launey and D. L. Flannery: *Algebraic design theory*. Math. Surveys. Monogr. 175, American Mathematical Society, Providence, RI, 2011.

[5] C. Carlet and S. Mesnager: "Four decades of research on bent functions", *Des. Codes Cryptogr.*, vol. 78, no. 1, pp. 5–50, 2016.

[6] R. Egan: "Generalizing pairs of complementary sequences and a construction of combinatorial structures", *Discrete Math.*, vol. 343, 111795, 2020.

[7] K. J. Horadam: *Hadamard matrices and their applications*, Princeton University Press, Princeton, NJ, 2007.

[8] G. Hughes: "Non-splitting abelian $(4t, 2, 4t, 2t)$ relative difference sets and Hadamard cocycles". *Europ. J. Combin.*, vol. 21, no. 3, pp. 323–331, 2000.

[9] J. Jedwab: "Generalized perfect arrays and Menon difference sets", *Des. Codes Cryptogr.*, vol. 2, no. 1, pp. 19–68, 1992.

[10] P. V. Kumar, R. A. Scholtz, and L. R. Welch, " Generalised bent functions and their properties", *J. Combin. Theory Ser. A*, vol. 40, pp. 90–107, 1985.

[11] K. H. Leung and B. Schmidt, "Nonexistence results on generalized bent functions $\mathbb{Z}_q^m \to \mathbb{Z}_q$ with odd $m$ and $q \equiv 2 \pmod 4$", *J. Combin. Theory Ser. A*, vol. 163, pp. 1–33, 2019.

[12] S. Mesnager, C. Tang, and Y. Qi: "Generalized plateaued functions and admissible (plateaued) functions", *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6139–6148, 2017.

[13] B. Schmidt: "A survey of group invariant Butson matrices and their relation to generalized bent functions and various other objects", *Radon Ser. Comput. Appl. Math.*, vol. 23, pp. 241–251, 2019.

[14] K.U. Schmidt: "$\mathbb{Z}_4$-valued quadratic forms and quaternary sequence families, *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5803–5810, 2009.

[15] N. Tokareva: "Bent functions: results and applications to cryptography", London, UK : Elsevier Science, 2015.

[16] X. Wang and J. Zhou: "Generalized partially bent functions", In: *Future Generation Communication and Networking (FGCN 2007)*, vol. 1, pp. 16–21, IEEE, 2007.

# Analysis and Improvements of the Sender Keys Protocol for Group Messaging

David Balbás
IMDEA Software Institute,
Universidad Politécnica de Madrid
Spain
david.balbas@imdea.org

Daniel Collins
EPFL
Switzerland
daniel.collins@epfl.ch

Phillip Gajland
Max Planck Institute for Security & Privacy,
Ruhr-University Bochum
Germany
phillip.gajland@mpi-sp.org

***Abstract*—Messaging between two parties and in the group setting has enjoyed widespread attention both in practice, and, more recently, from the cryptographic community. One of the main challenges in the area is constructing secure (end-to-end encrypted) and efficient messaging protocols for group conversations. The popular messaging applications WhatsApp and Signal utilise a protocol in which, instead of sharing a single group key, members have individual *sender keys*, which are shared with all other group members. The Sender Keys protocol is claimed to offer forward security guarantees. However, despite its broad adoption in practice, it has never been studied formally in the cryptographic literature.**

**In this paper we present the first analysis of the Sender Keys protocol along with some prospective improvements. To this end, we introduce a new cryptographic primitive, develop a game-based security model, present a security analysis in the passive and active settings, and propose several improvements to the protocol.**

***Index Terms*—Secure Messaging, Group Messaging, Signal, WhatsApp, Sender Keys.**

## I. Introduction

Messaging applications such as WhatsApp, Facebook Messenger, Signal and Telegram have enjoyed widespread adoption and form an integral part of communications for billions of people. All of the aforementioned applications rely, to a varying degree, on cryptography to provide diverse forms of authenticity and secrecy.

Among end-to-end encrypted messaging solutions (this excludes Telegram and Facebook Messenger by default, among others), there exist diverse cryptographic solutions. For two-party messaging, Signal's Double Ratchet Protocol [1] is the most popular choice in practice, and many solutions also exist in the cryptographic literature [2], [3], [4], [5], [6]. For group messaging, the naive solution, as used by Signal Messenger for small groups, of adopting Double Ratchet sessions among every pair of group members does not scale well. Thus, recent work such as the Messaging Layer Security (MLS) standardization effort [7] aims to construct secure group messaging protocols where the complexity of group operations (adding and removing members, updating key material) is sublinear in the group size [8], [9], [10], [11], [12].

Nevertheless, the popular messaging applications WhatsApp and Signal (for large groups) use a protocol for group messaging [13], [14] that does not involve sharing a unique group key that evolves over time. This differs from MLS, and from the group key agreement abstraction followed there [9], [10], [11]. This protocol, called Sender Keys, has not been formally studied in the literature despite its widespread adoption.

### A. Secure Group Messaging

Two standard security notions prevail in the literature both for two-party and group messaging. The first is *forward security* (FS), which protects the confidentiality of past messages in the event of a key exposure and can be achieved using just symmetric cryptography (for example by iteratively hashing symmetric keys). The second is *post-compromise security* (PCS), which ensures that security can be restored after a key exposure in certain adversarial settings [15], typically when the adversary is passive for some period of time. FS- and PCS-oriented key evolution mechanisms are commonly known as *ratcheting*.

Both properties apply to the confidentiality and authenticity of sent messages and can be captured formally in a security game. There exist different formalisations of security in the literature, but most of them model an adversarial Delivery Service (DS), the entity responsible for delivering messages between participants via the communication channel. The adversary (modelling the DS) can act as an eavesdropper (with extended yet limited capabilities) as in [9], as a semi-active adversary which can schedule messages arbitrarily [10], or as an active adversary that can inject messages [11], [16]. In many protocols, including Sender Keys and MLS, the DS relies mainly on some centralized infrastructure (the *central server* hereafter).

Some messaging protocols also require additional infrastructure to deal with user authentication or security. This may include Public Key Infrastructure (PKI), or, in the case of Sender Keys, secure two-party messaging channels established between each pair of users. Achieving security in multiple groups simultaneously is outside the scope of this work, and requires additional precautions detailed in [17].

### B. Sender Keys

In a Sender Keys group $\boldsymbol{G}$, every user $\mathsf{ID} \in \boldsymbol{G}$ owns a so-called *sender key* which is shared with all group members. A sender key is a tuple $\mathsf{send\text{-}k} = (\mathsf{spk}, \mathsf{ck})$, where $\mathsf{spk}$ is a public signature key (with a private counterpart $\mathsf{ssk}$), and $\mathsf{ck}$ is a symmetric *chain key*. Every time a user $\mathsf{ID}$ sends a message $m$ to the group, $\mathsf{ID}$ encrypts $m$ using a *message key* $\mathsf{mk}$ that is deterministically derived from its chain key $\mathsf{ck}$. Upon message reception, group members also derive $\mathsf{mk}$ to

decrypt the message. Messages are authenticated by appending the sender's signature.

Forward security is provided by using a fresh message key for every message; every time a message is sent, the chain key is hashed forward using a key derivation function. In other words, chain keys are symmetrically ratcheted.

The protocol also requires that there exist confidential and authenticated two-party communication channels between every pair of users. These are used for sharing sender keys in the event of parties being added or removed from the group.

### C. Contributions

The main scientific contributions of our paper are the following:

- We introduce a new cryptographic primitive, Group Messenger (GM), which is suitable for messaging protocols like Sender Keys that are not necessarily based on group key agreement.
- We formally describe Sender Keys, based on a code analysis of Signal's source code [14], and WhatsApp's security white paper [13].
- We present a security model for (single-group) Group Messenger. We do so via a security game that considers an active adversary who can interact with several oracles. The game is parametrised by a cleanness predicate that captures forward-secure group messaging.
- We carry out a security analysis for passive and active adversaries and detail prospective fixes and improvements to the Sender Keys protocol.

We note that this is a preliminary and shortened version of our work.

## II. Primitive Syntax

Unless otherwise stated, all algorithms are probabilistic, and $(x_1, \dots) \xleftarrow{\$} \mathcal{A}(y_1, \dots)$ is used to denote that $\mathcal{A}$ returns $(x_1, \dots)$ when run on input $(y_1, \dots)$. Blank values are represented by $\perp$. We denote the security parameter by $\lambda$ and its unary representation by $1^\lambda$. We also define the state $\gamma$ of a user $\mathsf{ID}$ as the data required by $\mathsf{ID}$ for protocol execution, including message records, group-related variables, and cryptographic material.

We introduce a cryptographic primitive that we call *Group Messenger* $\mathsf{GM} := (\mathsf{Init}, \mathsf{Send}, \mathsf{Recv}, \mathsf{Exec}, \mathsf{Proc})$, similar to other group messaging abstractions such as Continuous Group Key Agreement (CGKA) [9]. In contrast to CGKA, our primitive does not model key agreement (as this does not neatly capture the Sender Keys protocol), but rather sending and receiving messages. The syntax is as follows.

- $\gamma \xleftarrow{\$} \mathsf{Init}(1^\lambda, \mathsf{ID})$: Given the security parameter $1^\lambda$ and a user identity $\mathsf{ID}$, the probabilistic initialisation algorithm returns an initial state $\gamma$.
- $(C, \gamma') \xleftarrow{\$} \mathsf{Send}(m, \gamma)$: Given a message $m$, and a state $\gamma$, the probabilistic sending algorithm returns a ciphertext $C$ and a new state $\gamma'$.
- $(m, \gamma') \leftarrow \mathsf{Recv}(C, \gamma)$: Given a ciphertext $C$, and a state $\gamma$, the deterministic receiving algorithm returns a message $m$ and a new state $\gamma'$.
- $(C, \gamma') \xleftarrow{\$} \mathsf{Exec}(\mathtt{cmd}, \boldsymbol{IDs}, \gamma)$: Given a command $\mathtt{cmd} \in \{\mathtt{crt}, \mathtt{add}, \mathtt{rem}\}$, a list of user identities $\boldsymbol{IDs}$, and a state $\gamma$, the probabilistic execution algorithm returns a ciphertext $C$ and a new state $\gamma'$.
- $\gamma' \leftarrow \mathsf{Proc}(C, \gamma)$: Given a ciphertext $C$, and a state $\gamma$, the deterministic processing algorithm returns a new state $\gamma'$.

Note that there are separate algorithms for sending / receiving application messages and for executing / processing changes to the group. Furthermore, should two-party protocols be required under the hood to implement the group primitive (this is not the case for CGKAs such as TreeKEM [9]), then these are outside the scope of this definition.

## III. Protocol Description

In this section, we introduce a formal description of the Sender Keys protocol in accordance with our Group Messenger syntax.

### A. Protocol Setup

*1) Central server:* The Delivery Service (DS) relies on a central server which provides total ordering of messages and authenticates users initially (i.e. it acts as a PKI). In practice, the DS is also responsible for managing two-party channels.

*2) Two-party channels:* The protocol assumes that there exist authenticated and secure two-party communication channels (for example using Signal's Double Ratchet protocol [1] as done in WhatsApp [13]) between every pair of protocol users. This assumption can be realized via the Delivery Service and asynchronous, PKI-aided key-exchange mechanisms such as X3DH [18].

*3) Primitives:* The protocol uses standardised [19], [20] underlying cryptographic primitives:

- Two differnt Key Derivation Functions (KDF) $H_1, H_2 : \{0,1\}^\lambda \to \{0,1\}^\lambda$. These are used to derive message keys $\mathsf{mk}$ and chain keys $\mathsf{ck}$, respectively.
- A symmetric encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$.
- A digital signature scheme $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$.

In Signal's implementation of Sender Keys [14], the KDFs are instantiated as $H_1(m) := \mathsf{HMAC}(\mathtt{0x01}, m)$ and $H_2(m) := \mathsf{HMAC}(\mathtt{0x02}, m)$. We note that in Signal's two-party sessions signatures are not used. Instead, messages are authenticated by computing a MAC based on a function of the message key.

In WhatsApp [13], the HMAC used for the KDF is `HMAC-SHA256`, the symmetric encryption scheme is `AES-256` in CBC mode, and the signature scheme is `ECDSA` with `Curve25519`.

### B. State

The state $\gamma$ of a user $\mathsf{ID}$ contains a sender key $\mathsf{send\text{-}k} := (\mathsf{spk}, \mathsf{ck})$, where $\mathsf{spk}$ denotes the the signature public key and $\mathsf{ck}$ the chain key, as well as a secret signature key $\mathsf{ssk}$, each belonging to $\mathsf{ID}$. The state also maintains a list of current group members $\boldsymbol{G}$. Additionally, for each user $\mathsf{ID} \in \boldsymbol{G}$ the sender key $\mathsf{send\text{-}k}_{\mathsf{ID}}$, a list of skipped message keys $\{\mathsf{mk}^i_{\mathsf{ID}}, \dots\}$, used for out-of-order delivery, and the counter $i$ are stored. If $\mathsf{ID}$ leaks its state, we say that it suffered a *state compromise*.

### C. Algorithms

We describe the Sender Keys protocol according to our Group Messenger primitive defined in Section II. The description follows Signal's reference implementation [14] regarding sender key ratcheting as well as message encryption and decryption. The details of the Exec and Recv algorithms are inferred from [13], but we cannot assert that our interpretation is entirely faithful to WhatsApp's implementation. A simplified example of a 3-message conversation is shown in Figure 1.

*1) State initialization:* The Init algorithm initializes the state variables of users; in practice this is done at install time.

*2) Group creation:* The creation of a group is carried out via the $\mathsf{Exec}(\texttt{crt}, \boldsymbol{G}, \gamma)$ algorithm which takes a list of prospective members $\boldsymbol{G} := \{\mathsf{ID}_1, \ldots, \mathsf{ID}_{|\boldsymbol{G}|}\}$ as input. All parties are assumed to have pre-established two-party communication sessions. The group creator $\gamma$.ME generates a chain key $\mathsf{ck} \xleftarrow{\$} \{0,1\}^\lambda$ and a signature key pair $(\gamma.\mathsf{spk}, \gamma.\mathsf{ssk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$. Then, it sends its sender key $\gamma.\mathsf{send\text{-}k} = (\gamma.\mathsf{spk}, \mathsf{ck})$ to each $\mathsf{ID} \in \boldsymbol{G}$ individually via their secure two-party channel.

*3) Message sending:* To send the $i^{\text{th}}$ message $m_i$, a user $\gamma$.ME calls the $\mathsf{Send}(m_i, \gamma)$ algorithm which does the following:

- Derive a new message key mk from the symmetric part of its sender key (i.e. the chain key) as $\mathsf{mk}^i_{\mathsf{ME}} \leftarrow H_1(\mathsf{ck}^i_{\mathsf{ME}})$.
- Encrypt the message $m_i$ as $c_i \xleftarrow{\$} \mathsf{Enc}(\mathsf{mk}^i_{\mathsf{ME}}, m_i)$.
- Ratchet the chain key ck[1] as $\mathsf{ck}^{i+1}_{\mathsf{ME}} \leftarrow H_2(\mathsf{ck}^i_{\mathsf{ME}})$.
- Jointly sign the ciphertext $c_i$, the message index $i$ and the sender's identity ME as $\sigma_i \xleftarrow{\$} \mathsf{Sig}(\mathsf{ssk}, (c_i, i, \mathsf{ME}))$
- Send $C := (c_i, i, \mathsf{ME}, \sigma_i)$ to all group members via the DS.

If $\gamma.\mathsf{send\text{-}k} = \bot$, then $\gamma$.ME must first generate a fresh sender key and distribute it as in the group creation. This occurs every time a member sends their first message after entering the group or after a member removal. We emphasise that ciphertexts are not sent over the preexisting two-party channels that are used to communicate sender keys, but rather over the network (i.e., via the Delivery Service) itself.

*4) Message reception:* Upon reception of a message $C = (c_i, i, \mathsf{ID}, \sigma_i)$, the receiver calls $\mathsf{Recv}(C, \gamma)$. First, Recv verifies ID and the signature as $\mathsf{Ver}(\mathsf{send\text{-}k}[\mathsf{ID}].\mathsf{spk}, \sigma_i, (c_i, i, \mathsf{ID}))$ (note that if $\mathsf{send\text{-}k}[\mathsf{ID}] = \bot$, the receiver must wait until a new sender key is sent by ID). If the check passes, then the algorithm proceeds as follows:

- If send-k[ID].ck is at iteration $i$:
  derive $\mathsf{mk}_i \leftarrow H_1(\mathsf{send\text{-}k}[\mathsf{ID}.\mathsf{ck}])$,
  decrypt $c_i$ as $m_i \leftarrow \mathsf{Dec}(\mathsf{mk}_i, c_i)$ and erase $\mathsf{mk}_i$,
  and refresh $\mathsf{send\text{-}k}[\mathsf{ID}.\mathsf{ck}] \leftarrow H_2(\mathsf{send\text{-}k}[\mathsf{ID}.\mathsf{ck}])$.
- If send-k[ID].ck is at iteration $j < i$, refresh the chain key $i - j$ times as $\mathsf{send\text{-}k}[\mathsf{ID}.\mathsf{ck}] \leftarrow H_2(\mathsf{send\text{-}k}[\mathsf{ID}.\mathsf{ck}])$ while storing message keys $\mathsf{mk}_j, \ldots \mathsf{mk}_{i-1}$ (up to $N_{\mathsf{max}}$ keys). Then, obtain $\mathsf{mk}_i$, decrypt $c_i$, and erase $\mathsf{mk}_i$.
- If send-k[ID].ck is at iteration $j > i$, search for a stored $\mathsf{mk}_j$, attempt to decrypt $c_i$, and erase $\mathsf{mk}_j$. If unsuccessful, output $\bot$.

[1]Note that this practice is safer (better FS) than first evolving the chain key and then deriving a message key, since it allows for the immediate deletion of the chain key used to derive the message key.

*5) Membership changes:* To add a new user ID to the group, a member calls $\mathsf{Exec}(\texttt{add}, \{\mathsf{ID}\}, \gamma)$. This produces a notification message $T$ sent to all group members including ID. Separately, $\gamma$.ME sends every member's sender key to ID using their two-party channel. As mentioned earlier, ID only generates and sends its own sender key when they send their first message.

To remove a user ID from the group, a member calls $\mathsf{Exec}(\texttt{rem}, \{\mathsf{ID}\}, \gamma)$ and sends the notification $T$ to all members.

*6) Message processing:* Group changes are processed via $\mathsf{Proc}(T, \gamma)$. If a user ID is added, $\gamma$.ME simply updates the list of group members $\gamma.G$. If ID is removed, $\gamma$.ME deletes all sender keys, including his own. In this scenario, the group "starts over", namely all members must generate and send a new sender key (i.e. a new signature key pair and symmetric key) to the members.

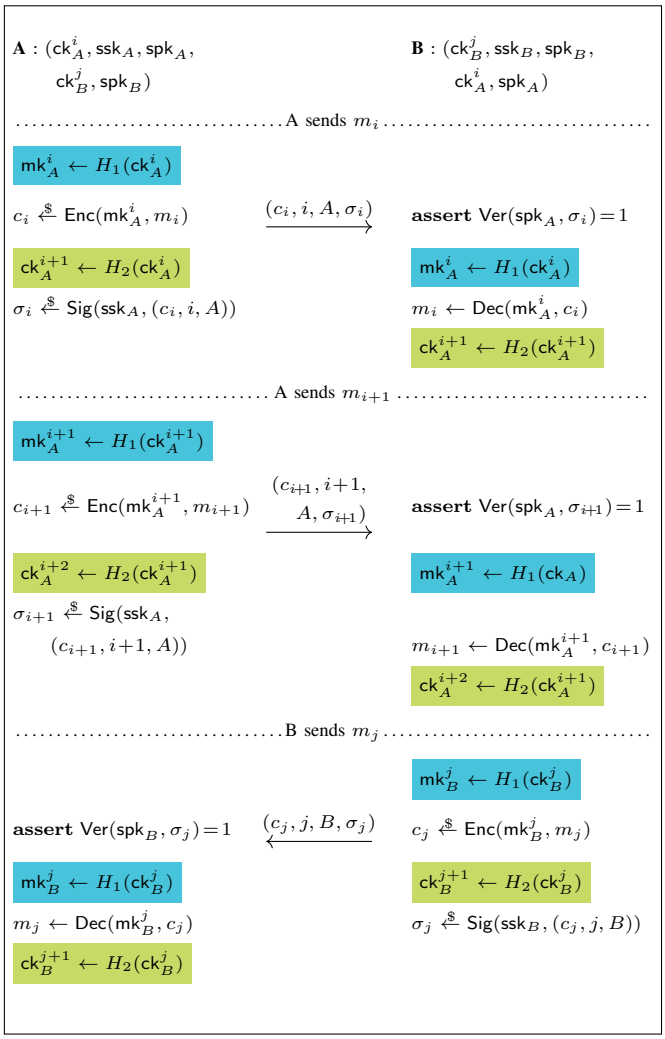



Figure 1: Sending/receiving messages between two group members for a three-message (in-order) conversation. Ephemeral message keys mk are deleted immediately after use. A's initial sender key is $(\mathsf{ck}^i_A, \mathsf{ssk}_A)$ and B's initial sender key is $(\mathsf{ck}^j_B, \mathsf{ssk}_B)$.

## IV. Security Model

We propose a model of security for our Group Messenger primitive that captures the security suitable for an authenticated and forward-secure group messaging scheme. We introduce a

game played between a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger.

### A. Game Description

At the beginning of the game, a bit $b$ is uniformly sampled which parametrises the game. To win, the adversary either has to guess $b$ or carry out a successful forgery in a *clean* protocol run. The game is parameterised by a *cleanness* predicate (sometimes safety [9] predicate) that captures the exact security of the protocol, namely the authenticity and confidentiality of group messages.

In this work, we assume that the two-party communication channels used for sending sender keys are perfectly secure; i.e., always confidential and authenticated. This assumption is not easily met in practice, but allows us to capture the essence of Sender Keys alone.

For simplicity, the game starts with a pre-established group $\boldsymbol{G}$ where every member $\mathsf{ID} \in \boldsymbol{G}$ already has everyone's honestly generated sender key. Then, $\mathcal{A}$ can interact with several oracles:

- $\mathcal{O}^{\mathsf{Challenge}}(\mathsf{ID}, m_0, m_1)$. The adversary receives a ciphertext $C_b$ corresponding to $\mathsf{ID}$ sending $m_b$, i.e., the output of $C_b \leftarrow \mathsf{Send}(m_b, \gamma_{\mathsf{ID}})$.
- $\mathcal{O}^{\mathsf{Send}}(\mathsf{ID}, m)$. $\mathsf{ID}$ sends an application-level message $m$ using the $\mathsf{Send}$ algorithm, producing a ciphertext $C$.
- $\mathcal{O}^{\mathsf{Receive}}(\mathsf{ID}, \mathsf{ID}', C)$. $\mathsf{ID}$ calls $\mathsf{Recv}$ on an application-level ciphertext $C$ claimed to be from user $\mathsf{ID}'$. If $C$ has not been generated honestly via the $\mathcal{O}^{\mathsf{Send}}(\mathsf{ID}, m)$ oracle and receiving is successful, $b$ is leaked to $\mathcal{A}$.
- $\mathcal{O}^{\mathsf{Add}}(\mathsf{ID}, \mathsf{ID}')$. $\mathsf{ID}$ adds $\mathsf{ID}'$ to the group via $\mathsf{Exec}$, generating a control message $T$.
- $\mathcal{O}^{\mathsf{Remove}}(\mathsf{ID}, \mathsf{ID}')$. $\mathsf{ID}$ removes $\mathsf{ID}'$ from the group via $\mathsf{Exec}$, generating a control message $T$.
- $\mathcal{O}^{\mathsf{Deliver}}(\mathsf{ID}, T)$. A control message $T$ is delivered to $\mathsf{ID}$ who calls $\mathsf{Proc}$.
- $\mathcal{O}^{\mathsf{Expose}}(\mathsf{ID})$. The current state $\gamma$ of $\mathsf{ID}$ leaks to $\mathcal{A}$.
- $\mathcal{O}^{\mathsf{ExpMK}}(\mathsf{ID}, i)$. The $i$-th message key $\mathsf{mk}$ of $\mathsf{ID}$ leaks to $\mathcal{A}$. No message encrypted under this key can be challenged (neither before nor after exposure).

After $q$ oracle queries, the adversary outputs a guess $b'$ of $b$. Note that $\mathcal{A}$ can win the game either by guessing a challenge correctly or by injecting a forged message via $\mathcal{O}^{\mathsf{Receive}}$ successfully.

### B. Cleanness

For the particular case of Sender Keys, we describe the cleanness predicate which defines the following conditions for a valid game:

- We define the event $\mathsf{refresh}(\mathsf{ID})$ that occurs when: 1) some member has been removed from the group, and 2) member $\mathsf{ID}$ processes this change (note the lack of a PCS update option).
- After exposing *any* user $\mathsf{ID}$, all adversarial calls to $\mathcal{O}^{\mathsf{Challenge}}$ on future messages are disallowed until $\mathsf{refresh}(\mathsf{ID})$ occurs for *every* $\mathsf{ID} \in \boldsymbol{G}$. This extends to all challenges on skipped messages (out-of-order) that $\mathsf{ID}$ has not received at exposure time.
- After exposing *a specific* user $\mathsf{ID}'$, $\mathcal{A}$ cannot win the game by impersonating $\mathsf{ID}'$ via $\mathcal{O}^{\mathsf{Receive}}(\mathsf{ID}, \mathsf{ID}', C)$ for a forgery $C$ until a new $\mathsf{refresh}(\mathsf{ID})$ event occurs.

We remark that our security notion is adaptive insofar as users can adaptively expose users. Under our cleanness predicate, we consider limited injection queries, i.e., partially active security. Our modelling further assumes that the underlying two-party channels are perfectly secure, and thus we leave it as important future work to examine security where, e.g., state exposures on the underlying channels are allowed and the consequent security guarantees are captured.

## V. Security

We claim that Sender Keys, as described in Section III, is secure with respect to our security model. However, the security captured by our cleanness predicate is sub-optimal, in the sense that forward security can be strengthened for authentication, as we introduce in Section V-C. Here we introduce a security analysis, but leave a security proof and more accurate modelling for future work.

### A. Passive adversaries

For a (semi-)passive adversary which does not attempt to inject messages via $\mathcal{O}^{\mathsf{Receive}}$ (but can still schedule messages arbitrarily), we claim that Sender Keys is secure with respect to the cleanness predicate, given that the symmetric encryption scheme is IND-CPA secure. Towards proving this:

- If the $\mathsf{KDF}$ is a one-way function, message keys $\mathsf{mk}_i$ can be exposed independently; the compromise of a message key never affects the confidentiality of other keys or messages. Hence, giving adversarial access to $\mathcal{O}^{\mathsf{ExpMK}}(\mathsf{ID}, i)$ does not impact the cleanness predicate.
- Also assuming one-wayness of the $\mathsf{KDF}$, forward secrecy holds trivially except for out-of-order messages.
- Assuming that the two-party channels are secure, all users recover from state exposure via $\mathcal{O}^{\mathsf{Expose}}(\mathsf{ID})$ after a removal is made effective. Note that, outside our model, security of the two-party channels may also degrade after a state exposure, leaving room for further attacks.

### B. Single- vs multi-key

In a Sender Keys group, each user is associated with a different symmetric key and thus the state comprises $O(n)$ secret material at all times. Since users encrypt and then hash forward using their own key when sending each message, users can safely send messages concurrently and with some inter-member message reordering.

For large groups, however, this scaling behaviour may represent a bottleneck. Consequently, one can envision trade-offs between the amount of concurrency supported and the amount of secret material required to be stored at a given point a time. The other extreme of the spectrum would be when all users maintain the *same* single symmetric chain. In situations where users are not expected to concurrently send messages this allows the secret state size to reduce to $O(1)$ without degrading security. To deal with concurrency in this setting, a central server which rejects all but the first (for example) message and requests re-transmission for other users could then be employed.

In MLS, a new group secret (chosen by a single user) is established each epoch, from which point all $O(n)$ application keys are derived for a given point in time; MLS additionally supports out-of-order message delivery within a given epoch.

In Sender Keys, each user chooses their own key; thus, the security of ciphertexts in the presence of a passive adversary is contingent upon users initially sampling their key with enough entropy.

### C. Active attacks

In an active adversary scenario, the security of Sender Keys is sub-optimal. In particular, we note two issues. First, consider a simple group $\boldsymbol{G} = \{\mathsf{ID}_1, \mathsf{ID}_2\}$ and the following sequence of oracle queries:

- $q_1 = \mathcal{O}^{\mathsf{Send}}(\mathsf{ID}_1, m)$ which generates the $i$-th message $C$ encrypted under $\mathsf{mk}$ and signed under $\mathsf{ssk}_1$.
- $q_2 = \mathcal{O}^{\mathsf{Expose}}(\mathsf{ID}_1)$, where $\mathcal{A}$ obtains $\mathsf{ssk}_1$, but not $\mathsf{mk}$.
- $q_3 = \mathcal{O}^{\mathsf{ExpMK}}(\mathsf{ID}_1, i)$, leaking $\mathsf{mk}$.
- $\mathcal{A}$ crafts $c' = Enc(m', \mathsf{mk})$, signs it under $\mathsf{ssk}_1$ and forges a $C'$.
- $q_4 = \mathcal{O}^{\mathsf{Deliver}}(\mathsf{ID}_2, C')$, as the $i$-th message.

Note that $q_4$ is a forbidden query by our cleanness predicate in Section IV-B. $q_4$ attempts to inject a message that corresponds to key material utilized *before* the state exposure, hence one can envision stronger forward security where queries like $q_4$ are allowed. In this case, the Sender Keys adversary would win the game. We describe how to achieve such stronger security in Section V-D by strengthening signatures.

Technically, the query $q_3$ can be replaced by $\mathcal{O}^{\mathsf{Expose}}(\mathsf{ID}_k)$, or even be omitted as the adversary can still create a valid forgery by altering the metadata in $C$ (such as the sender's identity) without crafting a new $c'$. This attack can occur naturally if $\mathsf{ID}_2$ is offline when $m$ is first sent.

An attack of a similar nature can also occur if the same signature key is re-used across groups, and they are refreshed at different times, as pointed out in [17].

The second issue we note is that the implementation of the Exec algorithm can be problematic if messages are not authenticated correctly. This led to attacks in the past such as the *burgle into the group* or the *acknowledgement forgery* attacks in [21]. Securing control messages and group membership changes is possible as introduced in [16].

### D. Proposed Modifications and Tweaks

*1) Ratcheting signature keys:* The attack shown in the previous section can be mitigated if signature keys are also ratcheted. A simple fix is to introduce a *chain* of signature keys, where an ephemeral signature key is created every time a message or block of messages is sent.

Let $(\mathsf{ssk}, \mathsf{spk})$ be $\mathsf{ID}$'s signature key pair, where $\mathsf{spk}$ is part of its Sender Key. Then, before sending a new message $m$ to the group, $\mathsf{ID}$ can generate a new key pair $(\mathsf{ssk}', \mathsf{spk}') \xleftarrow{\$} \mathsf{Gen}(1^\lambda)$. Then, $\mathsf{ID}$ can do as follows:

- Encrypt the $i^{\text{th}}$ message $m$ with the corresponding $\mathsf{mk}$, $c \xleftarrow{\$} \mathsf{Enc}(\mathsf{mk}, m)$.
- Sign $C \leftarrow (c, i, \mathsf{ID}, \mathsf{spk}')$ as $\sigma \xleftarrow{\$} \mathsf{Sig}(\mathsf{ssk}, C)$
- Send the tuple $(\sigma, C)$ to the group.
- Replace $\mathsf{ssk}$ by $\mathsf{ssk}'$.

Upon reception of a message, $\mathsf{ID}'$ will:

- Verify the signature as $\mathsf{Ver}(\mathsf{spk}, \sigma, C)$ and decrypt the ciphertext $c$.
- Replace $\mathsf{spk}$ by $\mathsf{spk}'$ in $\mathsf{ID}$'s sender key.

Note that this countermeasure involves a notable overhead and entropy consumption (although only for the sender), so it may not be desirable in all scenarios, or for all sent messages. Users could also replace their signature keys on-demand or on a time schedule to trade more post-compromise security for performance.

*2) Randomness manipulation:* We note that Sender Keys, as described in Section III, is susceptible to randomness exposure and randomness manipulation attacks. Namely, the adversary does not need to leak a member's state, but simply control the randomness used by the device, inhibiting any form of PCS. Protection against this family of attacks can be attained at small cost if freshly generated keys are hashed with the state or with part of the state, as in [2] and the classic NAXOS trick for authenticated key exchange [22].

*3) Refresh option for PCS:* Post compromise security (PCS) is generally achieved when introducing fresh randomness by establishing new group secrets. In the case of Sender Keys, it could be beneficial to establish new sender keys across the group at some intervals, for instance via an *update* group operation, as noted in [17]. However, if only Alice updates her sender key, a passive adversary would still be able to eavesdrop on messages sent by any other group member. Furthermore, as the sender key is group-specific, any messages sent by Alice in another group are also susceptible to eavesdropping. We refer the reader to Appendix A of [17] for a more detailed discussion and conclude that the sender keys is not a suitable approach towards achieving a PCS-secure group messenger.

## VI. Conclusion and Future Work

The Sender Keys protocol is a convenient protocol for group chats which is simple to implement, achieves a fair degree of end-to-end forward security, and deals well with concurrency. Nevertheless, we remark that no security is gained by employing multiple sender keys with respect to a single group key given randomness used to generate the keys is honest (i.e., in our model). Furthermore, forward security is sub-optimal for message authentication, which can be fixed by ratcheting signature keys.

Our results are under the assumption that two-party channels are secure, which is clearly not the case in practice. Therefore, more powerful attacks may arise under a more realistic model where exposure of two-party channels is possible. We also note that, in the real world, the security of messaging apps can also be broken in different ways than attacking the protocol. Additional features such as conversation transcript backups and multi-device support highly increase the attack surface and should be avoided in applications where security is critical.

Our analysis leaves multiple directions for future work, such as a rigorous formalization of the security model with a fine-grained analysis of the two-party channels and user compromise, more precise cleanness predicates, and especially concrete security reductions.

## Acknowledgements

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under project PICOCRYPT (grant agreement No. 101001283), and by a grant from Nomadic Labs and the Tezos foundation. The last author

was supported by DFG under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

## REFERENCES


[1] M. Marlinspike and T. Perrin, "The double ratchet algorithm," 2016. [Online]. Available: https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf

[2] J. Jaeger and I. Stepanovs, "Optimal channel security against fine-grained state compromise: The safety of messaging," in *Advances in Cryptology – CRYPTO 2018, Part I*, ser. Lecture Notes in Computer Science, H. Shacham and A. Boldyreva, Eds., vol. 10991. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 19–23, 2018, pp. 33–62.

[3] B. Poettering and P. Rösler, "Asynchronous ratcheted key exchange," Cryptology ePrint Archive, Report 2018/296, 2018, https://eprint.iacr.org/2018/296.

[4] F. B. Durak and S. Vaudenay, "Bidirectional asynchronous ratcheted key agreement with linear complexity," in *IWSEC 19: 14th International Workshop on Security, Advances in Information and Computer Security*, ser. Lecture Notes in Computer Science, N. Attrapadung and T. Yagi, Eds., vol. 11689. Tokyo, Japan: Springer, Heidelberg, Germany, Aug. 28–30, 2019, pp. 343–362.

[5] J. Alwen, S. Coretti, and Y. Dodis, "The double ratchet: Security notions, proofs, and modularization for the Signal protocol," in *Advances in Cryptology – EUROCRYPT 2019, Part I*, ser. Lecture Notes in Computer Science, Y. Ishai and V. Rijmen, Eds., vol. 11476. Darmstadt, Germany: Springer, Heidelberg, Germany, May 19–23, 2019, pp. 129–158.

[6] F. Balli, P. Rösler, and S. Vaudenay, "Determining the core primitive for optimally secure ratcheting," in *Advances in Cryptology – ASIACRYPT 2020, Part III*, ser. Lecture Notes in Computer Science, S. Moriai and H. Wang, Eds., vol. 12493. Daejeon, South Korea: Springer, Heidelberg, Germany, Dec. 7–11, 2020, pp. 621–650.

[7] R. Barnes, B. Beurdouche, R. Robert, J. Millican, E. Omara, and K. Cohn-Gordon, "The Messaging Layer Security (MLS) Protocol," Internet Engineering Task Force, Internet-Draft draft-ietf-mls-protocol-14, May 2022, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-mls-protocol-14

[8] K. Bhargavan, R. Barnes, and E. Rescorla, "TreeKEM: Asynchronous Decentralized Key Management for Large Dynamic Groups A protocol proposal for Messaging Layer Security (MLS)," Inria Paris, Research Report, May 2018. [Online]. Available: https://hal.inria.fr/hal-02425247

[9] J. Alwen, S. Coretti, Y. Dodis, and Y. Tselekounis, "Security analysis and improvements for the IETF MLS standard for group messaging," in *Advances in Cryptology – CRYPTO 2020, Part I*, ser. Lecture Notes in Computer Science, D. Micciancio and T. Ristenpart, Eds., vol. 12170. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, Aug. 17–21, 2020, pp. 248–277.

[10] K. Klein, G. Pascual-Perez, M. Walter, C. Kamath, M. Capretto, M. Cueto, I. Markov, M. Yeo, J. Alwen, and K. Pietrzak, "Keep the dirt: Tainted TreeKEM, adaptively and actively secure continuous group key agreement," in *2021 IEEE Symposium on Security and Privacy*. San Francisco, CA, USA: IEEE Computer Society Press, May 24–27, 2021, pp. 268–284.

[11] J. Alwen, S. Coretti, D. Jost, and M. Mularczyk, "Continuous group key agreement with active security," in *TCC 2020: 18th Theory of Cryptography Conference, Part II*, ser. Lecture Notes in Computer Science, R. Pass and K. Pietrzak, Eds., vol. 12551. Durham, NC, USA: Springer, Heidelberg, Germany, Nov. 16–19, 2020, pp. 261–290.

[12] J. Alwen, S. Coretti, Y. Dodis, and Y. Tselekounis, "Modular design of secure group messaging protocols and the security of MLS," in *ACM CCS 2021: 28th Conference on Computer and Communications Security*, G. Vigna and E. Shi, Eds. Virtual Event, Republic of Korea: ACM Press, Nov. 15–19, 2021, pp. 1463–1483.

[13] WhatsApp, "WhatsApp Encryption Overview Technical white paper, v.3," Oct. 2020, https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf.

[14] M. Marlinspike *et al.*, "Signal protocol." [Online]. Available: https://github.com/signalapp/libsignal-protocol-java/tree/master/java/src/main/java/org/whispersystems/libsignal

[15] K. Cohn-Gordon, C. J. F. Cremers, and L. Garratt, "On post-compromise security," in *CSF 2016: IEEE 29st Computer Security Foundations Symposium*, M. Hicks and B. Köpf, Eds. Lisbon, Portugal: IEEE Computer Society Press, jun 27-1 2016, pp. 164–178.

[16] D. Balbás, D. Collins, and S. Vaudenay, "Cryptographic Administrators for Secure Group Messaging," Cryptology ePrint Archive.

[17] C. Cremers, B. Hale, and K. Kohbrok, "The complexities of healing in secure group messaging: Why cross-group effects matter," in *USENIX Security 2021: 30th USENIX Security Symposium*, M. Bailey and R. Greenstadt, Eds. USENIX Association, Aug. 11–13, 2021, pp. 1847–1864.

[18] M. Marlinspike and T. Perrin, "The x3dh key agreement protocol," 2016. [Online]. Available: https://signal.org/docs/specifications/x3dh/x3dh.pdf

[19] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," IETF Internet Request for Comments 2104, Feb. 1997.

[20] H. Krawczyk and P. Eronen, "Hmac-based extract-and-expand key derivation function (hkdf)," Internet Requests for Comments, RFC Editor, RFC 5869, May 2010, http://www.rfc-editor.org/rfc/rfc5869.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5869.txt

[21] P. Rösler, C. Mainka, and J. Schwenk, "More is less: On the end-to-end security of group chats in signal, whatsapp, and threema," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018, pp. 415–429.

[22] B. A. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *ProvSec 2007: 1st International Conference on Provable Security*, ser. Lecture Notes in Computer Science, W. Susilo, J. K. Liu, and Y. Mu, Eds., vol. 4784. Wollongong, Australia: Springer, Heidelberg, Germany, Nov. 1–2, 2007, pp. 1–16.

# Transferencia de aprendizaje en redes neuronales para mejora de un IDS

José Ignacio Bengoechea-Isasa
Estudios de Informática,
Multimedia y Telecomunicación
Universitat Oberta de Catalunya
jbengoecheai@uoc.edu

Carles Ventura
Estudios de Informática,
Multimedia y Telecomunicación
Universitat Oberta de Catalunya
cventuraroy@uoc.edu

Helena Rifà-Pous
Internet Interdisciplinary Institute (IN3)
CYBERCAT
Universitat Oberta de Catalunya
hrifa@uoc.edu

***Resumen*—Los sistemas de detección de intrusos nos permiten detectar y clasificar las amenazas que existen en nuestro entorno de red. La aparición de nuevos tipos de vulnerabilidades hace necesario que estos sistemas puedan detectar amenazas hasta ahora desconocidas de manera automática. El uso de técnicas de inteligencia artificial, especialmente el uso de modelos basados en redes neuronales profundas nos permite mejorar el rendimiento de estos sistemas. En este artículo presentaremos las tendencias actuales para mejorar los sistemas de detección de intrusos mediante redes neuronales, y presentaremos un modelo pre-entrenado mediante transferencia de aprendizaje que ha sido usado en varios conjuntos de datos y puede ser adaptado para detectar nuevas amenazas. Este modelo ha sido entrenado a partir de los conjuntos de datos CIC-IDS-2017 y CIC-IDS-2018.**

***Index Terms*—aprendizaje profundo, sistemas de detección de intrusos**

## I. Introducción

Un sistema de detección de intrusos (IDS) usa dos técnicas para identificar los ataques. La primera técnica trata de identificar la amenaza buscando patrones y firmas (p.e. un conjunto específico de bytes). Estos sistemas tienen una ratio baja de falsos positivos, pero no pueden detectar nuevos ataques. La segunda técnica, basada en anomalías, trata de detectar, e identificar, un comportamiento no esperado del flujo de tráfico en la red (p.e. un incremento del ancho de banda usado por una aplicación). Este tipo de técnicas pueden detectar un ataque nuevo, pero tiene una ratio alta de falsos positivos y de falsos negativos.

Para mejorar el rendimiento en la detección de anomalías de los IDS se han realizado varias investigaciones que muestran que el uso de técnicas de inteligencia artificial es apropiado para mejorar el rendimiento en este problema. Dentro del área de la inteligencia artificial existe un campo llamado aprendizaje profundo, o deep learning, que usa diversas técnicas basadas en el uso de redes neuronales para hallar características relevantes de los datos.

Por ejemplo, el modelo de detección de malware desarrollado por Saxe [1] es un modelo basado en deep learning que ha sido usado en aplicaciones comerciales.

Podemos encontrar varias compañías que están usando métodos de deep learning para mejorar la detección de anomalías en sus sistemas de seguridad informática.

- BluVector[1]. Proporciona servicios de detección de anomalías en el tráfico, basados en deep learning.
- SignalSense[2]. No se conoce mucho sobre el funcionamiento interno de su producto, pero usan técnicas basadas en deep learning para las aplicaciones de seguridad informática.
- Deep Instinct[3]. Proporciona soluciones de seguridad para corporaciones, y usa técnicas de deep learning para detección de malware.

En deep learning distinguimos dos tipos de metodologías para el desarrollo de modelos. Por una parte, las metodologías supervisadas, que usan un dataset que ha sido etiquetado. Los estudios basados en estas metodologías tratan de hallar modelos de deep learning que se adapten como un guante a estos datasets. Por otra parte, las metodologías no supervisadas, que usan un dataset que no ha sido etiquetado. En este caso, los estudios tratan que su modelo sea capaz de detectar un comportamiento anómalo en el dataset que no se corresponde con un proceso de tráfico normal.

Hay muchos estudios basados en metodologías supervisadas que obtienen modelos muy buenos pero solo válidos para un dataset concreto, con resultados pobres al tratar de usarlos en otros conjuntos de datos. Hay otros estudios basados en metodologías no supervisadas que obtienen modelos que se pueden adaptar a varios datasets pero con un rendimiento muy limitado. El reto de la investigación del deep learning en detección de anomalías es encontrar modelos que se puedan adaptar a varios datasets teniendo un rendimiento elevado, y sin necesidad de entrenar el modelo completo en su totalidad por cada dataset.

En este artículo tratamos de ampliar el enfoque usado en las metodologías supervisadas. Para ello, vemos que en otros campos, como la visión artificial, se han usado con mucho éxito modelos entrenados previamente, que permiten transferir el aprendizaje que hemos realizado en un dataset a otros datasets distintos. En este articulo presentamos un modelo basado en una metodología supervisada, que puede ser usado en varios datasets con un rendimiento elevado, y sin necesidad de realizar un entrenamiento completo, sino parcial en los datasets nuevos.

## II. Background

### *II-A. Deep learning*

Es habitual que los IDS generen datasets masivos, de varias decenas de gigabytes por día. Para este tipo de datasets se ha

[1]https://www.bluvector.io/

[2]https://www.welcome.ai/tech/security/signalsense

[3]https://www.deepinstinct.com/

comprobado que la opción para obtener los mejores resultados es el uso de redes neuronales.

Andrew Ng, director del laboratorio de Inteligencia Artificial en Stanford, comento en la EasternConf conference de 2015[4] sobre la importancia del uso de redes neuronales en proyectos de alta escalabilidad, los cuales incluyen datasets masivos:

*"Deep learning is the first class of algorithms where performance just keeps getting better as we feed them more data".*

Una solución basada en técnicas clásicas de machine learning puede ser más rápida que una solución basada en técnicas de deep learning, pero su porcentaje de fallos será superior, y escala peor cuanto más grande es el dataset.

### *II-B. Datasets*

Esta es una lista de los datasets que han sido creados por la comunidad para el entrenamiento de modelos que quieren mejorar el rendimiento de los IDS.

- KDD CUP 1999[5]. Este dataset se creó en 1999. Fue uno de los primeros datasets creados, y es el más usado en los artículos. Sin embargo, presenta varios defectos; los ataques no son realistas, y las clases de los ataques no están balanceadas, ya que tiene más ataques de los habituales.
- NSL KDD[6]. Este dataset es una versión balanceada del anterior, pero sigue teniendo el resto de defectos.
- CTU-13[7]. Es un dataset de tráfico normal y de tráfico bot creado en 2011.
- CIC-IDS-2012, 2017, 2018. Son un conjunto de datasets creados por el Canadian Institute for Cybersecurity, CIC[8], en los años 2012, 2017 y 2018. Son ataques realistas, actualizados y que tienen en cuenta la presencia de un tráfico normal para complementar al tráfico de ataques.

En un modelo supervisado, estos datasets se dividen en tres partes: (1) entrenamiento, (2) validación, y (3) evaluación. Con (1) y (2) podemos entrenar el modelo, usando validación cruzada en 5 pliegues, lo que nos permite obtener el mejor modelo posible. Para ello, se usa como entrada un paquete y como salida la clase a la que pertenece. Para la evaluacion, con (3) se usa como entrada un paquete, sin conocer la clase, y el modelo predice a que clase pertenece. Luego, se compara la clase que ha estimado el modelo con la real, y se calcula una métrica, que es un valor numérico, que nos indica como de bueno es el rendimiento de ese modelo comparado a otros modelos, que usan el mismo dataset y la misma métrica.

### *II-C. Métricas*

Vamos a describir las métricas que son usadas en aplicaciones, o en artículos, relacionados con este tema.

**Matriz de confusión**

La matriz de confusión nos permite ver el rendimiento de un modelo en un problema de aprendizaje supervisado. La matriz de confusión (ver Tabla I) es una matriz con 2 columnas y 2 filas, que nos permite visualizar el número de positivos verdaderos (PV), negativos verdaderos (NV), positivos falsos (PF), y negativos falsos (NF).

Tabla I
MATRIZ DE CONFUSIÓN

| | Malicioso | No malicioso |
|---|---|---|
| Malicioso | PV | NF |
| No malicioso | PF | NV |

A partir de la matriz de confusión se definen 4 métricas relevantes:

1. Exactitud, o accuracy, que está relacionada con el sesgo de la medición.

$$Exactitud = \frac{PV + NV}{PV + PF + NV + NF}$$

2. Precisión, que está relacionada con la dispersión del conjunto de valores obtenidos.

$$Precisión = \frac{PV}{PV + PF}$$

3. Sensibilidad, es una métrica que nos indica como de bueno es nuestro modelo para discriminar los casos positivos.

$$Sensibilidad = \frac{PV}{PV + NF}$$

4. F1-Score, es una métrica estable, que no se ve afectada si la distribución de las clases no está balanceada, como ocurre en estos datasets.

$$F1 - Score = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

### *II-D. Técnicas de Deep Learning*

En esta apartado vemos qué técnicas de deep learning se están utilizando en la literatura.

**DNN: Red neuronal profunda**

La base de las técnicas de deep learning es la red neuronal artificial, la cual está formada por neuronas, que se organizan en tres etapas. La etapa de entrada, la etapa de salida, y la etapa intermedia. Si hay varias etapas intermedias se le llama red neuronal profunda, o DNN. Estas redes se usan en todo clase de problemas relacionados con clasificación o con regresión.

**CNN: Redes neuronales convolucionales**

Una CNN es un tipo de red neuronal que trabaja bien en tareas relacionadas con la visión artificial, (p.e. clasificación o detección de objetos en imágenes). Para ello extrae las características más relevantes de la imagen y las envía a una capa donde se realiza la clasificación de la misma.

**RNN: Redes neuronales recurrentes**

Este tipo de redes son usadas en reconocimiento de voz a texto, en traducción de idiomas, y en análisis financieros. En esta red los nodos tiene una entrada, una salida, y una realimentación recurrente, que transmite la salida del instante temporal anterior a la entrada.

**LSTM: Memoria de largo y corto plazo**

Una LSTM es una red neuronal recurrente que solventa una limitación de las RNN, que olvida las dependencias a largo termino. Su área de aplicación es similar a las RNN.

[4]https://www.youtube.com/watch?v=O0VN0pGgBZM
[5]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[6]https://www.unb.ca/cic/datasets/nsl.html
[7]https://www.stratosphereips.org/datasets-ctu13
[8]https://www.unb.ca/cic

### II-E. Comparación de técnicas

En la tabla II podemos ver una selección de los artículos más relevantes de la literatura que usan técnicas de deep learning con aprendizaje supervisado. En negrita, están los modelos más interesantes de los datasets más actualizados: CIC-IDS-2017 y CIC-IDS-2018.

Tabla II
ARTÍCULOS PUBLICADOS DE MODELOS CON APRENDIZAJE SUPERVISADO

| DL | Cita | Dataset usado | Accuracy | F1-Score |
|---|---|---|---|---|
| ANN | (Gam. et al, 2020)[5] | IDS-2017 | 99.58 % | 99.57 % |
| ANN | (Gam. et al, 2020)[5] | IDS-2018 | 98.30 % | 98.60 % |
| DNN | (Tang et al,2016)[6] | Custom | 75.75 % | 75.00 % |
| DNN | (Dir. et al, 2018b)[7] | NSL KDD | 98.27 % | - |
| DNN | (Vin. et al, 2019)[8] | IDS-2017 | 95.60 % | 95.70 % |
| DNN | (Fe. et al, 2020)[9] | IDS-2018 | 97.28 % | - |
| CNN | (Wan. et al, 2017)[10] | CTU-13 | 99.17 % | 98.00 % |
| LSTM | (Kim et al,2016)[11] | KDD 99 | 96.93 % | - |
| LSTM | (Lou. et al,2018)[12] | Custom | 86.94 % | - |
| LSTM | (Diro et al, 2018a)[13] | IDS-2012 | 99.91 % | - |
| **LSTM** | **(Gam. et al, 2020)[5]** | **IDS-2017** | **99.59 %** | **99.58 %** |
| **LSTM** | **(Gam. et al, 2020)[5]** | **IDS-2018** | **98.88 %** | **98.65 %** |
| RNN | (Staud.,2015)[14] | KDD 99 | 93.82 % | - |
| RNN | (Yin et al,2017)[15] | NSL KDD | 81.29 % | - |

### II-F. Transferencia del aprendizaje

La transferencia del aprendizaje es un proceso por el cual se transfiere un conocimiento ya aprendido para que sea enfrentado en una nueva situación.

En el área de la visión artificial, la transferencia del aprendizaje es usada para reconocimiento de objetos. Existen modelos pre-entrenados en 2015 para el concurso de reconocimiento de objetos ImageNet, como VGG-16 o Inception V3[16]. Estos modelos se usan como modelos base en nuevas arquitecturas para detectar objetos que no existían en el dataset original de ImageNet.

## III. METODOLOGÍA E IMPLEMENTACIÓN

### III-A. Metodología

Como hemos indicado anteriormente queremos crear un modelo basado en deep learning que sea útil para la comunidad. Este modelo tiene que cumplir los siguientes requisitos.

- Mejorar una o varias métricas, con respecto a los modelos que ya hemos presentado.
- Poder usarlo en más de un dataset con buenos resultados. Queremos que pueda ser usado en escenarios más generales.

El primer objetivo es fácil de conseguir. Basta con observar el mejor modelo que se haya publicado en la literatura y tratar de optimizarlo. El problema con esta implementación es que estamos mejorando las métricas del modelo para un dataset único. No podemos reutilizar directamente el mismo modelo en otro dataset y que mantenga el mismo rendimiento.

El segundo requisito nos conduce a usar transferencia del aprendizaje. Para ello, es necesario partir de un modelo, ya optimizado, y usarlo como modelo base en una nueva arquitectura. Las capas de ese modelo base se congelan, esto quiere decir que los pesos de las neuronas no se modificarán, y se realiza un entrenamiento, usando validación cruzada, en la nueva arquitectura. Al usar un modelo base congelado, el entrenamiento es parcial y el tiempo de entrenamiento se reduce.

### III-B. Datasets

Hemos planteado usar nuestra arquitectura en dos datasets

- CIC-IDS-2017. Tiene casi 3 millones de instancias (son las líneas del dataset), con 83 características (son las columnas del dataset) y 14 clases de ataques. En la tabla III podemos ver la división por clases del dataset.
- CIC-IDS-2018. Tiene más de 16 millones de instancias, con 81 características y 14 clases de ataques. En la tabla IV podemos ver como es la distribución de clases. Por limitaciones de memoria usaremos un 60 % de este dataset.

Tabla III
DISTRIBUCIÓN DE CLASES EN CIC-IDS-2017

| Clase | Instancia | Frecuencia |
|---|---|---|
| Normal | 2271320 | 0.80318939 |
| Bot | 1956 | 0.00069168 |
| DDoS | 128025 | 0.04527249 |
| DoS GoldenEye | 10293 | 0.00363983 |
| DoS Hulk | 230124 | 0.08137697 |
| DoS Slowhttptest | 5499 | 0.00194456 |
| DoS slowloris | 5796 | 0.00204959 |
| FTP-Patator | 7935 | 0.00280599 |
| Heartbleed | 11 | 0.00000388 |
| Infiltration | 36 | 0.00001273 |
| PortScan | 158804 | 0.05615663 |
| SSH-Patator | 5897 | 0.00208531 |
| Web Attack Brute Force | 1507 | 0.00053290 |
| Web Attack Sql Injection | 21 | 0.00000742 |
| Web Attack XSS | 652 | 0.00023056 |

Tabla IV
DISTRIBUCIÓN DE CLASES EN CIC-IDS-2018

| Clase | Instancia | Frecuencia |
|---|---|---|
| Normal | 13390249 | 0.829776 |
| Bot | 286191 | 0.017735 |
| Brute Force-Web | 611 | 0.000038 |
| Brute Force-XSS | 230 | 0.000014 |
| DDOS attack-HOIC | 686012 | 0.042511 |
| DDOS attack-LOIC-UDP | 1730 | 0.000107 |
| DDoS attacks-LOIC-HTTP | 576191 | 0.035706 |
| DoS attacks-GoldenEye | 41508 | 0.002572 |
| DoS attacks-Hulk | 461912 | 0.028624 |
| DoS attacks-SlowHTTPTest | 139890 | 0.008669 |
| DoS attacks-Slowloris | 10990 | 0.000681 |
| FTP-BruteForce | 193354 | 0.011982 |
| Infiltration | 160639 | 0.009955 |
| SQL Injection | 87 | 0.000005 |
| SSH-Bruteforce | 187589 | 0.011625 |

Ambos datasets presentan una distribución de clases no balanceada, siendo la clase mayoritaria, la clase normal, donde no hay ataques. Es muy fácil que esta distribución lleve a resultados erróneos, por lo que es conveniente usar métricas que no se vean tan afectadas, como F1-Score.

Asimismo, ambos datasets han sido creados mediante un software llamado CICFloWMeter[9] que ha convertido los ficheros PCAP originales, en los ficheros CSV que tienen un total de 81 caracteristicas comunes entre ambos.

### III-C. Arquitectura

Nuestra arquitectura se basa en transferencia del aprendizaje. Entrenamos un modelo, llamado modelo base, a partir

[9]https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter

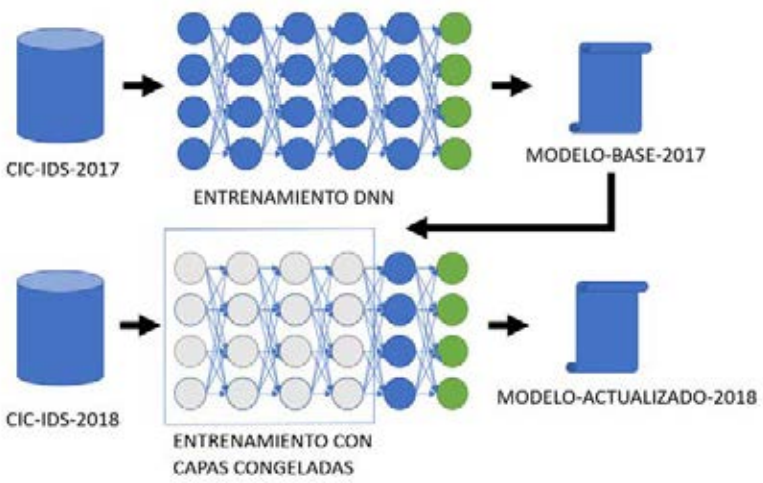



Figura 1. Esquema de la arquitectura

del dataset CIC-IDS-2017. Este modelo se actualizará usando el dataset CIC-IDS-2018, con las capas del modelo base congeladas, lo cual minimizará el tiempo de entrenamiento. Un esquema del flujo de esta arquitectura se puede ver en la Figura 1, donde existen una serie de fases:

1. Preprocesamiento del dataset CIC-IDS-2017. Se analiza el dataset, se limpia de elementos sin datos o con datos no válidos, se anonimiza, y por último se normaliza el contenido del dataset.
2. Preprocesamiento del dataset CIC-IDS-2018. Se sigue el proceso anterior, y luego se adapta la estructura de datos de un dataset al otro, en este caso es muy sencillo, ya que únicamente hay que respetar el orden de las entradas.
3. Se dividen los datasets de CIC-IDS-2017 y CIC-IDS-2018 en 3 partes. Entrenamiento (50 %), validación (20 %) y evaluación (30 %).
4. Se entrena mediante un proceso de validación cruzada, con CIC-IDS-2017 en 5 pliegues, y nos quedamos con el mejor modelo obtenido, que lo llamaremos modelo-base-2017. Evaluamos el modelo, en la parte de evaluación del mismo dataset, y anotamos los valores obtenidos. Hacemos el mismo proceso en CIC-IDS-2018 y obtendremos el modelo-base-2018.
5. Se actualiza el modelo-base-2017 usando esta vez como entrada el dataset CIC-IDS-2018. Se realiza un entrenamiento con validación cruzada y con las capas congeladas en ese dataset. Se obtiene un modelo actualizado llamado modelo-actualizado-2018. Se hace el proceso complementario de entrenar CIC-IDS-2017 con el modelo-base-2018 y se obtiene el modelo-actualizado-2017.
6. Se compara el modelo-base-2017 con el modelo-actualizado-2017, y el modelo-base-2018 con el modelo-actualizado-2018.

### *III-D. Construcción del modelo base*

En la figura 2 se puede observar un esquema del modelo adaptado a CIC-IDS-2017. El modelo base que hemos planteado tiene las siguientes características:

- Es un modelo de 6 capas. Se hicieron pruebas usando un número variable de capas, y se observa que con 6 capas se obtiene un buen balance entre el tiempo de entrenamiento y las métricas.
- Está basado en técnicas DNN. El motivo de usar DNN es que tras realizar varias comparaciones entre técnicas como DNN, CNN, LSTM y RNN se vio que los resultados eran muy similares, pero que el tiempo de entrenamiento es siempre inferior con el uso de esta técnica.
- El modelo en la figura 2 presenta 79 posibles entradas y 15 salidas, una por cada clase de ataque. Sin embargo, hay que indicar que este modelo se adapta al número de clases que maneje nuestro dataset.

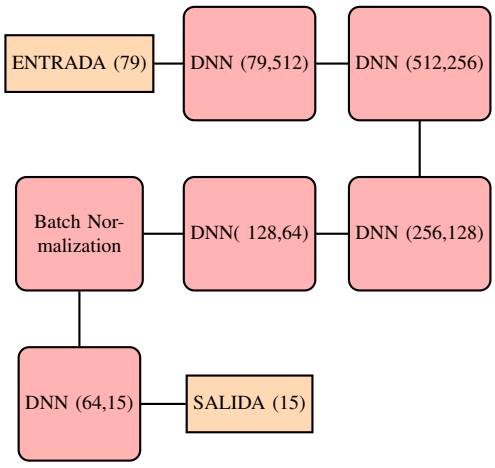



Figura 2. Modelo base

## IV. Evaluación de resultados

### *IV-A. Características del Hardware*

Se ha usado un servicio llamado Google Colab Pro para realizar los experimentos. Las medidas de tiempo fueron obtenidas usando esta especificación de máquina.

- CPU: Intel Xeon processor, 2 cores at 2.3 GHz.
- RAM: 32 GB RAM DDR4.
- GPU: NVIDIA Tesla P100, 16 GB RAM, 3584 CUDA.

### *IV-B. Evaluación de modelos base*

En las tablas V y VI se pueden observar los resultados de los modelos base usados en CIC-IDS-2017 y CIC-IDS-2018. Destacamos 2 promedios globales que son relevantes:

- Promedio por clases. El promedio por clases es un promedio de cada métrica a lo largo de todas las clases del modelo, en este caso 15 clases, la normal más 14 tipos de ataques.
- Promedio por instancias. Las instancias son las líneas del dataset con las que se alimenta el modelo. Este indicador nos muestra la media de cada métrica a lo largo de todas las entradas del modelo.

En cada promedio tenemos 3 métricas. La métrica más relevante es F1-Score, ya que a este valor no le afecta tanto la distribución no balanceada de clases de los datasets.

Podemos observar que en la columna de soporte está anotada la cantidad de instancias procesadas, por clase, y que la suma de esos valores no representa el total de instancias del dataset. El motivo es que la evaluación se realiza sobre un 30 % de las instancias del dataset, el restante 70 % ha sido asignado al entrenamiento y a la validación cruzada. Así tenemos que la clase Heartbleed pasa de 11 instancias a 3.

Estudiando estas métricas podemos hacer varias observaciones:

- El rendimiento es parecido al de los mejores modelos de la literatura. En el caso de CIC-IDS-2017 el valor del promedio por instancias de F1-Score es 0.9964993, el cual es mejor que el valor del mejor modelo de la literatura [5] (ver Tabla IV), que era de 0.9958. En el caso de CIC-IDS-2018 ese valor es de 0.9832973, el cual es ligeramente inferior que el del mejor modelo en este dataset (ver Tabla IV) que es de 0.9865. Esto nos indica que estos modelos son muy similares a los del estado del arte.
- Entrenamiento completo. El tiempo de entrenamiento es de 1170 segundos en CIC-IDS-2017 y 3550 segundos en CIC-IDS-2018. Este tiempo se debe a que debemos entrenar el sistema completo, el cual es una matriz con más de 200.000 parámetros.

Estos resultados son interesantes, pero no cumplen los objetivos planteados, ya que estos modelos se adaptan solo a un dataset, y en el resto obtienen malos resultados. En la próxima sección se analizaran los resultados de la transferencia de aprendizaje para ver si es viable su uso en escenarios más generales.

Tabla V
RESULTADOS DEL MODELO BASE EN CIC-IDS-2017

| Clase | Precision | F1-Score | Soporte |
|---|---|---|---|
| Normal | 0.9988754 | 0.9980274 | 681396 |
| Bot | 0.9949239 | 0.5000000 | 587 |
| DDoS | 0.9987506 | 0.9988676 | 38408 |
| DoS GoldenEye | 0.9951028 | 0.9910584 | 3088 |
| DoS Hulk | 0.9831589 | 0.9908525 | 69037 |
| DoS Slowhttptest | 0.9516035 | 0.9699851 | 1650 |
| DoS slowloris | 0.9895592 | 0.9852729 | 1739 |
| FTP-Patator | 0.9676617 | 0.9741235 | 2380 |
| Heartbleed | 0.0000000 | 0.0000000 | 3 |
| Infiltration | 0.0000000 | 0.0000000 | 11 |
| PortScan | 0.9997059 | 0.9992965 | 47641 |
| SSH-Patator | 0.9924021 | 0.9758621 | 1769 |
| Web Attack Brute Force | 0.3866782 | 0.5559701 | 452 |
| Web Attack Sql Injection | 0.0000000 | 0.0000000 | 6 |
| Web Attack XSS | 0.0000000 | 0.0000000 | 196 |
| Promedios globales | | | |
| Accuracy | | 0.9965522 | 848363 |
| **Promedio por clase** | 0.6838948 | **0.6626211** | 848363 |
| **Promedio por instancias** | 0.9968284 | **0.9964993** | 848363 |
| Epochs | 10 | | |
| Tiempo | 1170 sg | | |

Tabla VI
RESULTADOS DEL MODELO BASE EN CIC-IDS-2018

| Clase | Precisión | F1-Score | Soporte |
|---|---|---|---|
| Normal | 0.9894850 | 0.9935833 | 2409543 |
| Bot | 0.9996897 | 0.9997188 | 51564 |
| Brute Force-Web | 0.9807692 | 0.6071429 | 116 |
| Brute Force-XSS | 1.0000000 | 0.6250000 | 44 |
| DDOS attack-HOIC | 0.9999433 | 0.9999717 | 123541 |
| DDOS attack-LOIC-UDP | 0.7230047 | 0.5811321 | 317 |
| DDoS attacks-LOIC-HTTP | 0.9521882 | 0.9747966 | 103655 |
| DoS attacks-GoldenEye | 0.4389676 | 0.6067388 | 7481 |
| DoS attacks-Hulk | 0.9999322 | 0.9401098 | 83110 |
| DoS attacks-SlowHTTPTest | 1.0000000 | 1.0000000 | 25141 |
| DoS attacks-Slowloris | 0.9939179 | 0.9931628 | 1976 |
| FTP-BruteForce | 0.9998852 | 0.9999426 | 34842 |
| Infiltration | 0.9086426 | 0.2325749 | 29013 |
| SQL Injection | 0.0000000 | 0.0000000 | 16 |
| SSH-Bruteforce | 0.9999704 | 0.9998815 | 33767 |
| Promedios globales | | | |
| Accuracy | | 0.9860102 | 2904126 |
| **Promedio por clase** | 0.8657597 | **0.7702504** | 2904126 |
| **Promedio por instancias** | 0.9871591 | **0.9832973** | 2904126 |
| Epochs | 10 | | |
| Tiempo | 3550 sg | | |

### *IV-C. Evaluación de modelos actualizados*

En las tablas VII y VIII se pueden observar los resultados de los modelos actualizados mediante transferencia de aprendizaje usados en CIC-IDS-2017 y CIC-IDS2018. Podemos hacer varias observaciones:

- Mejora en el promedio por instancias. En el caso de CIC-IDS-2017 el promedio del F1-Score es 0.9976513, el cual es mejor que el del modelo base que es 0.9964993. En el caso de CIC-IDS-2018 el valor es 0.9862191, lo que aumenta el obtenido en el modelo base que era 0.9832973. Hay que tener en cuenta que el modelo actualizado en CIC-IDS-2017 parte del modelo base en CIC-IDS-2018, y no ha llegado a ser entrenado en todas sus capas, solo se ha entrenado en la capa de salida, por lo que ambos modelos son muy interesantes como modelos pre-entrenados para otros datasets, ya que mejoran el promedio por instancias.
- Mejora en el promedio por clases. En CIC-IDS-2017 tenemos un promedio de F1-Score de 0.7702504, en lugar de 0.6626211. Vemos que algunas clases han mejorado, como ”Bot” y ”Web Attack Brute Force”. Estos son ataques presentes en ambos datasets.
- Detección de nuevos ataques. Un caso especial es ”Heartbleed”, ya que este ataque, sin haber sido entrenado en el modelo base, es ahora detectado en el modelo actualizado. Se trata de una nueva anomalía en el dataset que ha podido ser detectada.
- Entrenamiento parcial. El tiempo de entrenamiento es de 179 segundos en CIC-IDS-2017 y 45 segundos en CIC-IDS-2018. Este tiempo tan reducido, en comparación al de los modelos base, se debe a que en lugar de entrenar

el sistema completo, con su matriz de más de 200.000 parámetros, solo hemos entrenado la última capa, de 9.000 parámetros. Los modelos que usan transferencia de aprendizaje son más rápidos que los mejores modelos de la literatura.

Tabla VII
RESULTADOS DEL MODELO ACTUALIZADO 2017

| Clase | Precisión | F1-Score | Soporte |
|---|---|---|---|
| Normal | 0.9990778 | 0.9987742 | 681396 |
| Bot | 0.9907834 | 0.5348259 | 587 |
| DDoS | 0.9980495 | 0.9986209 | 38408 |
| DoS GoldenEye | 0.9877419 | 0.9896574 | 3088 |
| DoS Hulk | 0.9940041 | 0.9964744 | 69037 |
| DoS Slowhttptest | 0.9542254 | 0.9695886 | 1650 |
| DoS slowloris | 0.9883721 | 0.9829430 | 1739 |
| FTP-Patator | 0.9974747 | 0.9966358 | 2380 |
| Heartbleed | 1.0000000 | 1.0000000 | 3 |
| Infiltration | 1.0000000 | 0.1666667 | 11 |
| PortScan | 0.9995800 | 0.9994017 | 47641 |
| SSH-Patator | 0.9649416 | 0.9725182 | 1769 |
| Web Attack Brute Force | 0.4363817 | 0.6021948 | 452 |
| Web Attack Sql Injection | 0.0000000 | 0.0000000 | 6 |
| Web Attack XSS | 0.0000000 | 0.0000000 | 196 |
| Promedios globales | | | |
| Accuracy | | 0.9977451 | 848363 |
| **Promedio por clase** | 0.8207088 | **0.7472201** | 848363 |
| **Promedio por instancias** | 0.9978770 | **0.9976513** | 848363 |
| Epochs | 10 | | |
| Tiempo | 179 sg | | |

Tabla VIII
RESULTADOS DEL MODELO ACTUALIZADO 2018

| Clase | Precisión | F1-Score | Soporte |
|---|---|---|---|
| Normal | 0.9893688 | 0.9940376 | 2409904 |
| Bot | 0.9943052 | 0.9963922 | 51585 |
| Brute Force-Web | 0.0000000 | 0.0000000 | 118 |
| Brute Force-XSS | 0.0000000 | 0.0000000 | 39 |
| DDOS attack-HOIC | 0.9997329 | 0.9998503 | 123527 |
| DDOS attack-LOIC-UDP | 0.7593052 | 0.8619718 | 307 |
| DDoS attacks-LOIC-HTTP | 0.9854774 | 0.9913758 | 103963 |
| DoS attacks-GoldenEye | 0.9844321 | 0.8962326 | 7534 |
| DoS attacks-Hulk | 0.9831440 | 0.9908967 | 83100 |
| DoS attacks-SlowHTTPTest | 0.9990904 | 0.9995450 | 25264 |
| DoS attacks-Slowloris | 0.9409687 | 0.9341347 | 2011 |
| FTP-BruteForce | 0.9982782 | 0.9991384 | 34787 |
| Infiltration | 0.8087971 | 0.2164305 | 28848 |
| SQL Injection | 0.0000000 | 0.0000000 | 16 |
| SSH-Bruteforce | 0.9994075 | 0.9991559 | 33773 |
| Promedios globales | | | |
| Accuracy | | 0.98953142 | 904776 |
| **Promedio por clase** | 0.7628205 | **0.7252774** | 2904776 |
| **Promedio por instancias** | 0.9879650 | **0.9862191** | 2904776 |
| Epochs | 1 | | |
| Tiempo | 45 sg | | |

## V. CONCLUSIONES

En este articulo hemos presentado un modelo basado en una metodología supervisada, que puede ser usado en varios datasets con un rendimiento elevado, y sin necesidad de entrenarlo en la totalidad del modelo. Para ello hemos usado la transferencia de aprendizaje, que es un enfoque interesante para realizar detección de anomalías. El modelo que usamos es un modelo pre-entrenado que puede ser usado en otros datasets, como base, siempre que se adapte el fichero PCAP original del dataset al formato de entrada que hemos usado. Para ello es necesario usar el software CICFlowMeter mencionado anteriormente.

Como trabajo futuro nos vamos a centrar en 2 líneas de investigación: (1) estudiar la capacidad del modelo de detectar nuevos ataques utilizando otros datasets; (2) estudiar la capacidad de detección de anomalías de modelos no supervisados.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Gobierno español a través del proyecto RTI2018-095094-B-C22 ”CONSENT” y PID2021-125962OB-C31 “SECURING”.

## REFERENCIAS

[1] Saxe, J.; Berlin, K. Deep neural network based malware detection using two dimensional binary program features. In *Proceedings of the 10th International Conference Malicious and Unwanted Software (MALWARE), Washington, DC, USA, 20–22 October 2015* pages 11–20.

[2] https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html

[3] Williams, Ronald J.; Hinton, Geoffrey E.; Rumelhart, David E. Learning representations by back-propagating errors”. In *Nature. 323 (6088), 1986* pages 533–536.

[4] Hochreiter, Sepp; Schmidhuber, Jürgen (1997-11-01). Long Short-Term Memory. In *Neural Computation. 9 (8), 1997* pages 1735–1780

[5] Gamage S., Samarabandu J. Deep learning methods in network intrusion detection: A survey and an objective comparison. To appear in *Journal of Network and Computer Applications*

[6] Tang, T.A.; Mhamdi, L.; McLernon, D.; Zaidi, S.A.R.Ghogho, M. Deep learning approach for network intrusion detection in software defined networking. In *Proceedings of the 2016 International Conference Wireless Networks and Mobile Communication (WINCOM), Fez, Morocco, 26–29 October 2016*, pages 258–263

[7] Diro, A.A., Chilamkurti, N., 2018b. Distributed attack detection scheme using deep learning approach for Internet of Things. In *Future Generation Computer Systems 82, 761-768.* doi:10.1016 j.future.2017.08.043

[8] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman. Deep Learning Approach for Intelligent Intrusion Detection System.In *IEEE Access, vol. 7, pp. 41525-41550, 2019*

[9] Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H., 2020. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. In *Journal of Information Security and Applications 50, 102419.* pages 690-710

[10] Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In *Proceedings of the IEEE 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017*, pages 712–717.

[11] Kim, J.; Kim, J.; Thu, H.L.T.; Kim, H. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. In *Proceedings of the 2016 International Conference Platform Technology and Service (PlatCon), Jeju, Korea, 15–17 February 2016*, pages 1–5.

[12] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon and D. Gan. Cloud-Based Cyber-Physical Intrusion Detection for Vehicles Using Deep Learning. In *IEEE Access, vol. 6, pp. 3491-3508, 2018*, doi: 10.1109/ACCESS.2017.2782159.

[13] . Diro, A., Chilamkurti, N., 2018a. Leveraging LSTM Networks for Attack Detection in Fog-to-Things Communications. In *IEEE Communications Magazine 56, 124-130*, doi:10.1109 MCOM.2018.1701270.

[14] S. Althubiti, W. Nick, J. Mason, X. Yuan and A. Esterline. Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection. In *SoutheastCon 2018, St. Petersburg, FL, 2018, pp. 1-5*, doi: 10.1109 SECON.2018.8478898.

[15] C. Yin, Y. Zhu, J. Fei and X. He. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. In *IEEE Access, vol. 5, pp. 21954-21961, 2017*, doi: 10.1109 ACCESS.2017.2762418.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the Inception Architecture for Computer Vision,”2016. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826*, doi: 10.1109/CVPR.2016.308.

# Auto-Aligned Remote Power Analysis through Ring Oscillator-based Sensors

Lilian Bossuet
Univ Lyon, UJM-Saint-Etienne, CNRS
Laboratoire Hubert Curien UMR 5516
F-42023, SAINT-ETIENNE, France
lilian.bossuet@univ-st-etienne.fr

Anis Fellah-Touta
Univ Lyon, UJM-Saint-Etienne, CNRS
Laboratoire Hubert Curien UMR 5516
F-42023, SAINT-ETIENNE, France
anis.fellah.touta@univ-st-etienne.fr

Carlos Andres Lara-Nino
Univ Lyon, UJM-Saint-Etienne, CNRS
Laboratoire Hubert Curien UMR 5516
F-42023, SAINT-ETIENNE, France
carlos.lara@univ-st-etienne.fr

***Abstract*—In recent years, the field of side-channel analysis has observed a revolution in the design of the attack methodology. Conventional approaches which require the use of highly specialized equipment like oscilloscopes and spectrum analyzers, despite highly-precise, might be regarded as impractical in some scenarios. On the other hand, the use of less-accurate internal sensors which can monitor the power footprint of a circuit has risen in popularity. In particular, delay sensors such as those based in Time-to-Digital converters and Ring-Oscillators have shown promising results. These structures are interesting since they can be implemented from regular hardware resources available in most circuits. This means that components already available in the target might be leveraged to implement a side-channel attack. Moreover, it is not really necessary to have direct access to the platform to carry out such an attack; which implies that if there is a remote link such as Ethernet, an adversary might be able to perform Remote Power Analysis (RPA) of the system. So far, the main challenge for the success of this kind of attacks is cutting and aligning the power traces. This is usually achieved through secondary digital channels which carry some trigger information. In this paper, we propose to use a single channel to encode both the power trace and the alignment information. This is achieved by exploiting architectural vulnerabilities of the platform. Our results demonstrate, for the very first time, that RPA traces can be auto-aligned. As a case study we attempt to perform RPA on a serialized implementation of Photon-Beetle, a finalist in the NIST lightweight cryptography standardization process.**

***Index Terms*—Remote Power Analysis, Ring Oscillator Sensors, Trace Alignment, Photon-Beetle**

## I. Introduction

Most of the modern applications of cryptography are considered secure from an algorithmic point of view. It is understood that a formal or statistical proof of security warranties that an adversary cannot compromise the security of the system under reasonable assumptions. However, if the attack model supposes that the attacker has physical access to the platform these notions of security decrease [1]. Under such scenarios, it is necessary to design and implement protections which can mitigate the information that can be retrieved from the hardware platform [2].

A side channel is, in a simplified way, a physical magnitude which can be measured and correlated with the operations being performed in the platform. Electromagnetic emanation, power dissipation, thermal irradiation, energy consumption, and noise are prime suspects for information leakage. Then, we can envision side channel attacks as the application of a sensing strategy with a subsequent information-processing step. For the first part, it is necessary to possess a sensor which can transform the physical magnitude into a digital signal which can be read from a computing system. Such system will then process the samples in a given time. With the adequate quality and volume of data it might be possible to break just any cryptographic implementation. Our work focuses on the first of the two aforementioned tasks, we investigate the process for obtaining these data.

According to Nyquist's theorem [3], the first requirement for the appropriate acquisition of samples is to use a sampling frequency at least two times greater than that of the magnitude under analysis. This is the so called sampling theorem. Secondly, the sensor has to perform a quantization step which will encode the sample into a finite array of possible values, generally through binary representation. The more bits we use, the greater the sampling resolution will be, and subsequently we will obtain a greater fidelity to represent the signal of interest. However, more bits and more samples also imply that we must store and transmit more data. In the end, the goal of a sampling scheme is to adequately represent the physical magnitude with the minimal storage and bandwidth costs.

These are not trivial challenges when the targets of the sensing are digital circuits. Most of the time, the operational frequency of these designs is in the order of megahertz (MHz). Furthermore, the signals to be sampled generate transceiver outputs in the order of micro-volts ($\mu$V). Consequently, a sampling scheme for a digital system must produce a few millions of samples per second with the adequate resolution to represent fluctuations of the millionth part of a volt! Thankfully, the quantization step depends not so much on the scale of the physical magnitude to be measured, but rather on the dynamic range of the signal. Usually less than 20 bits are sufficient to represent such measurements.

It should be evident that performing side channel analysis of a chip requires specialized equipment. Multi-channel digital oscilloscopes and high-frequency spectrum analyzers are some of the most popular tools for this task. Yet, their monetary costs are so high that only private companies and large laboratories can acquire these systems. Then, even if the adversary has the resources to acquire the sensing hardware (which would be the case in state-sponsored attacks), they must also gain physical access to the target platform. These two limitations could be sufficient for a designer to declare that their implementation is totally safe from side channel attacks. Nonetheless, recent works [4] have demonstrated that neither expensive equipment or physical access to the chip are necessary to perform power analysis on a digital circuit.

Using internal sensors created from digital components is not a particularly novel idea [5], [6]. These constructions have been used to monitor the operation of the circuit and assess whether everything is working as intended. However, until recently it was not considered viable that such sensors could be used to retrieve sensitive information from the platform. It is now known that Time-to-Digital converters (TDC) [7] and Ring-Oscillator [8] based sensors (ROS) can be exploited to perform power analysis with moderate accuracy. Moreover, since these circuits can be implemented and operated remotely, performing Remote Power Analysis (RPA) on a chip is just a matter of exploiting some software vulnerability. It is no longer required to assume that the attacker has physical access to the device under attack.

Despite the evident advantages, RPA must still cope with the original sensing problems: obtain more samples with more resolution, while reducing storage and transmission costs. The latter is a particularly interesting problem, since a remote origin of the attack also means that the sampled data must travel back to this origin. Ubiquitous connectivity might provide viability for this approach. The Internet-of-Things (IoT) supposes that everyday objects will be connected to the internet. If the attacker finds a way to access the circuit remotely, then they can leverage hardware components already in the platform [7] to mount an attack and possibly compromise the security of the system.

A problem so far not addressed in the literature is that RPA, just as classical power analysis, requires some additional information to cut and align the traces. When the attacker has access to the platform we assume that they can poke around until they find some *trigger* signal which can be used to determine the start of a *trace*. However, in a remote-attack model this is not so trivial. We cannot assume that the *start* and *done* signals of the architecture under attack will be connected to our sensor. Therefore, the problem of remotely determining the point for cutting and aligning the traces of significant relevance.

In this work, we propose that the same sensors that are used to perform RPA [8] can also be used to encode the necessary information for the alignment of the traces. Thus, our attack model considers that the adversary must only retrieve a train of samples from the internal sensor to perform RPA. We achieve this feat by leveraging strategies previously used for covert-channel communications [9]. To demonstrate the viability of this approach we propose as case study the RPA of a serialized implementation of Photon-Beetle [10], a finalist in the NIST lightweight cryptography standardization process.

Our findings suggests that the proposed approach is viable as we managed to align the power traces with statistical validity. However, there are multiple limitations that must be addressed to reach a point where such attack becomes a practical concern, for example for an IoT platform. In particular, the problem of obtaining and transmitting large volumes of data is critical for carrying out a successful attack on the target circuit. Nevertheless, this is a characteristic of remote-attack which falls out of the scope of this work.

The rest of the paper is structured as follows. In Section II we describe our methodology and the materials used in our experimentation. In this section we also provide a formal description of the proposed attack scenario. The derived results are subsequently reported in Section III. Finally, our findings and conclusions are summarized in Section IV.

## II. Materials and Methods

In this Section we provide details regarding our experimental setup. We describe the different components of our system and outline the guidelines for the proposed attack.

### A. Remote sensors

A delay sensor is a circuit capable of measuring the variation in the delay of an oscillatory digital signal. The propagation delay of an oscillator through digital components will fluctuate as a function of the temperature and voltage of the circuit. Therefore, as the chip performs different processing tasks, these will influence the delay propagation of the target oscillator. This delay will be quantified and then sampled as a digital signal. Both TDCs and ROs can be used to implement delay sensors. In this work, we use the ROS from [8] to perform the data acquisition. Figure 1 illustrates the general architecture of this circuit.

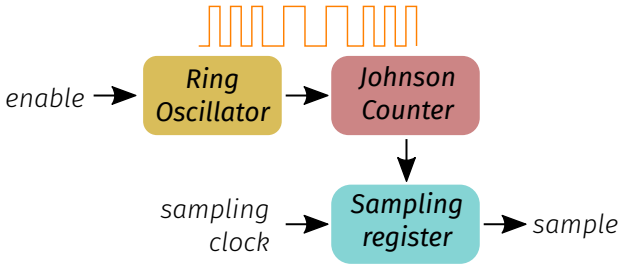



Figure 1: A delay sensor based on a ring oscillator

The main advantage of the ROS from [8] is their acquisition rate. When implemented in the reconfigurable fabric, these sensors allow to use sampling frequencies up to 250 MHz in the Zynq-7000 family of SoC-FPGAs, and greater in newer technologies. This is achieved thanks to the use of a Johnson counter which mitigates the need for carry propagation found in generic binary counters. The sampling register is 8-bits wide and captures the difference between counter states through a combinatorial function. These values are subsequently accumulated into a 32-bit registers which contains the output of the sensor matrix. The dynamic range of this output depends on the sampling frequency, normally less than 10 bits are used.

Under normal operation, a sampling frequency is chosen to clock the sampling register and *read* the resulting values. However, this oscillator can be modified and (for some frequencies) the result will be samples with a different offset but that still convey the change in delay propagation derived from the internal operation of the circuit. We employ this oscillator to produce a frequency-encoding which is then used to align the traces.

### B. Covert channels

Frequency-based covert channels have been explored in [9] with the goal of bypassing Trust-Zone protections in a Zynq-7000 SoC-FPGA. The authors demonstrated that it was possible to modify the output of Phase Locked Loops (PLL) in

the circuit through different modulation strategies in order to encode a message. This would create a covert channel between different components that were not supposed to communicate with each other. For example, a trusted application would exchange information with an untrusted hardware accelerator.

FPGA-enabled SoCs such as the Zynq-7000 boards feature different clocks which flow from the processing system into the programmable logic. These are sourced from a group of main PLLs and through a set of multipliers and dividers produce the desired frequency output. These multipliers and dividers are simply digital values stored in registers which can be modified from the processors of the SoC. In our work, we employ a Zynq Ultrascale+ SoC-FPGA as implementation platform. For these systems the clock modulation can be performed in a similar way as in previous generations of the technology, see Fig. 2.

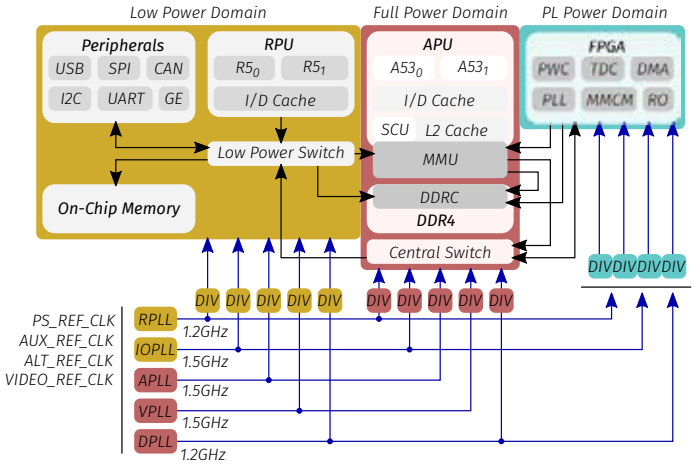


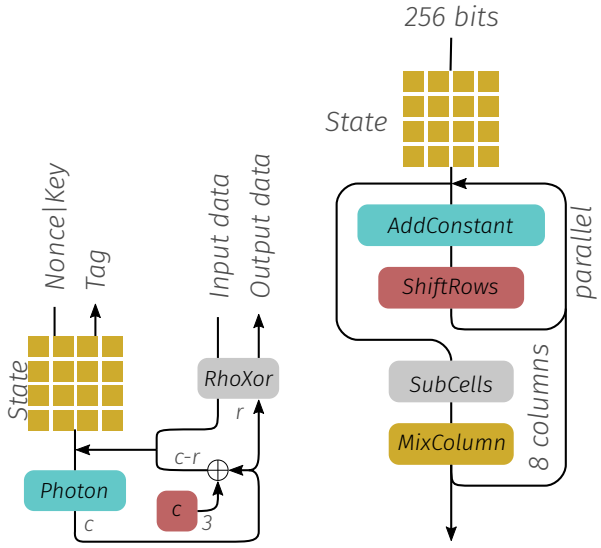

Figure 2: The clock tree of Zynq Ultrascale+ SoC-FPGAs

However, we do not encode a complex message in the covert channel. We use frequency modulation to modify the sampling clock of the ROS and produce a discernible pattern in the resulting sample train. As the circuit under attack is called from the processor, the malicious application modifies the target clock with each call. This pattern will encode the necessary information to perform the alignment of the traces.

### C. Target architecture

As a case study we attempt to perform RPA on a serialized implementation of Photon-Beetle [10]. The algorithm under attack is illustrated in Fig. 3. The main operations of this authenticated cipher are similar to those of AES. In the underlying permutation ($P_{256}$), the state, arranged as a matrix of eight rows and eight 4-bit columns, is first XOR-ed with some round constants (*AddConstant*), then substituted (*SubCells*), shifted (*ShiftRows)* and finally mixed (*MixColumn*). These operations are repeated over 12 rounds. Photon-Beetle uses $P_{256}$ to process each block of the message. The rate for the architecture under study is 32-bits.

In the architecture under attack, the application of the *ShiftRows* and *SubCells* operations have been swapped in order to merge *AddConstant* and *ShiftRows* into *AddShift* and *SubCells* and *MixColumn* into *SubMix*. This change in the order of operations is applied in order to serialize *SubMix* in a single step. Given the characteristics of the *SubCells*




(a) Photon-Beetle (b) Photon ($P_{256}$)

Figure 3: The specification for Photon-Beetle

transformation and the granularity of *ShiftRows*, it does not have any effect in the output of the algorithm or its resilience against power analysis.

The *AddShift* operation is performed in a single cycle and the *SubMix* operation is serialized over eight cycles. The additional operation in Photon-Beetle is the mathematical component $\rho$ which is an XOR and permutation (*RhoXor*) performed before every call to the underlying permutation.

Every round of $P_{256}$ takes 9 cycles, since there are 12 of such rounds, the latency of this permutation is 108 cycles. Counting the application of *RhoXor*, a total of 109 cycles are required per call. Then a total of $109\times(\lceil a\rceil/32+\lceil m\rceil/32+1)$ cycles are spent to process an $m$-bits message with $a$-bits of associated data. Our experiments take 1,853 cycles for a 256-bits message with 256-bits of associated data.

### D. Attack scenario

Our attack scenario consists of two main actors, see Fig. 4. These are two C-language applications running on bare metal. First, we have an application (*A0*) on the ARM CortexA53-0 processor which can query the hardware architecture under attack. This actor can also modify the frequency of the FPGA clocks. The second party is another application (*A1*) on the ARM CortexA53-1 which can query the ROS.

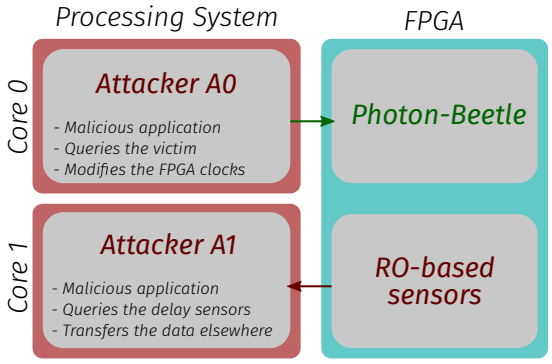



Figure 4: The proposed attack scenario

The proposed scenario assumes that *A0* has the necessary access level to modify the control registers of the SoC. We assume also that *A0* can query the hardware accelerator, which holds the cryptographic secrets but only replies to legitimate requests. For *A1* the assumptions are more usual. We have an application which comes with a custom hardware acceleration and will perform some given task. Except that the accelerator contains a bank of ROS and *A1* can retrieve the samples and transfer them to the remote origin. The exact mechanism for transferred the data is not addressed for brevity.

## III. Experimental results

We used a TE0802 development board for our experimentation. This platform features a Zynq Ultrascale+ SoC-FPGA (xczu2cg-sbva484-1-e). We used the AMD-Xilinx 2022.1 toolchain, creating the hardware specification in Vivado and programming and launching the applications through Vitis.

### A. PLL's transition delay

We first studied the PLL response times to assess whether it was viable to modify the frequency of these components from the processing system. Figure 5 shows the results for this experiment. We first enabled a digital trigger (TRIGGER) from the processor and then requested a frequency change from 100 MHz to 150 MHz. We sampled the PLL response, as well as the MSB in the output of the sensor which would indicate that the new frequency had been detected. We estimated that it takes approximately $400ns$ for the PLL to start the process to modify its output. Then $200ns$ are used to perform the requested change; during this time the output of the PLL is unstable. Finally, it takes a few additional nanoseconds for the processor to be notified of the change.

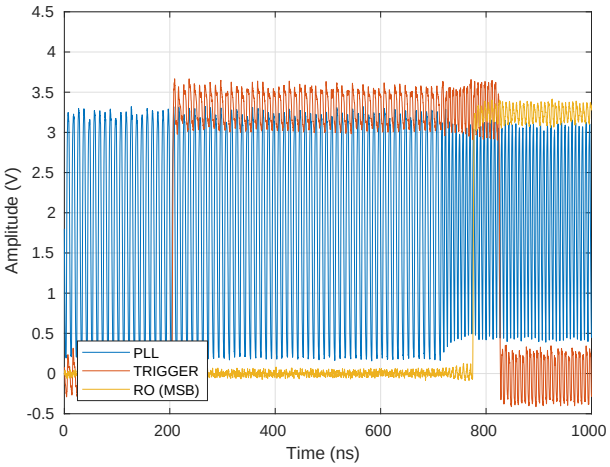



Figure 5: The step response of a PLL in the Zynq Ultrascale+ SoC-FPGAs. Obtained with a digital oscilloscope at 10 GSps.

These findings suggest that it is possible to employ frequency modulation to align the traces. Since the PLL calls are blocking, it is not necessary to account for the transition delay in the acquisition. Nonetheless, if the latency of the architecture under attack is much smaller than $600ns$, the traces will contain mostly spurious data.

### B. Acquisition rate

As illustrated in Fig. 4, we assumed that the *A1* application could retrieve the samples from the ROS. These two modules were connected through an AXI-HP channel clocked at 300 MHz; a typical link in Zynq Ultrascale+ chips.

Recall from Subsection II-C that the underlying operations of Photon-Beetle include *RhoXor* (one cycle), *AddShift* (one cycle), and *SubMix* (eight cycles). To determine the acquisition rate of the processor we sampled some digital triggers that correspond with the processing of these operations. The details for this experiment are shown in Fig. 6.

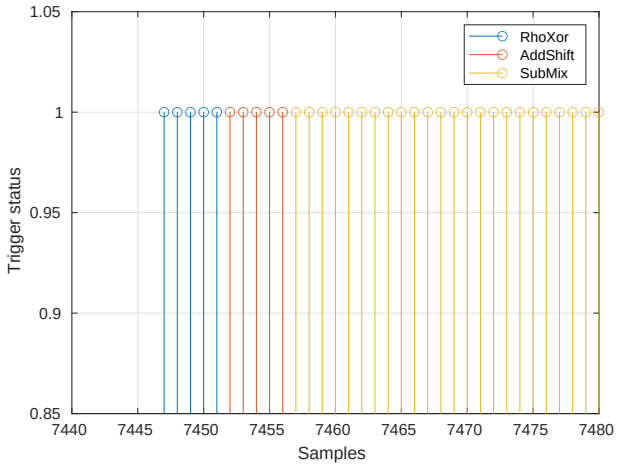



Figure 6: The sampling rate of the processor on the digital triggers of the Photon-Beetle architecture

By clocking the architecture under attack at 1 MHz we could retrieve five samples per cycle. These results indicated that the processor could retrieve 32 bits of data with a delay of 200 ns. Since we only included one sampler per transaction, our sampling rate was approximately 5 MSps.

### C. The covert channel

We used the algorithm in Fig. 7 to encode the synchronization information into the covert channel. Before each call to the cipher, the *A0* application modified the sampling frequency of the ROS. This value would iterate between two predefined values. For simplicity, *A0* would perform this task in an infinite loop.

**Require:** $f_1$, $f_2$ a pair of sampling frequencies
  $f_{RO} = f_1$
  **while** TRUE **do**
    $f_{RO} \leftarrow f_{RO} = f_1 ? \quad f_2 : f_1$
    `PhotonBeetle`(ENCRYPT)
  **end while**

Figure 7: The channel modulation strategy

Figure 8 illustrates the results from this modulation strategy. In this graph we plot a series of 50,000 samples retrieved by the *A1* application from the ROS. On first sight it was possible to clearly differentiate between the iterative stages of the operation of the chip. We could identify fragments of the channel which had a mean of $\sim$140 counts and others with

a mean of ∼60 counts. Nonetheless, we could also note that there was a significant overlap between the sample windows. These outliers could be mitigated with the application of a moving average filter over 16 samples. This filer was only used for segmenting the channel into traces, however. Thanks to the evident offset in the windows a simple threshold-detection method could be used for this task.

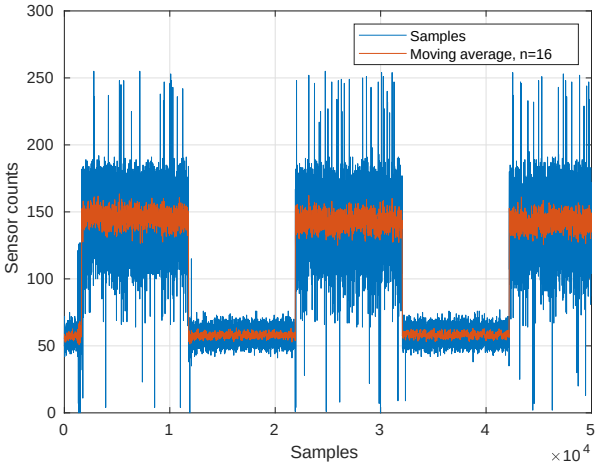



Figure 8: A sequence of samples when the sampling frequency of the ROS is modified from the application

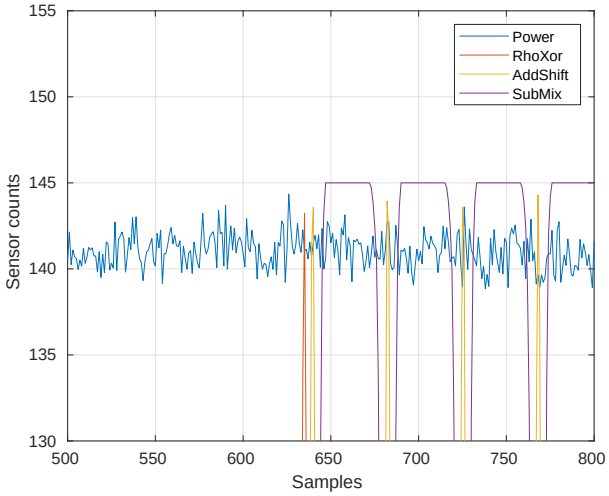
### D. Two sets of traces

A result of modifying the sampling rate of the ROS is that we obtained two sets of samples which were fundamentally different. Yet, thanks to the sub-sampling process performed by the processor, in the end we obtained traces of the same length. Since cryptanalysis methods like differential power analysis do not depend on the magnitude or period of the samples, it would be possible to process the traces as a single set. However, to be rigorous about their study we processed both sets separately. In the following, we use the notation $traces_H$ to identify the set with $\bar{x} \approx 140$ and $traces_L$ to identify the set with $\bar{x} \approx 60$.

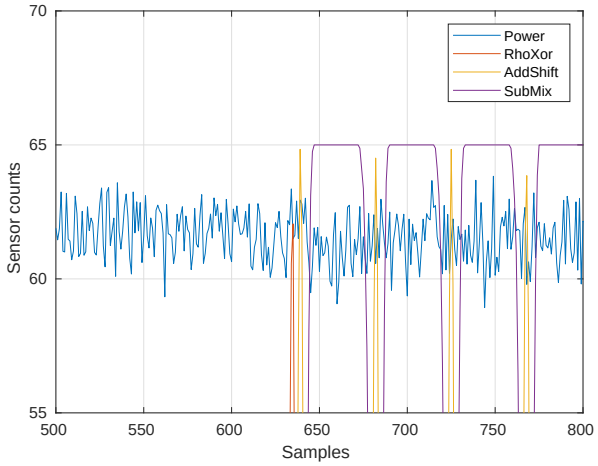
### E. Validation and analysis

To determine whether the alignment was successful we sampled the digital triggers shown in Fig. 6 and attempted to perform the automatic alignment of these traces. In the same samples, we included the output of the ROS and also performed their automatic alignment. By automatic alignment we mean that we only had to cut the sample into segments and classify the traces into the corresponding set. Then we could obtain the average value for each set. Recall that the processor could retrieve 32-bit samples from the FPGA, but the output of the sensor was under 10 bits; thus we could retrieve the triggers from the exact sampling intervals by appending them to each sample.

As shown in Fig. 9, the triggers were easily identifiable after the automatic alignment, even those with only five samples (*RhoXor*, *AddShift*). However, as it was also evident, the proposed alignment method was not perfect. Nonetheless, as the number of observations is increased, the similarity between the sets of auto-aligned traces and a *golden* set of




(a) Automatic alignment of $traces_H$




(b) Automatic alignment of $traces_L$

Figure 9: Visual results for the automatic alignment, the graphs shown represent the average of 500 traces

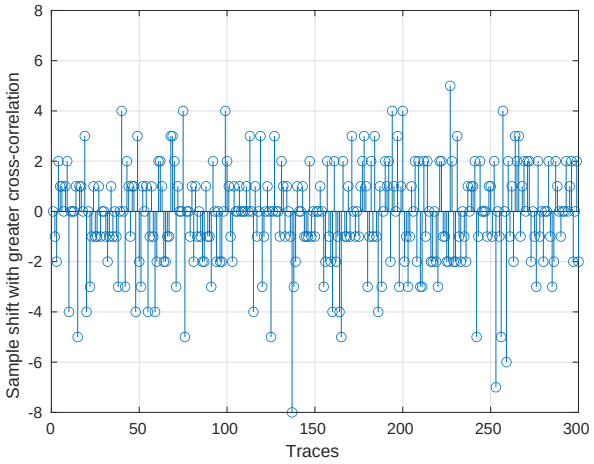
aligned traces should grow. To measure this similarity we used the digital triggers to perform the alignment and create two golden sets of traces ($golden_H$ corresponding to $traces_H$, and $golden_L$ corresponding to $traces_L$). Then we evaluated their similarity to our resulting sets. First we obtained simple statistics such as their mean ($\bar{x}$) and standard deviation ($\sigma_x$). These results re provided in Table I. Despite the similarity in the results, these values are not really meaningful for unstructured signals.

| Metrics | $\bar{x}$ | $\sigma_x$ |
|---|---|---|
| $traces_H$ | 140.8311 | 0.0071 |
| $golden_H$ | 140.8315 | 0.0071 |
| $traces_L$ | 61.6604 | 0.0150 |
| $golden_L$ | 61.6600 | 0.0151 |

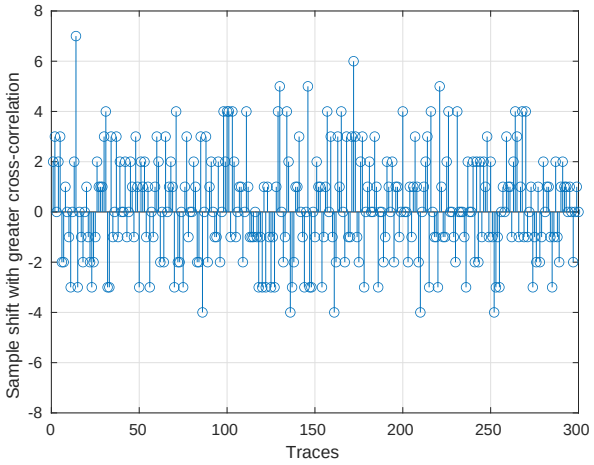
Table I: Basic statistical analysis

Next, we used cross-correlation to determine whether the alignment proposed was optimal or not. For each one of the traces in the auto-aligned sets we computed their cross-correlation with the respective golden-aligned trace. For this

kind of experiment we expected our results to be close to zero, as that would imply that the traces were aligned properly. Our preliminary results showed that the traces in the $traces_H$ set were better aligned than those in $traces_L$. This suggested that the transition from a lower to a higher sampling frequency was more consistent than the inverse operation. Therefore, we simply aligned all the traces in $traces_L$ to the rising edge.




(a) Between $traces_H$ and $golden_H$




(b) Between $traces_L$ and $golden_L$

Figure 10: Cross-correlation of the auto-aligned traces

The results of this analysis, illustrated in Fig. 10, indicate that a 81% of the traces in $traces_H$ were well-aligned with only a miss-alignment of two samples or less. The 73% of the traces in $traces_L$ satisfied the same condition. Bear in mind that five samples represent a single cycle of the architecture under attack, and that our case study has a processing latency of 1,853 cycles. Therefore, if we are more lenient and accept a miss-alignment of under a cycle of this architecture (five samples or less) we obtain that over 99% of the traces in both sets are properly aligned.

## IV. Conclusions and Future Work

In this paper we have described a new strategy for the automatic alignment of traces in the scope of RPA attacks. Our findings suggest that the proposed method is viable under certain assumptions. For example, the processing latency of the architecture under attack should not be smaller than the transition delay of the PLL. As case study we attempted to perform RPA on the authenticated cipher Photon-Beetle. At this point, the limitations of RPA made it difficult to conduct a power analysis attack, nonetheless the proposed alignment method can brings us closer to this end. See Appendix.

## Acknowledgements

The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-19-CE39-0008 (project ARCHI-SEC).

## References

[1] F.-X. Standaert, *Introduction to Side-Channel Attacks*. Boston, MA: Springer US, 2010, pp. 27–42.

[2] A. Moradi, M. Kasper, and C. Paar, "Black-Box Side-Channel Attacks Highlight the Importance of Countermeasures," in *Topics in Cryptology – CT-RSA 2012*, O. Dunkelman, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–18.

[3] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing (4th Edition)*. USA: Prentice-Hall, Inc., 2006.

[4] J. Gravellier, J.-M. Dutertre, Y. Teglia, P. L. Moundi, and F. Olivier, "Remote Side-Channel Attacks on Heterogeneous SoC," in *Smart Card Research and Advanced Applications*, S. Belaïd and T. Güneysu, Eds. Cham: Springer International Publishing, 2020, pp. 109–125.

[5] J. J. L. Franco, E. Boemo, E. Castillo, and L. Parrilla, "Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage," in *2010 VI Southern Programmable Logic Conference (SPL)*, 2010, pp. 133–137.

[6] S. Henzler, *Time-to-Digital Converter Basics*. Dordrecht: Springer Netherlands, 2010, pp. 5–18.

[7] J. Gravellier, J.-M. Dutertre, Y. Teglia, and P. L. Moundi, "SideLine: How Delay-Lines (May) Leak Secrets from Your SoC," in *Constructive Side-Channel Analysis and Secure Design*, S. Bhasin and F. De Santis, Eds. Cham: Springer International Publishing, 2021, pp. 3–30.

[8] J. Gravellier, J.-M. Dutertre, Y. Teglia, and P. Loubet-Moundi, "High-Speed Ring Oscillator based Sensors for Remote Side-Channel Attacks on FPGAs," in *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, 2019, pp. 1–8.

[9] E. M. Benhani and L. Bossuet, "DVFS as a Security Failure of TrustZone-enabled Heterogeneous SoC," in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2018, pp. 489–492.

[10] D. Bao, A. Chakraborti, N. Datta, J. Guo, M. Nandi, T. Peyrin, and K. Yasuda, "PHOTON-Beetle Authenticated Encryption and Hash Family," National Institute of Standards and Technology, NIST Lightweight Cryptography – Finalists, 2021.

## Appendix

With roughly 4,000 traces we can distinguish some patters which are assumed to be correlated with the operation of the circuit under analysis. This is illustrated in Fig. 11.

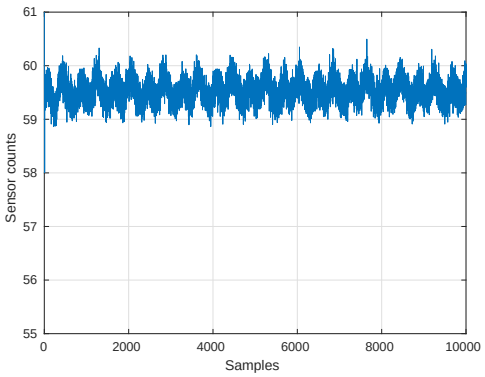



Figure 11: The average of ∼4,000 traces obtained remotely

# Análisis de ciberseguridad para cerraduras inteligentes

Cándido Caballero-Gil
Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna
ccabgil@ull.edu.es

Jezabel Molina-Gil
Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna
jmmolina@ull.edu.es

***Resumen*—El auge de las cerraduras inteligentes ha hecho que estén presentes en muchos hogares. Esto se debe a su diseño fácil de usar y a la variedad de opciones que ofrecen. Sin embargo, por defecto, el uso de estas cerraduras no hace que los hogares sea más seguro ya que su uso, ha llevado a la aparición de nuevas técnicas de robo basadas en ciberataques. Esto implica que su uso pueda verse afectado por la desconfianza del público. El objetivo de este trabajo es realizar una investigación sobre la seguridad de dos cerraduras inteligentes existentes en el mercado y que funcionan vía Bluetooth. En concreto, se realiza un análisis sobre el modelo de seguridad que presenta cada una de ellas y se intentan realizar ataques para poder llevar acabo la apertura de las mismas.**

***Index Terms*—cerraduras inteligentes, ciberseguridad, Bluetooth, ciberataque**

## I. Introducción

Las cerraduras inteligentes no dependen de llaves tradicionales. De esta manera se puede abrir o desbloquear una puerta mediante el uso de un dispositivo móvil, un control remoto, una tarjeta NFC o incluso con la huella dactilar o un código numérico. Además de este tipo de accesos, muchas de estas cerraduras se pueden considerar inteligentes dado que permiten guardar y consultar un registro de accesos, comunicarse y sincronizarse con calendarios para permitir o retirar accesos dependiendo de condiciones establecidas, abrir de forma remota o incluso, alguna de estas cerraduras permiten el acceso mediante un código al tocar la puerta. Por todo esto, muchas familias y empresas han empezado a utilizar este tipo de cerraduras dada la gran flexibilidad que proporcionan. Además, el auge del alquiler vacacional también está ayudando a que más viviendas o establecimientos utilicen esta forma de apertura de puertas, debido a la posibilidad de dar y quitar el acceso bajo demanda.

El objetivo de este trabajo es estudiar la seguridad [1] que presentan las cerraduras inteligentes. La técnica utilizada para este propósito consiste en la captura de paquetes con la herramienta Wireshark [2] y un Bluetooth Sniffer [3], mientras se abre o desbloquea la puerta donde está instalada. Con los datos obtenidos se intenta replicar la información en la cerradura y comprobar si es posible abrirla.

En concreto, para el desarrollo de este trabajo se han utilizado dos modelos de cerraduras diferentes, con funcionamiento Bluetooth. Se realiza un análisis previo, comparando los sistemas de seguridad que presentan cada una de ellas. Posteriormente, se lleva a cabo un proceso de recolección y analisis de datos y se realiza una investigación de los resultados obtenidos. Por último, se estudia la viabilidad de ataque a cada una de ellas.

El resto del artículo se estructura como sigue: el capítulo II aborda los antecedentes y el estado actual de los diferentes tipos de cerraduras inteligentes disponibles en el mercado. El capítulo III introduce el problema a tratar en el proyecto y las tecnologías y herramientas utilizadas para la investigación. El capítulo IV trata de las características de las dos cerraduras que se van a analizar. En dicho apartado se proporcionará información de cada cerradura y su funcionamiento, así como de la implementación realizada para llevar a cabo un posible ataque. Además, se muestra una comparación entre las dos cerraduras, específicando las diferencias y similitudes en sus modelos de seguridad. El capítulo V presenta las conclusiones sobre los resultados obtenidos en esta investigación.

## II. Estado del arte

Actualmente existe una gran variedad de cerraduras inteligentes en el mercado, entre las que el usuario puede elegir según sus preferencias. Según datos del ministerio de Interior de España, el número de hurtos se reduce cada año, pero el acceso con violencia a viviendas aumenta [4]. En este sentido, los propietarios optan por instalar mecanismos de seguridad en sus hogares y empresas, entre los que se pueden encontrar este tipo de cerraduras. La seguridad depende del sistema de cierre que se instale, pero en general, este tipo de cerraduras son más seguras que las tradicionales porque es posible tener un registro de cuándo se ha accedido y por medio de qué dispositivos. Además, tienen la ventaja de que no es necesario cambiar la llave si se pierde, ya que en su lugar, simplemente se tendría que cambiar el código si se sospecha que alguien conoce la contraseña. Este tipo de cerradura proporciona comodidad y seguridad, además de ser sencilla de manejar.

El artículo [5] presenta ataques basados en vulnerabilidades de hardware de microarquitectura y los efectos secundarios que producen en el sistema. Además, en este trabajo se presentan mecanismos de seguridad que pueden implementarse para hacer frente a algunos de estos ataques. La mayoría de los mecanismos de seguridad se dirigen a un pequeño conjunto de vectores de ataque o a un único vector de ataque específico. Dado que existen muchos vectores de ataque, es necesario encontrar soluciones que protejan contra una gran variedad de amenazas. Este estudio pretende informar a los diseñadores sobre los efectos secundarios relacionados con los ataques y los mecanismos de detección que se han descrito en la literatura. Para ello, se presenta en este artículo dos tablas en las que se enumeran y clasifican los efectos secundarios

y los mecanismos de detección en función de los criterios indicados.

Los autores de [6] presentan ataques a aplicar sobre las cerraduras, con el fin de responder a las deficiencias existen en las cerraduras inteligentes de las puertas en la actualidad. Los resultados concluyen que existen varios problemas en las cerraduras inteligentes actuales, donde la más grave ha sido reportada bajo divulgación responsable al fabricante. Las cerraduras investigadas muestran vulnerabilidades en la consistencia del estado, las políticas de contraseñas y el mecanismo de restablecimiento de contraseñas.

## III. Tecnologías utilizadas

El problema al que se enfrenta esta tecnología, es que a pesar de que la mayoría de las cerraduras Bluetooth incorporan sistemas de encriptación en sus aplicaciones, es posible que, si no se adaptan las medidas de seguridad necesarias por parte del fabricante, sus claves puedan ser desencriptadas. Esto pone en peligro tanto la seguridad del hogar como la del negocio.

Hablar de seguridad 100 % no sería correcto ya que siempre existe el riesgo de que algo salga mal, incluso cuando se toman todas las precauciones necesarias. Las cerraduras inteligentes no son una excepción, a pesar de que en muchos casos no pueden ser forzadas con técnicas como el bumping. El bumping consiste en introducir en el cilindro de la cerradura una llave manipulada y golpear la misma con el fin de hacer ”bailar” los pistones del cilindro. Esto consigue que los pistones del cilindro salten simultáneamente cuando la llave es golpeada, permitiendo el giro de la llave y por tanto la apertura de la puerta de seguridad.

Vivimos en una época en la que la piratería informática y las brechas de seguridad van en aumento [7], lo que también pone en riesgo las cerraduras electrónicas o digitales. Por ello, en este trabajo se ha estudiado un método a través del cual se podría intentar acceder a las comunicaciones de la cerradura para comprobar si es posible romper su encriptación y, en consecuencia, desbloquear la puerta.

### *III-A. Herramientas*

*III-A1. Wireshark:* Wireshark es el analizador de paquetes más conocido y utilizado en el mundo [8]. Gracias a este programa, se puede capturar y analizar en detalle todo el tráfico de red que entra y sale de un PC. Este programa gratuito permite realizar una inspección profunda de cientos de protocolos, ya que soporta protocolos de capa física, protocolos de enlace, protocolos de red, capa de transporte y también capa de aplicación.

Esta herramienta nos permite capturar el tráfico de red en tiempo real y, una vez capturados todos los paquetes, realizar un análisis detallado. Wireshark es el programa utilizado en este trabajo para analizar los paquetes enviados entre el dispositivo móvil y ambas cerraduras.

*III-A2. Adafruit bluefruit LE Sniffer:* Adafruit bluefruit LE Sniffer está programado con una imagen de firmware que lo convierte en un sniffer de Bluetooth Low Energy (BLE) fácil de usar. Puede capturar los intercambios de datos entre dos dispositivos BLE, introduciendo los datos en Wireshark donde se puede visualizar la información a nivel de paquete, con descriptores útiles para una lectura más cómoda y sencilla. Este sniffer Bluetooth es el empleado para capturar los paquetes enviados entre ambos dispositivos.

*III-A3. nRF Sniffer para Bluetooth LE:* nRF Sniffer for Bluetooth LE es una herramienta útil para depurar y aprender sobre aplicaciones de Bluetooth Low Energy. Permite visualizar casi en tiempo real los dispositivos Bluetooth LE cercanos disponibles. Gracias a esta herramienta y con el sniffer conectado, podemos capturar el tráfico en Wireshark enviado desde la cerradura.

*III-A4. Antena Bluetooth CBT40NANO:* El nanoadaptador Bluetooth CBT40NANO se utiliza para crear una conexión inalámbrica con otros dispositivos Bluetooth. En este trabajo se utiliza para crear una conexión a través de una máquina virtual, que ejecuta el sistema operativo Linux, con la cerradura. De este modo, se puede replicar el tráfico necesario para desbloquear la cerradura.

*III-A5. Gatttool:* Gatttool [9] es una herramienta que permite obtener información o manipular atributos de un dispositivo BLE. En este trabajo se utiliza para realizar la escritura de claves en las cerraduras.

### *III-B. Cerraduras Inteligentes*

Las cerraduras inteligentes aportan una forma de aumentar la flexibilidad y en menor medida, la seguridad en el hogar u oficina. Son un tipo de cerradura electrónica que utiliza un teclado encriptado o un lector de tarjetas RFID para permitir el acceso a una puerta cerrada. Ofrecen una serie de ventajas con respecto a las cerraduras mecánicas tradicionales, como la posibilidad de añadir o eliminar acceso a usuarios de manera más fácil y flexible, y la posibilidad de recibir notificaciones cuando alguien accede por una puerta.

En concreto, las cerraduras utilizadas en este trabajo son motorizadas, es decir, hay un motor que hace girar la llave física automáticamente. Su instalación es muy sencilla, se adapta al cilindro que tiene la puerta y se introduce una de las llaves físicas para abrir la puerta. No es necesario desmontar la cerradura original ya que se pega directamente sobre la que ya existe y se adhiere a la puerta mediante un adhesivo 3M muy resistente o tornillos.

Las cerraduras inteligentes que se han analizado en este trabajo para hacer una comparación entre ellas se presentan a continuación.

*III-B1. Cerradura inteligente Sherlock S2:* Una de las cerraduras analizadas en este trabajo es la Sherlock S2 de Xiaomi [10] (ver Figura 1). Es una de las cerraduras más vendidas y mejor valoradas del mercado, siendo además una de las más baratas y fáciles de instalar. La cerradura Sherlock tiene múltiples formas de desbloqueo. Utilizando la llave original, una SmartKey (un mando a distancia que se configura desde la app móvil para accionar la cerradura), el desbloqueo por huella dactilar (deslizando el dedo por el lateral de la cerradura) y, por último, utilizando la app móvil Sherlock. Esta cerradura ha sido seleccionada para su análisis por la falta de información de seguridad que aporta el fabricante. En concreto se utiliza el método de apertura con app móvil para analizar el tráfico en este trabajo.

*III-B2. Cerradura inteligente Nuki:* La segunda cerradura analizada en esta investigación es la cerradura inteligente Nuki [11] (véase la figura 1), producto de Nuki Home Solutions GmbH, de Austria. Esta cerradura ha ido ganando

Figura 1. Cerraduras inteligentes Sherlock y Nuki

considerable popularidad en los últimos años ya que destaca por su seguridad y su fácil instalación. La cerradura Nuki es la primera cerradura inteligente de Europa que abre las puertas con la ayuda de un teléfono. Además, es la primera cerradura más flexible, es decir, puede instalarse en casi todas las cerraduras europeas existentes. Nuki desbloquea automáticamente la puerta al llegar a casa a través de Bluetooth y también la vuelve a bloquear al salir. Por ello, Nuki es bastante famosa en el mercado europeo, ya que aporta gran comodidad y sencillez al usuario. Nuki es compatible con otros métodos de apertura, además de con un teléfono inteligente, la cerradura puede abrirse con un mando a distancia, un teclado numérico o con la llave normal que también puede utilizarse si la puerta tiene un cilindro de doble embrague. La cerradura Nuki ofrece una gran comodidad de uso, ya que la puerta puede abrirse desde un teléfono móvil, un smartwatch o una Tablet. Además, se puede dar acceso temporal o permanente a otras personas, como familiares, amigos o servicios de limpieza para un horario determinado.

Además, gracias a su sistema de manos libres, la puerta se abrirá cuando detecte el móvil, proporcionando una gran comodidad al usuario cuando venga cargado o simplemente para una mayor rapidez. En el caso de que se trate de una empresa o alojamiento turístico como Airbnb, se puede proporcionar un código temporal a los usuarios para evitar la pérdida y copia de llaves. Esta cerradura permite tener varios usuarios registrados al mismo tiempo, enviándoles un código de invitación, que puede ser retirado en cuanto se desee. Esta cerradura ha sido seleccionada para este trabajo por ser una de las más usadas en el mercado europeo.

## IV. Análisis de seguridad de las cerraduras inteligentes

El proceso de bloqueo y desbloqueo de la cerradura es muy sencillo. El usuario debe entrar en la app con el nombre de usuario utilizado para registrarse previamente. A continuación, la app le llevará al menú principal de la aplicación, donde si ya hay un bloqueo asociado, aparecerá con las opciones de desbloquear, deslizando el dedo hacia la derecha, o bloquear, deslizando el dedo hacia la izquierda. Además, esta aplicación tiene la ventaja de permitir tener varios bloqueos para un mismo usuario y gestionarlos en función de dónde se encuentre.

A continuación, se estudiará cómo funciona el envío de información entre la aplicación y la cerradura. El objetivo de este proceso es poder capturar el tráfico cuando la cerradura se conecta con el móvil y le envía la clave para desbloquear la puerta, para luego intentar replicar ese tráfico.

En primer lugar, es necesario tener a disposición la primera herramienta para realizar la conexión entre el ordenador, la aplicación y la cerradura. En este trabajo se ha utilizado el sniffer Adafruit bluefruit LE, que permite capturar y analizar los paquetes en tránsito entre los dispositivos.

Para analizar el tráfico capturado en la aplicación Wireshark, primero se tiene que enlazar la información recibida por el sniffer. Para ello se utiliza el programa nRF Sniffer for Bluetooth LE. Esta herramienta permite ver en tiempo real todos los dispositivos que están siendo capturados por el sniffer con el fin de capturar los datos para posteriormente filtrar ese contenido y ver la información en detalle. Gracias a esta herramienta es posible ver todos los dispositivos Bluetooth cercanos disponibles con sus respectivas direcciones MAC.

### *IV-A. Análisis de seguridad de la cerradura Sherlock*

El programa nRF Sniffer detecta la cerradura Sherlock. Una vez localizado el dispositivo, el siguiente paso es analizar su tráfico en Wireshark. Para ello, se selecciona la cerradura y se pulsa la tecla 'w'. Esto redireccionará directamente a la herramienta Wireshark, filtrando sólo el tráfico del dispositivo bluetooth seleccionado.

Otro método para capturar el tráfico entre los dos dispositivos, implementado en principio para el desarrollo de este trabajo, es a través de la función HCI snoop log [12]. Se trata de un archivo de registro que contiene todas las transmisiones bluetooth que se han realizado desde un teléfono.

Para poder utilizar esta funcionalidad, es necesario activar el modo desarrollador en el dispositivo móvil y, una vez activado:

- Activar el registro HCI de Bluetooth, para habilitar los registros.
- Activar depuración USB, para luego poder extraer los registros vía USB y verlos en un ordenador.

A continuación, se activa y desactiva el Bluetooth para habilitar la recogida de datos. Una vez habilitado el Bluetooth, se puede desbloquear la cerradura para capturar el tráfico.

Para analizar los datos recogidos en un ordenador, el registro se pasa por una herramienta de línea de comandos *adb (Android Debug Bridge)*, que permite la comunicación con un dispositivo, y finalmente se genera un archivo *.log* que puede ser analizado en Wireshark. Este método se descartó porque el uso de un sniffer Bluetooth agiliza el proceso de recogida de datos.

Una vez se tiene la iformación en Wireshark, se puede empezar el análisis. Para filtrar el tráfico de forma más eficaz, se utiliza el filtro *'btatt'* [13]. Este filtro muestra todos los protocolos relacionados con el tráfico Bluetooth y, por tanto, se puede obtener sólo los paquetes que se envían entre la

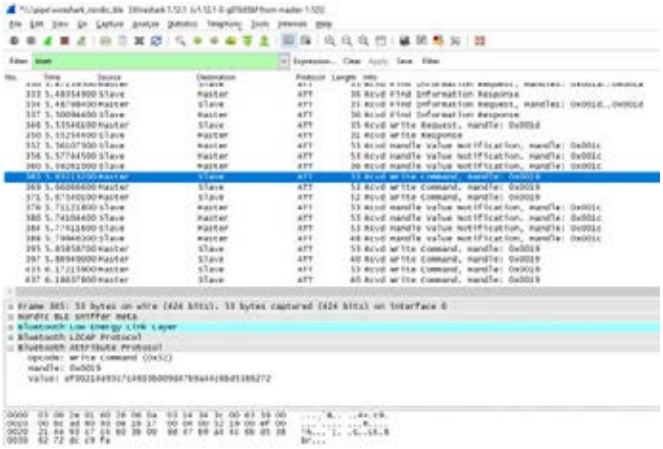


Figura 2. Captura de paquetes de Sherlock utilizando Wireshark

cerradura y la aplicación. Una vez introducido el filtro, se empieza a capturar el tráfico bloqueando o desbloqueando la cerradura. De esta forma, empezará a llegar información al programa, que posteriormente utilizaremos para replicar el tráfico y comprobar si es posible atacarlo. A continuación se explica brevemente el detalle de la lista de paquetes capturados en el proceso.

La Figura 2 muestra la comunicación entre el Maestro (la cerradura Sherlock) y el Esclavo (la aplicación móvil), es decir, el recurso que envía o recibe la información y su destino.

El protocolo utilizado para la conexión es ATT (Attribute Protocol) [14], un protocolo basado en atributos perteneciente a los protocolos BLE, con arquitectura cliente-servidor, que permite el intercambio de información. Este protocolo define cómo se representan los datos y los métodos por los que se pueden leer o escribir estos datos. En este caso, actúa como servidor, reteniendo los datos hasta que el teléfono los solicite. Estos datos se almacenan en el servidor BLE como atributos. La última columna muestra la información que se adquiere en cada punto de la conexión. El punto que es interesante para esta investigación es, cuando se envía la clave desde Maestro a Esclavo:

$$Rvcd\ write\ command,\ Handle:\ 0x0019 \qquad (1)$$

Este es un paquete que está haciendo una petición de escritura con un Handle 0x0019. Los Handles [15] o Manejadores son los componentes que procesan cada uno de los paquetes que pertenecen a las capturas. El manejador extrae del paquete la información necesaria, como en este caso el valor que se envía. Se puede observar que se envían tres peticiones de escritura de claves desde la cerradura al móvil. Posteriormente, la aplicación móvil responde a la cerradura enviando una serie de notificaciones informando de que se ha recibido el valor. Por último, Sherlock vuelve a enviar cuatro peticiones de escritura de claves al dispositivo. Esto implica que el desbloqueo de la cerradura requiere la introducción de siete claves en total.

El siguiente paso es ver los valores que se escriben en cada paquete de escritura. Se puede ver esta información seleccionando el paquete que se quiere analizar y de este modo aparecerán una serie de características de ese paquete. Para ver el valor, se analiza la información que proporciona el protocolo ATT. El protocolo ATT proporciona los tres puntos más importantes para resolver el problema de este trabajo:

- **Opcode** (Código de operación) [16]. En este caso se trata de una operación de escritura*'Write Command'*
- **Handle** (Puntero). Indica el atributo al que apunta el script.
- **Value**. Informa del valor de la clave que se está escribiendo.

Cuando se tienen todas las claves enviadas, se puede pasar a replicar los paquetes. Para ello, se utiliza el sistema operativo Linux o una máquina virtual que lo contenga. En este caso, se ha optado por esta última opción. Para ello, es necesario un adaptador Bluetooth. En el trabajo aquí propuesto se ha usado el Adafruit bluefruit LE Sniffer, ya que la máquina virtual no dispone de conexión Bluetooth. En primer lugar, se comprueba que se recibe una señal de la cerradura. Para ello, se utiliza la herramienta hcitool para buscar dispositivos Bluetooth cercanos. Ejecutando el siguiente comando *'sudo hcitool lescan'* se pueden ver los dispositivos BLE disponibles con sus respectivas direcciones MAC y comprobar que existe conexión con la cerradura.

El siguiente paso es hacer uso del comando hciconfig, que se utiliza para configurar los dispositivos Bluetooth. El nombre del dispositivo Bluetooth instalado en el sistema es hciX. En concreto, el que se tiene instalado es el hci0. Para poder trabajar con dispositivos Bluetooth es necesario inicializarlo con los siguientes comandos:

- hciconfig hci down
- hciconfig hci up

Con los comandos anteriores cerramos el dispositivo HCI en caso de que haya sido iniciado por un proceso anterior no finalizado, y lo volvemos a abrir e iniciar.

La última herramienta utilizada y la más importante para esta investigación es *Gatttool*. Esta herramienta permite conectar a dispositivos BLE con la dirección MAC del dispositivo y manipular sus atributos con una serie de comandos disponibles desde la herramienta. De este modo, será posible replicar los paquetes que se hayan capturado previamente.

En el caso de que no se conozca la dirección MAC de la cerradura a analizar, existen dos opciones para buscarla. Por un lado, se puede obtener esta información a partir de la captura de paquetes en Wireshark, que muestra las direcciones de los dos dispositivos en comunicación. Por otro lado, gracias a la herramienta hcitool, mencionada anteriormente, se puede ver fácilmente los dispositivos BLE encontrados y su dirección MAC asociada.

La herramienta Gatttool dispone de varios comandos interesantes para manipular los dispositivos Bluetooth o simplemente adquirir información relevante de los mismos. A continuación se listan los argumentos más útiles para este trabajo:

- -i. Especifica el nombre del dispositivo Bluetooth instalado en el sistema.
- -b. Especifica la dirección MAC.
- –characteristics. Muestra todos los manejadores asociados y sus propiedades.
- –char-read. Permite leer las características del dispositivo conectado. Por ejemplo, leer el valor / descriptor de un

manejador.

- –char-write-req. Permite hacer una solicitud de escritura en el dispositivo.
- -a. Lee o escribe las características del manejador especificado.
- -n. Contiene el valor a enviar en la petición de escritura.

Una vez que los comandos necesarios para replicar los paquetes están claros, se realiza una conexión a la cerradura y se envian las claves obtenidas. En una primera aproximación, para conectar con el dispositivo, se introdujeron cada una de las claves de forma individual, tal y como se muestra en la Figura 3.

```
$gatttool -i hci0 -b AC:9A:22:60:8F:7E -I
[ ] [AC:9A:22:60:8F:7E] [LE]> connect
[CON] [AC:9A:22:60:8F:7E] [LE]> char-write-req -a 0x0019 -n
ef00214e420ed2603b0097f74644bd69894f867d
[CON] [AC:9A:22:60:8F:7E] [LE]>
Characteristic value was written successfully
```

Figura 3. Conexión con la herramienta Gatttool

Esto no era factible porque la conexión con la cerradura se perdía unos segundos después de la inicialización. Como era necesario introducir 7 claves, era inviable establecer una conexión por cada clave a introducir. Para solventar este problema, se implementó un sencillo script bash (ver Figura 4) que incluye todos los comandos necesarios para la conexión y escritura, facilitando y agilizando el proceso de envío de paquetes.

```
#!/bin/bash
sudo hciconfig hci0 down
sudo hciconfig hci0 up
sleep 2

gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef00214e93c7c4603b009d47b9a44c6bd5386272
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
8eaf5aee126d0f84148e12d47b74db517e18c194
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
91e4ae93853752a0c062c1f339ebbbcbeb9d01
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef00234e420ed2601b007570512b8f7a9c1f3557
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n b9d1c3d50b3e01
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n
ef003175420ed2601b00296b72d64ad5c640575d
gatttool -i hci0 -b AC:9A:22:60:8F:7E --char-write-req -a 0x0019 -n bb2562ea8baa00

sleep 2
```

Figura 4. Script bash para automatizar el envío de claves

De este modo, se ejecutan conjuntamente todos los comandos explicados anteriormente, ahorrando así tiempo al atacar el bloqueo. Con esta medida es posible desbloquear la cerradura replicando los paquetes. Es un proceso bastante sencillo y que plantea dudas sobre la seguridad de esta marca de cerraduras inteligentes. Sin embargo, aunque es fácil y rápido de atacar, Sherlock cuenta con algunas medidas de seguridad. La cerradura realiza un cambio de claves cuando han pasado varios minutos desde que se abrió la puerta aunque, todavía hay suficiente tiempo para realizar la replicación de paquetes. Además, en el futuro se podría implementar un programa u optimizar el script para que la recogida de datos sea más efectiva y rápida, y así desbloquear la cerradura en cuestión de segundos.

### *IV-B. Análisis de seguridad de la cerradura Nuki*

Esta es la segunda cerradura que se ha estudiado en este trabajo. A continuación, se presentan algunos conceptos técnicos sobre ella y su funcionamiento, para luego proceder al análisis de su seguridad. Es especialmente destacable que la cerradura Nuki opera al más alto nivel de seguridad de cifrado, ya que utiliza AES con claves de 256 bits. AES es un cifrado simétrico por bloques, lo que significa que cifra y descifra datos en bloques de 128 bits cada uno. Para ello, utiliza una clave criptográfica específica, que es efectivamente un conjunto de protocolos para manipular la información. Esta clave puede tener un tamaño de 128, 192 o 256 bits. AES-256, la versión de clave de 256 bits de AES, es el estándar de encriptación utilizado por LE VPN. Es la forma más avanzada de encriptación y consiste en 14 rondas de sustitución, transposición y mezcla para un nivel de seguridad excepcionalmente alto. Su mayor tamaño de clave lo hace esencialmente irrompible, lo que significa que incluso si se piratea, los datos serían imposibles de descifrar.

El candado Nuki destaca en Europa por sus buenas críticas en cuanto a su seguridad, por lo que se ha analizado si con el mismo método de replicación de paquetes realizado anteriormente con el candado Sherlock también es posible atacarla.

Los pasos a seguir son prácticamente los mismos que los implementados con la cerradura Sherlock, por lo que no se explicará en profundidad para esta implementación.

Con el sniffer Adafruit bluefruit LE y su programa nRF Sniffer for Bluetooth para detectar dispositivos BLE cercanos, se realiza la conexión a la cerradura Nuki para posteriormente analizar los paquetes enviados y recibidos en Wireshark. Una vez iniciado Wireshark e introducido el filtro ”btatt” para adquirir sólo el tráfico Bluetooth, podemos desbloquear la cerradura y comprobar qué información nos llega y si será posible atacarla.

En la Figura 5 podemos apreciar la comunicación entre la cerradura Nuki y nuestro dispositivo móvil Samsung.

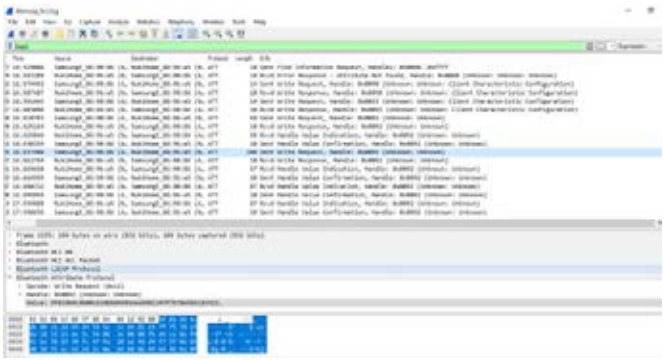


Figura 5. Captura de tráfico de Nuki usando Wireshark

El protocolo utilizado para la conexión también es ATT, donde se almacenan los atributos enviados, pero en la columna que indica la información del paquete se observan ligeros cambios de formato que se comentarán más adelante. En este caso, se peude ver que se realizan varias peticiones de escritura a la cerradura para enviar las claves con Manejador ”Handle 0x0092”:

$$Sent\ Write\ Request,\ Handle\ 0x0092 \quad (2)$$

Por lo tanto, se deben recoger todos los valores que se envían para posteriormente realizar su réplica. Para ver estos valores, interesa la información que proporciona el protocolo ATT. Una vez que que se tienen todas las claves que se envían, se puede volver a replicar el tráfico desde la máquina virtual. En primer lugar, se comprueba que se está recibiendo la conexión del candado Nuki con la herramienta hcitool y, al obtener una señal, se puede ejecutar el script bash, previamente modificado con los valores encontrados en Wireshark. Sin embargo, aunque los valores se escriben correctamente en la cerradura, no consigue desbloquearla. Esto se debe a que, al desbloquear la puerta, Nuki cambia instantáneamente la clave para la siguiente apertura, siendo inviable el ataque a la cerradura con este método, ya que las claves recogidas anteriormente quedarían automáticamente obsoletas.

A continuación, se explica con detalle cómo funciona el cifrado en la cerradura Nuki [17]. Este candado utiliza el principio de encriptación de extremo a extremo, es decir, aplica un cifrado a la clave de tal forma que sólo el dispositivo receptor puede descifrarla. Para establecer la comunicación entre la aplicación Nuki y la cerradura inteligente, se utiliza una clave propia que sólo conocen ambos dispositivos. Para protegerse de los atacantes, los datos se encriptan antes de ser transmitidos por el emisor (la app Nuki). Para ello se utiliza el proceso NaCl (Networking and Cryptography library) [18]. En este proceso, las combinaciones únicas de números y letras se utilizan sólo una vez. Estos datos se transfieren a través de Bluetooth y se descodifican de nuevo cuando los recibe el receptor (Nuki Smart Lock).

1. La app Nuki envía la instrucción de ”desbloqueo” y la encripta de tal manera que sólo la app y el Nuki Smart Lock conocen la clave.
2. La aplicación Nuki transfiere el mensaje encriptado a través de Bluetooth a la cerradura.
3. Nuki Smart Lock conoce la clave y por lo tanto puede descifrar el mensaje contenido y ejecutar la orden de ”desbloqueo”.
4. En el proceso de desbloqueo, la app Nuki recibe un número aleatorio. La instrucción de ”desbloqueo” sólo puede enviarse a la cerradura cuando ésta contiene necesariamente un número aleatorio idéntico.

Si posteriormente se envía otra instrucción de desbloqueo con el mismo número aleatorio a la cerradura de la puerta, la cerradura inteligente Nuki rechaza la instrucción. Este análisis muestra que el nivel de seguridad de Nuki es alto, tal y como informan los fabricantes, y en general es una cerradura en la que el mercado puede confiar actualmente. Sin embargo, dado que el método de replicación de paquetes utilizado en este trabajo, es sólo una de las opciones de ataque. Los resultados de esta investigación no implican que no pueda haber otro procedimiento para desbloquear la cerradura y que ésta pueda acabar siendo vulnerable a algún ataque.

### *IV-C. Comparativa de seguridad*

Una vez analizado el funcionamiento de ambas cerraduras y visto cómo se envían los paquetes entre los dos dispositivos, es posible realizar un análisis de las diferencias que se han encontrado durante el procedimiento de ataque de ambas cerraduras.

En primer lugar, se ha podido comprobar que la cerradura Nuki tiene mayor seguridad que la Sherlock, ya que durante la investigación se consiguió desbloquear la Sherlock, pero no la Nuki.

Además, hay más información abierta sobre el cifrado utilizado en el candado Nuki que en el caso del Sherlock, y de hecho se sabe que el candado Nuki utiliza AES con claves de 256 bits.

En conclusión, hay una serie de puntos que muestran las principales diferencias entre ambas cerraduras y por qué una se ha podido atacar, siendo, en este caso, imposible realizar el mismo ataque en la segunda.

- **Formato de escritura de paquetes.** Aunque en ambas conexiones se realizan peticiones de escritura y se almacenan claves en el protocolo ATT, la petición no se envía en el mismo formato. En la cerradura Sherlock se utiliza *”Rvcd write command”* que significa que ha recibido la orden de escritura con el valor de la clave y en la cerradura Nuki se utiliza *”Sent Write Request”* que implica que se ha enviado la escritura solicitada con ese valor. Además, en el primer caso, es la cerradura la que envía el valor siendo en el segundo caso, la aplicación móvil la que realiza el envío. Con esta información se puede asumir que ha sido posible desbloquear la cerradura Sherlock porque el ataque se ha realizado frente a la cerradura. Sin embargo, no fue posible realizar este ataque a la cerradura Nuki porque es necesario el dispositivo móvil, que no estaría presente en el momento del ciberataque.
- **Momento de cambio de clave.** Como ya hemos se ha comentado, Sherlock mantiene la misma clave durante unos minutos después de desbloquear la puerta, lo que permite atacar su seguridad sin problemas. Sin embargo, Nuki controla el tiempo de forma mucho más segura. Esta cerradura cambia la clave en el instante en que se desbloquea la puerta. De este modo, no hay margen de tiempo para el robo. Este es el punto más importante y fuerte de Nuki en cuanto a su seguridad y el que la diferencia de Sherlock.

## V. Conclusiones

En este trabajo se investiga la seguridad de las cerraduras inteligentes, que se están popularizando en todo el mundo. Dado su auge en el mercado, se ha comprobado cómo de seguras son estas cerraduras.Se ha investigado en detalle cómo funciona el proceso de apertura de la puerta y qué información se envía a través de los dispositivos implicados.

Además, se ha probado la replicación de los paquetes obtenidos con la intención de desbloquear la cerradura. A lo largo de la investigación se ha comprobado que no todas las cerraduras inteligentes tienen el nivel de seguridad que los fabricantes indican a los compradores. Por un lado, con un método bastante sencillo, la cerradura Sherlock S2 ha sido atacada usando herramientas de fácil acceso para poder llevarlo a cabo.

Por otro lado, se demuestra la existencia de otras cerraduras que incluyen un mayor nivel de seguridad. En concreto la cerradura Nuki, que no ha sido posible atacarlas con el

método propuesto en este trabajo. Sin embargo, no se descarta la posibilidad de encontrar nuevos métodos para realizar el ataque y comprometer su seguridad.

El trabajo actual es bastante flexible y podría ampliarse mucho más en el futuro. Es posible optimizar el script bash realizado o incluso crear un nuevo programa que recoja automáticamente las claves enviadas, ahorrando así mucho tiempo en el ataque.

Además de esto, en esta investigación sólo se contempla un método para desbloquear la cerradura, sin embargo, pueden existir o crearse muchos más procedimientos para desbloquear la cerradura a partir de otra información o atacando otro punto de la conexión. Teniendo en cuenta el avance de la tecnología y la competencia del mercado, es esperabel que los fabricantes refuercen la seguridad de sus cerraduras inteligentes en los próximos lanzamientos. Aun así, hay una gran variedad de marcas en el mercado que pueden ser analizadas en cuanto a su vulnerabilidad con el objetivo de poder concluir cuál es la opción más segura.

## AGRADECIMIENTOS

Investigación apoyada por el Ministerio de Ciencia, Innovación y Universidades (MCIU), la Agencia Estatal de Investigación (AEI) y el Fondo Europeo de Desarrollo Regional (FEDER) bajo el proyecto RTI2018-097263-B-I00.

## REFERENCIAS

[1] T. V. B. Sastry and P. Amritha, "Bluetooth low energy devices: Attacks and mitigations," *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2020*, vol. 711, p. 381, 2021.

[2] U. Lamping and E. Warnicke, "Wireshark user's guide," *Interface*, vol. 4, no. 6, p. 1, 2004.

[3] D. Spill and A. Bittau, "*BlueSniff: Eve Meets Alice and Bluetooth.*" *WooT*, vol. 7, pp. 1–10, 2007.

[4] "Según datos de interior, los robos con fuerza en domicilios en españa aumentaron casi un 2 % en 2018," https://www.europapress.es/comunicados/sociedad-00909/-casi-2018-20190204095856.html.

[5] N.-F. Polychronou, P.-H. Thevenon, M. Puys, and V. Beroulle, "A comprehensive survey of attacks without physical access targeting hardware vulnerabilities in iot/iiot devices, and their detection mechanisms," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 27, no. 1, pp. 1–35, 2021.

[6] A. Viderberg, "Security evaluation of smart door locks," https://www.kth.se/polopoly_fs/1.914054.1600689128!/Examensarbete\%20Final.pdf, 2019, [Online; accessed 07-July-2022].

[7] "The biggest data breaches and hacks of 2020," https://hacked.com/largest-breaches-and-hacks-of-2020-the-year-of-the-digital-pandemic/.

[8] "Aprende cómo utilizar wireshark para capturar y analizar el tráfico de red," https://www.redeszone.net/tutoriales/redes-cable/wireshark-capturar-analizar-trafico-red/.

[9] Gatttool, "Gatttool," http://manpages.ubuntu.com/manpages/cosmic/man1/gatttool.1.html.

[10] Xiaomi, "Sherlock smart lock," https://cerradurasinteligentes.net/xiaomi/xiaomi-sherlock-s2/.

[11] N. H. Solutions, "Nuki smart lock," https://nuki.io/es/smart-lock/, 2013.

[12] *HCI snoop log*, "Hci snoop log," https://www.mybluetoothreviews.com/what-is-bluetooth-hci-snoop-log/.

[13] Wireshark, "btatt filter," https://www.wireshark.org/docs/dfref/b/btatt.html.

[14] "*ATT (Attribute Protocol)*," https://programmerclick.com/article/68241335665/.

[15] "Handle," https://es.wikipedia.org/wiki/Handle.

[16] "Opcode," https://es.wikipedia.org/wiki/Cdigo_de_operacin.

[17] "Nuki encryption," https://nuki.io/es/blog/sientete-seguro/seguridad-en-primer-plano-concepto-de-cifrado-de-nuki-explicado-de-forma-sencilla/.

[18] "*NaCl*," https://en.wikipedia.org/wiki/NaCl_(software).

# Sistema de Votación Electrónica basado en Blockchain con Encriptación Homomórfica

Cándido Caballero-Gil, Universidad de La Laguna, ccabgil@ull.edu.es
Pino Caballero-Gil, Universidad de La Laguna, pcaballe@ull.edu.es
Néstor Álvarez-Díaz, Expero, Tenerife, naddiaz.92@gmail.com
Moti Yung, Columbia University, United States, motiyung@gmail.com

***Resumen*—El objetivo de este trabajo consiste en diseñar y desarrollar una aplicación web en la que se emplee la blockchain en el ámbito de las votaciones electrónicas, pero utilizando el cifrado homomórfico para encriptar el voto y garantizar su confidencialidad e integridad en todas las operaciones de recuento e identificación a lo largo del proceso electoral. En concreto, se desea enfocar el uso de la aplicación a la capacidad de crear elecciones, registrar candidatos, votantes y administradores, y realizar el recuento y custodia de los votos de forma totalmente descentralizada y segura, garantizando el anonimato de los votos. Se busca encontrar aceptación ante las autoridades, ya que suponen un compromiso para la transparencia, autenticidad e integridad de todo el proceso. De esta forma se pueden superar los retos de la votación electrónica, y reducir las dudas y sospechas de fraude que reciben las elecciones en general. La blockchain permitirá seguir el recuento de manera directa, y rastrear el destino de cada voto. Esto gracias a la estructura de árboles de Merkle enlazados criptográficamente, lo que otorga propiedades de inmutabilidad a todo el proceso, una cualidad exigible en toda votación.**

***Index Terms*—Votación Electrónica, Blockchain, Cifrado Homomórfico**

## I. Introducción

Hoy en día, muchos procesos gubernamentales han migrado o lo están implementando hacia soluciones electrónicas [1]. Desgraciadamente, este no es el caso del voto electrónico (e-voting), especialmente si se realiza a través de Internet. La tecnología se utiliza parcialmente en los sistemas de voto electrónico, como las máquinas de votación de grabación electrónica directa (DRE), el reconocimiento óptico de caracteres (OCR) o las impresoras de boletas electrónicas (EBP). Sin embargo, sólo unos pocos países, como Estonia, Noruega o Canadá, permiten a los ciudadanos votar en línea y la mayoría de ellos en escenarios específicos (votantes en el extranjero, elecciones municipales o federales, votantes militares, personal diplomático). Las principales barreras a las que se enfrenta el voto online en su proceso de adopción son, entre otras, la falta de transparencia, la violación del secreto del voto o la manipulación [2].

La solución aquí propuesta se ha centrado en cubrir las barreras mencionadas, prestando especial atención al diseño de un esquema práctico, no sólo teórico. ¿Qué significa práctico en este contexto? Algunos de los principales retos en la construcción de un sistema de votación en línea no es sólo hacer un sistema robusto que garantice el más alto nivel de seguridad. Implica que los ciudadanos no expertos también deben poder utilizarlo. Además, debe funcionar en escenarios reales en los que participen millones de personas de forma simultánea y en un corto espacio de tiempo.

Como solución a estos retos, en el desarrollo de esta propuesta se han considerado tres pilares fundamentales. En primer lugar, la interacción de cada votante debe ser la menor posible para evitar fricciones durante el proceso de votación. También se ha tenido en cuenta que la interacción del usuario debe ser mínima porque muchos de ellos pueden no estar adaptados al uso de la tecnología en general. En segundo lugar, el proceso de recuento debe ser rápido ya que en un escenario real el número de votantes se eleva a millones. Para solucionarlo, se ha desarrollado un sistema de Cifrado Homomorfico (HE) [3], que permite contar cada voto sin mucho esfuerzo debido a las propiedades de este tipo de criptosistema.

Además, el proceso de recuento se ha diseñado como en un sistema electoral estándar por dos razones. Por un lado, permite que el proceso de recuento sea práctico, ya que los niveles inferiores de la jerarquía sólo cuentan un pequeño número de votos, los emitidos en primera instancia por los votantes, propagando el resultado a los niveles superiores hasta llegar al nivel más alto, que sólo conocerá el resultado agregado. Por otro lado, esta jerarquía permite, desde el punto de vista de la seguridad, que la confianza no recaiga en una única autoridad central, que muy probablemente será el propio gobierno, sino que se distribuya entre todos los niveles de la propia jerarquía. Por último, para aumentar la confianza en el sistema, cada votante podrá verificar que su voto se ha incorporado al recuento y que no ha sido modificado durante el proceso gracias al uso de pruebas de conocimiento cero (ZKP).

La fiabilidad de los procesos plebiscitarios es una cualidad muy importante en cualquier estándar democráticos [4]. La tecnología expuesta en el presente trabajo intenta transformar la votación a un proceso telemático donde:

- Cada voto se apila en una cadena de bloques restringidos a modificaciones mediante un sistema de claves criptográficas. Gracias a esta característica se reduce la sospecha de tamaño de manera considerable.
- Todos los votantes pueden observar los votos acumulados por cada candidato en tiempo real, sin ser revelado el sentido de voto de cada elector.
- Toda la información de usuarios autorizados a emitir su preferencia se encuentra almacenada de manera distribuida, y se verifica que el voto haya sido emitido de manera única para evitar falsificaciones.
- La Comisión de Administradores es pública, y tiene disponibles una serie de poderes para organizar el sufragio.
- El plazo de apertura y cierre de la elección se puede consultar de manera anticipada, y una vez finalizado, se bloquea el acceso a la aplicación para votar, y se transita

a la fase de recuento.

## II. Estado del arte

El objetivo de esta sección es principalmente presentar el estado actual del voto electrónico [5], [6], [7], centrándose en el voto en línea porque tiene requisitos especiales que están haciendo que su adopción sea más lenta de lo deseado. En la literatura se encuentran dos términos principales: voto electrónico (e-voting), y voto online o voto por internet (i-voting) [8].

Muchos sistemas de votación utilizan máquinas electrónicas para ayudar en parte del proceso, como las máquinas de votación DRE que permiten registrar cada voto individual para un proceso de recuento automático. Esto no se limita a una máquina específica para realizar parte del proceso, un ordenador es una máquina electrónica que puede acceder a Internet. Por ello, el voto electrónico es un superconjunto del i-voting que también utiliza máquinas electrónicas pero implica que el proceso se realiza desde fuera de las instalaciones de votación gracias a una conexión a través de Internet.

Uno de los países de referencia en la aplicación del voto a distancia es Estonia [9] porque en 2005 se convirtió en la primera nación en permitir el uso de Internet en el proceso electoral. Sin embargo, después de ellos, otros países como Noruega introdujeron la posibilidad de votar a distancia con su propio protocolo [10], diferente al caso estonio [11]. No obstante, criptográficamente hablando, comparten una propiedad común, el uso de criptografía de clave pública como base del sistema. El uso de la criptografía de clave pública en los casos mencionados implica dos fases esenciales, la fase de configuración en la que una autoridad central genera los pares de claves necesarios y los distribuye por todo el sistema, y la fase de recuento que requiere el descifrado de cada voto individual tanto para la verificación como para el recuento. En este caso, la confianza se deposita en una única autoridad central que gestiona los pares de claves, incluidas las claves privadas que permiten descifrar cada voto. En términos de seguridad, está garantizada por el sistema criptográfico subyacente, pero la confianza está fuertemente centralizada. El hecho de que el sistema se corrompa y su consiguiente pérdida de confianza no implica directamente que se conozca la relación entre el votante y su voto. Se han añadido algunas capas adicionales de criptografía a este tipo de sistemas. Por ejemplo, en el caso noruego, utilizan el concepto de mix-net [12] para desvincular la relación entre el votante y el voto de éste, así como para la eficiencia a la hora del recuento. El uso de redes mixtas es útil, pero conlleva un problema que puede mermar la reputación de las elecciones. Este sistema requiere que todos los votos se descifren al final de las elecciones, es decir, cuando las papeletas se han cerrado y no se pueden recibir más votos. En esta parte del proceso, se descartan los votos duplicados y no válidos. Los votos que han sido alterados o introducidos intencionadamente en el sistema como forma de ataque pueden ser entonces invalidados, lo que es positivo para evitar la alteración de los resultados electorales mediante ataques maliciosos. Sin embargo, los fallos en la emisión de los votos, una mala codificación o un borrado intencionado de los mismos, también se considerarán inválidos. Como la validación no se realiza hasta el cierre de la elección, en el segundo escenario, los votos serán descartados sin posibilidad de recuperarlos [13].

El barajado de votos es un enfoque común para el diseño de sistemas de votación en línea, pero implica varios problemas de confianza como los mencionados anteriormente. Por ello, otra tendencia comúnmente utilizada es el uso de las propiedades homomórficas de algunos esquemas criptográficos para dotar al sistema de un proceso de recuento ”automático”. En la literatura existen varios ejemplos sobre su uso en votaciones online [14], [15], [16]. Sin embargo, suelen combinarse con otras herramientas criptográficas como las firmas ciegas propuestas por Chaum [17] u otras técnicas ZKP. Este hecho se debe a que cada voto debe ser verificado. En un escenario normal en el que cada votante puede votar una sola vez por un candidato, la validación debe garantizar que el voto pertenece al conjunto $\{0, 1\}$, tomando como inválidos los votos negativos o múltiples.

Como desventaja, estos sistemas se basan, como muchos otros, en un criptosistema de clave pública. Esto requiere que cada usuario cifre su voto con una clave pública precompartida. En el caso de que la clave pública pertenezca a una entidad ”curiosa”, ésta puede ser capaz de descifrar cada uno de los votos si éstos son accesibles públicamente incluso en forma cifrada.

Todas las preocupaciones mencionadas nos han llevado a evaluar sistemas con la posibilidad de no depositar toda la confianza en una entidad. En concreto, en la propuesta definida, cada usuario puede cifrar los datos con su propia clave pública y un tercero puede seguir realizando una evaluación homomórfica sobre estos datos cifrados. La única interacción necesaria entre los usuarios es la obtención de una clave derivada de sus claves. A continuación, los datos cifrados se descifran utilizando la nueva clave secreta, que se obtiene utilizando todas las claves secretas implicadas. Conseguimos estas interesantes propiedades a través de un tipo de esquema de Cifrado Totalmente Homomórfico (FHE) que permite aplicar propiedades homomórficas incluso con datos cifrados con claves independientes. De esta forma, el sistema de votación online se hace más fiable, ya que gestiona las claves de forma flexible, delegando la confianza en lugar de en una única entidad en múltiples participantes.

## III. Preliminares

Como se ha mencionado anteriormente, la propuesta se basa en dos herramientas criptográficas principales: El Cifrado Homomórfico y la Prueba de Conocimiento Cero, cuyas bases se explican en esta sección. Se introduce una pequeña historia de la evolución de ambas herramientas, así como una breve explicación de los fundamentos teóricos.

### *III-A. ZKP- Pruebas de Conocimiento Cero o nulo*

El concepto general fue introducido por el esquema ”Blind Signatures” de David Chaum [18]. La firma ciega es un protocolo de firma digital que permite a una persona obtener un mensaje firmado o sellado concedido por otra entidad para poder presentarlo a terceros, sin revelar ningún contenido específico del mensaje. Sin embargo, las firmas ciegas fueron las precursoras de la formalización de este tipo de pruebas de la mano de Shafrira Goldwasser, Silvio Micali y Charles Rackoff tres años después, en 1985 [19].

Como ellos definieron, una prueba de conocimiento cero es un protocolo criptográfico probabilístico que permite a un proveedor ($P$) convencer a un verificador ($V$) sobre el conocimiento de alguna información secreta sin revelar nada sobre ella. Estas pruebas se caracterizan por las siguientes propiedades:

- Completitud de los elementos. Con una declaración verdadera, un $V$ honesto será convencido con alta probabilidad por un $P$ honesto. La honestidad implica que el esquema es seguido correctamente por ambas partes.
- Solidez del elemento. Con una afirmación inválida, $V$ se convence de lo contrario con una probabilidad muy pequeña.
- Conocimiento cero. Para una declaración válida, $V$ no aprende ninguna información excepto que la declaración es válida.

Dependiendo del propósito que se quiera cubrir en un diseño específico, la prueba puede variar mucho de una a otra. Una división común de este tipo de pruebas se basa en la interactividad necesaria para cumplir la prueba. Pruebas de Conocimiento Cero Interactivas, o simplemente Pruebas de Conocimiento Cero, que implican la sincronización entre el proveedor y el verificador. Algunos ejemplos de este tipo de protocolos son el esquema de identificación y firma de Schnorr [20], el protocolo Chaum-Pedersen [21], o el protocolo de testigos indistinguibles diseñado por Cramer, Damgård, y Schoenmakers [22].

Pruebas de conocimiento cero no interactivas (NIZK), que permiten al verificador comprobar las pruebas generadas por el proveedor sin comunicaciones sincrónicas [23]. En general, los sistemas ZKP se diseñan teóricamente sin restricciones de interactividad, pero en la práctica se convierten en sistemas NIZK. La heurística Fiat-Shamir [24] permite obtener un NIZK a partir de un ZKP en los siguientes pasos:

- El proveedor genera un mensaje de compromiso que muestra que conoce el secreto.
- El proveedor toma el compromiso y otra información como entradas y devuelve el desafío aplicando alguna función hash criptográfica [25].
- El proveedor calcula la respuesta y envía el resultado, incluyendo el compromiso, el desafío y la respuesta al verificador.

La necesidad de realizar un gran número de verificaciones sin un alto consumo de red es un requisito para desarrollar un sistema viable en el mundo real. Por lo tanto, este mecanismo es absolutamente interesante porque permite construir sistemas que utilizan el concepto general de ZKP para una amplia variedad de propósitos sin las limitaciones de la interactividad. Con esta puerta abierta a la no interactividad, las aplicaciones de ZKP han seguido creciendo desde los sistemas de autenticación [26] hasta la criptomoneda [27] pasando por el soporte del voto electrónico [28], [29].

### *III-B. Criptografía Homomórfica*

La criptografía homomórfica es un esquema de encriptación que permite realizar un conjunto de funciones sobre los datos encriptados que tienen una función equivalente en el álgebra. Por ejemplo, para dos mensajes $m_1$ y $m_2$ existe una operación que satisface que $E(m_1 + m_2) = E(m_1) \circ E(m_2)$ donde, $\circ$ denota la función homomórfica que representa una adición homomórfica en este esquema particular HE y $E()$ denota la función de cifrado.

La encriptación es un mecanismo extendido para preservar la privacidad de la información sensible. Sin embargo, los esquemas de cifrado convencionales no pueden trabajar con datos cifrados, lo que les obliga a descifrarlos para poder trabajar con ellos. Dependiendo de la finalidad del sistema, no tiene por qué ser un requisito que un tercero descifre la información si sólo el propietario es el que va a trabajar con ella. Muchos de estos escenarios están relacionados con servicios en la nube como el almacenamiento o los entornos colaborativos donde el tercero, en este caso, sería el servidor que lo soporta. Este servidor no tiene que descifrar la información ya que es un mero gestor que almacena los datos y permite sincronizarlos entre un grupo de personas, por ejemplo. Sin embargo, en los esquemas clásicos, los usuarios tienen que sacrificar su privacidad para hacer uso de los servicios mencionados. En un intento de evitar la necesidad de descifrar los datos para poder trabajar con ellos, fue estudiado por Rivest, Adleman y Dertouzos en 1978 [30] y sirvió de germen para todo un campo en la criptografía que ha ido avanzando con los años.

Teniendo en cuenta el tipo de operaciones que admite un determinado esquema de HE, podemos clasificar los esquemas en tres categorías. En primer lugar, el cifrado parcialmente homomórfico (PHE) sólo admite operaciones aditivas o multiplicativas. Hay varios ejemplos útiles de PHE, como RSA en 1978 [30], Goldwasser-Micali en 1982 [31], ElGamal en 1985 [32], Benaloh en 1994 [33], Paillier en 1999 [34], entre otros.

En segundo lugar, el cifrado homomórfico SomeWhat (SWHE) admite un número limitado de operaciones o circuitos específicos. Tal vez el ejemplo más destacado sea el esquema de Yao [35], ya que se utiliza ampliamente para resolver problemas de computación multipartita.

Por último, FHE fue introducido por primera vez por Gentry [36] en 2010 permitiendo un conjunto ilimitado de operaciones sobre los datos cifrados. Este esquema fue el precursor del uso de celosías para intentar resolver o mejorar los esquemas FHE. Después de este se desarrollaron varios pero cabe destacar los esquemas NTRU-like que vinieron de la mano de López en 2012 [37]. Los esquemas NTRU-like no son realmente modernos, se basan en trabajos anteriores [38], pero sus propiedades homomórficas fueron realizadas recientemente.

Dependiendo del propósito de la aplicación, algunos esquemas pueden ser más adecuados que otros, pero todos ofrecen una amplia gama de posibilidades [39], [40]. En el caso que nos ocupa, la votación en línea, un criptosistema particular ha sido comúnmente utilizado, teóricamente al menos, porque se queja de la operación aditiva. La votación online requiere sumar votos a cada candidato, y el criptosistema Paillier permite calcular una operación homomórfica equivalente a la suma algebraica de la siguiente manera: $E(M_1) \cdot E(M_2) = E(M_1 + M_2 \ mod \ n)$.

## IV. Propuesta de Sistema de Votación electrónica

Para el desarrollo del sistema se han utilizado las siguientes tecnologías, React y Material-UI para el entorno web, Truffle y Ganache para el desarrollo de aplicaciones descentralizadas. Firebase para la base de datos y Metamask como extensión del navegador para ejecutar Dapps sin necesidad de un servidor.

Por lo tanto, en este trabajo se ha creado una aplicación web para la plataforma Ethereum que a través de contratos inteligentes suministra los mecanismos necesarios para celebrar una elección. En la aplicación existen tres tipos de usuarios: Votante, Candidato, y Administrador. Los permisos y limitaciones de cada rol son los siguientes:

- Votante (V): Solo puede enviar el voto una única vez, ha de estar inscrito por el Administrador en una o varias elecciones para disponer del derecho a votar.
- Candidato (C): Tiene fijada como incompatibilidad ejercer de Administrador de manera simultánea. Puede hacer uso de su derecho al voto. Ha de estar inscrito por el Administrador para presentar su candidatura.
- Administrador (A): Dispone de permisos para crear elecciones, e inscribir candidatos y votantes en las mismas.

La manera de funcionar de la aplicación consta de tres pasos. Primero se ha de crear la elección, seguidamente se declaran los candidatos y votantes autorizados, y finalmente, se da comienzo al proceso de votación, y una vez vencido el plazo, se publican los resultados finales.

Para desplegar la aplicación y poder llevar a cabo pruebas y observar los resultados del desarrollo fue necesario realizar los pasos que a continuación se enumeran:

1. Desplegar una blockchain de prueba con Ganache. Al desplegar la cadena de bloques, Ganache provee un número de cuentas de prueba con Ether ficticio. En el caso de la aplicación desarrollada para este trabajo, el Ether es necesario para poder abonar los gastos asociados a la ejecución de las transacciones de los contratos inteligentes (Gas).
2. Configurar un dominio en Firebase, y una base de datos NoSQL en tiempo real para gestionar el servicio de autenticación de usuarios.
3. Configurar Metamask desde el ordenador para hacer uso de la blockchain privada y poder acceder a las cuentas de usuario de prueba que provee Ganache.
4. Compilar y desplegar los contratos inteligentes haciendo uso de Truffle. Una vez seguidos estos pasos, la aplicación es totalmente funcional. Es relevante aclarar que la gestión de cuentas de BlockVote se delega a Metamask que, como se ha expuesto con anterioridad, es capaz de almacenar de forma segura y distribuida las claves privadas de las cuentas de usuarios de la blockchain.

### *IV-A. Vistas Web*

La aplicación cuenta con cuatro vistas: Inicio de sesión, Registro, Panel para crear una elección y la Interfaz de Votación. Además de las vistas principales, la aplicación incluye mecanismos auxiliares para notificación de errores (vistas secundarias, mensajes en pantalla, ...). También es posible desplegar BlockVote localmente para poder realizar las pruebas y comprobaciones.


Figura 1. Panel para crear elecciones

En el panel "Creación de Elecciones" (ver figura 1) se encuentra la funcionalidad para crear una elección con un identificador único, seleccionar los administradores de la misma, los candidatos, y finalmente, los votantes. Este panel incorpora una herramienta para enviar invitaciones a los usuarios seleccionados para participar de la elección. En el "Panel de votación" el usuario dispone de la interfaz para enviar el voto, y obtener la certificación de haber votado. Una vez el usuario emite su voto, se bloquea la interfaz en el navegador cliente para evitar que haya fraude.

### *IV-B. Contractos Inteligentes*

En esta sección se exponen los distintos contratos inteligentes implementados para el desarrollo de BlockVote. Para cada uno se comentarán y mostrarán los métodos y atributos más relevantes, junto con su funcionamiento u objetivo. A pesar de que no se profundice totalmente en los detalles de implementación de cada método por cuestiones de longitud del presente trabajo, se ha de destacar que se han utilizado cláusulas require de Solidity con el propósito de garantizar el correcto funcionamiento de los contratos inteligentes, cotejando previamente los parámetros recibidos.

*IV-B1. Election:* Es el contrato vertebrador de la aplicación, contiene los métodos para crear una elección, crear candidatos, registrar votantes, comenzar una elección, la lógica asociada al proceso de votación, y un método para cerrar la votación de manera automática una vez finaliza el plazo. Finalmente, en él también se implementa la funcionalidad para el recuento de votos, agregación de administradores, almacenamiento de parámetros de la elección a fin de enviar información de transparencia a la cadena de bloques.

*IV-B2. FirstPastThePost:* El segundo contrato implementado calcula el ganador de la elección, y los resultados globales de todo el proceso, retornando el número de votos obtenidos por cada candidato. La metodología de recuento utilizada se denomina FirstPastThePost, y consiste en declarar como ganador el candidato que recibe un mayor número de votos.

*IV-B3. VotingSystem:* Se trata de una clase abstracta que implementa los métodos del sistema de votación. De ella hereda la clase FirstPastThePost, que desarrolla el algoritmo calculate para obtener el candidato con la mayor suma de votos.

### *IV-C. Análisis de seguridad y rendimiento*

Los requerimientos de cada elección se especifican en un contrato electrónico que recoge los permisos y privilegios de todos los participantes de los comicios: votantes, candidatos y administradores.

Cuando se inicia la votación, los usuarios con rol de votante reciben invitaciones por correo electrónico con un código de seguridad que caduca después de unas horas. Acceden al evento, registran su voto, y pueden seguir todas las interacciones de la cadena de bloques relacionadas con la elección. Una vez finalizado el plazo, se pasa a declarar los candidatos ganadores. Esta es una de las ventajas de la descentralización, los usuarios finales pueden ver todas las transacciones de la votación en directo, cuando se registra un nuevo voto, la fórmula para el recuento de votos, el número de votantes, etc.

## V. Conclusiones

Partiendo de los sistemas de votación tradicionales donde podría haber ataques que hagan que las elecciones fuesen un fraude, se ha realizado una investigación con el propósito de aplicar las tecnologías blockchain, la criptografía homomórfica y las pruebas de conocimiento nulo para crear un sistema de votación electrónica descentralizado y seguro. Dicho sistema de votación electrónica basado en blockchain con encriptación homomórfica permite mejorar la seguridad y la transparencia de los procesos electorales. La encriptación homomórfica permite garantizar la privacidad de los votantes. además, el blockchain garantiza la integridad de los datos y evita el fraude. Gracias al trabajo realizado ha sido posible entender cómo al aplicar correctamente estas tecnologías se consiguen resultados que proporcionan a un sistema de votación electrónica múltiples ventajas. La implementación de la aplicación web con conexión a la plataforma Ethereum ha servido para demostrar el funcionamiento de dichas tecnologías. A pesar del grado de simplicidad de las acciones que los usuarios de la plataforma web *BlockVote* deben llevar a cabo, se ha demostrado que estas tecnologías conjuntamente son capaces de alterar sustancialmente el modo en el que se realizan y controlan los procesos de votación electrónica, logrando un estado de confianza máxima entre los interesados.

## Agradecimientos

Investigación apoyada por el Ministerio de Ciencia, Innovación y Universidades (MCIU), la Agencia Estatal de Investigación (AEI) y el Fondo Europeo de Desarrollo Regional (FEDER) bajo el proyecto RTI2018-097263-B-I00.

## Referencias

[1] A. Ingrams, A. Manoharan, L. Schmidthuber, and M. Holzer, "Stages and determinants of e-government development: a twelve-year longitudinal study of global cities," *International Public Management Journal*, vol. 23, no. 6, pp. 731–769, 2020.

[2] P. Wolf, R. Nackerdien, and D. Tuccinardi. (2011) Introducing electronic voting: essential considerations. International Institute for Democracy and Electoral Assistance. [Online]. Available: https://www.idea.int/publications/catalogue/introducing-electronic-voting-essential-considerations

[3] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[4] M. PAGE, P. ANTENUCCI, and M. LEIRAS, "Mapa de confiabilidad de las elecciones," *Documento de Trabajo*, no. 182, 2019.

[5] N. Kshetri and J. Voas, "Blockchain-enabled e-voting," *Ieee Software*, vol. 35, no. 4, pp. 95–99, 2018.

[6] F. Hjálmarsson, G. K. Hreióarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-based e-voting system," in *2018 IEEE 11th international conference on cloud computing (CLOUD)*. IEEE, 2018, pp. 983–986.

[7] A. B. Ayed, "A conceptual secure blockchain-based electronic voting system," *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01–09, 2017.

[8] S. Heiberg, P. Laud, and J. Willemson, "The application of i-voting for estonian parliamentary elections of 2011," in *E-Voting and Identity*, A. Kiayias and H. Lipmaa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 208–223.

[9] E. Maaten, "Towards remote e-voting: Estonian case," in *Electronic Voting in Europe*, 2004.

[10] V. Cortier and C. Wiedling, "A formal analysis of the norwegian e-voting protocol," in *Principles of Security and Trust*, P. Degano and J. D. Guttman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 109–128.

[11] M. J. M. Chowdhury, "Comparison of e-voting schemes: Estonian and norwegian solutions," *International Journal of Applied Information Systems*, vol. 6, no. 2, pp. 60–66, September 2013, published by Foundation of Computer Science, New York, USA.

[12] M. Abe, "Mix-networks on permutation networks," in *Advances in Cryptology - ASIACRYPT'99*, K.-Y. Lam, E. Okamoto, and C. Xing, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 258–273.

[13] P. Bibiloni, A. Escala, and P. Morillo, "Vote validatability in mix-net-based evoting," in *E-Voting and Identity*, R. Haenni, R. E. Koenig, and D. Wikström, Eds. Cham: Springer International Publishing, 2015, pp. 92–109.

[14] S. M. Anggriane, S. M. Nasution, and F. Azmi, "Advanced e-voting system using paillier homomorphic encryption algorithm," in *2016 International Conference on Informatics and Computing (ICIC)*, 2016, pp. 338–342.

[15] T. Sharma, "E-voting using homomorphic encryption scheme," *International Journal of Computer Applications*, vol. 141, no. 13, pp. 14–16, 2016.

[16] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard, "Practical multi-candidate election system," in *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing*, ser. PODC '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 274–283. [Online]. Available: https://doi.org/10.1145/383962.384044

[17] L. R. Rivest, S. Ledlie *et al.*, "Lecture notes 15: Voting, homomorphic encryption," 2002.

[18] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Boston, MA: Springer US, 1983, pp. 199–203.

[19] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[20] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology — CRYPTO' 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 239–252.

[21] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Advances in Cryptology — CRYPTO' 92*, E. F. Brickell, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 89–105.

[22] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Advances in Cryptology — CRYPTO '94*, Y. G. Desmedt, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 174–187.

[23] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 103–112. [Online]. Available: https://doi.org/10.1145/62212.62222

[24] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology — CRYPTO' 86*, A. M. Odlyzko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194.

[25] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.

[26] J. Brandt, I. Damgård, P. Landrock, and T. P. Pedersen, "Zero-knowledge authentication scheme with secret key exchange," *J. Cryptology*, vol. 11, pp. 147–159, 1998.

[27] V. W. C. Koens T., Ramaekers C., "Efficient zero-knowledge range proofs in ethereum," in *Tech. Rep., ING, blockchain@ing.com*, 2018.

[28] S. Panja and B. K. Roy, "A secure end-to-end verifiable e-voting system using zero knowledge based blockchain." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 466, 2018.

[29] S. Panja and B. Roy, "A secure end-to-end verifiable e-voting system using blockchain and cloud server," *Journal of Information Security and Applications*, vol. 59, p. 102815, 2021.

[30] R. L. Rivest, L. Adleman, M. L. Dertouzos *et al.*, "On data banks and privacy homomorphisms," *Foundations of secure computation*, vol. 4, no. 11, pp. 169–180, 1978.

[31] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, ser. STOC '82. New York, NY, USA: Association for Computing Machinery, 1982, p. 365–377. [Online]. Available: https://doi.org/10.1145/800070.802212

[32] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[33] J. Benaloh, "Dense probabilistic encryption," in *Proceedings of the workshop on selected areas of cryptography*, 1994, pp. 120–128.

[34] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238.

[35] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, 1982, pp. 160–164.

[36] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 24–43.

[37] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1219–1234. [Online]. Available: https://doi.org/10.1145/2213977.2214086

[38] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *Lecture Notes in Computer Science*. Springer-Verlag, 1998, pp. 267–288.

[39] J. Sen, "Homomorphic encryption: Theory & applications," *CoRR*, vol. abs/1305.5886, 2013. [Online]. Available: http://arxiv.org/abs/1305.5886

[40] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–35, 2018.

# Una guía metodológica para la elaboración de libros de jugadas (playbooks) para riesgos cibernéticos.

Jeimy J. Cano M.
Universidad de los Andes
Cra 1E. No. 18A-10
jcano@uniandes.edu.co

*Abstract*- **Las organizaciones modernas cada vez más se encuentran en medio de escenarios inestables e inciertos. El aumento creciente de la densidad digital y las mayores exigencias de sus clientes de experiencias distintas, demandan repensar sus prácticas de seguridad y control para contar con la preparación y coordinación necesaria para responder y atender el incierto cuando se materializa un riesgo cibernético. En ese sentido, los libros jugadas o *playbooks*, establecen un marco de acción concreto de coordinación, comunicación, respuesta y aprendizaje, que limita las intenciones del adversario: confusión, inestabilidad y respuestas erráticas. En consecuencia, este artículo desarrolla y detalla la aplicación de una guía metodológica para la elaboración de *playbooks* para riesgos cibernéticos desde el diálogo y la construcción conjunta entre el área de seguridad/ciberseguridad y los procesos de negocio, como un ejercicio inclusivo y propio de la dinámica de los procesos, que explora y aprende de la inevitabilidad de la falla.**

***Index Terms*- playbooks, adversario, incidentes**

## I. INTRODUCCIÓN

Las inestabilidades globales y las tensiones geopolíticas vigentes se convierten en elementos concretos y claves para analizar y retar los planes de las organizaciones del siglo XXI. En este sentido, la complejidad que implica operar en un entorno como el actual, exige una transformación de la forma como se concibe la seguridad y la ciberseguridad comoquiera que es ahora desde la incertidumbre donde se entiende el concepto de confianza y confiabilidad de los clientes y las operaciones [1].

Las organizaciones modernas y centradas en el reto de la inevitabilidad de la falla, saben que un ciberataque o evento adverso de seguridad y control ya no es un probabilidad, sino una certeza, y por lo tanto, deberán prepararse para operar en medio de las tensiones que estas situaciones generan, lo que demanda no sólo desplegar el proceso de atención de incidentes, sino aplicar un "libro de jugadas" (*playbook*) previsto para la situación, donde todos los participantes suman para movilizar la organización y coordinar sus acciones en medio del incierto [2].

Conceptualizar un "libro de jugadas en el contexto del gobierno y gestión de la ciberseguridad/seguridad de la información de las empresas, implica un cambio de postura y tratamiento de los incidentes o brechas de seguridad en las empresas. Esto sugiere que la organización sabe que ahora está en un campo de juego, donde con frecuencia no conoce a su adversario, y debe moverse de forma apropiada para mantenerse activo y vigilante tanto de sus movimientos como los de su atacante. La postura de seguridad deja de ser estática y basada exclusivamente en las prácticas estándares y se traslada a un escenario de cambios y sorpresas permanentes donde no sólo debe "actuar", sino aprender [3].

Así las cosas, construir un "libro de jugadas" para riesgos cibernéticos concretos se convierte en una tarea no sólo de los profesionales de ciberseguridad/seguridad de tecnología de información, sino una práctica de alcance corporativo que entiende los impactos y efectos de la materialización de un ciberataque o un ciberriesgo, y cómo cada uno de las áreas debe coordinar y comunicar las acciones pertinentes, de cara a la atención y actuación antes, durante y después de un ciberataque exitoso [4].

En este sentido, este artículo detalla una guía metodológica para construir "libros de jugadas" para riesgos cibernéticos específicos, que desde una perspectiva sistémica conecta las diferentes posturas de los participantes corporativos (técnicos y no técnicos), de tal forma que la visual del ejercicio mantiene en foco la protección de la organización y su reputación, mientras cada uno de ellos aporta sus reflexiones y sugerencias de acción que se deben adelantar para disminuir los impactos de la materialización del evento adverso.

Con el fin de concretar el diseño metodológico mencionado este documento se estructura de la siguiente manera. Una primera sección para comprender y diferenciar los riesgos de tecnologías y los riesgos cibernéticos, luego un detalle de la estructura de los "libros de jugadas", donde se especifica su alcance y entregables. Seguidamente la presentación de la guía metodológica para desarrollar los "playbooks" con sus diferentes etapas y productos esperados. Luego se ilustra la aplicación práctica de la guía mencionada para un riesgo específico en una organización y finalmente se tienen las conclusiones del ejercicio realizado.

## II. LOS RIESGOS DE TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIONES, Y LOS RIESGOS CIBERNÉTICOS. DOS VISTAS COMPLEMENTARIAS.

En un escenario de inestabilidad y volatilidad las organizaciones deben reconocer con los diferentes cambios y tendencias que se presentan en el desarrollo de sus estrategias de negocio. En consecuencia, es claro que la tecnología de información y comunicaciones se convierte en un elemento relevante que le permite a las empresas movilizarse y afianzarse en el nuevo entorno hiperconectado, donde las oportunidades se ubican en zonas de interacción e innovación que retan las prácticas vigentes de seguridad y control de las organizaciones [5].

Mientras los riesgos de tecnologías de información y comunicaciones (TIC) tienen su foco en la operación y la reducción de sus costos, el riesgo cibernético tiene como fundamento el apetito de riesgo de la empresa y el negocio en sí mismo. Lo anterior sugiere que los riesgos TIC son parte

natural y conocida de la gestión de riesgos empresariales los cuales son tratados siguiendo los estándares y buenas prácticas generalmente asociadas con los controles generales de tecnología que buscan mantener la continuidad del negocio [6].

Los riesgos cibernéticos son riesgos sistémicos (basados en las interacciones y acoplamiento de los diferente componentes), emergentes (son fruto de las interacciones entre los elementos y no parte en sí misma de ellos) y disruptivos (cambian y se modifican a sí mismo en la dinámica del negocio) lo que necesariamente demanda salir de la perspectiva tradicional de prácticas, para concentrarse en el desarrollo de capacidades (patrones de aprendizaje) que permitan a la organización reconocer tendencias y patrones de interacción en entorno inestables, que permitan habilitar los umbrales de operación necesarios para aumentar la resiliencia del negocio [7].

Los riesgos TIC están orientados a mitigar los efectos de eventos no previstos en el desarrollo de las operaciones, para lo cual los mecanismos de seguridad instalados y asegurados buscan detectar y detener posibles amenazas conocidas, y advertir mediante alertas específicas eventos previamente identificados. El riesgo cibernético se enmarca en el contexto de la incertidumbre y la volatilidad del entorno, por lo que más que proteger y asegurar, que es lo que se busca en el riesgo TIC, se requiere defender y anticipar. Esto es, establecer las medidas para demorar y ganar tiempo frente al atacante, y por otro lado, establecer patrones de movimientos y estrategias del adversario, con el fin de seguirlo y estudiarlo hasta antes de que tenga éxito [8].

En consecuencia, si bien el riesgo TIC es un riesgo relevante que requiere atención y aseguramiento por parte de las organizaciones, el riesgo cibernético con sus características mencionadas establece un marco de acción diferente, pues exige de aquellos que son sus responsables, una vista más holística del entorno y las relaciones de la empresa, así como el reconocimiento del adversario como un elemento fundamental para comprender su dinámica y establecer las capacidades necesarias para dar cuenta con los retos e inestabilidades que buscan sorprender a la organización de diferentes formas, en distintos momentos y lugares inesperados [9].

Por tanto, para atender los riesgos cibernéticos, los "libros de jugadas" terminan siendo una estrategia de interés para las organizaciones, pues invita a los diferentes perfiles corporativos a comprender estos riesgos, como lo afirma un ejecutivo de alto nivel de una empresa internacional, "el riesgo de ciberseguridad es como un perro rabioso que está siempre pendiente a quien morder". Lo que implica, mantenerse alerta, consciente de la realidad de la organización y sobremanera con una perspectiva coordinada y de comunicación organizada que limite aquello que el atacante quiere: distracción, desorden y actuaciones erráticas por parte de la empresa.

## III. Libros de jugadas (*playbooks*). Conceptos y estructura general.

Un libro de jugadas o *playbook* (LJ/PB) es un concepto tomado del futbol americano, donde cada entrenador establece un conjunto de jugadas y estrategias tanto defensivas como ofensivas para ganar terreno sobre su adversario y así conseguir una ventaja táctica que le permita lograr el mayor número de anotaciones. Nótese que para lograr su propósito cada persona que participa tiene un papel y responsabilidades que deberán estar coordinadas con el propósito general. En el escenario de la ciberseguridad/seguridad de la información los LJ/PB son: [10]

- Una forma de gestionar riesgos,
- Una estrategia para actuar de forma coordinada,
- Una estructura para la toma de decisiones,
- Una respuesta a escenarios conocidos y latentes,

que busca dar respuesta al menos a cinco interrogantes claves:

1. ¿Qué estamos tratando de proteger?
2. ¿Cuáles son las amenazas claves?
3. ¿Cómo las detectamos?
4. ¿Cómo debemos responder?
5. ¿Cómo nos organizamos?

Si bien existen diferentes estructuras y aproximaciones para el desarrollo de LJ/PB, en general se tienen tres momentos claves que se deben tener en cuenta: el antes, el durante y el después. A continuación basados en [2] se detallan cada una de estas etapas y sus entregables.

Las actividades previstas en el "**antes**" se configuran en tres momentos:

- *Preparación*: vincular a las diferentes personas que participan en la atención del evento, el detalle de las actividades previas que se deben tener, las actividades de concientización sobre el evento, reportes de eventos recientes, la estructura de notificación establecida y los planes de respuesta legal asociados con el evento.
- *Detección*: Alertas identificadas locales o internacionales, reporte de usuarios internos, advertencias de centro de operaciones de seguridad, actividades inusuales, reporte de clientes vía el "call center".
- *Análisis preliminares*: Basados en los resultados de la detección ubicar la posible fuente inicial del evento, el posible vector de ataque, los posibles impactos y afectaciones, el tipo de información o servicios que puede estar comprometido y activar la estrategia de comunicaciones inicial para los diferentes grupos de interés establecidos.

El momento del "antes" termina con las notificaciones concretas del caso propias de la estrategia de comunicaciones prevista, la activación del orquestador de las actividades del LJ/PB y la convocatoria de todos los actores y la activación de sus roles en el desarrollo de las actividades posteriores.

En el "**durante**" se llevan a cabo tres fases a saber:

- *Contención*: Establece las acciones iniciales previstas para mitigar o demorar las acciones del adversario con dos propósitos: detener la propagación del ataque y prevenir más daños a los sistemas afectados. Emitir los reportes de avance de esta actividad.
- *Erradicación*: Consiste en las actividades y acciones de mitigación a largo plazo que incluyen medidas para inhabilitar la fuente del ataque, validando que el adversario no haya efectuado movimientos laterales con otros puntos o pivotes escondidos para evitar ser detectado

de forma posterior al evento. Emitir los reportes de avance de esta actividad.

- *Remediación*: Define las acciones de reconfiguración y actualización de las medidas de seguridad actuales, ajuste de las alertas de los sistemas de monitoreo y las listas de chequeo de aseguramiento de los sistemas, así como la modificación de los planes de concientización respecto del evento materializado y sus impactos.

El "durante" mantiene un estado de reporte permanente al orquestador de cada una de las actividades realizadas en las tres fases hasta que se concreten los ajustes de la remediación previstos. Terminada esta iteración se notifica el cierre del "durante" a los participantes por parte del orquestador.

Finalmente en el "**después**" se tienen los siguientes pasos:

- *Recuperación*: El orquestador notifica sobre la activación del plan de regreso a la "nueva normalidad" donde cada uno de los participantes informa a sus áreas el inicio del retorno, informando los ajustes y acciones realizadas, así como los planes de entrenamiento que se requieren para su aplicación. De igual forma, se activa el plan legal y el trabajo conjunto con el asegurador para evaluar los impactos del evento y sus consecuencias de mediano y largo plazo.
- *Lecciones aprendidas*: Realizar una sesión de trabajo con todos los participantes de la atención del evento donde se validen al menos tres interrogantes:
  - ¿Qué cosas salieron bien que vale la pena mantener?
  - ¿Qué cosas definitivamente se deben dejar de hacer que no facilitan el proceso?
  - ¿Qué cosas se van a hacer diferentes para mejorar el desempeño en la atención del incidente?

Los resultados de este ejercicio deben llevar a la actualización del LJ/PB y a una posterior simulación para asegurar la apropiación de los cambios incorporados luego de las lecciones aprendidas.

La práctica y simulación de los LJ/PB generan competencias claves en el comportamiento de las personas y la dinámica de la organización frente a los incidentes que se ve reflejada en:

- Mejores interacciones y comunicaciones internas y externas. Tono y foco en los mensajes.
- Coordinación de las operaciones y equipos de trabajo.
- Comprensión del evento y sus relaciones.
- Calidad de las respuestas y de los resultados.
- Conciencia de las situaciones y sus implicaciones a nivel organizacional.

Con este marco de trabajo es claro que se requiere un ejercicio de construcción colectiva donde los diferentes participantes no sólo saben qué hacer, sino que se reconocen a sí mismos como parte del ejercicio y establecen nuevas relaciones antes inexistentes, cuando conocen los alcances de las acciones de sus otros colegas durante el desarrollo y aplicación del LJ/PB. En este contexto, se plantea guía metodológica para construir "libros de jugadas" para riesgos cibernéticos específicos donde el reto más que contar con un documento formal (que por demás es siempre vivo y cambiante), es encontrar lugares y lenguajes comunes que habiliten una cooperación y coordinación fluida en medio de las inestabilidades que provoca un ciberataque exitoso.

## IV. GUÍA METODOLÓGICA PARA CONSTRUIR "LIBROS DE JUGADAS" PARA RIESGOS CIBERNÉTICOS ESPECÍFICOS.

La elaboración de LJ/PB demanda necesariamente la participación de diferentes partes interesadas de la organización, lo que implica que la vista técnica especializada de los profesionales de ciberseguridad/seguridad deberá ser traducida y leída por aquellos que no están en este dominio. Lo anterior, exige apertura tanto de las áreas de negocio para conocer lo que sus colegas de seguridad hacen, así como las necesidades y dinámicas que ocurren a nivel de los procesos de la operación de la empresa. Esto es, motivar una vista sistémica de la organización para encontrar puntos de encuentro y relaciones que inicialmente no son evidentes ni visibles en la dinámica actual de la organización [11].

El discurso metodológico demanda que se tenga un patrocinador o líder de alto nivel (nivel ejecutivo) que convoque y motive las sesiones de trabajo previstas, con la vista centrada en la protección de la organización y su reputación, con el fin de darle la importancia al ejercicio y los resultados esperados que conecten a cada uno de los participantes. De igual forma, se requiere un facilitador especializado que oriente y dirija las conversaciones para conectar los diferentes puntos de vista, para lo cual es clave contar con herramientas tecnológicas para trabajo colaborativo [12].

La guía metodológica está compuesta de cinco fases como se indica en la figura 1 y detallan a continuación.


Fig. 1. Guía metodológica para construir "libros de jugadas" para riesgos cibernéticos específicos (Elaboración propia)

La primera fase de la guía metodológica es lo que se denomina el contexto. Luego de seleccionado por parte del equipo ejecutivo del proyecto (generalmente conformado por el ejecutivo corporativo, el ejecutivo de ciberseguridad y su equipo, así como el gerente del proyecto) el riesgo cibernético específico, basado en una valoración y revisión interna que es guiada por la probabilidad de la materialización del riesgo y sus posibles impactos, se adelanta a todos los participantes de las áreas convocadas una presentación conceptual del incidente: definición, características, algunos indicadores y posibles impactos.

Esta presentación tiene como objetivos establecer un lenguaje común sobre el riesgo y las diferentes formas en las cuales se puede materializar en una organización. Durante este momento, igualmente interviene el ejecutivo de seguridad para incorporar ejemplos internos y los casos que se han presentado con los impactos respectivos. La presentación incluye eventos en el sector específico de la organización y algunas estadísticas recientes que muestran la relevancia del tema y los efectos que se han presentado por la acción exitosa

de los atacantes. Esta presentación toma 20 minutos.

Luego de la presentación, se tiene previamente preparada y configurada una herramienta tecnológica de trabajo colaborativo, donde se tiene establecidos los tres momentos del LJ/PB (antes, durante y después) donde de forma individual cada persona que participa, comienza a incorporar sus propuestas de los que se debería hacer desde su área en cada uno de los momentos. Cada participante es libre de incorporar allí su visión de lo que es necesario hacer y quiénes más de su proceso o área debe intervenir. Esta etapa de trabajo individual toma alrededor de 60 minutos.

El siguiente paso se denomina revisión cruzada. Este espacio busca que cada persona de cada área revise los apuntes o propuestas de las otras, para complementar el ejercicio realizado en la etapa anterior. Este proceso implica reconocer la vista y las estrategias de las otras dependencias frente al tratamiento del riesgo materializado. En este punto, se descubren perspectivas desconocidas hasta el momento o impactos que no se tenían previstos al materializarse el riesgo. Esta etapa de trabajo individual y colectivo tiene una duración promedio de 30 minutos.

Surtida la revisión cruzada se inicia el momento de la construcción conjunta donde cada una de las personas participantes de las áreas, establece los puntos de encuentro de las acciones detalladas en la plataforma las cuales consolidan una postura uniforme de acción organizacional. De igual forma, se detallan los puntos de desencuentro donde las diferencias y vistas diferentes se analizan teniendo como fondo la mejor opción posible para proteger la organización durante la atención del incidente que se ha materializado. Finalmente, se identifican las brechas, esto es, los puntos que no han sido tratados por ninguno de los participantes, que pueden ser de interés y relevantes para efectos de complementar las acciones previamente detalladas. En este punto, el facilitador especializado puede intervenir formulando preguntas que lleven a la revisión de esos posibles puntos ciegos que aún no se han concretado ni revisado por los participantes del ejercicio. Esta etapa toma en promedio 30 minutos de interacción.

La dinámica termina con una sesión de lecciones aprendidas donde se consolidan los aprendizajes y reflexiones realizadas durante las cuatro fases anteriores. Esto es, confirmar aquellos puntos ciegos o actividades que no se conocían que se realizan, o aquellas que no se hacen, descubriendo interacciones con otras áreas hasta ahora desconocidas, lo que habilita nuevos canales de comunicación y coordinación que permiten fortalecer las acciones claves que se deben asegurar y simular para efectos de hacer a la organización más resistente y mejor preparada frente a eventos inesperados. Este momento concluye luego de 20 minutos de interacción.

La aplicación del proceso metodológico toma alrededor de 160 minutos efectivos, con unos 30 minutos adicionales de descanso para no saturar ni distraer la atención de los participantes y hacer del espacio de trabajo un momento de concentración y conversación amable y abierta.

## V. APLICACIÓN DE LA GUÍA METODOLÓGICA PARA CONSTRUIR "LIBROS DE JUGADAS" PARA RIESGOS CIBERNÉTICOS ESPECÍFICOS.

La aplicación de este proceso metodológico se adelantó durante el año 2021 con una organización del sector financiero centroamericano que goza de reconocimiento en la región. El ejercicio realizado es parte del plan y estrategia de ciberseguridad que tiene la empresa, la cual se encuentra apalancada desde la Vicepresidencia Ejecutiva de Operaciones y Tecnología, dependencia a la cual reporta directamente el Director de Ciberseguridad. Esta iniciativa al contar con el apoyo decidido de la vicepresidencia en mención, permitió la participación decidida de cada una de las áreas invitadas (tecnología de información, riesgos, legal, mercadeo, servicio al cliente, auditoría, fraude, el asegurador (invitado externo), continuidad) y el fortalecimiento de la práctica de seguridad y control, como una lectura más de negocio que técnica u operativa.

Adicionalmente, es importante anotar el reconocimiento de la temática en la organización, lo cual permite habilitar canales de comunicación y relacionamiento más cercanos entre los procesos que hablan de la dinámica que ha desarrollado el área de seguridad al interior de la dinámica de los negocios de esta empresa.

Previo al desarrollo de la guía metodológica, se adelantó una sesión de trabajo inicial para la validación de expectativas, la coordinación del trabajo, establecer las personas de contacto y enlace con las áreas y los acuerdos formales del cronograma de actividades y sus fechas respectivas. Esta sesión se convierte en un elemento claves de éxito, comoquiera que se requiere este equipo para canalizar las inquietudes y asegurar las agendas de las personas que van a participar.

La primera fase de la guía metodológica que se refiere al contexto, requiere una valoración y revisión interna de los riesgos cibernéticos que es guiada por la probabilidad de la materialización del riesgo y sus posibles impactos, un ejercicio que se realiza con base en la información y práctica de seguridad/ciberseguridad disponible en la organización. Un ejemplo del ejercicio se plantea en la figura 2. El resultado de este ejercicio estableció que los riesgos cibernéticos a trabajar para concretar sus LJ/PB fueron: denegación de servicio, fuga de información y ransomware (secuestro de datos).

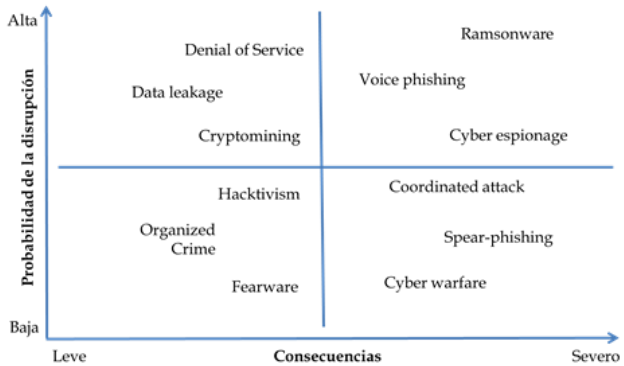



Fig. 2. Priorización de vulnerabilidades cibernéticas (Basado en: [13], p.20)

Luego el facilitador especializado adelanta la presentación del contexto del riesgo cibernético (para este caso, el de

ransomware), donde se detalla la definición, sus características, algunos indicadores y posibles impactos. Durante dicha presentación, el ejecutivo de seguridad/ciberseguridad comentó detalles de eventos que se había manifestado recientemente y algunas estadísticas disponibles del sector que hablaban del incremento de la amenaza y sus repercusiones cada vez más adversas. Luego de la presentación, los participantes de la Vicepresidencia de Tecnología, dieron más detalles de casos del sector y los dos elementos claves más relevantes para atender este riesgo que son: el control del contagio y la agilidad de la reacción y coordinación de actividades.

Se activa y comparte el enlace a la herramienta de trabajo y construcción colaborativa disponible de acceso abierto (*Padlet*, aplicación web desarrollada por empresa con base en San Francisco, USA y Singapur) con cada uno de los momentos del LJ/PB (antes, durante y después) y con los espacios disponibles para cada una de las áreas participantes con el fin de que cada una de las personas puedan escribir y aportar desde su perspectiva que se debe hacer desde su posición frente a un evento adverso, en este caso el "secuestro de datos". Una vista de este momento se puede observar en la figura 3.


Fig. 3. *Padlet* – Herramienta de apoyo y trabajo colaborativo

Terminada esta fase del ejercicio, el facilitador especializado invita a los participantes a iniciar el siguiente momento que es la revisión cruzada, donde las personas repasan y complementan los aportes de sus compañeros de las otras áreas, donde de forma paralela van escribiendo sus reflexiones y recomendaciones, de viva voz van participando comentando lo que han visto en los aportes realizados por sus pares en el ejercicio. En este punto para el tema del *ransomware*, el área de continuidad de negocio descubrió que si los equipos que tienen dispuestos para la contingencia se contaminan no lo pueden usar, lo que implica cambiar el procedimiento previsto hasta ese momento y ajustar los planes de movilidad de las personas a los sitios alternos de operación.

Concluida esta fase de la guía, el facilitador orienta para iniciar la construcción conjunta donde se consolidan los aportes de cada área participante. Se hizo evidente punto de desencuentro en los momentos de actuación entre las áreas legales y mercadeo, en cuanto la notificación a los entes de supervisión, lo que significó establecer un vía de consenso donde ambas partes se abren a cooperar una en la construcción del mensaje y la otra en su consolidación y envío. Es importante aclarar que la relación con el órgano supervisor se hace con el área jurídica exclusivamente.

De otra parte se identificó un aspecto que no se había concretado, que era el momento en que se debía notificar al asegurador para activar el mecanismo previsto en la póliza de riesgo cibernético. En este punto, el representante del asegurador comentó que la empresa tiene las primeras 24 horas luego de validar el incidente para contactarlos e iniciar el protocolo de atención que se tiene previsto, el cual consiste entre otros temas, el apoyo de un negociador especializado si la empresa decide hacer este proceso con el atacante.

Finalizada la sesión anterior, se lleva a los participantes a incluir en otro *padlet*, sus reflexiones y opiniones del ejercicio, orientados por tres preguntas base:

- ¿Qué cosas vamos a seguir haciendo, que son útiles?
- ¿Qué cosas vamos a dejar de hacer que no suman al ejercicio?
- ¿Qué cosas vamos a hacer distintas de ahora en adelante?

Mientras algunos participantes registran sus comentarios en el *padlet*, otros de viva voz van comentando los que aprendieron y desaprendieron con el ejercicio. Particularmente, las áreas fuera de tecnología se dieron cuenta de la multitud de procesos y estrategias que se activan cuando ocurre un *ransomware* y la necesidad de coordinación y apoyo que se requiere de ellos para su contención y aseguramiento. Adicionalmente, se insistió por parte de los participantes en que este tipo de ejercicios se deben seguir haciendo pues permite crear nuevos puentes de comunicación y mejores formas de cooperación para hacer más resistente a la entidad en estos momentos.

La auditoría, como participante observador, igualmente pudo detallar aspectos que pueden ser susceptibles de mejora en la coordinación entre las áreas de servicio al cliente, mercadeo y seguridad de la información, como quiera que las expectativas y noticias fluyen rápidamente al cliente por diferentes medios, y por lo tanto hay que asegurar los mensajes en forma, tiempo y modo, para mantener informados a los distintos públicos tanto internos como externos.

El área de seguridad de la información consolidó su red contactos y su papel como orquestador de la aplicación del LJ/PB resultado, toda vez que como área especializada conoce los alcances del evento y puede establecer canales de coordinación con los encargados de las áreas participantes, para asegurar los entregables requeridos y así poder canalizar los esfuerzos que habiliten una respuesta ordenada, adecuada y ajustada a las necesidades y retos que el momento demanda.

## VI. Conclusiones

La inevitabilidad de la falla es una condición base que todas las organizaciones deben tener clara en un contexto digital y digitalmente modificado. La inestabilidad geopolítica y la asimetría del flujo de información en la actualidad crean escenarios inesperados que plantean retos para la dinámica de los negocios, los cuales deben ser atendidos no sólo desde la perspectiva técnica, sino desde la visión global y sistémica de la organización. Esto implica reconocer el nuevo entorno cibernético como una realidad emergente donde es posible concretar nuevas propuestas de creación de valor para los clientes y al tiempo, un ambiente de amenazas novedosas y emergentes que pueden inhabilitar a la organización de forma real y concreta [14].

En este sentido, los equipos ejecutivos de las organizaciones deben sintonizar la lectura estratégica de la compañía con las incertidumbres y ambigüedades que se derivan de un aumento de la densidad digital, donde los objetos físicos incrementan su conectividad y habilitan flujos de información, creando una mayor superficie de posibles ataques, los cuales terminarán afectando a los diferentes grupos de interés de la empresa [15]. Así las cosas, más allá de la atención del incidente que se pueda derivar de un evento adverso, la empresa deberá habilitar un LJ/PB que le permita no sólo atender técnicamente el ataque, sino desarrollar un esfuerzo coordinado de acciones que mantengan la operación y cuiden su reputación.

La construcción de un LJ/PB se convierte en un reto empresarial, dada la necesidad de fundar una vista conjunta y holística de las actividades que se requieren para orquestar los diferentes momentos de inestabilidad que genera un incidente de seguridad. Esto implica habilitar lugares comunes de conversación entre las diferentes áreas de la organización, para lo cual se deben establecer espacios de colaboración y cooperación que sumen desde diferentes ópticas y así, se elabore un plan de acción con un lenguaje común, con un fin específico y unas responsabilidades claves que den cuenta del fin último del "libro de jugadas": coordinación, comunicación, respuesta y aprendizaje frente a los incidentes de seguridad [4].

La guía metodológica propuesta en este artículo reconoce en la intervención abierta, activa y transparente de cada uno de los participantes del ejercicio realizado, un insumo base para la construcción del LJ/PB, toda vez que al consolidar y concretar las reflexiones realizadas, cada área se ve representada de forma real y explícita lo que habilita un escenario de cooperación que genera una respuesta ágil y una mayor conexión con el área de seguridad de la información. Las cinco fases de la guía y el uso de la herramienta de trabajo colaborativo privilegian el diálogo entre los diferentes actores organizacionales, lo cual permite de forma natural revelar posibles puntos ciegos vigentes en el modelo de seguridad y control, lo que es un efecto colateral que termina enriqueciendo los resultados del ejercicio.

Con la aplicación de guía metodológica detallada en este artículo, la organización del sector financiero logró concretar el desarrollo de tres (3) LJ/PB básicos (actividades básicas claves definidas para actuar) en un marco de siete (7) semanas de trabajo, en las cuales el equipo del proyecto, el compromiso ejecutivo y la dinámica de diálogo y construcción colectiva, fueron fundamentales para crear un espacio de apertura y conexión con los participantes en cada una de las sesiones desarrolladas.

Si bien concretar los documentos básicos de los LJ/PB fueron logros claves que apalancan la vista estratégica de la compañía, es claro que el reto siguiente será la simulación de los eventos adversos que se tuvieron como alcance de los "libros de jugadas", para desarrollar el músculo de memoria colectiva requerido para responder y actuar de forma adecuada frente al ataque, y de este modo, mantener actualizado los LJ/PB y así permanecer en una espiral de conocimiento y aprendizaje ascendente que permita a la compañía reconocer y limitar la falsa sensación de seguridad.

## AGRADECIMIENTOS

El autor expresa su agradecimiento al equipo de trabajo y los ejecutivos de la empresa del sector financiero centroamericano por su compromiso estratégico con la seguridad/ciberseguridad empresarial y por la oportunidad para acompañarlos en su ejercicio de construcción de los libros de jugadas siguiendo la propuesta metodológica detallada en este documento.

## REFERENCIAS

[1] EY: "Are you reframing your future or is the future reframing you? Megatrends 2020 and beyond". *EY Megatrends*. https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/megatrends/ey-megatrends-2020-report.pdf, 2020.

[2] Vasella, T: "Incident response playbooks. Indispensable in future crisis situation". https://www.scip.ch/en/?labs.20190103, 2020.

[3] Caltagirone, S., Pendergast, A. & Betz, C.: "The Diamond Model of Intrusion Analysis". US Department of Defense. *Technical Report*. https://apps.dtic.mil/sti/pdfs/ADA586960.pdf, 2013.

[4] Cano, J.: "De las incertidumbres claves, los "libros de jugadas" y la gestión dinámica de riesgos. Conceptos que retan el statu quo de la ciberseguridad empresarial". *Global Strategy*. Global Strategy Report. No.8. https://global-strategy.org/de-las-incertidumbres-claves-los-libros-de-jugadas-y-la-gestion-dinamica-de-riesgos-conceptos-que-retan-el-statu-quo-de-la-ciberseguridad-empresarial/, 2021.

[5] Hoffman, W. & Levite, A.: "Private sector cyberdefense. Can active measures help stabilize cyberspace". Washington, D.C., USA: *Carnegie Endowment for International Peace*. https://carnegieendowment.org/2017/06/14/private-sector-cyber-defense-can-active-measures-help-stabilize-cyberspace-pub-71236, 2017.

[6] W. Stallings, *Effective cybersecurity. A guide to using best practices and standards*. New York, USA: Addison Wesley, 2018.

[7] Cano, J., "Ciberriesgo. Aprendizaje de un riesgo sistémico, emergente y disruptivo". *Revista SISTEMAS*. Asociación Colombiana de Ingenieros de Sistemas. 63-73. https://doi.org/10.29236/sistemas.n151a5, 2019.

[8] Donaldson, S., Siegel, S., Williams, C. & Aslam, A., *Enterprise Security. How to build a successful cyberdefense program against advanced threats*. New York, USA: Apress, 2015

[9] Eling, M. & Schnell, W., "What do we know about cyber risk and cyber risk insurance?" *The Journal of Risk Finance*. 17(5). 474-491. Doi: https://doi.org/10.1108/JRF-09-2016-0122, 2016.

[10] Bollinger, J., Enright, B. & Valites, M., *Crafting the InfoSec Playbook*. Sebastopol, CA. USA: O'Reilly, 2015.

[11] Espejo, R. & Reyes, A., *Sistemas organizacionales. El manejo de la complejidad con el modelo del sistema viable*. Bogotá, Colombia: Universidad de los Andes-Universidad de Ibagué, 2016.

[12] Angafor, G., Yevseyeva, I. & He, Y., "Game-based learning: A review of tabletop exercises for cybersecurity incident response training". *Security and Privacy*. 3:e126. https://doi.org/10.1002/spy2.126, 2020.

[13] Sheffi, Y., *The Resilient Enterprise. Overcoming Vulnerability for Competitive Advantage*. Cambridge, MA. USA. MIT Press, 2005.

[14] Daniel, M., "Why Is Cybersecurity So Hard?" *Harvard Business Review*. https://hbr.org/2017/05/why-is-cybersecurity-so-hard, 2017.

[15] Sieber, S. & Zamora, J., "The Cybersecurity Challenge in a High Digital Density World". *European Business Review*. November. https://www.europeanbusinessreview.com/the-cybersecurity-challenge-in-a-high-digital-density-world/, 2018.

.

# PN-secuencias entrelazadas de polinomios diferentes

Sara D. Cardell
Centro de Matemática, Computação e Cognição, Universidade Federal do ABC
09210-580, Santo André-SP, Brasil
s.cardell@ufabc.edu.br

Amparo Fúster-Sabater
Instituto de Tecnologías Físicas y de la Información (ITEFI)
CSIC, Serrano 144, 28006, Madrid, España
amparo@iec.csic.es

Verónica Requena
Dept. de Matemáticas, Universidad de Alicante
03690, Alicante, España
vrequena@ua.es

***Resumen*—Las PN-secuencias generadas por los LFSRs exhiben buenas propiedades estadísticas; sin embargo, debido a su linealidad intrínseca, no son adecuados para aplicaciones criptográficas. Para romper esta linealidad, se pueden implementar distintas estrategias. Por ejemplo, se pueden entrelazar distintas PN-secuencias para incrementar la complejidad lineal. En este trabajo, comparamos las secuencias resultantes al entrelazar PN-secuencias binarias obtenidas a partir de polinomios característicos diferentes con el mismo grado y las obtenidas con los desplazamientos de una PN-secuencia fija. Además, analizamos el periodo y la complejidad lineal, además de otras propiedades criptográficas importantes de tales secuencias.**

***Index Terms*—PN-secuencia, secuencias entrelazadas, complejidad lineal, aleatoriedad.**

## I. Introducción

El rápido desarrollo y la evolución de internet ha hecho posible la conectividad entre multitud de dispositivos de uso diario y, consecuentemente, la irrupción de lo que conocemos como el internet de las cosas (IoT, sus siglas en inglés). Además, muchos servicios como: e-banking, e-govern, e-health o e-commerce se basan en infraestructuras IoT. La presencia de dichos servicios crece de forma exponencial, y con ello todos los riesgos asociados a su seguridad [1]. En este contexto, la criptografía ligera, en general, y los cifrados de flujo, en particular, son la base sobre la que se diseñan protocolos de comunicación y dispositivos IoT cuya seguridad esté garantizada.

Los cifrados de flujo están relacionados con la idea de pseudo-aleatoriedad. De hecho, el propósito de los Generadores de Números Pseudo-Aleatorios (PRNG, sus siglas en inglés) es producir secuencias de números que parecen comportarse como si fueran generados aleatoriamente a partir de una distribución de probabilidad específica. Los PRNG deben ser rápidos y fáciles de implementar, mostrando pequeños requisitos de memoria y buenas propiedades estadísticas. Por ello, la aplicación de estos generadores en IoT se perfila como algo cada vez más necesario [2], [3].

Muchas de las secuencias de bits pseudoaleatorias con aplicación en criptografía se generan mediante registros de desplazamiento con realimentación lineal (LFSR, sus siglas en inglés) de longitud máxima [4]. Sus secuencias de salida son las PN-secuencias que exhiben buenas propiedades estadísticas. Sin embargo, su linealidad, es decir, su previsibilidad, las hace vulnerables frente a ataques criptoanalíticos. Una forma común de romper esta linealidad es a través de la decimación irregular, que ha dado lugar a una amplia familia de generadores de secuencias basados en esta técnica. Un elemento representativo de esta familia es el generador shrinking. En [11], se demuestra que la secuencia de salida de este generador se puede obtener entrelazando secuencias desplazadas de una misma PN-secuencia. Además, estos desplazamientos se pueden deducir fácilmente de los polinomios característicos de los LFSR, lo cual se puede utilizar para implementar ataques criptoanalíticos [6]. En [7], se generaliza esta construcción proponiendo el entrelazado de secuencias desplazadas de una única PN-secuencia, considerando estos cambios (diferentes de los usados en la secuencia shrunken) como parte de la clave. Esta idea dificultaría aún más el criptoanálisis de tales secuencias. Sin embargo, dependiendo del estado inicial del LFSR, algunas de las secuencias resultantes mostraron una baja complejidad lineal. En este trabajo, proponemos un análisis similar al desarrollado en [7], pero considerando el entrelazado de diferentes PN-secuencias del mismo grado. Las secuencias aquí analizadas presentan una complejidad lineal bastante mayor e independiente de los estados iniciales.

Este artículo se organiza de la siguiente manera: En la sección II, recordamos algunos conceptos básicos relacionados con las secuencias binarias, necesarios para comprender el resto del artículo. En la sección III, estudiamos la complejidad lineal y el polinomio característico de las secuencias obtenidas entrelazando PN-secuencias de diferentes LFSR. Posteriormente, en la Sección IV, comparamos nuestras secuencias con las obtenidas mediante otros generadores con parámetros similares, a la vez que realizamos un análisis en profundidad de la aleatoriedad de las secuencias obtenidas. Finalmente, en la Sección V presentamos las conclusiones de nuestro trabajo y las líneas de investigación futuras.

## II. Preliminares

Consideremos $\mathbb{F}_2 = \{0, 1\}$ el cuerpo de Galois de dos elementos. Sea $\{u_i\}_{i\geq 0} = (u_0, u_1, u_2, \ldots)$ una secuencia binaria, es decir, cada término cumple que $u_i \in \mathbb{F}_2$, para todo $i \geq 0$. La secuencia $\{u_i\}_{i\geq 0}$ (o simplemente $\{u_i\}$) se dice que es periódica si existe un entero positivo $T$ tal que $u_{i+T} = u_i$, para todo $i \geq 0$. Este número $T$ se conoce como el periodo de la secuencia. Sea $L$ un entero positivo y $a_0, a_1, \ldots, a_{L-1}$ elementos de $\mathbb{F}_2$. La secuencia $\{u_i\}$ es una secuencia binaria de recurrencia lineal de orden $L$ si satisface para todo $i \geq 0$

$$u_{i+L} = a_{L-1}u_{i+L-1} + a_{L-2}u_{i+L-2} + \cdots + a_1u_{i+1} + a_0u_i, \quad (1)$$

La expresión en (1) se conoce como la relación de recurrencia lineal de orden $L$. El polinomio de grado $L$ dado por

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{L-1}x^{L-1} + x^L \in \mathbb{F}_2[x],$$

se conoce como el polinomio característico de la relación de recurrencia lineal así como también el polinomio característico de $\{u_i\}$.

La generación de estas secuencias de recurrencia lineal puede ser implementada por los LFSRs [4]. Un LFSR de longitud $L$ es un generador de secuencia binaria con celdas $L$ o etapas interconectadas. Los términos $(a_0, a_1, a_2, \ldots, a_{L-1})$ son coeficientes binarios asignados a las etapas correspondientes. El estado inicial (contenido de las celdas en el instante cero) es la semilla, y dado que el registro opera de forma determinista, la secuencia resultante está completamente determinada por el estado inicial. En cada pulso de reloj, el contenido binario de cada etapa se desplaza una posición hacia la izquierda y se emite un bit del registro. La entrada de cada ronda es un bit resultante de aplicar una función de transformación lineal a un estado anterior (ver Figura 1).

Si el polinomio característico $p(x)$ es primitivo, entonces se dice que el LFSR es de longitud máxima y la secuencia resultante, conocida como PN-secuencia, tiene periodo $T = 2^L - 1$ (con $2^{L-1}$ unos y $2^{L-1} - 1$ ceros) [4].

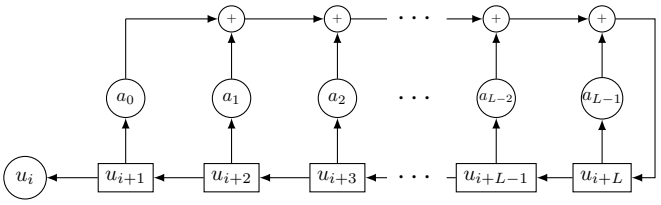



Figura 1. LFSR de longitud $L$ (o LFSR con $L$ etapas)

La complejidad lineal de una secuencia, denotada por $LC$, se define como la longitud del LFSR más corto que genera dicha secuencia, es decir, el grado de su polinomio característico. En criptografía, $LC$ debe ser lo más grande posible.; el valor esperado es aproximadamente la mitad del periodo $LC \simeq T/2$ (ver [5]). Hoy en día, los valores de $T$ dentro del rango $T \geq 2^{128}$, es decir, $LC \simeq 2^{127}$, parecen ser suficientes para propósitos criptográficos (ver especificaciones de los candidatos en la convocatoria del NIST para primitivas criptográficas ligeras [8]). Nótese que todos los ejemplos incluidos en este trabajo son meramente ilustrativos, ya que no alcanzan los valores requeridos para aplicaciones criptográficas.

### *II-A. Generador Shrinking*

Comenzaremos recordando el concepto de decimación. La decimación de la secuencia $\{s_i\}$ por (distancia) $\delta$ es la nueva secuencia $\{u_i\} = \{s_{\delta \cdot i}\}$, obtenida tomando cada $\delta$-ésimo término de dicha secuencia [9].

El generador de secuencias binarias conocido como *Generador Shrinking* (SG, sus siglas en inglés), véase [10], se construye a partir de dos LFSRs de máxima longitud, $R_1$ y $R_2$, con longitudes $L_1$ y $L_2$, respectivamente, y $\text{mcd}(L_1, L_2) = 1$. Denotemos por $p_k \in \mathbb{F}_2[x]$, con grado $L_k$, el polinomio característico $R_k$, y $T_k = 2^{L_k} - 1$, el periodo de la correspondiente PN-secuencia, para $k = 1, 2$. La PN-secuencia $\{a_i\}$ generada por $R_1$ decima la PN-secuencia $\{b_i\}$ producida por el otro registro $R_2$. La regla de decimación satisface que: dado $a_i$ y $b_i$, $i = 0, 1, 2, \ldots$, la secuencia de salida $\{s_j\}$ se obtiene como

$$\begin{cases} \text{Si } a_i = 1, \text{ entonces } s_j = b_i. \\ \text{Si } a_i = 0, \text{ entonces } b_i \text{ se descarta.} \end{cases}$$

La secuencia $\{s_j\}$ se conoce como *secuencia shrunken* cuyo periodo es $T = (2^{L_2} - 1)2^{L_1 - 1}$. Su complejidad lineal [11] satisface la desigualdad $L_2 2^{L_1 - 2} < LC \leq L_2 2^{L_1 - 1}$ y su polinomio característico tiene la forma $p(x)^m$, donde $2^{L_1 - 2} < m \leq 2^{L_1 - 1}$ y $p(x)$ es un polinomio primitivo de grado $L_2$ [12]. El polinomio $p(x)^m$ denota la $m$-ésima potencia del polinomio $p(x)$ con coeficientes módulo 2.

*Ejemplo 1:* Consideremos $R_1$ y $R_2$, dos LFSRs con polinomios característicos $p_1(x) = 1 + x + x^2$ y $p_2(x) = 1 + x^2 + x^3$, y estados iniciales $(11)$ y $(111)$, respectivamente. La secuencia shrunken se obtiene de la siguiente manera:

$$\begin{array}{rl} R_1: & 1\,1\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0\,1\,1\,0 \\ R_2: & 1\,1\,\cancel{1}\,0\,1\,\cancel{0}\,0\,1\,\cancel{1}\,1\,0\,\cancel{1}\,0\,0\,\cancel{1}\,1\,1\,\cancel{0}\,1\,0\,\cancel{0} \\ \{s_j\}: & \mathbf{1\,1 \quad 0\,1 \quad 0\,1 \quad 1\,0 \quad 0\,0 \quad 1\,1 \quad 1\,0} \end{array}$$

La secuencia resultante tiene periodo 14 y es fácil comprobar que su polinomio característico es $p(x)^2 = (1 + x + x^3)^2$, es decir, la complejidad lineal es $LC = 6$.

Los siguientes resultados indican que la secuencia shrunken se puede obtener entrelazando versiones desplazadas de una PN-secuencia dada.

*Teorema 1:* [11, Teorema 3.1] Las secuencias obtenidas decimando por (distancia) $2^{L_1 - 1}$ la secuencia shrunken son PN-secuencias con periodo $T_2$. A estas secuencias las llamamos PN-secuencias entrelazadas de la secuencia shrunken.

*Teorema 2:* [11, Teorema 3.3] El polinomio primitivo $p(x)$ que genera las PN-secuencias entrelazadas de la secuencia shrunken es

$$p(x) = (x + \alpha^{T_1})(x + \alpha^{2T_1})(x + \alpha^{4T_1}) \cdots (x + \alpha^{2^{L_2 - 1} T_1}), \quad (2)$$

donde $\alpha \in \mathbb{F}_{2^{L_2}}$ es una raíz primitiva del polinomio $p_2(x)$.

*Corolario 1:* [11, Corollary 1] Si $L_2 = L_1 + 1$, entonces el polinomio $p(x)$ obtenido en (2) es el polinomio recíproco de $p_2(x)$.

*Ejemplo 2:* Sean $R_1$ y $R_2$ dos LFSRs con polinomios característicos $p_1(x) = 1 + x^2 + x^3$ y $p_2(x) = 1 + x^3 + x^4$, con $L_1 = 3$ y $L_2 = 4$, y estados iniciales $(111)$ y $(1111)$, respectivamente. Las correspondientes PN-secuencias tienen periodos $T_1 = 7$ y $T_2 = 15$, respectivamente. La secuencia shrunken obtenida es

$$\{s_j\} = (111011010111011000111010000101 \\ 011001101101001100001011111000)\,.$$

Tiene periodo $T = (2^{L_2} - 1)2^{L_1 - 1} = 60$ y polinomio característico $p(x)^4 = (1 + x + x^4)^4$, es decir, la complejidad lineal es $LC = 16$. Si decimamos la secuencia shrunken por $\delta = 4$, entonces, obtenemos las 4 PN-secuencias

$$\begin{array}{rl} \{s_{4 \cdot j}\}: & (110001001101011) \\ \{s_{4 \cdot j + 1}\}: & (111100010011010) \\ \{s_{4 \cdot j + 2}\}: & (101111000100110) \\ \{s_{4 \cdot j + 3}\}: & (011010111100010) \end{array}$$

El polinomio característico de estas 4 PN-secuencias entrelazadas es

$$p(x) = \left(x + \alpha^7\right)\left(x + \alpha^{14}\right)\left(x + \alpha^{28}\right)\left(x + \alpha^{56}\right) = 1{+}x{+}x^4$$

donde $\alpha \in \mathbb{F}_{2^{L_2}}$ es una raíz de $p_2(x)$ y $p(x)$ es el polinomio recíproco de $p_2(x)$. Observemos que las 4 PN-secuencias son versiones desplazadas de la misma PN-secuencia.
El polinomio $p(x)$ depende de $L_1$ (el grado de $p_1(x)$) y del polinomio $p_2(x)$. Por tanto, cada polinomio primitivo de grado $L_1$ producirá el mismo polinomio $p(x)$, una vez fijado el polinomio $p_2(x)$.

Notemos que si $p(x)$ genera las PN-secuencias entrelazadas de la secuencia shrunken, entonces $p(x)^{2^{L_1-1}}$ genera dicha secuencia. Sin embargo, aunque $p(x)^{2^{L_1-1}}$ siempre genera la secuencia shrunken, no tiene porqué ser el polinomio característico de ésta.

### *II-B. Entrelazando PN-secuencias con un único polinomio característico*

En la Sección II-A, vimos que la secuencia shrunken se puede generar entrelazando versiones desplazadas de la misma PN-secuencia, y el polinomio característico de estas PN-secuencias se obtiene a partir de los polinomios de entrada del generador shrinking. Los desplazamientos de las secuencias desplazadas también se pueden obtener a través de los LFSRs de entrada (ver [11], [6]) y este hecho se usa para poder atacar el generador shrinking [6]. Una forma de enmascarar esta debilidad del generador, sería considerar desplazamientos aleatorios.

En esta subsección comentamos brevemente los resultados obtenidos en [7]. Primero, necesitamos introducir el concepto de *secuencia $t$-entrelazada*. Decimos que la sucesión $\{s_j\}$ se obtiene entrelazando las sucesiones $\{u_i^{(1)}\}$, $\{u_i^{(2)}\}$, ..., $\{u_i^{(t)}\}$, todas ellas de periodo $T$, si tiene la siguiente forma

$$\{s_j\} = \Big(u_0^{(1)}, u_0^{(2)}, \ldots, u_0^{(t)}, u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(t)}, \ldots, u_{T-1}^{(1)}, u_{T-1}^{(2)}, \ldots, u_{T-1}^{(t)}\Big).$$

En [7], las autoras consideran que estas $t$ secuencias $\{u_i^{(j)}\}$ para $j = 1, 2, \ldots, t$, son PN-secuencias obtenidas a partir del mismo polinomio primitivo, es decir, secuencias desplazadas de la misma PN-secuencia. Si el LFSR correspondiente tiene una longitud $L$, entonces la secuencia $t$-entrelazada resultante es casi equilibrada y su número de 1s es $t \cdot 2^{(L-1)}$. Además, la complejidad lineal de esta secuencia satisface $LC \leq t \cdot L$ y su periodo $T \leq t \cdot (2^L - 1)$. Para un valor fijo de $t$, casi el 90 % de las secuencias $t$-entrelazadas (recorriendo todas las secuencias trasladadas posibles) alcanzan el máximo valor del periodo y de $LC$. En [7], se profundiza en los casos donde $t$ es una potencia de 2 y se realiza un análisis preliminar sobre la aleatoriedad de estas secuencias. También se proporcionan algunas herramientas para identificar los casos en los que $LC$ es bajo y, por tanto, las secuencias no son adecuadas para fines criptográficos. Se puede encontrar más información sobre estas secuencias y una comparativa con las secuencias construidas en este trabajo en la Sección IV.

A partir de ahora, consideraremos que las secuencias $t$-entrelazadas con las que trabajamos se obtienen entrelazando PN-secuencias de diferentes polinomios primitivos del mismo grado. Tengamos en cuenta que estas secuencias $t$-entrelazadas pueden verse como las secuencias de salida de un generador de secuencia donde, a cada pulso de reloj, obtenemos al mismo tiempo la salida de $t$ LFSRs distintos. Es decir, en cada instante $t_i$ los bits de salida son $\left(u_{t_i}^{(1)}, u_{t_i}^{(2)}, \ldots, u_{t_i}^{(t)}\right)$, para $i = 1, 2, \ldots, t$. Por tanto, el método de entrelazado, en este caso, podría considerarse como la concatenación de la salida de $t$ LFSRs en cada instante de tiempo. Por otro lado, dicho método es muy similar al método de generación de un DLFSR (Dynamic Linear Feedback Shift Register). Un DLFSR es un tipo de LFSR en el que el polinomio característico cambia en cierto instante [14], [13]. En la Figura 2, representamos un DLFSR que consta de un LFSR principal y un módulo de control adicional. Este módulo gestiona el polinomio característico utilizado en cada instante de tiempo. Las secuencias generadas por un DLFSR pueden considerarse como la concatenación de segmentos de diferentes PN-secuencias. El propósito de un DLFSR es generar secuencias con mayor periodo y complejidad lineal que las producidos por un LFSR [15], [16]. Para llevar a cabo esta tarea, el módulo de control modifica diferentes parámetros de retroalimentación para generar una secuencia diferente. Nuestro método de entrelazado puede verse como un DLFSR donde el polinomio característico cambia según el módulo contador, es decir, en cada pulso de reloj consideramos un polinomio primitivo diferente. En la Figura 3, podemos comprobar la generación de una secuencia 4-entrelazada. En cada pulso de reloj, se genera un bit del LFSR correspondiente en ese instante y luego saltamos del polinomio actual al siguiente. Así, obtenemos nuestra secuencia entrelazada concatenando las salidas individuales de cada uno de los LFSRs en cada instante de tiempo.

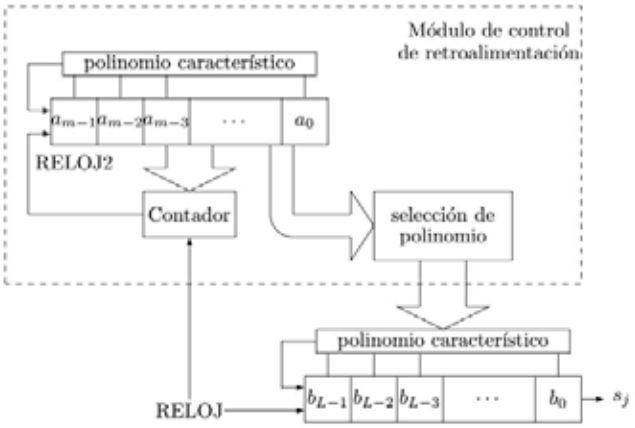



Figura 2. DLFSR

## III. Entrelazando PN-secuencias con polinomios característicos diferentes

En esa sección analizaremos el entrelazado de PN-secuencias obtenidas a partir de polinomios primitivos diferentes pero con el mismo grado.

Decimos que la secuencia $\{s_j\}$ se obtiene entrelazando las secuencias $\{u_i^{(1)}\}$, $\{u_i^{(2)}\}$, ..., $\{u_i^{(t)}\}$, todas de periodo $T$, si

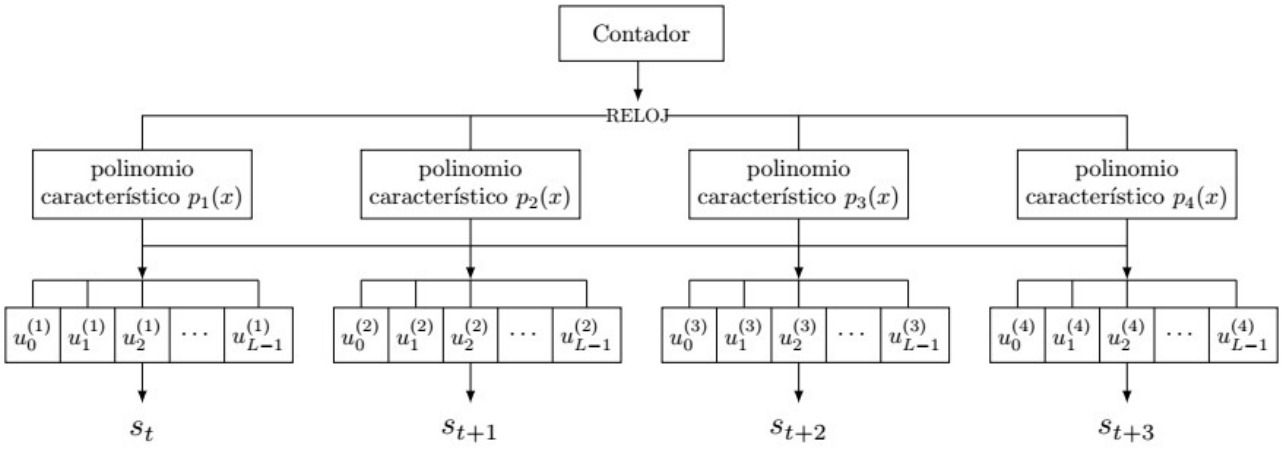



Figura 3. Generar una secuencia 4-entrelazada a partir de un DLFSR

se obtiene de la siguiente manera

$$\{s_j\} = \Big(u_0^{(1)}, u_0^{(2)}, \ldots, u_0^{(t)}, u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(t)}, \ldots, u_{T-1}^{(1)}, u_{T-1}^{(2)}, \ldots, u_{T-1}^{(t)}\Big),$$

definiéndola como la *secuencia $t$-entrelazada*.

Consideremos $t$ LFSRs de longitud máxima, $R_1$, $R_2$,..., $R_t$, con polinomios característicos primitivos $p_1(x)$, $p_2(x), \ldots, p_t(x))$, respectivamente, y todos ellos de grado $L$. Dada la PN-secuencia $\{a_i^{(k)}\}$, generada por $R_k$, para $k = 1, 2, \ldots, t$, la correspondiente secuencia $t$-entrelazada $\{s_j\}$ se obtiene de la siguiente manera

$$\{s_j\} = \Big(a_0^{(1)}, a_0^{(2)}, \ldots, a_0^{(t)}, a_1^{(1)}, a_1^{(2)}, \ldots, a_1^{(t)}, \ldots, a_{L-1}^{(1)}, a_{L-1}^{(2)}, \ldots, a_{L-1}^{(t)}, \ldots\Big).$$

A partir de ahora, sólo consideraremos secuencias $t$-entrelazadas obtenidas con diferentes polinomios del mismo grado.

El siguiente resultado proporciona el valor de $LC$ para las secuencias $t$-entrelazadas y sus polinomios característicos.

*Teorema 3:* [17, Teorema 1] La complejidad lineal de la secuencia generada entrelazando $t$ PN-secuencias producidas por diferentes polinomios primitivos $p_1(x), \ldots, p_t(x)$ de grado $L$ es $LC = t^2 L$. Además, el polinomio característico es

$$p(x) = \prod_{i=1}^{t} p_i(x^t).$$

Observemos que el $LC$ y el periodo no se ven afectados por los estados iniciales.

*Ejemplo 3:* Consideremos 3 registros con polinomios primitivos $p_1(x) = 1 + x^2 + x^5$, $p_2(x) = 1 + x + x^2 + x^4 + x^5$ y $p_3(x) = 1 + x + x^2 + x^3 + x^5$, con estados iniciales $\{11101\}$, $(10001)$ y $(10101)$, respectivamente. Las correspondientes PN-secuencias son

$$\{a_i^{(1)}\} : (11101010000100101100111111000110)$$
$$\{a_i^{(2)}\} : (10001001010110000111001101111 10)$$
$$\{a_i^{(3)}\} : (10101000111011111001001100000101).$$

Si las entrelazamos, obtenemos una secuencia de periodo $T = 93$ y $LC = 45$:

$$(11110010100011100010001000101100111001100110100111001001111111000100100101111110001)$$

Usando el algoritmo de Berlekamp-Massey [18], obtenemos su polinomio característico

$$p(x) = 1 + x^9 + x^{24} + x^{27} + x^{39} + x^{42} + x^{45} = p_1(x^3) \cdot p_2(x^3) \cdot p_3(x^3)$$

con

$$p_1(x^3) = 1 + x^6 + x^{15} = (x^5 + x^4 + x^3 + x + 1) \cdot (x^{10} + x^9 + x^7 + x^5 + x^2 + x + 1)$$
$$p_2(x^3) = 1 + x^3 + x^6 + x^{12} + x^{15} = (x^5 + x^4 + x^3 + x^2 + 1) \cdot (x^{10} + x^9 + x^7 + x^2 + 1)$$
$$p_3(x^3) = 1 + x^3 + x^6 + x^9 + x^{15} = (x^5 + x^3 + 1) \cdot (x^{10} + x^8 + x^6 + x^5 + 1)$$

donde los 3 polinomios de grado 5 son primitivos y los de grado 10 son irreducibles.

El siguiente resultado es un caso particular del Teorema 3 para el caso en que $t$ es una potencia de 2.

*Corolario 2:* Sea $t$ una potencia de dos. Entonces, el polinomio característico de una secuencia $t$-entrelazada, generada a partir de $t$ polinomios primitivos distintos $p_1(x), \ldots, p_t(x)$ de grado $L$, se obtiene de la siguiente forma

$$p(x) = [p_1(x) \cdot p_2(x) \cdots p_{t-1}(x) \cdot p_t(x)]^t.$$

*Proof:* Sea $t = 2^r$ para $r$ un entero positivo. El resultado es consecuencia inmediata de que $p_i(x^{2^r}) = p_i(x)^{2^r}$ en $\mathbb{F}_2$. ■

A continuación, mostramos diferentes ejemplos de la generación de secuencias $t$-entrelazadas. Analizamos su $LC$ y sus polinomios característicos en función de la elección de los polinomios primitivos iniciales.

En el siguiente ejemplo, obtenemos una secuencia 4-entrelazada correspondiente a dos polinomios primitivos y sus correspondientes polinomios recíprocos.

*Ejemplo 4:* Consideremos 4 LFSRs con polinomios primitivos $p_1(x) = 1 + x^2 + x^5$, $p_2(x) = 1 + x^3 + x^5$, $p_3(x) = 1+x^2+x^3+x^4+x^5$ y $p_4(x) = 1+x+x^2+x^3+x^5$. Observemos que $p_2(x)$ y $p_4(x)$ son polinomios recíprocos de $p_1(x)$ y $p_3(x)$, respectivamente. Tomemos los estados iniciales (01101), (10001), (10001) y (00110), respectivamente. Las correspondientes PN-secuencias son

$$\{a_i^{(1)}\} : (01101110101000010010111001111100)$$
$$\{a_i^{(2)}\} : (10001111100110100100000101011101)$$
$$\{a_i^{(3)}\} : (10001010110100001100100111111011)$$
$$\{a_i^{(4)}\} : (00110000101101010000111011111001)\,.$$

Si entrelazamos esas 4 PN-secuencias, obtenemos una secuencia 4-entrelazada de periodo $T = 124$ y complejidad lineal $LC = 80$, dada por

$$(01101000100100011110110011100100111100101001011101000001010010010010011010000001101110010100000111111101111111111110000100111)$$

Usando el algoritmo de Berlekamp-Massey [18], se puede verificar que el polinomio característico de esta secuencia es

$$\begin{aligned} p(x) &= [p_1(x) \cdot p_2(x) \cdot p_3(x) \cdot p_4(x)]^4 \\ &= (1 + x^2 + x^5)^4(1 + x^3 + x^5)^4 \\ &\quad (1 + x^2 + x^3 + x^4 + x^5)^4(1 + x + x^2 + x^3 + x^5)^4 \\ &= 1 + x^4 + x^8 + x^{12} + x^{20} + x^{32} + x^{36} + x^{40} + x^{44} \\ &\quad + x^{48} + x^{60} + x^{68} + x^{72} + x^{76} + x^{80}. \end{aligned}$$

En este ejemplo, el valor de $LC$ no depende de los estados iniciales, su valor es siempre 80. Además, si consideramos diferentes polinomios primitivos de grado 5, el valor de $LC$ siempre es el mismo.

El siguiente ejemplo muestra que los polinomios deben ser todos diferentes para lograr la máxima complejidad. En este caso consideraremos dos polinomios primitivos iguales y comprobaremos que $LC$ no obtiene el valor máximo.

*Ejemplo 5:* Consideremos los polinomios primitivos $p_1(x) = p_2(x) = 1+x^2+x^5$, $p_3(x) = 1+x+x^2+x^3+x^5$ y $p_4(x) = 1+x^2+x^3+x^4+x^5$ y los estados iniciales $\{10111\}$, $\{11010\}$, $\{00011\}$, y $\{10110\}$, respectivamente para cada polinomio. La secuencia 4-entrelazada correspondiente tiene la siguiente forma

$$(11010100100111111010011110000010101000100111001100000011000011011100001110111101010101011001111100110111010010001100001110)\,.$$

Dicha secuencia tiene periodo $T = 124$ y complejidad lineal $LC = 60$, este último valor no sería su valor máximo para dicho parámetro, pues $LC \leq 80$. El polinomio característico de dicha secuencia viene dado por

$$\begin{aligned} p(x) =& [p_1(x)p_3(x)p_4(x)]^4 \\ =& 1 + x^4 + x^8 + x^{16} + x^{20} + x^{24} + x^{36} + x^{56} + x^{60}. \end{aligned}$$

Observemos que, en este caso, el polinomio primitivo no es el producto de los 4 polinomios; y esto se debe a que $p_1(x) = p_2(x)$.

En la Tabla I mostramos los valores de $LC$ de las secuencias $t$-entrelazadas obtenidas a partir de PN-secuencias con polinomios distintos, todos ellos de grado $L$. Recordemos

Tabla I
$LC$ DE LAS SECUENCIAS $t$-ENTRELAZADAS CON POLINOMIOS PRIMITIVOS DE GRADO $L$

| $t$ \ $L$ | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|
| 4 | 80 | 96 | 112 | 128 | 144 |
| 5 | 125 | 150 | 175 | 200 | 225 |
| 6 | 180 | 216 | 252 | 288 | 324 |
| 7 | 245 | 294 | 343 | 392 | 441 |
| 8 | 320 | 384 | 448 | 512 | 576 |

que sólo hay 6 polinomios primitivos de grados 5 y 6. Significa que, cuando construimos secuencias 7-entrelazadas o secuencias 8-entrelazadas, tenemos que considerar al menos un polinomio repetido. Por lo tanto, los valores mostrados en rojo en la Tabla I son sólo cotas superiores para $LC$, ya que, como vimos en el ejemplo 5, cuando los polinomios no son diferentes, el valor $LC$ de la secuencia no tiene porqué ser máximo.

## IV. COMPARACIÓN CON OTROS GENERADORES DE SECUENCIAS

En esta sección analizamos brevemente las ventajas de nuestras secuencias $t$-entrelazadas en comparación con las secuencias obtenidas con generadores con parámetros similares.

*Generador Shrinking*

Dados dos polinomios primitivos de grado $L_1$ y $L_2$, la complejidad lineal de la secuencia shrunken satisface que $2^{L_1-2} < LC \leq L_2 \cdot 2^{L_1-1}$ y $T = (2^{L_2} - 1)2^{L_1-1}$. En este caso, la secuencia se obtiene entrelazando $2^{L_1-1}$ versiones desplazadas de la misma PN-secuencia.

Si intercalamos $2^{L_1-1}$ PN-secuencias generadas por diferentes polinomios primitivos de grado $L_2$, la complejidad lineal de la secuencia resultante es $LC = L_2 \cdot 2^{2(L_1-1)}$, que es mucho mayor que la del SG. Observemos que el periodo y el número de unos siguen siendo los mismos.

En el siguiente ejemplo, comparamos la secuencia shrunken y la correspondiente $t$-entrelazada con parámetros similares. Vemos que el valor de $LC$ de la secuencia $t$-entrelazada es mayor.

*Ejemplo 6:* Consideremos el generador shrinking compuesto por dos registros de longitud $L_1 = 3$ y $L_2 = 5$. En este caso, la secuencia shrunken está formada por el entrelazamiento de 4 secuencias desplazadas de una misma PN-secuencia generada por un polinomio primitivo de grado 5. El periodo de las secuencias shrunken en este caso es de $T = 124$ y $LC \leq 20$.

Si consideramos de nuevo el ejemplo 4, entrelazábamos 4 PN-secuencias generadas a partir de polinomios primitivos de grado 5. La secuencia resultante tenía periodo 124 (igual que el de la SG con polinomios de grado 3 y 5) y $LC = 80$, que es 4 veces mayor que la de la SG.

*Secuencias $t$-entrelazadas con el mismo polinomio*

En [7], se analizan las secuencias $t$-entrelazadas obtenidas entrelazando versiones desplazadas de una PN-secuencia generada por un polinomio primitivo $p(x)$ de grado $L$. Se determina el periodo y la complejidad lineal de dichas secuencias, para algunos casos particulares de $t$. Además, se obtiene una cota superior para $LC$ y el periodo de las secuencias $t$-entrelazadas. En [17], los autores estudian diferentes casos

Tabla II
COMPARACIÓN DE LOS VALORES DE $LC$ Y $T$ DE SECUENCIAS $t$-ENTRELAZADAS OBTENIDAS USANDO UN ÚNICO POLINOMIO PRIMITIVO DE GRADO $L$ Y TOMANDO DISTINTOS POLINOMIOS DE GRADO $L$.

| | $t$ polinomios diferentes | | un único polinomio | |
|---|---|---|---|---|
| $(t, L)$ | $LC$ | $T$ | $LC$ | $T$ |
| (8,16) | 1024 | 524280 | 128 | 524280 |
| (8,17) | 1088 | 1048568 | 136 | 1048568 |
| (8,18) | 1152 | 2097144 | 144 | 2097144 |
| (8,19) | 1216 | 4194296 | 152 | 4194296 |
| (8,20) | 1280 | 8388600 | 160 | 8388600 |

de secuencias entrelazadas, analizando el $LC$ y los polinomios característicos de las secuencias resultantes. El siguiente teorema es una consecuencia del Teorema 2 en [17] y los resultados obtenidos en [7].

*Teorema 4:* [7], [17] Consideremos un polinomio primitivo $p(x)$ de grado $L$. Si entrelazamos $t$ secuencias trasladadas de una PN-secuencia de periodo $T = 2^L - 1$, entonces la secuencia $t$-entrelazada resultante cumple que: $LC \leq t \cdot L$ y $T \leq t \cdot \left(2^L - 1\right)$.

En el siguiente ejemplo comparamos una secuencia $t$-entrelazada obtenida a partir de un polinomio primitivo $p(x)$ de grado $L$, con la correspondiente secuencia $t$-entrelazada, con parámetros similares, obtenida a partir de $t$ polinomios primitivos distintos $p_i(x)$ de grado $L$, $i = 1, \ldots, t$.

*Ejemplo 7:* Consideremos cualquier secuencia 4-entrelazada generada a partir de un polinomio primitivo de grado 5, y 4 secuencias trasladadas de la PN-secuencia correspondiente. En este caso, el periodo de las secuencias cumple que $T \leq 124$ y $LC \leq 20$.

Consideremos, ahora, los 4 polinomios primitivos de grado 5 dados en el ejemplo 4 y entrelacemos las 4 PN-secuencias diferentes producidas por estos polinomios. La secuencia resultante tiene periodo 124 y $LC = 80$; mejorando los valores para $LC$, siendo en este caso cuatro veces mayor.

En la Tabla II realizamos un análisis comparativo entre los valores de $LC$ y $T$ para las secuencias $t$-entrelazadas obtenidas con polinomios distintos del mismo grado, y los valores de las secuencias $t$-entrelazadas obtenidas en [7] (usando versiones desplazadas de una PN-secuencia, es decir, usando un único polinomio característico). Observemos que los valores para el mismo polinomio son cotas superiores (dependen del estado inicial), mientras que los valores para nuestras secuencias son máximos (independientes del estado inicial). Además, los valores de $LC$ son mayores considerando estas nuevas secuencias $t$-entrelazadas.

## V. CONCLUSIONES

Entrelazar secuencias es una forma de aumentar la complejidad lineal de las secuencias y de romper la linealidad. En este artículo calculamos la complejidad lineal y el periodo de las secuencias obtenidas al entrelazar PN-secuencias generadas por diferentes polinomios característicos del mismo grado; analizándolas y comparándolas con las secuencias obtenidas al entrelazar secuencias desplazadas de una misma PN-secuencia. Como trabajo futuro, nos gustaría realizar un análisis de la aleatoriedad de dichas secuencias, a través de diversas baterías estadísticas. Asimismo estudiar qué sucede si entrelazamos PN-secuencias con diferentes periodos.

## AGRADECIMIENTOS

Esta publicación es parte del proyecto de I+D+i P2QProMeTe (PID2020-112586RB-I00), financiado por MCIN/ AEI/10.13039/501100011033. La tercera autora fue parcialmente financiada por el proyecto VIGROB-287 de la Universitat d'Alacant.

## REFERENCIAS

[1] Gallegos-Segovia, P.L., Bravo-Torres, J.F., Argudo-Parra, J.J. Internet of things as an attack vector to critical infrastructures of cities *International Caribbean Conference on Devices, Circuits and Systems (ICCDCS), Mexico*, pp. 117–120, 2017.

[2] Zia, U., McCartney, M., Scotney, B., Martínez, J., Sajjad, A. A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map, *SN Applied Sciences* , vol. 4, n. 48, 2022.

[3] Kietzmann, P., Schmidt, T. C., Wählisch, M. A Guideline on Pseudorandom Number Generation (PRNG) in the IoT, *Association for Computing Machinery*, vol. 54, n. 6, 2021.

[4] Golomb, S.W. Shift Register-Sequences , *Aegean Park Press: Laguna Hill, CA, USA,* 1982.

[5] Rueppel, R.A. Linear Complexity and Random Sequences, *Advances in Cryptology — EUROCRYPT 1985, Lecture Notes in Computer Science*, vol. 219, pp.167–188, 1986.

[6] Cardell, S. D., Climent, J.-J., Fúster-Sabater, A., Requena, V. Representations of Generalized Self-Shrunken Sequences, *Mathematics*, n. 8, 2020.

[7] Cardell, S.D., Fúster-Sabater, A., Requena, V. Interleaving Shifted Versions of a PN-Sequence, *Mathematics*, vol. 9, n.68, pp. 1–23, 2021.

[8] National Institute of Standards and Technology (NIST). NIST Lightweight Cryptography Project, *Technology Administration*, https://csrc.nist.gov/Projects/Lightweight-Cryptography, 2022.

[9] Duvall, P.F., Mortick, J.C. Decimation of Periodic Sequences. *SIAM J. Appl. Math.* vol. 21,pp. 367–372, 1971.

[10] Coppersmith, D., Krawczyk, H., Mansour, Y. The shrinking generator, *Advances in Cryptology—CRYPTO '93, Lecture Notes in Computer Science*, vol. 773, pp. 22–39, 1994.

[11] Cardell, S.D., Fúster-Sabater, A. Modelling the shrinking generator in terms of linear CA. *Adv. Math. Commun.* vol.10, pp. 797–809, 2016.

[12] Fúster-Sabater, A., Caballero-Gil, P. Linear solutions for cryptographic nonlinear sequence generators. *Phys. Lett. A* vol. 369,pp. 432–437, 2007.

[13] Eljadi,A., Mohamed, F., Shaikhli, T.A., Fakhri, I. Dynamic linear feedback shift registers: A review, *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*, pp. 1-5, 2014.

[14] Mita, R., Palumbo, G., Pennisi, S., Poli, M., Pseudorandom bit generator based on dynamic linear feedback topology, *Electronic Letters*, vol. 28, n. 19, pp. 1097–1098, 2002.

[15] Peinado, A., Munilla, J., Fúster-Sabater, A. EPCGen2 Pseudorandom Number Generators: Analysis of J3Gen. *Sensors*, vol. 14,pp. 6500–6515, 2014.

[16] Stepień, R., Walczak, J. Comparative analysis of pseudo random signals of the LFSR and DLFSR generators, *Proceedings of the 20th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2013*, pp.598-602, 2013.

[17] Xiong, H., Qu, L., Li, C., Fu, S. Linear complexity of binary sequences with interleaved structure, *IET Communications*, vol. 7, n. 15, pp.1688-1696, 2013.

[18] Massey, J. L. Shift-register synthesis and BCH decoding, *IEEE Transactions on Information Theory*, vol. 15, n. 1, pp. 122–127, 1969.

# Protegiendo la identidad de las denuncias en un sistema abierto y auditable

Sergio Chica
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
sergio.chica@uc3m.es

Andrés Marín
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
andres.marin@uc3m.es

David Arroyo
ITEFI / Consejo Superior de Investigaciones Científicas
david.arroyo@csic.es

Jesús Díaz
IOHK / Input Output
jesus.diaz.vico@gmail.com

***Resumen*—En determinadas ocasiones para conocer y actuar frente a vulneraciones de la ley por parte de organizaciones, es necesario contar con información a la que únicamente tienen acceso personas con vinculación laboral o relación comercial. Cuando una de estas personas tiene acceso legítimo a las pruebas de un delito, es necesario proteger la identidad del denunciante. Es necesario proporcionar canales que garanticen el anonimato de la fuente. Aquí presentamos un trabajo en desarrollo de un sistema que proporcione anonimato a las fuentes en un sistema abierto y auditable, orientado a los sistemas de auditorías de infraestructuras críticas y basándonos en nuestro trabajo previo** AUTOAUDITOR **[1].**

***Index Terms*—*blockchain* permisionada, denuncias anónimas, firmas de grupo, ECDHE**

## I. Introducción

Las personas trabajadoras de las organizaciones suelen ser las primeras en tener conocimiento de amenazas o perjuicios para el interés público que puedan surgir en sus organizaciones. Al informar sobre infracciones del derecho perjudiciales para el interés público, dichas personas actúan como denunciantes (*whistleblowers*) y son fundamentales para poner esas infracciones en conocimiento de las autoridades protegiendo el bienestar de la sociedad. Sin embargo, el temor a las represalias suele disuadir a los potenciales denunciantes.

En la Directiva (UE) 2019/1937 [2] se reconoce la importancia de garantizar una protección equilibrada y efectiva a los denunciantes en diversas áreas: contratación pública; servicios financieros; fabricación, importación o distribución de armas de fuego, defensa y explosivos; seguridad en el transporte; en la protección del medio ambiente; seguridad nuclear; alimentos y piensos; consumo y salud pública; y en la seguridad de las redes y los sistemas de información. En este último ámbito, la directiva introduce: *el requisito de notificar incidentes, incluidos los que no pongan en peligro los datos personales, y requisitos de seguridad para las entidades que prestan servicios esenciales en numerosos sectores, por ejemplo la energía, ...*

En la protección de las empresas del sector de la energía están apareciendo herramientas que facilitan la realización de auditorías de seguridad a los dispositivos, redes y sistemas en red de las empresas. Un ejemplo es [3], que además propone un sistema para almacenar los resultados de las auditorías en una *blockchain* permisionada, que ejecuta de forma distribuida en los nodos de distintas organizaciones. El sistema de permisos facilita cierta compartición de información entre las empresas, de manera que puedan mejorar la respuesta en caso de detección de un incidente de seguridad, pues se podría conocer qué pruebas se habrían efectuado antes del incidente y con qué resultados. Esto facilita identificar el punto del sistema por el que se ha producido el ciberataque, y ayuda a otras organizaciones a protegerse de ataques similares.

Con la motivación de facilitar la transparencia y garantizar la protección de potenciales denuncias, estamos trabajando en un sistema que utilizando firmas de grupo [4] y acuerdos efímeros con clave pública, ofrezca garantías suficientes de anonimato a los potenciales denunciantes. Las firmas de grupo ofrecen primitivas para que una autoridad de grupo registre nuevos usuarios como miembros del grupo. A cada miembro se les da una identidad de grupo, de forma que puedan emitir firmas que posteriormente se puedan verificar como firmas válidas del grupo, pero sin poder saber cuál de las identidades del grupo ha emitido la firma. En algunos esquemas existe la posibilidad de que la autoridad del grupo desvele cuál de los miembros del grupo ha emitido la firma. Los acuerdos efímeros de clave pública son una negociación de una clave efímera que se utiliza para una única interacción o sesión, en nuestro caso para cada denuncia se utiliza una de estas claves efímeras entre denunciante y destinatario de la denuncia. La negociación adicional de estas claves efímeras permite garantizar que aunque la clave pública utilizada se vea comprometida en el futuro, será necesario comprometer todas y cada una de las claves efímeras negociadas por separado, a esto también se le denomina *forward secrecy*.

En nuestro esquema los potenciales denunciantes serían personas trabajadoras de una organización, por ejemplo una empresa de generación o distribución de energía. En nuestra propuesta, las potenciales denuncias darían a conocer informes con resultados de auditorías que solo están accesibles de forma completa a trabajadores de la organización. Nuestra propuesta utiliza una *blockchain* permisionada que aloja todos los resultados de las auditorías, de forma que el informe vinculado a una denuncia potencial es una prueba totalmente auditable que no admite manipulación o fabricación de forma alguna, y que esto puede ser demostrable matemáticamente. Por esa razón, en nuestro esquema no caben las denuncias falsas que utilicen informes falsos o manipulados. Del mismo modo, la naturaleza distribuida de la *blockchain* evita la

eliminación de evidencias, algo difícilmente alcanzable sin dicha distribución.

A continuación en la Sección II se referencian y explican brevemente trabajos relacionados con el presente para exponer las diferencias con nuestra propuesta que se concreta en la Sección III. Finalmente en la Sección IV se dan unas conclusiones y los próximos pasos a seguir del presente trabajo.

## II. Estado del arte

De entre los proyectos de software abierto para facilitar denuncias anónimas, podemos citar GlobaLeaks y SecureDrop. GlobaLeaks cumple con el estándar ISO 37002 y la Directiva (UE) 2019/1937. Además, permite configurar las políticas de retención de datos del RGPD, y ofrece un sistema de fácil distribución en distintos nodos, cada uno de ellos almacenando los documentos de sus propias denuncias. SecureDrop es un sistema más centralizado, pensado para que se instale en un periódico y que el denunciante utilice un *pen drive* con una distribución Tails que enruta todo su tráfico por Tor y no utiliza el disco del ordenador ni tampoco el sistema operativo con el fin de no dejar trazas de la sesión. En ambos proyectos, periodistas y denunciantes se comunican e intercambian mensajes exclusivamente a través del sistema, de forma que ni la empresa de comunicaciones, ni otros proveedores tipo GAFA (Google Apple Facebook Amazon) puedan identificar la fuente de una denuncia.

Por defecto ambos sistemas proporcionan cifrado de servidor, y ninguno de estos sistemas permite a los denunciantes dar una prueba de su fiabilidad como fuentes de información. Para superar esta limitación, se pueden emplear sistemas de firma de grupo, de forma que se requiera probar pertenencia a una organización para entrar en el grupo. Al firmar con la clave de grupo, se está demostrando que pertenece a la organización, y por tanto que tiene acceso legal a la documentación que ahora está denunciando/revelando.

En [5] se propone una comparativa entre distintas formas para proteger las denuncias anónimas: formularios web, WikiLeaks, GlobaLeaks y SecureDrop. Finalmente el artículo propone el uso de *blockchain* y contratos inteligentes junto con IPFS (*InterPlanetary File System*), aunque no da detalles.

En [6] se propone un sistema que combina *blockchain* y firmas de grupo que proporciona auténtico anonimato al denunciante, de forma que dicha persona sea la única que decide desvelar o no su identidad. Sin embargo al utilizar un esquema de firmas en anillo con RSA, la complejidad y el tiempo de cálculo para firmar y enviar una denuncia son elevados, y también para comprobar la veracidad de una reivindicación del denunciante, especialmente si está reclamando el crédito de una denuncia, es decir, revocando su anonimato.

Uno de nuestros objetivos es conseguir mejorar dicha complejidad, para ello proponemos cambiar la revocación del anonimato basado en firmas de grupo a uno basado en acuerdos efímeros. De esta forma, en caso de que el denunciante quiera revocar su identidad podrá presentar la clave privada efímera usada para generar la clave simétrica con la que se cifró la denuncia. Además queremos que las denuncias solamente puedan ser accedidas por las entidades a las que están destinadas. Esto está totalmente en línea con el borrador de la futura ley nacional que transpondrá la Directiva (UE) 2019/1937. En dicho borrador se menciona explícitamente una figura de referencia, el responsable del sistema interno de información. El sistema que proponemos y describimos en la siguiente sección podría perfectamente encajar en el sistema interno de información al que se refiere el borrador de la ley.

Una pieza clave para garantizar el anonimato de los denunciantes es el esquema de firmas grupales PS16 [7]. Este es un esquema de tipo *Sign-Randomize-Prove* basado en firmas de tipo Camenisch-Lysyanskaya [8]. Es decir, tiene como base un esquema de firmas sobre vectores de mensajes, compatible con protocolos eficientes de conocimiento cero. Dada una firma de tipo PS16 emitida por el gestor del grupo (es decir, la credencial de miembro del grupo), estos protocolos de conocimiento cero adicionales permiten al usuario demostrar el conocimiento de dicha firma, sin desvelar la propia firma en sí misma – consiguiendo así el anonimato requerido en los esquemas de firmas grupales. El gestor de grupo mantiene una lista con los registros del proceso de unión al grupo de cada miembro. Para abrir una firma grupal, debe iterar dicha lista.

## III. Arquitectura

Nuestra propuesta está basada en una *blockchain* permisionada, concretamente HyperLedger Fabric, en adelante simplemente fabric. Fabric se utiliza para ofrecer auditabilidad a los informes de las distintas auditorías técnicas que las organizaciones van realizando a lo largo del tiempo. Cada organización mantiene su propio nodo (*Peer*) integrado en la red fabric.

En la Fig. 1 se presenta la arquitectura, dividida en dos partes, una llevada a cabo por el Denunciante, y otra por el Destinatario.

La parte del Denunciante se divide en dos fases: la primera fase de registro, Fase 1 Dn, se corresponde con la Sección III-B2, donde se obtienen las credenciales de grupo; en un sistema en producción se debería exigir el registro de un mínimo de $k$-personas de cada organización para obtener credenciales de grupo, con la finalidad de conseguir $k$-anonimato en caso de denuncia. La segunda fase de publicación de denuncias, Fase 2 Dn, se corresponde con la Sección III-C.

La parte del Destinatario, Fase 1-2 Ds, se divide a su vez en una primera fase de registro que se corresponde con la Sección III-B1, necesaria para suscribirse como receptor de denuncias anónimas, de manera que los Denunciantes tengan acceso a su certificado; y otra fase de lectura de las mismas. En un escenario ideal, todos los miembros de las organizaciones participantes estarán registrados como Destinatarios. En el resto de casos, de acuerdo con el borrador de ley, al menos el responsable del canal interno deberá estarlo.

### *III-A. Componentes*

Nuestro sistema se divide en cinco componentes principales: contrato inteligente, proveedor, verificador, denunciante y destinatario.

- **Contrato inteligente:** Software que se ejecuta en un nodo de la red fabric y ofrece una interfaz para poder interactuar con la *blockchain*. Entre las funciones a disposición de Denunciantes y Destinatarios se encuentran métodos como: *Subscribe/Unsubscribe*, permite a los

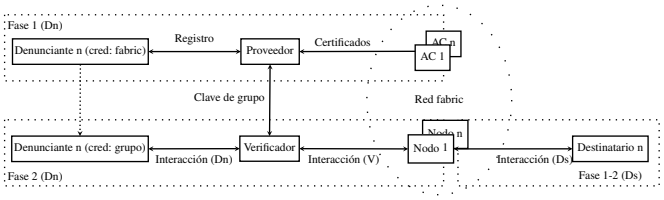



Figura 1. Interacción entre los componentes del sistema

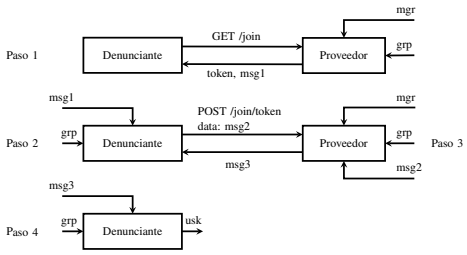



Figura 2. Proceso de obtención de credencial de grupo

Destinatarios almacenar su certificado en la *blockchain* de manera que sea accesible para los Denunciantes; *GetPublications*, permite a los Destinatarios ver la lista de denuncias anónimas publicadas; *PublishDisclosure*, permite a los Denunciantes publicar denuncias anónimas; *GetSubscribers*, permite a los Denunciantes ver la lista de Destinatarios y los certificados de estos. Además, existen funciones que permiten filtrar la información como la fecha en la que se publicaron las denuncias, la organización a la que pertenecen los Destinatarios, etc. Para una definición detallada del contrato inteligente consultar [9].

- **Proveedor (Emisor de credenciales de grupo):** Entidad independiente a la red fabric y de confianza. Posee los certificados de las autoridades certificadoras de la red fabric y confía en ellos. Se encarga de ofrecer credenciales de grupo a aquellos usuarios que pertenezcan a la red fabric, para ello los Denunciantes deberán autenticarse contra el Proveedor usando su identidad de fabric. Se comprobará si el certificado del Denunciante ha sido emitido por alguna de las autoridades certificadoras que tiene en su base de datos. En este proceso no se almacena información personal del Denunciante, únicamente un *digest* del certificado para comprobar que no posee una identidad en el grupo. Una explicación más detallada sobre el proceso de registro en el grupo se encuentra en la Sección III-B2.
- **Verificador:** Permite a los Denunciantes interactuar con la red fabric, Interacción V, mediante el uso del contrato inteligente disponible en la red fabric, siendo posible obtener la lista de Destinatarios y la publicación de denuncias. Debe ser administrada por una parte de confianza y perteneciente a la red de fabric. Las denuncias se comprobarán antes de ser publicadas: la firma de la denuncia deberá haber sido emitida por un miembro del grupo, para ello el Verificador contactará con el Proveedor que le facilitará la clave pública del grupo y poder así cerciorarse de que realmente ha sido emitida por un miembro del grupo y que es válida.
- **Denunciante:** Publica información confidencial de una manera totalmente anónima. Para que sus denuncias sean publicadas, es necesario que el Denunciante posea una identidad de grupo válida. Los detalles de publicación se explican en la Sección III-C.
- **Destinatario:** Almacena su certificado en la *blockchain* mediante el uso del contrato inteligente en la red fabric, permitiendo que así que sea accesible por los Denunciantes. Además, puede leer mediante el contrato inteligente aquellas denuncias publicadas de las cuales sea el receptor, realizando el proceso inverso al Denunciante. Debe poseer una identidad de fabric.

### *III-B. Registro*

*III-B1. Destinatario:* Cualquier miembro de fabric podrá registrarse como Destinatario, Fase 1 Ds, pudiendo así recibir denuncias anónimas usando el contrato inteligente instalado en los nodos de la red fabric. El proceso de registro consiste en el almacenamiento del certificado del Destinatario en la *blockchain*, permitiendo a los Denunciantes acceder a dichos certificados y usarlos para publicar denuncias.

*III-B2. Denunciante:* El protocolo de registro del Denunciante, Fase 1 Dn, utiliza como base fundamental el esquema PS16 implementado en la biblioteca `libgroupsig` [10]. Consta de cuatro pasos, descritos en la Fig. 2. El registro requiere el intercambio de tres mensajes entre Denunciante y Proveedor, encargado de emitir las credenciales de grupo. Es necesario mencionar que el proceso de registro sólo se realiza entre Denunciante y Proveedor, no existe interacción por parte del Proveedor con otras partes ni con la red fabric, excepto en una primera instancia cuando obtiene los certificados raíz de las autoridades certificadoras.

El registro comienza con una conexión TLS en la que ambas partes se identifican mutuamente con su certificado: el Proveedor con un certificado emitido por una autoridad certificadora de confianza y el Denunciante con un certificado emitido por alguna autoridad certificadora de la red fabric.

La Fig. 2 está compuesta por cuatro pasos:

- **Paso 1**: Un Denunciante interesado en empezar el registro envía una petición HTTP GET contra el *endpoint* del Proveedor, autenticándose usando su certificado de fabric. El Proveedor genera un *token*, basado en UUIDv4 acorde a la especificación RFC4122, asignado al *digest* del certificado. Además, genera un número aleatorio de uso único, a modo de desafío, usando la clave de grupo (*grp*) para obtener el rango apropiado. Se envía de vuelta al Denunciante el *token* y el desafío (*msg1*).
- **Paso 2**: El Denunciante responde al desafío, generando una clave privada de miembro si no lo había hecho antes, y usándola para computar una firma de conocimiento cero [11] de dicha clave privada sobre el desafío recibido del Proveedor. Enviará una petición HTTP POST contra el *endpoint* y el *token* recibido en el paso anterior, incluyendo la respuesta al desafío (*msg2*).
- **Paso 3**: El Proveedor verificará que la firma de conocimiento cero recibida es una respuesta válida al desafío. En caso afirmativo, reutilizará la estructura interna de la firma de conocimiento cero para producir la clave de

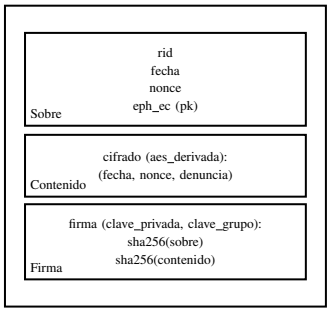



Figura 3. Esquema de mensaje enviado al Verificador

miembro para el Denunciante, la cual consiste esencialmente en una firma PS16 emitida de forma ciega sobre la clave privada del Denunciante, utilizando su clave de administrador de grupo (*mgr*). Se envía dicha firma al Denunciante (*msg3*).

- **Paso 4**: El Denunciante verifica que la clave de miembro recibida es una firma PS16 válida, asociada a la clave pública del grupo. Desde este momento, el Denunciante podrá utilizar su clave de miembro (*usk*) para generar firmas anónimas en nombre del grupo.

### *III-C. Publicación*

El proceso de publicación de una denuncia, Fase 2 Dn, empezará con la obtención de los identificadores de los Destinatarios. Para ello, el Denunciante enviará una petición HTTP GET al Verificador que responderá con la lista de identificadores de los Destinatarios interesados en recibir denuncias anónimas. El Denunciante enviará una petición HTTP POST, usando el identificador del Destinatario en el cuerpo de la petición, a la que el Verificador responderá con el certificado del Destinatario. Acto seguido el Denunciante preparará un par de claves efímeras (*eph_ec*), usando criptografía de curvas elípticas, en concreto la curva SECP256R1, cuyo propósito será derivar la clave simétrica para cifrar y descifrar la denuncia.

La Fig. 3 presenta el esquema del mensaje que se envía al Verificador a la hora de publicar una denuncia. El mensaje se compone de tres partes: sobre, contenido y firma.

- **Sobre:** Incluye el identificador del Destinatario (*rid*), la fecha de publicación de la denuncia, un *nonce* y la clave pública del par de claves efímeras generadas de antemano, todo esto codificado en BASE64, tal y como se indica en Ec. 1:

$$Base64(rid, fecha, nonce, eph\_ec_{pk}) \quad (1)$$

- **Contenido:** Incluye la fecha, el *nonce* y el texto plano de la denuncia. El contenido se cifra mediante la implementación Fernet [12], la cual usa AES con una clave de 128-bits en modo CBC. La clave para cifrar y descifrar el contenido se obtendrá mediante Diffie-Hellman aplicado a curvas elípticas y el uso de una función de derivación SHA256. En el caso de cifrado, representado en Ec. 2, la clave se obtiene con la clave privada efímera y la clave pública del Destinatario.

$$K_{enc} = kdf(SHA_{256}, eph\_ec_{sk} \times rec_{pk}) \quad (2)$$

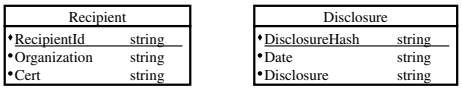



Figura 4. Esquema de la información en la *blockchain*

Para el descifrado, representado en Ec. 3, la clave se obtiene de la clave privada del Destinatario y la clave pública efímera presente en el sobre.

$$K_{dec} = kdf(SHA_{256}, eph\_ec_{pk} \times rec_{sk}) \quad (3)$$

El contenido se expone en Ec. 4, donde *E* es el algoritmo de cifrado Fernet y *K* la clave simétrica compartida de cifrado.

$$E(K_{enc}, fecha, nonce, denuncia), \quad (4)$$

- **Firma:** Incluye la firma de grupo sobre el *hash* del Sobre y el *hash* del Contenido, tal y como se indica en Ec. 5.

$$sign(SHA_{256}(sobre), SHA_{256}(contenido)). \quad (5)$$

Si la firma ha sido emitida por un miembro del grupo y es válida, el Verificador ejecutará la función *PublishDisclosure* presente en el contrato inteligente desplegado en la red fabric, almacenándose la denuncia en la *blockchain*, generándose una transacción con las credenciales del Verificador.

### *III-D. Blockchain*

La Fig. 4 representa el formato en el que se encuentran almacenados los Destinatarios y las denuncias en la *blockchain*.

Los Destinatarios están definidos por tres elementos:

- **RecipientId:** Identificador único siguiendo la estructura $x509 :: asunto :: emisor$, donde *asunto* y *emisor* se corresponden con sus homónimos en el certificado del Destinatario siguiendo el formato RFC4514. El identificador se almacena codificado en BASE64.
- **Organization:** Identificador del MSP (*Membership Service Provider*) que gestiona la identidad del Destinatario.
- **Cert:** Certificado del Destinatario codificado en BASE64.

Las denuncias se definen mediante tres elementos:

- **DisclosureHash:** *Hash* de la denuncia, usando SHA256.
- **Date:** Fecha en la que se publica la denuncia, bajo el formato $año-mes-día$.
- **Disclosure:** Compuesto por sobre y contenido codificados en BASE64.

## IV. Conclusiones

En este artículo proponemos un sistema que proporcione anonimato a las denuncias en un sistema abierto y auditable, basado en Hyperledger Fabric, una *blockchain* permisionada. Nuestro sistema proporciona confidencialidad a las denuncias, de forma que únicamente los Destinatarios sean capaces de descifrarlas. Aunque se podría dar soporte a denuncias públicas fácilmente.

Un segundo objetivo es que únicamente el Denunciante sea capaz de revocar su anonimato, aunque esto no se puede garantizar criptográficamente con PS16, de forma que asumimos que el Proveedor de identidades de grupo no

almacenará ninguna información de los miembros cuando estos se registren. En trabajos futuros propondremos usar el esquema de firmas grupales DL21 [13], disponible también en `libgroupsig`. Este esquema, de tipo User-Controlled Linkability (UCL), ofrece un control total al Denunciante sobre cuáles de sus firmas grupales van a ser trazables, dado que no existe una autoridad de apertura de firmas (*Opener*). Al mismo tiempo, permite al Denunciante reclamar *a posteriori* la autoría de una denuncia. Un caso de uso sería, por ejemplo, un Denunciante que crea una firma no enlazable a todas las firmas que haya creado hasta ese momento, y un mes después, reclamar que fue él quien la generó. Además, nuestro proceso de revocación del anonimato ofrece una complejidad menor que la de otros trabajos similares como [6].

El uso de curvas elípticas nos aporta una serie de beneficios sobre RSA como pueden ser: un menor tamaño de clave con seguridad equivalente a una de mayor tamaño en RSA, lo que reduce drásticamente la carga de procesamiento, permitiendo así generar claves efímeras robustas para cada denuncia sin suponer una desventaja notable; o compatibilidad directa con las identidades de Hyperledger Fabric al usar el mismo tipo de claves.

El uso de firmas de grupos garantiza el anonimato de los Denunciantes dentro del conjunto de anonimato definido por el grupo al que pertenecen. El único momento en el que el Denunciante necesita mostrar su identidad, es al unirse al grupo. Cualquier acción posterior es completamente no enlazable con el proceso de registro (propiedad de anonimato de las firmas grupales) pero, al mismo tiempo, hay máxima garantía de que únicamente quien haya pasado por ese proceso de registro puede generar firmas grupales válidas (propiedad de trazabilidad). Es preciso tener en cuenta que la variedad de implementaciones de firmas grupales permite definir un conjunto muy amplio de procedimientos de trazabilidad y enlazado [14], lo que en último término puede servir de base a la generación de marcos de gobernanza adecuados al caso concreto de las denuncias anónimas [15]. Por último, no es posible crear firmas grupales que incriminen a un miembro del grupo que no las emitió (propiedad de no inculpación, *non-frameability*), siendo el propio Denunciante el único que puede desvelar haber sido quien generó una firma grupal en concreto.

Como puntos a mejorar, se ha pensado en la supresión de los certificados de los Destinatarios y las denuncias de la *blockchain* debido a la carga que supone en el sistema a medida que aumente su uso (poco escalable). Para ello se propone la sustitución de esos elementos por una referencia a un sistema de almacenamiento externo (por ejemplo IPFS o similar) y un *hash* al elemento referenciado para garantizar la inmutabilidad. Otro posible punto a mejorar sería distribuir el proceso de registro de credenciales de grupo, de forma que no haya una única autoridad, sino varias. Esto reduciría la confianza necesaria en el Proveedor, aunque `libgroupsig` no ofrece este soporte por lo que implicaría cambios mayores que los anteriores.

Tenemos un primer prototipo listo y nuestros próximos pasos irán encaminados a mostrar las prestaciones del sistema y hacer una comparativa cuantitativa con otras propuestas de la literatura. Entonces liberaremos el código de este prototipo, al igual que hicimos con AUTOAUDITOR [9].

## AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto COMPROMISE (PID2020-113795RB-C32) financiado por MCIN/AEI/10.13039/501100011033, por el proyecto SPIRS No. 952622 del H2020 de la Comisión Europea, y por la Comunidad de Madrid (España) dentro del proyecto CYNAMON (P2018/TCS-4566) con apoyo de fondos FSE y FEDER de la Unión Europea.

## REFERENCIAS

[1] S. Chica, A. Marín, D. Díaz, F. Almenares, “On the Automation of Auditing in Power Grid Companies,” en *Intelligent Environments 2020. Workshop Proc. 16th Int. Conf. Intelligent Environments*, vol. 28, 2020, pp. 331-340, doi: 10.3233/AISE200057.

[2] Unión Europea, “Directiva (UE) 2019/1937 del Parlamento Europeo y del Consejo de 23 de octubre de 2019 relativa a la protección de las personas que informen sobre infracciones del Derecho de la Unión,” en *Diario Oficial de la Unión Europea*, L. 305/17, 2019. Disponible: https://eur-lex.europa.eu/eli/dir/2019/1937/oj

[3] A. Marín, S. Chica, D. Arroyo, F. Almenares, D. Díaz, “Security information sharing in smart grids: Persisting security audits to the blockchain,” en *Electronics*, vol. 9, no. 11, p. 1865, 2020, doi: 10.3390/electronics9111865.

[4] D. Chaum, E. van Heyst, “Group signatures,” en D. W. Davies, editor, *Proceedings of Eurocrypt 1991*, vol. 547 of LNCS, pp. 257–65, Springer-Verlag, 1991.

[5] A. Habbabeh, P. Asprion, B. Schneider, “Mitigating the Risks of Whistleblowing-an Approach Using Distributed System Technologies,” en *PoEM Workshops*, 2020.

[6] A.E.B. Tomaz, J.C.d. Nascimento, J.N.d. Souza, “Blockchain-based whistleblowing service to solve the problem of journalistic conflict of interest,” en *Annals of Telecommunications 77.1*, pp. 101-118, 2022, doi: 10.1007/s12243-021-00860-0.

[7] D. Pointcheval, O. Sanders, “Short Randomizable Signatures,” en *Cryptology ePrint Archive, Report 2015/525*.

[8] J. Camenisch, A. Lysyanskaya, “A Signature Scheme with Efficient Protocols,” en *Security in Communication Networks*, 2002, doi: 10.1007/3-540-36413-7_20.

[9] S. Chica, “autoauditor: Semiautomatic vulnerabilities auditor using docker containers,” 2022. Disponible: https://gitlab.gast.it.uc3m.es/schica/autoauditor

[10] J. Díaz, D. Arroyo, F. Rodríguez, “`libgroupsig`: An extensible C library for group signatures,” en *Cryptology ePrint Archive, Report 2015/1146*.

[11] M. Chase, A. Lysyanskaya, “On Signatures of Knowledge,” en *CRYPTO*, 2006, doi: 10.1007/11818175_5.

[12] K. Parnell, H. Giménez, K. McDermot, “fernet: Delicious HMAC Digest(if) authentication and AES-128-CBC encryption ,” 2022. Disponible: https://github.com/fernet/spec

[13] J. Diaz, A. Lehman, “Group Signatures with User-Controlled and Sequential Linkability,” en *Public Key Cryptography*, 2021, doi: 10.1007/978-3-030-75245-3_14.

[14] M.N.S. Perera, T. Nakamura, M. Hashimoto, H. Yokoyama, C-M. Chengand, K. Sakurai, “A Survey on Group Signatures and Ring Signatures: Traceability vs. Anonymity,” en *Cryptography*, 2022, doi: 10.3390/cryptography6010003

[15] J. Díaz, D. Arroyo, F.B. Rodríguez, “New X.509-based mechanisms for fair anonymity management,” en *Computers & Security*, 2014, doi: 10.1016/j.cose.2014.06.009

# Un estudio del DNIe y de su infraestructura

Javier Correa-Marichal
Universidad de La Laguna
Tenerife, Spain
jcorreamarichal@gmail.com

Pino Caballero-Gil
Universidad de La Laguna
Tenerife, Spain
pcaballe@ull.edu.es

Carlos Rosa-Remedios
Gobierno de Canarias
Tenerife, Spain
crosa@gsccanarias.com

Rames Sarwat-Shaker
Telefónica Tech
Madrid, Spain
rames.sarwatshaker@telefonica.com

***Resumen*—El Documento Nacional de Identidad o DNI es una pieza de documentación fundamental para la identificación de los ciudadanos españoles. Su importancia se vio impulsada en 2006 con la incorporación de un chip para la autenticación de usuarios dentro de los servicios administrativos telemáticos, pasando a denominarse DNI electrónico o simplemente DNIe. En ese circuito integrado se almacena información de naturaleza sensible del portador, como sus datos personales y biométricos, junto a certificados de firma y autenticación. Algunas de las funcionalidades del DNIe en su versión actual a fecha de redacción del presente trabajo están implementadas desde hace años en el DNI 3.0 y por tanto ya han sido ampliamente estudiadas. Este trabajo aporta un estudio recopilatorio teórico y práctico de algunos de los mecanismos de seguridad incorporados en el actual DNIe y en algunas de las aplicaciones que requieren su uso; empleando únicamente dispositivos móviles y lectores de tarjeta genéricos. Se trata, por tanto, de un análisis exploratorio realizado con la intención de confirmar, con herramientas básicas, el nivel de robustez del token de seguridad más importante del país.**

***Index Terms*—DNIe, eMRTD, NFC, API hooking**

## I. Introducción

El Documento Nacional de Identidad es un documento emitido por el Ministerio del Interior, que permite acreditar la identidad y datos personales del titular. Dado que su obtención es obligatoria para todos los españoles mayores de 14 años, millones de DNI son expedidos anualmente en España [1]. Aprovechando su popularización, y con el fin de impulsar la digitalización de los servicios telemáticos ofrecidos por las administraciones públicas, en 2006 se lanzó una nueva versión del DNI con la incorporación de un chip que incorpora diversas funcionalidades relacionadas con la identidad del portador, denominado DNI electrónico o DNIe.

La seguridad tanto del documento físico como de sus componentes electrónicos y software asociado es mejorada en cada nueva revisión. Cada actualización del DNIe, antes de ser certificada por el Organismo de Certificación del Centro Criptológico Nacional, pasa un proceso de evaluación desarrollado por la Fábrica Nacional de Moneda y Timbre – Real Casa de la Moneda a solicitud de la Dirección General de la Policía, y llevado a cabo por un laboratorio acreditado que pasa auditorías SOG-IS. Dicha evaluación se realiza siguiendo la metodología Common Criteria (ISO/IEC 15408). Concretamente, el software del DNIe ha sido certificado con nivel de evaluación EAL4+ y EAL4 AVA_VAN.5, y los chips han sido certificados como Dispositivo Seguro de Creación de Firma, conforme a los estándares europeos [2]. A pesar de la tranquilidad que brindan estas certificaciones, a veces pasan desapercibidos errores de diseño o de implementación en productos certificados ya implementados en dispositivos desplegados. Por eso, siempre es necesario considerar la seguridad como un proceso y no como un estado.

En el lanzamiento del DNIe 3.0 en 2015 se incluyó una interfaz de uso a través de NFC, que permite usar el DNIe directamente a través de dispositivos móviles que incorporen esa tecnología, en un esfuerzo por popularizar su uso [3].

A fecha de redacción de este trabajo, la última revisión del DNI habría sido lanzada en agosto de 2021, como se recoge en el anuncio oficial "NUEVO DNIe 4.0 – FORMATO EUROPEO" publicado en el portal del DNI electrónico [4]. Una de las características más notables de esa versión es el diseño y la funcionalidad, buscando homogeneizar los documentos de identidad de los países de la Unión Europea para que su uso pueda ser estandarizado y homologado de acuerdo al reglamento eIDAS, de identificación digital en Europa. Además, según se anuncia en la web de la Policía Nacional [5], la versión actual del DNIe incorpora nuevas medidas de seguridad, tanto visibles como invisibles.

El presente estudio es una exploración de la implementación de las funcionalidades ofrecidas y mecanismos de seguridad asociados contenidos en la última versión del DNIe, realizado sobre un documento emitido a finales de 2021. Concretamente, esta línea de investigación pretende confirmar la resistencia del DNIe ante vectores de ataques frecuentes en pasaportes electrónicos y dispositivos NFC [6]. El análisis realizado ha sido presentado al organismo responsable del DNIe, como parte de la práctica habitual de investigación científica realizada por los autores.

El resto del artículo se organiza como sigue. La sección II describe la interfaz del DNIe para su interacción a través de lectores con o sin contacto. Las secciones III y IV introducen, respectivamente, la estructura lógica de los datos almacenados en el DNIe y los mecanismos de seguridad desarrollados para proteger su integridad. La sección V recoge algunos detalles de la investigación realizada sobre la versión actual del DNIe y varias aplicaciones que requieren su uso. Finalmente, en la sección VI se exponen las conclusiones de este trabajo.

## II. Interfaz de uso

El DNI analizado se encuentra equipado con un chip SLE78CLFX408AP del fabricante Infineon Technologies [7], donde se implementa la funcionalidad digital del documento. Este sería el mismo chip del DNI 3.0 afectado por la vulnerabilidad ROCA descubierta en 2017 [8] en la generación de claves RSA utilizadas por la biblioteca de software. Para solventar los efectos de dicha vulnerabilidad fue necesario en ese momento revocar un gran número de certificados.

Ese chip es accesible a través de una interfaz dual que soporta acceso a través de una toma de contacto estandarizada como ISO/IEC 7816 [9] y de forma inalámbrica por medio de una antena NFC, siguiendo el conjunto de protocolos definido

en el estándar ISO/IEC 14443 [10]. Para la realización de este estudio se ha utilizado preferentemente la interfaz NFC por ser compatible con dispositivos Android con esta tecnología. El estándar ISO/IEC 14443 se apoya a su vez en el protocolo a nivel de capa de aplicación definida en ISO/IEC 7816-4, donde la comunicación se lleva a cabo a través de pares de comandos y respuestas denominados *Application Protocol Data Units* (APDU). En este estándar, se define un sistema de archivos y los comandos necesarios para realizar consultas. Pueden encontrarse varias aplicaciones alojadas en el chip, separadas en distintos ficheros dedicados (*Dedicated Files* o DF) que cuelgan de la raíz del sistema de archivos, señalizadas por el fichero maestro (*Master File* o MF). En estos se almacena una colección de ficheros elementales (*Elementary Files* o EF), donde se guardan los datos del chip.

Para documentos de identidad europeos con una aplicación de pasaporte electrónico opcional, como es el caso del DNIe, se especifica en el estándar BSI TR-03110-4 que, para asegurar la interoperabilidad como token eIDAS, se han de implementar de forma obligatoria las aplicaciones de *eID* y de *eSign*; y, opcionalmente, la aplicación *ePassport* [11]. El chip del modelo del DNIe objeto de este estudio incluye tanto la aplicación de ePassport como la de eSign, si bien no implementa el servicio de eID.

## III. Estructura lógica de datos

El estándar ICAO 9303 establece los distintos aspectos por los que se rige un pasaporte electrónico (*electronic Machine Readable Travel Documents* o eMRTD). Con el fin de asegurar la interoperabilidad internacional de los datos almacenados en el DNIe, dicho estándar define una estructura lógica de datos (*Logical Data Structure* o LDS) para todos los pasaportes electrónicos. La información almacenada es asignada durante el proceso de creación del DNIe y no puede ser modificada a posteriori, sirviendo como mecanismo de protección de la información personal del portador contra posibles ataques de manipulación. Estos datos son accesibles a través de la aplicación *ePassport* del DNIe.

Los datos almacenados en el DNIe incluyen información personal sobre el portador como datos biométricos del mismo, junto a elementos utilizados durante la ejecución de los protocolos de seguridad establecidos para garantizar la legitimidad del documento y de la información almacenada.

Además, existen dos ficheros sin numeración, de implementación obligatoria, que contienen información sobre la propia estructura lógica de datos. El fichero *EF.COM* contiene las versiones de LDS y del estándar de codificación de caracteres Unicode utilizadas, así como una lista de los grupos de datos presentes en la aplicación. Por otra parte, el fichero *EF.SOD* contiene, para cada uno de los grupos de datos incluidos en el DNIe, sus hashes y firmas digitales realizadas por la entidad que firma el DNIe (*Document Signer* o DS) [12].

## IV. Mecanismos de seguridad

En la parte 11 del estándar ICAO 9303 se define un conjunto de medidas de seguridad para aumentar la resistencia de los pasaportes electrónicos ante los ataques más comunes contra ese tipo de documentos [13]. En la Tabla I se incluye un resumen de esos protocolos de seguridad, indicando la técnica utilizada en cada uno para mitigar cada ataque concreto. A continuación se describen con mayor detalle los protocolos que son de mayor interés en esta investigación, por estar implementados en el actual DNIe.

### *IV-A. Basic / Supplemental Access Control*

El propósito del protocolo *Basic Access Control* (BAC) es garantizar que el Sistema de Inspección (SI) ha mantenido contacto visual con el documento, protegiendo así los datos de naturaleza sensible almacenados en el DNIe. Este mecanismo de seguridad tiene un doble propósito. En primer lugar, dificulta el robo de información a través de la interfaz sin contacto de la tarjeta sin el conocimiento del portador. En segundo lugar, establece un canal seguro a través del cual poder enviar todo el tráfico posterior, protegiendo la comunicación de ataques de escucha y de corrupción de datos. Para la ejecución de este protocolo, primero es necesario que el SI obtenga las *Document Basic Access Keys* que se derivan de la zona legible por máquina (*Machine Readable Zone* o MRZ) del DNIe. En particular, se obtienen 3 campos del mismo.

Con el fin de subsanar debilidades reportadas en el protocolo BAC [14][15], se plantea el uso de *Suplemental Access Control* (SAC) como sustituto. Este mecanismo de autenticación utiliza el protocolo *Password Authenticated Connection Establishment* (PACE) para proveer una encriptación fuerte y permitir la compartición de códigos de autenticación de mensaje (*Message Authentication Codes* o MAC) para el establecimiento de un canal seguro.

### *IV-B. Autenticación Pasiva*

La autenticación pasiva (*Passive Authentication* o PA) es un mecanismo de seguridad que permite verificar la autenticidad e integridad de la información almacenada en el LDS. Durante la fase de personalización del chip en el proceso de expedición del DNIe, se calculan los hashes correspondientes a los distintos grupos de datos almacenados en el LDS y se almacenan en el fichero *EF.SOD*, junto a una firma digital realizada sobre ellos. En el proceso de inspección del documento, es labor del SI consultar el fichero *EF.SOD* y validar los hashes para cada uno de esos grupos, y verificar que el certificado del *Document Signer* utilizado está firmado, a su vez, por una *Country Signing Certification Authority* reconocida y válida.

### *IV-C. Extended Access Control*

El manejo de datos biométricos almacenados en el DNIe, como la información de la huella dactilar del portador, se considera más sensible que el de otros datos almacenados en el documento. Por eso existe un procedimiento más restrictivo para su acceso. Este protocolo, denominado *Extended Access Control* (EAC), se encuentra especificado en el documento BSI TR-03110 para pasaportes y tarjetas de identidad europeas [16]. Consta de tres partes: autenticación pasiva, autenticación del chip (*Chip Authentication* o CA), y autenticación del terminal (*Terminal Authentication* o TA).

## V. Evaluación de seguridad

Se introduce a continuación la investigación realizada sobre el DNIe y algunas de las aplicaciones que requieren de su uso.

Tabla I
ALGORITMOS DE SEGURIDAD IMPLANTADOS EN DOCUMENTOS EMRTD

| Protocolo | Abreviatura | Técnica | Ataque |
|---|---|---|---|
| Basic Access Control | BAC | Autenticación y canal seguro | Robo de información |
| Supplemental Access Control | SAC (PACE) | Autenticación y canal seguro | Robo de información |
| Passive Authentication | PA | Firma digital | Falsificación, manipulación de datos |
| Active Authentication | AA | Desafío-respuesta | Clonación |
| Chip Authentication | CA | Autenticación | Clonación |
| Terminal Authentication | TA | Autenticación mediante PKI | Robo de información sensible |

### *V-A. ePassport*

En la aplicación *ePassport* se almacenan datos personales y biométricos del portador del DNIe, estructurados dentro de la LDS mencionada, estandarizada para otros documentos eMRTD. El acceso indeseado por un atacante a esta funcionalidad, a través de la capacidad RFID habilitada en el DNIe, podría revelar información personal sobre un individuo sin su consentimiento, permitiendo el acceso potencial a datos de gran sensibilidad como son los datos biométricos del usuario. Con el fin de prevenir la lectura arbitraria de esos datos, se encuentran implementados protocolos de seguridad como BAC, PACE o EAC, detallados en la sección anterior. Debido a la importancia de estos protocolos dentro del marco de seguridad del DNIe, se decidió llevar a cabo un estudio exploratorio pormenorizado, para confirmar la robustez de su implementación en el DNIe. Para ello se utilizó el kit de desarrollo de aplicaciones Android proporcionado por el Cuerpo Nacional de Policía [17], tomando como base la implementación de la librería *DNIeDroid*. A través de la herramienta *Frida* [18], es posible modificar dinámicamente el comportamiento de los métodos definidos en esa librería y sustituirlos enteramente por funciones codificadas por el usuario, mediante una técnica denominada *API Hooking*. Aprovechando esta funcionalidad, es posible realizar un enganche a los métodos encargados de establecer una comunicación segura con el DNIe, permitiendo analizar e incluso modificar los paquetes APDU enviados entre el terminal y el documento a tiempo real.

A través de la malformación de paquetes enviados entre terminal y DNIe, se estudió la posibilidad de encontrar fallos en la implementación de las aplicaciones que componen el DNIe. En concreto, se prestó especial atención a la implementación del protocolo PACE, estudiando la robustez de las operaciones más críticas realizadas en el mismo, como la generación criptográfica de números aleatorios [14], o la validación final de la clave efímera de sesión negociada. Como cabría esperar, dadas las garantías proporcionadas por las evaluaciones Common Criteria a las que es sometido el DNIe antes de su lanzamiento, no se obtuvieron resultados prácticos significativos a través de ese enfoque práctico. La metodología utilizada y el código desarrollado para este estudio puede ser consultado en [19] y en [20].

En otra línea de investigación, se analizó la viabilidad de explotar vulnerabilidades conocidas que afectan a dispositivos NFC. En este sentido, se confirmó que la mayoría de los ataques descritos en la literatura, como la escucha a escondidas o la manipulación de datos, se encuentran mitigadas en el DNIe a través de la implementación de las medidas de seguridad descritas en la anterior sección. Además se analizaron las contramedidas incluidas para varios ataques conocidos como, por ejemplo, los ataques de retransmisión o de *relay* en dispositivos NFC.

Los ataques de *relay*, descritos en [21] bajo el sobrenombre de "fraude de la mafia" y en [22] como "ataque de agujero de gusano", siguen la técnica *Man-in-the-Middle* y consisten en la extensión de la comunicación entre dos participantes, utilizando dos dispositivos para reenviar los paquetes APDU de comando y respuesta entre tarjeta y lector. Puesto que estos paquetes pueden ser enviados a través de Internet, ambos dispositivos podrían estar a gran distancia, incumpliéndose así la restricción de distancia teórica máxima para la lectura de tarjetas NFC (ver Figura 1).

Figura 1. Representación gráfica de un ataque de *relay*

Las medidas de seguridad comentadas en la sección anterior permiten prevenir ataques de escucha y de corrupción de datos. Sin embargo, como es habitual en tarjetas que incorporan una interfaz NFC, es posible realizar un ataque de *relay* entre un lector y una tarjeta. En la investigación realizada en el presente estudio, se recoge la ejecución de este ataque utilizando la aplicación *NFCGate* [23], siendo esta la primera demostración documentada de la ejecución de este ataque en el caso específico del DNIe, según la literatura consultada.

Para la realización de este ataque utilizando las herramientas incluidas en el kit de investigación, es necesario disponer de dos móviles Android con versión igual o superior a la 4.4 y con la aplicación de *NFCGate* instalada, de los cuales, uno ha de encontrarse *rooteado* e incluir el módulo de *EdXposed* para poder acceder plenamente a las capacidades NFC del dispositivo. Es además necesario que un tercer equipo ejerza de servidor, reenviando los paquetes de comando y respuesta entre ambos terminales a través de Internet.

A diferencia de lo que sucede con otras tarjetas que incorporan tecnología NFC, como por ejemplo los ataques de carterismo en tarjetas de crédito sin contacto [24], cabe resaltar que las medidas de control de acceso discutidas en la sección anterior impiden que un atacante pueda acceder a los datos y certificados almacenados en el DNIe sin disponer de las contraseñas necesarias, como el código CAN o PIN. De hecho, para la ejecución de un ataque de *relay* significativo contra el DNIe, sería necesario que el adversario introdujese de antemano la contraseña correcta en el lector legítimo, dificultando enormemente la complejidad de explotación en un escenario beneficioso para un actor maligno.

En todo caso, dado que en los ataques de *relay* contra el DNIe intervienen mensajes cifrados, una posible mejora sería mediante métodos criptográficos que permitan detectar si se está realizando un ataque de *relay*. Un ejemplo de solución propuesta en la bibliografía son los protocolos de autenticación basados en acotamiento de la distancia (*distance-bounding*), los cuales utilizan una medición del tiempo de ida y vuelta de los paquetes enviados para estimar la separación existente entre el lector y la tarjeta [25].

### *V-B. eSign*

El acceso a la aplicación *eSign* es similar al de *ePassport*, pues es necesario llevar a cabo una ejecución del protocolo PACE utilizando el PIN asociado al DNIe como secreto compartido entre la tarjeta y el lector. Una vez este protocolo se ha ejecutado con éxito, se establece un canal seguro y se permite el acceso a la funcionalidad de firma. Las operaciones que utilizan estos certificados son ejecutadas siempre en el interior del DNIe, asegurando que las claves privadas nunca salen de la memoria del documento, de forma que se puede considerar un dispositivo seguro de creación de firma.

Dada la escasez de posibles vectores de ataques contra esta funcionalidad, se ha optado por analizar la implementación software oficial para la interacción de un ordenador con el documento usando el cliente @Firma [26]. @Firma es una aplicación de firma electrónica que se ejecuta en el PC del usuario y cuya función es ofrecer un sistema sencillo para la firma de documentación. Se planteó el estudio de un escenario en el que un atacante fuera capaz de capturar el PIN del DNIe mediante la interacción de un usuario legítimo con la aplicación @Firma y su posterior uso para la firma de documentos arbitrarios, forzando al usuario a comprometerse legalmente con contratos elegidos por el atacante a través de un malware ejecutándose en el PC del usuario.

En la investigación realizada se optó por explorar la viabilidad de este ataque, modificando lo menos posible el cliente de @Firma. Como trabajo futuro se podría plantear la introducción de cambios en el código fuente de esta aplicación [27] para añadir una rutina maliciosa que se encargara de reemplazar el documento a firmar por el usuario sin su consentimiento. Sin embargo, esta aproximación requeriría de permisos administrativos en el equipo para poder reemplazar el binario del aplicativo, implicando por tanto una mayor complejidad de explotación del ataque.

El enfoque seguido en este estudio práctico se ha basado principalmente en el uso del depurador JDB de Java incluido en las herramientas del *Java Development Kit*. Con ese depurador es posible conectarse a un proceso en ejecución para, mediante puntos de ruptura ubicados en funciones estratégicas, observar o modificar las variables locales almacenadas en memoria, pudiendo así usar un PIN introducido por el usuario para la posterior iniciación de una firma no autorizada. También sería posible modificar dinámicamente el fichero a firmar usando el mismo método, aunque en este caso las herramientas proporcionadas en el entorno de JDB son algo limitadas para manejar estructuras complejas de datos.

En esta investigación se ha desarrollado como prueba de concepto un script capaz de recuperar el PIN introducido por un usuario en la aplicación de @Firma, sin necesidad de modificar el binario original dentro de un entorno Linux. Además se ha estudiado la viabilidad de realizar un ataque de firma en segundo plano de documentos escogidos por el atacante, mediante las modificaciones descritas. El código desarrollado para la realización de este ataque puede ser consultado en un repositorio público de GitHub recogido en [28].

Cabe destacar que el ataque descrito vulnera la aplicación de @Firma como parte de la infraestructura del DNIe, lo que implica que el DNIe es un afectado colateral como también lo es cualquier certificado digital estándar que pueda ser utilizado a través de esa aplicación cliente. De hecho, este enfoque constituye una línea de investigación en desarrollo sobre las fortalezas y debilidades de la aplicación @Firma y su rol en ataques de firma o de exfiltración de documentos arbitrarios.

## VI. Conclusiones

En este trabajo se ha presentado un estudio exploratorio de la seguridad implementada en el DNIe en su versión actual, así como de algunas de las aplicaciones que requieren su uso. La línea de investigación seguida ha sido eminentemente práctica, con el objeto de comprobar la robustez de la tecnología y el software que rodean al DNIe.

En general se ha confirmado el alto nivel de seguridad del DNIe. En particular, la inclusión de una interfaz NFC desde la versión 3.0, aunque ofrece una mejor experiencia de uso, expone al DNIe a la posible realización de ataques de *relay*. En consecuencia, dicho tipo de ataques han sido especialmente estudiados en este trabajo, concluyéndose que en general en el DNIe han sido compensados con las medidas de seguridad requeridas para el acceso a la información almacenada. Por otra parte, del estudio de varias aplicaciones que requieren del uso del DNIe se ha concluido que el cliente @Firma podría llegar a suponer un punto débil dentro de la cadena de confianza si no se garantiza la imposibilidad de captura del PIN del usuario. Desde nuestro punto de vista, esta cuestión podría llegar a constituir una potencial amenaza si un malware logra tener acceso al documento.

En consecuencia, como conclusión se puede afirmar que, si bien en general a fecha de hoy el DNIe puede considerarse una plataforma segura sin amenazas activas contra su seguridad, es necesario tener en cuenta el gran impacto que tendría la posible confección y distribución de un malware que se aprovechara de debilidades de software para comprometer a víctimas en acuerdos legales sin su consentimiento. Por tanto, si bien la infraestructura hardware y software del DNIe ha demostrado ser suficientemente sólida frente a los ataques más directos, es importante mantener permanentemente auditada la seguridad de las aplicaciones asociadas, puesto que podrían convertirse en un punto de entrada clave para atacantes que busquen realizar un mal uso de las funcionalidades del DNIe.

## Agradecimientos

Esta investigación ha sido posible gracias a la Cátedra Institucional de Ciberseguridad financiada por Binter. Asimismo los autores agradecen el feedback recibido de la Fábrica Nacional de Moneda y Timbre-Real Casa de la Moneda.

## Referencias

[1] Anuario estadístico del Ministerio del Interior 2020. 2021, p. 547. [Online]. Available: http://www.interior.gob.es/web/archivos-y-documentacion/anuario-estadistico-de-2020. [Accessed: 02-Jun-2022].

[2] Resolución 1A0/38016/2018, de 15 de junio, del Centro Criptológico Nacional, por la que se certifica la seguridad del producto DNIe-DSCF (dispositivo seguro de creación de firma), versión 3.0. 2018.
[3] A. Pascual, "El DNI electrónico ha muerto: ¡larga vida al DNI 3.0!", El Confidencial, 2 Oct. 2013.
[4] NUEVO DNIe 4.0 – FORMATO EUROPEO. Policía Nacional, 2021. [Online]. Available: https://www.dnielectronico.es/PDFs/DNI_4.0.pdf. [Accessed: 02-Jun-2022].
[5] La Policía Nacional finaliza la implantación del DNI Europeo, la nueva versión del DNI electrónico, 2021. [Online]. Available: https://www.policia.es/_es/comunicacion_prensa_detalle.php?ID=9462#. [Accessed: 02-Jun-2022].
[6] D. Arroyo Guardeño, V. Gayoso Martínez, L. Hernández Encinas and A. Martín Muñoz. "Using smart cards for authenticating in public services: a comparative study", Computational Intelligence in Security for Information Systems Conference, pp. 437-446, Springer, 2015. DOI: 10.1007/978-3-319-19713-5_37
[7] "Descripción del Chip DNIe 3.0", Portal del DNI Electrónico. [Online]. Available: https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_1078. [Accessed: 02-Jun-2022].
[8] "Desactivados los DNI electrónicos por fallos en su firma digital", INCIBE-CERT Bitácora Ciberseguridad. [Online]. Available: https://www.incibe-cert.es/alerta-temprana/bitacora-ciberseguridad/desactivados-los-dni-electronicos-fallos-su-firma-digital. [Accessed: 02-Jun-2022].
[9] "ISO/IEC 7816-2:2007", ISO, 2007. [Online]. Available: https://www.iso.org/standard/45989.html. [Accessed: 02-Jun-2022].
[10] "ISO/IEC 14443-3:2011", ISO, 2011. [Online]. Available: https://www.iso.org/standard/50942.html. [Accessed: 02-Jun-2022].
[11] TR-03110 Part 4: Applications and Document Profiles, 2nd ed. Federal Office for Information Security, p. 18, 2016.
[12] Doc 9303 Part 10: Logical Data Structure (LDS) for Storage of Biometrics and Other Data in the Contactless Integrated Circuit (IC), 8th ed. ICAO, 2021.
[13] Doc 9303. Machine Readable Travel Documents. Part 11: Security Mechanisms for MRTDs, 8th ed. ICAO, 2021.
[14] R. Rodríguez and J. Garcia-Escartin, "Security assessment of the Spanish contactless identity card", IET Information Security, vol. 11, no. 6, pp. 386-393, 2017. DOI: 10.1049/iet-ifs.2017.0299
[15] G. Avoine, A. Beaujeant, J. Hernandez-Castro, L. Demay and P. Teuwen, "A Survey of Security and Privacy Issues in ePassports Protocols", ACM Computing Surveys, vol. 48, no 3, p. 1-37, 2016. DOI: 10.1145/2825026
[16] TR-03110 Part 2: Protocols for electronic IDentification, Authentication and trust Services (eIDAS), 2nd ed. Federal Office for Information Security, 2016.
[17] "Código Fuente de las aplicaciones", Portal del DNI Electrónico. [Online]. Available: https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_036&id_menu=21. [Accessed: 02-Jun-2022].
[18] "Frida. A world-class dynamic instrumentation framework", Frida. [Online]. Available: https://frida.re/. [Accessed: 12-Apr-2022].
[19] J. Correa, "alu0101233598/nfc-card-security", GitHub, 2022. [Online]. Available: https://github.com/alu0101233598/nfc-card-security. [Accessed: 13-Jul-2022].
[20] J. Correa, "alu0101233598/JMRTD-bruteforce-demo", GitHub, 2022. [Online]. Available: https://github.com/alu0101233598/JMRTD-bruteforce-demo. [Accessed: 13-Jul-2022].
[21] Y. Desmedt, C. Goutier and S. Bengio. "Special uses and abuses of the Fiat-Shamir passport protocol", Advances in Cryptology, Proc. of Crypto' 87 (Lecture Notes in Computer Science 293), pp. 21–39. Springer-Verlag, 1988. DOI: 10.1007/3-540-48184-2_3
[22] Y. Hu, A. Perrig and D. B. Johnson, "Wormhole Attacks in Wireless Networks", IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 370-380, 2006. DOI: 10.1109/JSAC.2005.861394
[23] "nfcgate/nfcgate: An NFC research toolkit application for Android", GitHub, 2022. [Online]. Available: https://github.com/nfcgate/nfcgate. [Accessed: 02-Jun-2022].
[24] A. Lotfi, "Could your contactless bank card be vulnerable to virtual pickpocketing?", The Conversation, 2022. [Online]. Available: https://theconversation.com/could-your-contactless-bank-card-be-vulnerable-to-virtual-pickpocketing-55264. [Accessed: 02-Jun-2022].
[25] S. Drimer and S. J. Murdoch, "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks", USENIX security symposium, Vol. 312, 2007. Available: https://static.usenix.org/events/sec07/tech/drimer/drimer_html/. [Accessed: 13-Jul-2022]
[26] "PAe - Cliente@Firma", Portal Administración electrónica, 2022. [Online]. Available: https://administracionelectronica.gob.es/pae_Home. [Accessed: 02-Jun-2022].
[27] "GitHub - ctt-gob-es/clienteafirma: Cliente @firma", GitHub, 2022. [Online]. Available: https://github.com/ctt-gob-es/clienteafirma. [Accessed: 02-Jun-2022].
[28] J. Correa, "alu0101233598/AutoStealer: AutoFirma unauthorized sign demonstration", GitHub, 2022. [Online]. Available: https://github.com/alu0101233598/AutoStealer. [Accessed: 13-Jul-2022].

# Reconocimiento Facial e Identificación de Somnolencia en Conductores

Alba Cruz-Torres
Universidad de La Laguna
acruztor@ull.edu.es

Carlos Rosa-Remedios
Universidad de La Laguna
crosarem@ull.edu.es

Pino Caballero-Gil
Universidad de La Laguna
pcaballe@ull.edu.es

Candelaria Hernández-Goya
Universidad de La Laguna
mchgoya@ull.edu.es

***Resumen*—Los accidentes de tráfico causan continuos fallecimientos en todo el mundo. A medida que se va desarrollando nueva tecnología aplicable y se van endureciendo las sanciones, se consigue reducir progresivamente el número de víctimas mortales, pero aun así hoy en día sigue habiendo demasiados fallecidos en las carreteras. Por ese motivo, en este trabajo se ha realizado una investigación preliminar de algunas de las soluciones existentes en el mercado para aumentar la seguridad vial en las carreteras mediante el reconocimiento facial de los conductores y la detección de signos de somnolencia. A partir de dicho análisis se ha estudiado el diseño óptimo para cumplir el objetivo establecido. Este documento presenta un estudio de técnicas de *Machine Learning* y herramientas de detección de rostros aplicadas a un programa para el reconocimiento facial y la detección de somnolencia en conductores.**

***Index Terms*—seguridad vial, reconocimiento facial, identificación de somnolencia, biometría**

## I. Introducción

Gracias a sucesivas mejoras en las tecnologías automovilísticas, y a las modificaciones en las normas de seguridad vial que han implicado el recrudecimiento de las sanciones por algunos delitos viales, se ha conseguido disminuir considerablemente el número de víctimas de accidentes de tráfico en los últimos años. Sin embargo, mientras sigan existiendo fallecimientos en accidentes viales, sigue siendo necesario seguir tomando más y mejores medidas de prevención.

Entre las causas más frecuentes de los accidentes de tráfico se encuentra el exceso de velocidad, seguido del consumo de alcohol y drogas, que causan más del 60 % de los accidentes al volante. Por otro lado, entre el 15 % y el 30 % de los accidentes está causado, directa o indirectamente, por la somnolencia de los conductores. Es ese un trastorno del sueño que conlleva una repentina e intensa necesidad de dormir, que conduce a quedarse dormido sin poder evitarlo [1].

Existe bastante bibliografía reciente relacionada con el reconocimiento del cansancio al volante [2] [3], pero dada la cantidad de consideraciones que se deben tener en cuenta en sistemas reales por la heterogeneidad de situaciones, sigue siendo de gran interés su estudio para aplicaciones concretas, como los sistemas de prevención anti-arranque.

El objetivo de este trabajo es diseñar una herramienta con dos funciones diferenciadas. Por un lado, el reconocimiento facial de los conductores permitirá acceder a consultas de información de tráfico relativa a la legalidad del permiso de conducir o a los delitos cometidos al volante por la persona identificada, impidiendo de esa forma el arranque del vehículo si así lo indican los datos recolectados. Por otro lado, la pronta detección de signos de somnolencia en el conductor permitirá alertar de forma temprana de que existe una alta probabilidad de que dicha persona se quede dormido al volante, evitando así que pueda suceder un accidente.

Para poder llevar a cabo el objetivo establecido, se hace necesario detectar los rostros de los conductores en las imágenes, de manera que se pueda averiguar tanto de quién se trata como si se está quedando dormido o no. Con esa meta, en el diseño propuesto en este trabajo se aplican varias técnicas de *Machine Learning* como el Histograma de Gradientes Orientados (HOG, *Histogram of Oriented Gradients*) para la detección de rostros [4], y la Estimación de *Landmark* Facial [5]. Estas técnicas permiten estimar la orientación del rostro para centrarlo al máximo posible, con objeto de poder luego identificar de quién se trata, comparando el *embedding* del conductor, generado con una Red Neuronal Convolucional, con los *embeddings* de los conductores registrados [6]. Después de haber detectado el rostro, se requiere, en primer lugar, comprobar si la persona tiene los párpados abiertos o cerrados, haciendo uso también de una Red Neuronal Convolucional, que permite clasificar en qué estado se encuentran. Después, se estudia si el conductor está bostezando o no y cuántas veces lo hace, haciendo uso del Modelo de Reconocimiento de Actividad Humana [7].

## II. Reconocimiento Facial de Conductores

Una de las maneras que se proponen en este trabajo para intentar disminuir los accidentes viales es mediante el uso de algunos datos de tráfico del conductor, como posesión de licencia de conducir, tenencia de todos los puntos, infracciones cometidas, multas y si están pagadas o no, contratación de seguro, información del vehículo, etc. De esta manera, si el usuario al volante no cumple los requisitos necesarios para conducir el vehículo, se pone en marcha un mecanismo que inutiliza el motor, enviando una señal al bus CAN (*Controller Area Network*) del vehículo [8], dando así lugar a que no se pueda conducir ese vehículo en ese momento.

El flujo de funcionamiento se muestra en la Figura 1. En primer lugar, se detectan las caras existentes en el vídeo que se captura a través de la cámara que se encuentra en el vehículo. Después el programa calcula el *embedding* de esa cara, es decir, los datos que permiten diferenciarla de cualquier otra cara y que le dan esa característica de unicidad. Luego se compara ese *embedding* con los *embeddings* previamente almacenados en la base de datos, es decir, se hace una comparación con las caras ya conocidas. Para la construcción de esa base de datos se propone que a la hora de obtener o renovar la licencia de conducir, se realicen varias fotografías del conductor que permitan calcular el *embedding* de su cara para almacenarlo de forma segura junto con su información

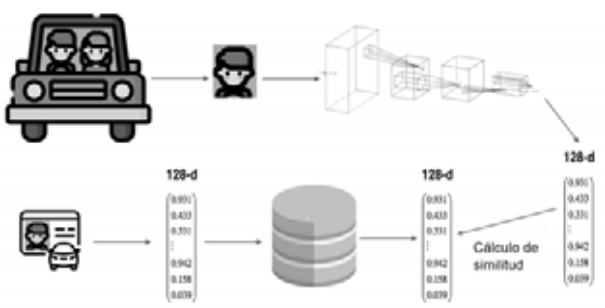



Figura 1: Flujo de funcionamiento del reconocimiento facial

de tráfico. Finalmente, una vez reconocido el conductor, se analizan sus datos de tráfico y si no es apto para conducir ese vehículo se envía señal al inmovilizador del motor.

Para llevar a cabo todo el proceso del reconocimiento facial son necesarios los siguientes pasos:

### *II-A. Detección de Rostros con Histograma de Gradientes Orientados*

El primer paso para el reconocimiento facial es la detección de los rostros presentes en una imagen. Para esto, se utiliza el Histograma de Gradientes Orientados [4], que es un tipo de descriptor de características que se encarga de representar los rasgos elementales de un objeto como la forma, el color o la textura. Este mecanismo funciona observando cada píxel que conforma la imagen y sus píxeles vecinos. El objetivo del HOG es calcular cuán oscuro es el píxel actual comparado con los píxeles vecinos. Para ello, se dibujan flechas mostrando la dirección en la que la imagen se hace más oscura. Estas flechas, llamadas gradientes, se utilizan para mostrar el flujo de tonos claros a oscuros en la imagen. Finalmente, con la repetición de este proceso se consigue convertir cada uno de los píxeles de la imagen en una flecha.

De esa manera se consigue extraer la estructura básica del objeto contenido en la imagen. Sin embargo, al realizar el proceso descrito, se obtienen como resultado muchísimas flechas que serían difíciles de evaluar de manera rápida y eficiente. Es por esto que la imagen se divide entonces en pequeños cuadrados de píxeles y en cada uno de ellos se cuenta cuántos gradientes apuntan en cada dirección principal, de manera que después se reemplaza cada cuadrado en la imagen con las direcciones más fuertes y visibles. De esta forma, se consigue convertir la imagen original en una representación de la estructura básica del objeto, en este caso, un rostro. Entonces para descubrir si ese objeto se corresponde con un rostro, basta con comparar esa nueva imagen HOG con un patrón HOG ya conocido, calculado gracias a una gran cantidad de rostros de entrenamiento.

### *II-B. Proyección de Rostros con Estimación de Landmark Facial*

Una vez detectado un rostro en una imagen, se debe tener en cuenta que no siempre esos rostros están de frente a la cámara, sino que pueden estar de perfil o en otras posiciones. Por eso se debe posicionar el rostro capturado de manera que sea más fácil de estudiar y comparar por el modelo. Para ello, se hace uso de un algoritmo denominado Estimación de *Landmark* Facial [9], que tiene como idea principal describir un rostro

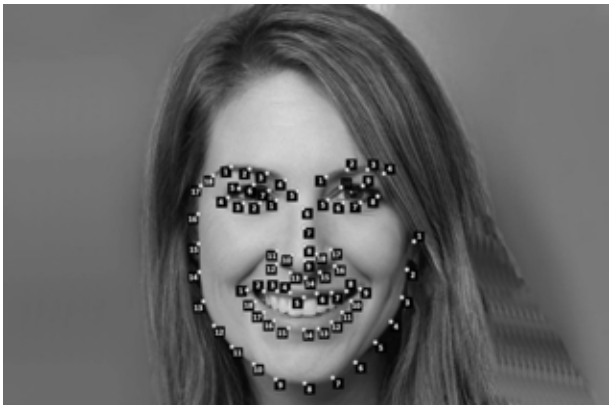


Figura 2: *Landmark* Facial

Fuente: latesttechnicalreviews.com

a través de 68 puntos específicos (ver Figura 2), que es lo que se denomina como *landmark* y que se puede encontrar en cualquier rostro. De esta manera, una vez se sabe dónde están ubicados cada uno de los puntos que detallan dónde se hallan los ojos, la nariz, la boca y demás partes que conforman un rostro, se hace una rotación y escalado, de forma que los ojos y la boca estén lo más centrados posibles, para hacer más sencillo el proceso de cálculo del *embbeding* facial.

### *II-C. Generación de Embedding Facial con Redes Neuronales Convolucionales*

El siguiente paso consiste en calcular el *embedding* del rostro detectado en la imagen, con el fin de compararlo con los demás *embeddings* almacenados en la base de datos de los rostros ya conocidos. Estos *embeddings* son, realmente, 128 números que representan las características únicas y más importantes de un rostro [6], y para calcularlos es necesario hacer uso de una Red Neuronal Convolucional (CNN, *Convolutional Neural Network*), que es un tipo de red neuronal artificial cuyo diseño ha sido pensado para sacar partido a la estructura espacial de una imagen, ya que tiene la capacidad de interpretar las formas y patrones más complejos presentes en grandes conjuntos de imágenes. Las neuronas de las CNNs son muy similares a las neuronas en la corteza visual primaria de un cerebro biológico, es decir, su procedimiento de aprendizaje se asemeja mucho al proceso de visión de un ser humano. Siguen un procesamiento mediante el cual se identifican primero elementos básicos y generales, que luego son combinados con el fin de generar patrones cada vez más complejos.

Este tipo de redes neuronales se caracterizan por realizar una serie de convoluciones. En procesamiento de imágenes, una convolución no es más que una operación matemática que, realizando combinaciones con los valores de los píxeles, es capaz de generar nuevas imágenes con las que estudiar formas y patrones que componen a la imagen original. Cada píxel nuevo que se vaya a generar se calcula aplicando una matriz de números, llamada filtro o *kernel*, sobre la imagen original, y luego se multiplican y suman los valores de cada píxel vecino para obtener así el nuevo valor. Por tanto, la principal característica de una CNN es aplicar los filtros necesarios para detectar todas las formas y patrones que conforman la imagen.

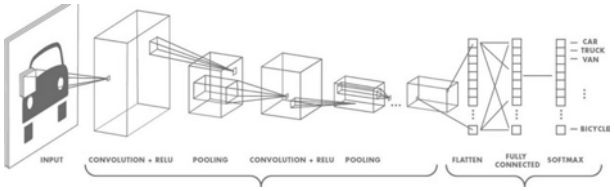



Figura 3: Arquitectura de una CNN

Fuente: es.mathworks.com/discovery/convolution.html

Cada vez que se utiliza un filtro, se genera una nueva imagen, denominada mapa de características, que actúa como un mapa donde se indica en qué parte de la imagen se ha detectado la característica buscada por el filtro.

Lo destacable de la CNN es que el proceso de búsqueda de características se realiza de forma secuencial, es decir, la salida de una de las capas se convierte en la entrada de la siguiente. Por eso la operación de convolución se hace cada vez más potente, puesto que cada vez que se realiza una convolución se está accediendo a más información espacial de la imagen original. Con las convoluciones lo que se está haciendo son nuevas detecciones sobre las detecciones recibidas de capas anteriores. En otras palabras: realizar las detecciones sobre detecciones anteriores permite componer cada vez patrones más complejos.

La arquitectura de una CNN (ver Figura 3) se representa normalmente como un embudo, ya que la imagen inicial se va comprimiendo espacialmente, es decir, su resolución va disminuyendo, al mismo tiempo que su grosor va en aumento, puesto que el número de mapas de características que se detectan crece. Finalmente, se consigue obtener todos los patrones necesarios de la imagen, que luego se introducen como *inputs* independientes dentro de una red neuronal multicapa para descubrir qué representa esa imagen. En el caso que nos ocupa, calcula el *embedding* del rostro que se esté analizando.

## III. Detección de Signos de Somnolencia

Otra propuesta para intentar disminuir el número de accidentes de tráfico y víctimas mortales es detectando de manera temprana los signos de somnolencia que pudiera estar presentando el conductor, con el fin de poner en marcha una alerta que le haga despertarse y tomar conciencia de su estado.

En la línea de varios trabajos previos, como [10] o [11], usaremos redes neuronales convoluciones para la detección de la somnolencia.

Para realizar este proceso, se estudian las imágenes recibidas a través de la cámara del vehículo, detectando rostros, ojos y boca, para observar el comportamiento de estas partes del cuerpo y determinar si están pasando por un proceso de somnolencia, ya sea por la frecuencia de parpadeo, por la regularidad con la que suceden los bostezos en un periodo de tiempo determinado o por movimientos involuntarios de cabeza que indiquen que una persona se está dejando dormir.

Este cometido se puede cumplir llevando a cabo los siguientes pasos (ver Figura 4). Primero, se toman las imágenes capturadas por la cámara de vídeo y se detecta el rostro del conductor, el cual se marca como una Región de Interés (ROI, *Region of Interest*). Dentro de esa ROI se detectan los ojos con los que se alimenta el clasificador, que en este caso es también

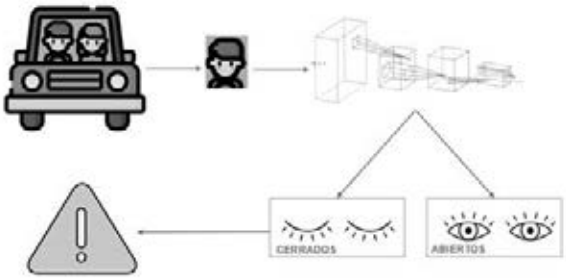



Figura 4: Flujo de Detección de Signos de Somnolencia

una CNN. Ésta se encarga de comprobar si los párpados del conductor están abiertos o cerrados.

Más adelante, se detecta si la persona que conduce el vehículo está bostezando o no. Para ello, se hace uso del Modelo de Reconocimiento de Actividad Humana [12] con OpenCV, que es una biblioteca libre de visión artificial [13]. Dicho modelo ha sido previamente entrenado con el *Dataset* de Vídeos de Acciones Humanas de *Kinetics* [14] de forma que detecta si el conductor está bostezando y cuántas veces lo hace. En este modelo, se han utilizado arquitecturas 2D existentes que han sido extendidas a través de *kernels* 3D para llevar a cabo la clasificación de vídeos.

Una vez clasificados estos datos, si se comprueba que el conductor tiene los párpados cerrados durante más segundos de lo normalmente establecido, y que presenta una frecuencia de bostezo bastante alta, se lanza una señal a una alarma conectada al bus CAN del vehículo, con el fin de alertar al conductor de que se está quedando dormido o de que tiene una alta probabilidad de hacerlo.

## IV. Inmovilización del Vehículo y envío de alerta por somnolencia a través del bus CAN

El bus CAN es un sistema diseñado por la compañía alemana *Robert Bosch GmbH*, que fue lanzado oficialmente en 1986 en el congreso de la Sociedad de Ingenieros Automotrices (SAE, *Society of Automotive Engineers*). Este sistema en serie basado en una topología bus permite intercambiar información entre los distintos componentes electrónicos de un vehículo, como son, por ejemplo, el motor, los asientos eléctricos, el climatizador, el techo solar, etc.

Por una parte, durante el arranque del vehículo, una vez se ha identificado al conductor mediante reconocimiento facial, se accede a sus datos de tráfico con el fin de saber si está en condiciones de conducir ese vehículo o no. Para esto, la cámara a través de la cual se capturan las imágenes debe estar conectada al bus CAN del vehículo, de forma que cuando se consiga reconocer al conductor y se sepa si puede conducir el vehículo, se envíe una señal al inmovilizador del motor indicando si puede ser arrancado o no.

Por otra parte, durante la conducción, en el supuesto de que se detecten signos de somnolencia en el conductor, se envía una señal de alarma también a través del bus CAN del vehículo, ya sea mediante un mensaje visual en la pantalla integrada, o un mensaje sonoro o bien la mezcla de ambos.

## V. Herramientas Utilizadas

Para diseñar el sistema con todos los pasos comentados, previamente se estudiaron las herramientas existentes en el mercado con funcionalidades similares.

Entre todas las herramientas analizadas destaca la librería de *Python*, *face_recognition* [16], que permite reconocer y manipular rostros. Dicha librería está creada con *Dlib*, un *kit* de herramientas que contiene algoritmos de *Machine Learning* y mecanismos para crear softwares complejos. Fue creada por Adam Geitgey en 2017, consiguiendo encapsular en una sola librería todos los pasos descritos en la sección II, con el fin de trabajar con rostros capturados en imágenes.

Para ambas fases de este trabajo se puede utilizar dicha librería, pues facilita mucho el trabajo y permite realizar el reconocimiento y la detección de rostros de una manera rápida y eficiente. La captura de las imágenes de la cámara de vídeo se hace gracias al uso de *OpenCV* y *Machine Learning*. Con *OpenCV* no solo se puede realizar la detección de rostros, sino que también se pueden detectar partes que conforman el rostro como los ojos, la nariz o la boca, además de clasificar acciones humanas captadas en vídeo, como, por ejemplo, la acción de bostezar.

A la hora de acceder a los datos de tráfico de un conductor, ha hecho falta simular la base de datos de la Dirección General de Tráfico. Por ello, en principio, se ha hecho uso de una base de datos no relacional con *MongoDB*, ya que lo natural es que crezca cada vez más, aumentando con rapidez su tamaño considerablemente. La API con la que se accede a la base de datos se ha realizado con *NestJS*, el cual es un *framework* de *Node.js* para generar aplicaciones web escalables y eficientes. En relación con la protección de la información almacenada, se han seguido las últimas consideraciones de seguridad relativas a bases de datos NoSQL [15].


Figura 5: Parametrización de la BBDD MOngoDB

## VI. Conclusiones y Trabajo Futuro

El número de víctimas mortales en las carreteras sigue siendo alarmante. Gran parte de estos accidentes son ocasionados por somnolencia. El principal objetivo de este trabajo es proporcionar una contribución en este sentido para hacer más seguras las carreteras, de manera que se consiga reducir el porcentaje de accidentes viales producidos con ese motivo.

Para ello, se ha realizado un estudio de algunas de las herramientas existentes en el mercado, de cómo funcionan y de cómo se pueden utilizar para cumplir el objetivo propuesto. En este sentido, si bien es cierto que existen varias herramientas que pueden usarse para conseguir el cometido, en este trabajo se ha optado por investigar dichas herramientas, elegir las más adecuadas y eficientes, y diseñar una propia a partir de la información obtenida y los medios disponibles.

En resumen, lo que se ha realizado en este trabajo es combinar algunas herramientas existentes, convenientemente elegidas, para dar lugar a un programa capaz de cumplir con los requisitos establecidos con objeto de alcanzar el objetivo propuesto. Por un lado, el reconocimiento facial del conductor permite comprobar sus datos de tráfico y ver si posee las condiciones adecuadas para conducir. Por otro lado, la detección temprana de signos de somnolencia en el conductor permite alertar de manera temprana la posibilidad de quedarse dormido al volante.

Por tanto, después de haber realizado un estudio del mercado y de haber elegido los instrumentos necesarios, se ha llevado a cabo el diseño del modelo contemplando la implementación tanto del reconocimiento facial de conductores como de la detección de signos de somnolencia, de manera que en un futuro se pueda llegar a hacer una demostración en vivo de la utilidad de la herramienta diseñada. Teniendo en cuenta que este es un trabajo actualmente en desarrollo, se hace necesaria una implementación optimizada así como la realización de tests simulados y en vivo que permitan comprobar el correcto comportamiento de las funcionalidades desarrolladas, así como su actualización y mejora.

## Agradecimientos

Esta investigación ha sido posible gracias al Acuerdo de Subcontratación entre Nokia Spain, S.A. y la Universidad de La Laguna, y a la Cátedra Institucional de Ciberseguridad Binter - Universidad de La Laguna.

## Referencias

[1] "Conducir con sueño o cansancio", Dirección General de Tráfico, 2022. [Online]. Available: https://www.dgt.es/muevete-con-seguridad/evita-conductas-de-riesgo/Conducir-con-sueno-o-cansancio. [Accessed: 02-Jun-2022].

[2] Feng You, Yunbo Gong, Haiqing Tu, Jianzhong Liang, and Haiwei Wang: "A fatigue driving detection algorithm based on facial motion information entropy", en *Journal of advanced transportation*, 2020.

[3] Albadawi, Yaman, Maen Takruri, and Mohammed Awad: "A review of recent developments in driver drowsiness detection systems", en *Sensors*, 22(5), p. 2069, 2022.

[4] Navneet Dalal and Bill Triggs: "Histograms of Oriented Gradients for Human Detection", en *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005.

[5] Yue Wu and Qiang Ji: "Facial landmark detection: A literature survey", en *International Journal of Computer Vision*, 127(2), 115-142, 2019.

[6] Florian Schroff, Dmitry Kalenichenko and James Philbin: "FaceNet: A Unified Embedding for Face Recognition and Clustering", en *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] Lin Fan, Zhongmin Wang and Hai Wang: "Human activity recognition model based on decision tree ", in *IEEE International Conference on Advanced Cloud and Big Data*, pp. 64-68, 2013.

[8] ISO 11898-1:2015 "Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling". Available: https://www.iso.org/standard/63648.html. [Accessed: 16-Jul-2022].

[9] Vahid Kazemi and Josephine Sullivan: "One Millisecond Face Alignment with an Ensemble of Regression Trees", en *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] Jonathan Flores-Monroy, Mariko Nakano-Miyatake, Gabriel Sanchez-Perez and Hector Perez-Meana: "Visual-based Real Time Driver Drowsiness Detection System Using CNN", en *18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)* pp. 1-5, 2021.

[11] Abid Ali Minhas, Sohail Jabbar and Muhammad Farhan: "A smart analysis of driver fatigue and drowsiness detection using convolutional neural networks", en *Multimedia Tools and Applications*, -05-27 2022

[12] Kensho Hara et al.: ”Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?”, en *arXiv from Cornell University*, 2017.
[13] OpenCV. [Online]. Available: https://opencv.org. [Accessed: 02-Jun-2022].
[14] Will Kay et al.: ”The Kinetics Human Action Video Dataset”, en *arXiv from Cornell University*, 2017.
[15] Sabrina Sicari, Alessandra Rizzardi and Alberto Coen-Porisini: ”Security & privacy issues and challenges in NoSQL databases”, en *Computer Networks (Amsterdam, Netherlands : 1999)*, vol. 206, Apr 07, 2022.
[16] Adam Geitgey: ”Face Recognition Documentation”, en *media.readthedocs.org*, Release 1.2, 2019.

# Sistema para la gestión automática de las políticas de privacidad y uso de las cookies

Cristòfol Daudén-Esmel
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT
Universitat Rovira i Virgili
Av. Països Catalans 26, E-43007 Tarragona, Spain
critofol.dauden@urv.cat

Jordi Castellà-Roca
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT
Universitat Rovira i Virgili
Av. Països Catalans 26, E-43007 Tarragona, Spain
jordi.castella@urv.cat

Alexandre Viejo
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT
Universitat Rovira i Virgili
Av. Països Catalans 26, E-43007 Tarragona, Spain
alexandre.viejo@urv.cat

***Resumen*—El incremento en el uso de Internet implica que cada vez más usuarios compartan información sensible sin darse cuenta de que se exponen a riesgos innecesarios. Para ello, la UE propuso el Reglamento General de Protección de Datos (RGPD), según el cual el consentimiento para la obtención y procesamiento de los datos personales de los usuarios debe darse de forma clara e inequívoca. Un claro ejemplo donde esto se lleva a cabo, son los pop-ups que aparecen mientras navegamos por internet. Sin embargo, una correcta configuración del consentimiento requiere de un esfuerzo para el usuario, llevándolo en la mayoría de casos a la aceptación todas las condiciones sin siquiera leerlas. Así, en este proyecto realizamos una propuesta de marco para la gestión automática de las políticas de privacidad y uso de las cookies de los usuarios, mediante el uso de contratos inteligentes y la tecnología blockchain.**

***Index Terms*—RGPD, criptología, seguridad de la información, blockchain, contratos inteligentes**

## I. Introducción

El uso de las tecnologías de información y comunicación (TIC) ha seguido crecido en los últimos años[1]. La popularidad de las redes sociales y la digitalización de las gestiones online [2] han contribuido significativamente a ello. No obstante, uno de los efectos colaterales de esta situación es que los usuarios comparten información sensible por Internet sin darse cuenta de que puede afectar a su seguridad, es decir, sus datos personales les exponen a riesgos innecesarios [1].

Para que los usuarios sean sensibles a esta exposición y puedan prevenirla es importante que sepan en que consiste su privacidad de datos, cómo protegerla y proporcionarles herramientas con las que puedan ejercer este derecho.
En este sentido, una de las claves es el consentimiento de datos, regulado por el Reglamento General de Protección de Datos (RGPD)[2]. Según esta normativa, el consentimiento para la obtención y procesamiento de los datos personales de los usuarios debe darse de forma clara e inequívoca por parte del usuario interesado, el cual debe haber recibido información detallada acerca de la gestión de sus datos por parte del servicio o página online.
Un claro ejemplo, en donde se pide un consentimiento para la recaudación y uso de datos, son los pop-ups que aparecen mientras navegamos por internet[3]. En ellos se informa al usuario del uso que hacen los proveedores de servicio sobre los datos que recaban mientras se navega por sus páginas. Con el acuerdo de los usuarios, estos proveedores de servicio y sus socios, usan cookies o tecnologías similares para almacenar, acceder y procesar datos personales como: almacenar o acceder a información en un dispositivo, anuncios y contenido personalizados, medición de anuncios y del contenido, información sobre el público y desarrollo de productos, datos de localización geográfica precisa e identificación mediante las características de dispositivos, uso de cookies técnicas o de preferencias, etc.

De acuerdo con la legislación previamente mencionada, el usuario puede retirar su consentimiento u oponerse al procesamiento de datos basado en intereses legítimos en cualquier momento. Sin embargo, tal como se menciona en estos mismos pup-ups, esto puede provocar un incorrecto funcionamiento de ciertos servicios e incluso el no acceso a la plataforma web.
En este proceso de aceptación de las condiciones para la recolección y uso de datos, los usuarios, por norma general, pueden aceptar todos los usos con un solo clic u optar por una configuración de usos más personalizada, rechazando algunos de estos. La segunda opción, sin embargo, requiere de un esfuerzo extra para el usuario, ya que debe leer detenidamente cada uno de estos usos y aceptar y/o rechazar cada uno de ellos individualmente. De este modo, en la mayoría de casos, los usuarios aceptan todas las condiciones sin siquiera leerlas[4].

Como ya hemos comentado, este proceso se debe llevar

[1]Individuals - internet use [ISOC_CI_IFP_IU] https://ec.europa.eu/eurostat/databrowser/bookmark/1614ab44-2592-48ee-99b1-01237e038e78

[2]Individuals - internet activities [ISOC_CI_AC_I] https://ec.europa.eu/eurostat/databrowser/bookmark/0a0bfe7e-85a1-4a0c-84d6-fc2f5bf006ab

[3]Didomi https://www.didomi.io/

[4]76 % Ignore Cookie Banners https://www.advance-metrics.com/en/blog/76-ignore-cookie-banners-the-user-behavior-after-30-days-of-gdpr/

a cabo cada vez que un usuario quiera acceder a un nuevo sitio web, por lo que resulta muy complicado para este, tener conocimiento/recordar de todos aquellos lugares donde ha dado su consentimiento y de que tipo es este (ha aceptado todas las condiciones, las ha rechazado o ha optado por una configuración personalizada). Además, estos sistemas de gestión de consentimientos siguen una estrategia centralizada guardando estos permisos en los servidores de los proveedores de servicios, de forma que el usuario debe confiar en que dicho proveedor es honesto en lo referente a sus preferencias.

Con todo lo que hemos mencionado podemos extraer una serie de necesidades:

i) Se requiere de una herramienta que de forma automática y transparente gestione las preferencias del uso de las cookies y del nivel de privacidad que desea cada usuario. De este modo, se evitará el "Aceptar Todo" al cual hacen uso la mayoría de los usuarios para acceder al contenido web deseado con la mayor brevedad posible.
ii) Se requiere de una alternativa a esta centralización de la gestión del consentimiento en los servidores de los proveedores de servicio. Como alternativa, algunos trabajos ([3], [4], [5], [6], [7], [8], [9], [10], [11]) sugieren descentralizar estos consentimientos mediante su inclusión en Contratos Inteligentes (SCs), los cuales son ejecutados en una blockchain (se realiza el "*deploy*" del contrato inteligente).

Cabe mencionar, que para realizar el *deploy* de un SC en una blockchain se requiere de una pareja de claves (una clave pública y una privada) para calcular una firma digital. El problema está en que si un usuario usa siempre la misma pareja de claves para firmar cada uno de los SCs sobre el consentimiento de uso de sus datos personales, un adversario podría re-identificar al usuario al cual pertenece una clave pública concreta. Esto se llevaría a cabo mediante el análisis de todas las plataformas web a las que ha accedido. Además, una vez identificado el usuario, se puede extraer información sensible del mismo, analizando las mismas páginas web a las que ha accedido.
Teniendo esto en cuenta, las propuestas previamente mencionadas no consideran o no presentan ninguna solución para la protección de la privacidad de los usuarios, en lo referente al uso de la misma pareja de claves para realizar el *deploy* de los SCs.

Para evitar esta situación, proponemos el uso de una nueva pareja de claves cada vez que se genera un nuevo SC entre un usuario y un nuevo proveedor de servicio. Así, se requiere de un sistema que permita la generación y gestión de todas estas parejas de claves de forma automática y transparente al usuario. Este sistema debe ser multi-plataforma, de modo que permita al usuario hacer uso de estas claves desde cualquier equipo para gestionar los SCs que contienen estos/sus consentimientos sobre el uso de sus datos personales por los proveedores de servicio.

### *I-A. Contribución*

En este trabajo realizamos una propuesta de marco para la gestión automática de las políticas de privacidad y uso de las cookies de los usuarios. De acuerdo con las necesidades mencionadas anteriormente, este marco:

a) gestiona de forma **automática** y **transparente** la aceptación de políticas y uso de las cookies en cuanto el usuario navega por internet;
b) registra de forma **descentraliza** estas políticas aceptadas mediante el uso de contratos inteligentes y la tecnología blockchain, y;
c) protege la **privacidad** del usuario en el caso de que un atacante pretendiese identificarlo y extraer información sensible sobre el mismo.

En cuanto al punto a), existen plug-ins que se encargan de la automática aceptación de las cookies en navegadores web [5] [6] [7]. Estos, sin embargo, se limitan a la simple interacción con el "*pop-up*"de la web. Del mismo modo, como ya hemos mencionado, respecto al punto b), también existen propuestas para la gestión del consentimiento de un usuario a proveedores de servicio mediante SCs.
Así, en este trabajo combinamos tecnologías existentes que abordan los puntos a) y b), y nos centramos en el diseño de un protocolo para la generación y gestión de las parejas de claves necesarias para completar el proceso de firma digital para el desarrollo de los contratos inteligentes en la blockchain. De este modo mantenemos la privacidad de los usuarios, en un marco para la automática gestión de las políticas de privacidad y uso de las cookies de los usuarios, en cuanto estos navegan por internet.

El resto del trabajo está organizado de la siguiente forma: en la sección II se muestra el marco propuesto y se explica el protocolo para la gestión de claves diseñado; en la sección III se analiza la propuesta presentada y; finalmente, se concluye el trabajo en la sección IV.

## II. Propuesta

En esta sección presentamos nuestro marco para la gestión automática de las políticas de privacidad y uso de las cookies de los usuarios. Primero, mostramos una visión general del marco y como funciona y, después, detallamos el protocolo desarrollado para la creación y gestión de las claves criptográficas.

### *II-A. Visión general del marco*

Una visión general de la arquitectura propuesta se muestra en la Figura 1. Como se puede observar, el usuario tiene dos puntos de acceso al sistema propuesto. Por un lado, un plug-in que se debe instalar en el navegador web. Este se encarga de la gestión de las políticas y el uso de las cookies, y de la generación de los contratos inteligentes donde estas quedan registradas. Por otro lado, una aplicación móvil desde la cual los usuarios tienen el control sobre los proveedores de servicio a los que han dado su consentimiento para el procesamiento de datos, cuáles son estos permisos y gestionar los contratos inteligentes generados por el plug-in (e.j. restringiendo el uso de los datos).
Otro componente esencial del sistema es el *Key Storage*, donde se guardan todas las parejas de claves generadas por el plug-in para poder realizar el *deploy* de los SCs en la

[5]Ninja Cookie https://addons.mozilla.org/es/firefox/addon/ninja-cookie/
[6]Super Agent https://addons.mozilla.org/es/firefox/addon/super-agent/
[7]Auto Cookie Optout https://addons.mozilla.org/es/firefox/addon/auto-cookie-optout/

blockchain. Este *Key Storage* consiste en una Base de Datos Personal Online (Personal Online Database - POD) la cual permite el almacenaje seguro y gestión de dichas claves mediante el protocolo diseñado.
Finalmente, el último componente es la blockchain sobre la que se llevará a cabo el *deploy* de los contratos inteligentes.

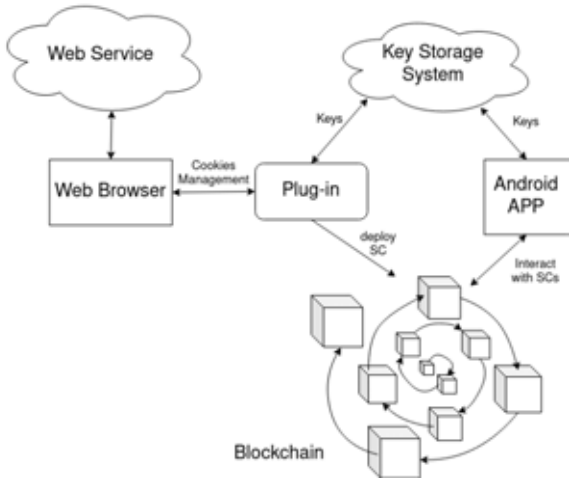



Figura 1. Framework

### *II-B. Protocolo*

El protocolo propuesto se divide en tres fases: i) fase de inicialización o set-up, donde se instala la aplicación móvil y el plug-in en el navegador; ii) fase de navegación, donde el usuario visita distintos sitios web y; iii) fase de control, donde el usuario puede consultar en que sitios web ha dado su consentimiento para el procesamiento de sus datos, de que tipo es este y revocar dicho permiso en el caso que lo crea oportuno. Una visión más general de este protocolo se puede observar en la Figura 2.

*II-B1. Fase de Set-up:* En esta primera fase, el usuario debe realizar una serie de tareas para poder usar el sistema propuesto:

1. Instalar y configurar en un dispositivo móvil (smartphone) la aplicación provista. Este proceso de configuración consta a su vez de distintos pasos:
   *a*) Generar un login seguro para el acceso a la aplicación mediante usuario y contraseña. Se ofrece al usuario la posibilidad de usar un Segundo Factor de Autenticación (2AF) para mejorar la seguridad de acceso.
   *b*) Sincronizar la aplicación con una POD. La aplicación provee al usuario de una POD gratuita; sin embargo, se le permite a este cambiarla.
   *c*) Generar una pareja de claves de criptografía asimétrica mediante un generador (clave pública $PK_0$ y clave privada $SK_0$). Este generador utilizará 12 palabras aleatorias como semilla para la generación de dichas claves. El usuario debe anotar y guardar estas palabras para restaurar el sistema en caso de que haya algún problema con el dispositivo móvil.
   *d*) Guardar la clave privada ($SK_0$) en el *Secure Storage* del dispositivo móvil.
   *e*) Guardar la clave pública ($PK_0$) en la POD, la cual ha sido previamente sincronizada con la aplicación.
2. Instalar y configurar el plug-in provisto en los navegadores que el usuario vaya a usar para navegar por internet. Este proceso, a su vez, también requiere de algunos pasos:
   *a*) Sincronizar el plug-in con la POD previamente escogida.
   *b*) Configurar el nivel de privacidad deseado por parte del usuario. Aquí, se le pregunta al usuario que nivel de privacidad desea, del 0 al 3, siendo el 0 en el que se aceptan todas las políticas y uso de las cookies y el 3 donde se rechazan todas.
   *c*) Descargar y almacenar en memoria la clave pública generada por la aplicación móvil ($PK_0$).

*II-B2. Fase de Navegación:* Una vez completada la fase de set-up, el usuario ya puede empezar a navegar por internet, de modo que todo el proceso de aceptación de políticas de privacidad y del uso de las cookies se realizará de forma automática y transparente mediante el plug-in instalado en el navegador. De este modo, cada vez que el usuario acceda a una nueva web donde se le requiera un consentimiento previo, el sistema realizará los siguientes pasos:

1. Generar una nueva pareja de claves asimétricas($PK_i$ y $SK_i$).
2. Aceptar las políticas y cookies en función del nivel de privacidad especificado por el usuario.
3. Exportar las políticas aceptadas a un contrato inteligente.
4. *Deploy* del contrato inteligente en la blockchain mediante la clave privada previamente generada ($SK_i$).
5. Generar una clave simétrica ($K_i$).
6. Cifrar la pareja de claves generada ($PK_i$ y $SK_i$) usando la clave simétrica $K_i$ ($Enc_{K_i}(PK_i, SK_i)$).
7. Cifrar la clave simétrica ($K_i$) usando la clave pública obtenida de la POD (($Enc_{PK_0}(K_i)$)).
8. Guardar la pareja de claves, cifradas con la clave simétrica ($Enc_{K_i}(PK_i, SK_i)$); y la clave simétrica, también cifrada con la clave pública $PK_0$ ($Enc_{PK_0}(K_i)$), en la POD.

*II-B3. Fase de Control:* En cualquier momento, el usuario debe tener acceso a todos aquellos sitios en los que ha dado su consentimiento y poder modificar o retirar dicho consentimiento de acuerdo con sus derechos sobre sus datos personales. Esto se lleva a cabo a través de la aplicación móvil provista. Así, en esta fase se explica el funcionamiento del sistema en cuanto un usuario quiera tener acceso y o modificar dichos permisos:

1. Descargar el bloque cifrado ($Enc_{K_i}(PK_i, SK_i)$, $Enc_{PK_0}(K_i)$).
2. Descifrar la clave simétrica ($K_i$) mediante la clave privada almacenada en el *secure storage* del dispositivo móvil ($SK_0$): $Dec_{SK_0}(K_i)$.
3. Descifrar la pareja de claves asimétricas (($PK_i$ y $SK_i$)) mediante la clave simétrica previamente obtenida ($K_i$): $Dec_{K_i}(PK_i, SK_i)$.
4. Obtener el contrato inteligente que contiene el acuerdo aceptado por el usuario sobre las políticas de privacidad

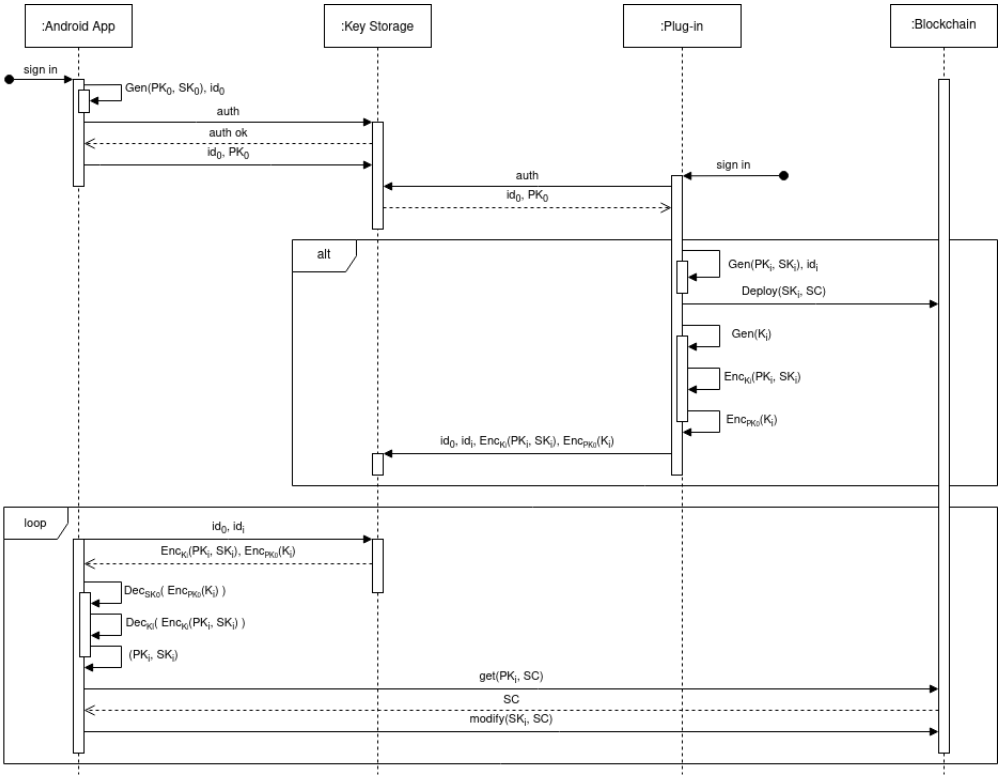



Figura 2. Protocol

y el uso de las cookies, mediante la clave pública obtenida anteriormente ($PK_i$).

5. Modificar el acuerdo en el contrato inteligente mediante una transacción firmada con la clave privada obtenida anteriormente ($SK_i$).

Cabe destacar que el contenido de los contratos inteligentes se muestra de una forma clara e intuitiva a los usuarios, del mismo modo que cualquier modificación a realizar sobre los consentimientos almacenados en los mismos.

## III. Discusión

En esta sección demostramos como el protocolo para la gestión de claves criptográficas permite mantener la privacidad de un usuario, en el caso de que un atacante intente re-identificarlo y obtener información sensible sobre él. Como no se guarda ningún tipo de información personal del usuario en los SCs, la única forma de que un atacante pueda re-identificar a dicho usuario es mediante el uso de casi-identificadores. En este caso, el atacante puede usar todas aquellas páginas web a las cuales el usuario ha accedido como estos casi-identificadores.

En una primera instancia, el atacante podría extraer de la BC todos los SCs asociados a una determinada clave pública y mediante ellos re-identificar al usuario. Para evitar este ataque, el sistema genera una nueva pareja de claves cada vez que el usuario acede a una nueva plataforma web, por lo que no habrá en la BC dos SCs asociados a una misma clave pública.

En una segunda instancia, el atacante podría obtener acceso a la POD donde se guardan todas estas parejas de claves generadas para la creación y gestión de los SCs. Para evitar que un atacante que ha obtenido acceso al POD pueda obtener acceso a las parejas de claves guardadas, estas son cifradas mediante criptografía simétrica (usando una nueva clave simétrica por cada pareja de claves asimétricas). Así mismo, estas claves simétricas son cifradas mediante una clave púbica, manteniéndose la clave privada asociada segura en el *secure storage* del dispositivo móvil donde se ha instalado la aplicación provista con el marco. Así, aunque un atacante pudiese acceder a la POD de un usuario concreto, este no podría acceder a las claves asimétricas y, por tanto, detectar cuáles son los SCs asociados a dicho usuario.

## IV. Conclusiones

En este trabajo hemos presentado un marco para la gestión automática de las políticas de privacidad y uso de las cookies de los usuarios, centrándonos en el protocolo desarrollado para la creación y gestión de las claves criptográficas. Este protocolo permite el uso del sistema a la vez que permite mantener la privacidad del usuario en el caso de que un atacante intentase re-identificar a dicho usuario y obtener información sensible sobre él, como se ha demostrado en la sección III.

### *IV-A. Future work*

Como trabajo futuro pretendemos añadir integridad en el almacenamiento de las parejas de claves asimétricas, de forma que un usuario pueda detectar si se ha llevado a cabo alguna modificación o se ha eliminado alguna de las claves.

## Agradecimientos

Esta investigación cuenta con el apoyo del Fondo de Desarrollo Regional de la Unión Europea en el marco del Programa Operativo FEDER de Cataluña 2014-2020 con una subvención del 50 % del coste total subvencionable, en el marco del proyecto FEM-IOT [001-P-001682]; de la Comisión Europea

(proyecto H2020-871042 ”SoBigData++”); y del Gobierno de España (proyecto RTI2018-095094-B-C21 Çonsentimiento”). El autor también cuenta con el apoyo del Gobierno español mediante una beca FPU (ref. FPU20/03254).

## REFERENCIAS

[1] A. Esteve, “The business of personal data: Google, Facebook, and privacy issues in the EU and the USA,” *International Data Privacy Law*, vol. 7, no. 1, pp. 36–47, 03 2017. [Online]. Available: https://doi.org/10.1093/idpl/ipw026

[2] “Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance),” *Official Journal of the European Union L 119*, vol. 59, p. 1–88, 5 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[3] C. Wirth and M. Kolain, “Privacy by blockchain design: A blockchain-enabled GDPR-compliant approach for handling personal data,” in *Reports of the European Society for Socially Embedded Technologies (EUSSET)*, 2018.

[4] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, “BPDIMS: A blockchain-based personal data and identity management system,” in *Conference: Hawaii International Conference on System Sciences*, January 2019.

[5] M. Zichichi, S. Ferretti, G. D’Angelo, and V. Rodríguez-Doncel, “Personal data access control through distributed authorization,” in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, Nov 2020, pp. 1–4.

[6] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, “GDPR-compliant personal data management: A blockchain-based solution,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1746–1761, 2020.

[7] M. Davari and E. Bertino, “Access control model extensions to support data privacy protection based on GDPR,” in *2019 IEEE International Conference on Big Data (Big Data)*, Dec 2019, pp. 4017–4024.

[8] M. Barati and O. Rana, “Tracking GDPR compliance in cloud-based service delivery,” *IEEE Transactions on Services Computing*, pp. 1–1, 2020.

[9] M. M. Merlec, Y. K. Lee, S.-P. Hong, and H. P. In, “A smart contract-based dynamic consent management system for personal data usage under gdpr,” *Sensors*, vol. 21, no. 23, 2021.

[10] S.-S. Jung, S.-J. Lee, and I.-C. Euom, “Delegation-based personal data processing request notarization framework for GDPR based on private blockchain,” *Applied Sciences*, vol. 11, no. 22, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10574

[11] C. Daudén-Esmel, J. Castellà-Roca, A. Viejo, and J. Domingo-Ferrer, “Lightweight blockchain-based platform for GDPR-compliant personal data management,” in *5th IEEE International Conference on Cryptography, Security and Privacy, CSP 2021, Zhuhai, China, January 8-10, 2021*, 2021, pp. 68–73.

# Detección de somnolencia en conductores con un reloj inteligente

Sonia Díaz-Santos
Dpto. de Ingeniería Informática y de Sistemas
Universidad de La Laguna
sdiazsan@ull.edu.es

Pino Caballero-Gil
Dpto. de Ingeniería Informática y de Sistemas
Universidad de La Laguna
pcaballe@ull.edu.es

***Resumen*—El objetivo principal de este trabajo es detectar de forma temprana si un conductor presenta síntomas de sueño que indiquen que se está quedando dormido para, en ese caso, generar una alerta que despierte al sujeto. Para dar solución al problema planteado, se ha diseñado una aplicación que recopila diversos parámetros, a través de un reloj inteligente, mientras se conduce. En primer lugar, la aplicación detecta la acción de la conducción. Entonces recoge información acerca de las variables fisiológicas más significativas de una persona mientras se encuentra conduciendo. Por otra parte, dado el alto nivel de sensibilidad de los datos gestionados en la aplicación diseñada, en este trabajo se ha prestado especial atención a la seguridad de la implementación. Mediante la solución propuesta se logra mejorar la seguridad vial, reduciendo el número de accidentes por somnolencia al volante.**

***Index Terms*—seguridad vial, reloj inteligente, somnolencia, variables fisiológicas, conducción segura, programación segura**

## I. Introducción

Los accidentes de tráfico suelen poner en peligro la vida no solo del conductor sino también de las demás personas en la vía. Por eso es necesario hacer todo lo posible para reducir su número. Entre las opciones para lograrlo, en este trabajo se ha optado por el desarrollo de tecnologías innovadoras que permitan abordar ese problema. Concretamente el presente estudio se ha centrado en identificar variables fisiológicas que caracterizan el sueño o la fatiga durante la conducción, con el fin de utilizarlas para reducir el número de accidentes provocados por ese motivo, ya que según la Dirección General de Tráfico, la somnolencia interviene, directa o indirectamente, en el 15 - 30 % de los accidentes de tráfico en España [1].

En este trabajo se ha realizado un estudio de las variables fisiológicas más relevantes que permitan concluir si una persona se está quedando dormida. Como se comenta en este documento, según diversas publicaciones, esas variables son: frecuencia cardíaca, electrocardiograma, función respiratoria y estrés. Concretamente, mediante el reloj inteligente utilizado en este trabajo se ha podido recoger los datos de algunas de esas variables fisiológicas de una persona, para localizar las señales de adormecimiento y comprobar, en tiempo real, si la persona se está quedando dormida [2].

Por otra parte, para detectar la acción de la conducción, el reloj inteligente cuenta con un conjunto de sensores como son el acelerómetro, el giroscopio, el podómetro y el sistema de posicionamiento global (GPS, *Global Positioning System*).

En este trabajo se ha partido de la conclusión de que se está conduciendo para luego utilizar los datos resultantes de la combinación del listado de las variables fisiológicas descriptivas del sueño con el de los sensores disponibles en el reloj inteligente utilizado. En particular, se han usado los datos del latido del corazón como la Variabilidad del Ritmo Cardíaco (VRC) y de otras de las variables mencionadas. Luego se han analizado esos datos recabados para detectar la somnolencia en los conductores. Para ello se han utilizado diferentes sensores como los de fotopletismografía (PPG, *PhotoPlethysmoGraphy*) y ElectroCardioGrama (ECG). El sensor PPG utiliza una tecnología basada en la luz para detectar la tasa de flujo sanguíneo controlada por la acción de bombeo del corazón. Además se han utilizado otros sensores como el acelerómetro, giroscopio, podómetro y GPS, integrados en el reloj, para monitorizar la actividad física del usuario [3].

Asimismo se han tenido en cuenta otros factores, como la hora del día, atendiendo al ritmo circadiano según el cual se producen cambios físicos, mentales y conductuales en ciclos de 24 horas. De hecho, se sabe que los accidentes relacionados con la fatiga dependen mucho de la hora del día, así como del tipo de carretera, especialmente en carreteras monótonas. Además, otros factores que se han tenido en cuenta son la edad, el género y el ejercicio físico habitual de una persona, para determinar los parámetros normales de cada individuo. Por ejemplo, la frecuencia cardíaca en deportistas suele ser inferior que en personas sedentarias, y la frecuencia cardíaca suele ser más baja en mujeres que en hombres. Teniendo en cuenta todos estos indicadores se ha podido realizar un mejor análisis de los datos de somnolencia [4].

Este documento se estructura de la forma siguiente. La sección II es un breve estado del arte. En la sección III aportan datos del reloj inteligente y la plataforma usados. La sección IV contiene una discusión sobre variables fisiológicas y sensores de interés para este trabajo. En la sección V se proporciona una descripción del diseño y funcionalidades de la aplicación propuesta. La sección VI describe algunas características de la implementación segura de la aplicación. Por último, la sección VII cierra el trabajo con algunas conclusiones y trabajos futuros.

## II. Estado del arte

En un simulador de conducción, se propuso una métrica de somnolencia llamada PERcentage of eye CLOSure (PERCLOS) como la proporción de tiempo en un minuto en que los ojos están cerrados al menos en un 80 por ciento [5]. Más tarde, se desarrolló un sistema de detección de somnolencia del conductor como una aplicación de dispositivo móvil para medir PERCLOS usando la cámara del dispositivo móvil, pero tenía el riesgo potencial de distraer la atención del conductor y causar accidentes.

El documento [6] propuso un modelo probabilístico para la detección de la somnolencia del conductor destinado a transformar el nivel de somnolencia a valores de 0-1, para ser utilizado en un sistema que consta de una diadema Bluetooth y un reloj inteligente.

El objetivo del paper [7] fue detectar el nivel de somnolencia del conductor con base en los datos recopilados de los sensores de movimiento incorporados en un reloj inteligente, como el acelerómetro y el giroscopio, que sirven como entrada a un algoritmo de aprendizaje supervisado obteniendo una precisión del 98,15 % tras el entrenamiento y las pruebas.

La revisión [8] sobre somnolencia del conductor cubriendo diferentes aspectos de la detección de la somnolencia, incluía aspectos como conjunto de datos de la expresión facial del conductor somnoliento, el método híbrido (integración de varios métodos, visual, no visual, vehicular) y el uso de los sensores de smartwatch y smartphone en la detección de somnolencia.

El sistema propuesto en [9] utiliza una pulsera de diseño propio, cuyos datos de sensores se envían a un dispositivo móvil que sirve como unidad de procesamiento de análisis principal, y esos datos se analizan junto con los sensores de movimiento, para derivar el estado de somnolencia y alertar al conductor mediante alarma gráfica y vibratoria generada por el dispositivo móvil.

Sensores y sistemas basados en cámaras en combinación con medidas de electroencefalograma se usan en [10].

Todos los trabajos mencionados apuntan hacia un interés creciente de la comunidad científica sobre el problema de la detección temprana de somnolencia, si bien, los enfoques llevados a cabo en todos ellos se diferencian en diversos aspectos con respecto al trabajo aquí presentado.

## III. Reloj inteligente y plataformas utilizados

Aquí se proporciona información sobre el reloj inteligente escogido para medir los datos del conductor, así como de la plataforma usada para desarrollar las aplicaciones [11].

### *III-A. Hardware*

Para el desarrollo de este trabajo se ha utilizado el reloj inteligente *Samsung Galaxy Watch 4 LTE*. Este reloj cuenta con el sistema operativo *Wear OS Powered by Samsung*, que permite monitorizar la salud las 24 horas del día. Cuenta con un sensor *BioActive* que mide el ECG y la presión arterial en tiempo real. Para medir la presión arterial utiliza un sensor PPG óptico de frecuencia cardíaca y para medir el ECG utiliza un sensor eléctrico del corazón. Asimismo permite medir la composición corporal a través del sensor de Análisis de Impedancia Bioeléctrica (BIA, *Bioelectrical Impedance Analysis*). Este dispositivo permite medir el oxígeno en sangre y los niveles de estrés para obtener un análisis de sueño completo. Tiene dos sensores de movimiento, el acelerómetro y el giroscopio, permite conocer la localización con el sensor GPS y calcula los pasos con el sensor del podómetro. En cuanto a conexiones, dispone de conexión Bluetooth 5.0 y conexión vía Wi-Fi [12].

### *III-B. Software*

En esta sección se define el software utilizado por los dispositivos, incluyendo la arquitectura de comunicación entre los sistemas, así como las tecnologías y los entornos de desarrollo. También se mencionan la plataforma *Health* y la comunidad *Wear OS*.

*III-B1. Esquema de comunicación de la arquitectura:* Para la comunicación entre los diferentes dispositivos del trabajo se requieren conexiones Bluetooth o Wi-Fi, como se muestra en la Figura 1. El reloj inteligente con sistema operativo *Wear OS Powered by Samsung* cuenta con conexión Bluetooth para conectarse al móvil. El móvil cuenta con la aplicación *Samsung Health*, la cual recoge los datos necesarios de las variables fisiológicas, así como la aplicación *Samsung Health Monitor* para poder obtener los datos de la tensión arterial y el electrocardiograma de la persona que utiliza el reloj. En el ordenador se crea la aplicación que se instala en el reloj inteligente a través del entorno de desarrollo *Android Studio*. La aplicación se instala en el reloj a través de las conexiones mencionadas. Además, con *One UI Watch* se instalan automáticamente las aplicaciones compatibles en el reloj cuando se descargan en el teléfono móvil.

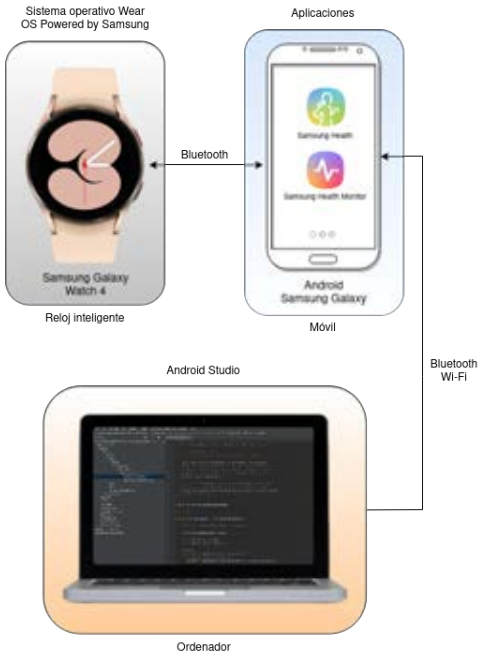



Figura 1. Esquema de comunicación de la arquitectura

*III-B2. Tecnologías:* Las tecnologías utilizadas cuentan con la herramienta de *Android Studio* para desarrollar la aplicación. Los lenguajes de programación que se pueden utilizar para llevarla a cabo son *Kotlin* y *Java*. Es necesario contar con las aplicaciones en el móvil para poder tener acceso a los datos del conductor y realizar su análisis con el fin de detectar si este se encuentra conduciendo y está teniendo síntomas de somnolencia.

*III-B3. Plataforma Health:* El *kit* de desarrollo de software (SDK, *Software Development Kit*) de *Samsung Health* para Android permite compartir los datos de salud entre *Samsung Health*, que se ejecuta en los teléfonos Android, y las aplicaciones asociadas, como puede verse en la Figura 2. También permite a las aplicaciones asociadas utilizar la función de seguimiento de *Samsung Health* mediante la creación de aplicaciones con el SDK. El SDK proporciona un

acceso seguro a los datos de *Samsung Health* con los tipos de datos aplicables. Sin embargo, el intercambio de datos sólo se habilita tras el consentimiento explícito del usuario. El usuario puede seleccionar una configuración detallada para compartir datos, incluyendo qué aplicación asociada accederá a los datos del usuario y qué tipo de datos se leerán o escribirán.

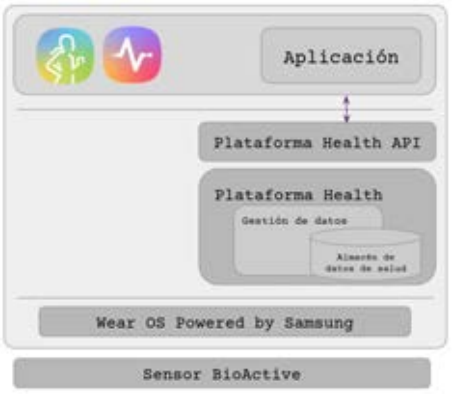



Figura 2. Esquema de la aplicación

*III-B4. Comunidad WearOS:* La comunidad *Wear OS Powered by Samsung* cuenta con una plataforma dedicada para los desarrolladores de Android con multitud de tutoriales acerca de cualquier tipo de dispositivo Android. Existen comunidades en YouTube: *Android Developers*, en Twitter: *AndroidDev* y en LinkedIn: *Android Developers*.

### *III-C. Compatibilidad*

La compatibilidad entre los dispositivos para disponer de todas las funcionalidades y acceso a todos los datos es compleja. El reloj inteligente *Samsung Galaxy Watch 4* tiene compatibilidad con cualquier dispositivo Android pero para poder obtener los datos de los sensores del ECG y de la tensión arterial se requiere que el móvil sea un *Samsung Galaxy* con una versión de Android superior a N, una capacidad de memoria RAM de al menos 1,5 Gb y tener los Servicios de Google para Móviles (GMS, *Google Mobile Services*). Esto quiere decir que aunque la aplicación *Samsung Health* sea compatible con dispositivos Android, incluidos los que no sean Samsung, si se quiere obtener esos datos fisiológicos e instalar la aplicación *Samsung Health Monitor* las restricciones cambian.

## IV. Variables fisiológicas y sensores

En esta sección se detalla cómo influyen las variables fisiológicas más relevantes para este trabajo y cómo funcionan los sensores utilizados, a la hora de recoger los datos [13].

### *IV-A. Fases del sueño*

La Figura 3 muestra el esquema de las fases del sueño, en el que la somnolencia inicial es la más relevante para este trabajo. La fase 1 es el grado más ligero de sueño y dura pocos minutos. En ella se disminuye la actividad fisiológica con una caída gradual de las constantes vitales y del metabolismo. Además, en esa fase es fácil despertarse con estímulos sensoriales.

Durante el sueño no solo ocurren los cambios más conocidos como las alteraciones del ElectroEncefaloGrama (EEG),


Figura 3. Esquema de las fases del sueño

Movimientos Oculares Rápidos (MOR) o ElectroMioGrama (EMG) sino también cambios importantes cardiovasculares, respiratorios, hormonales, renales, digestivos y en general de todo el organismo. Se disminuye la tensión arterial (5-16 %), el pulso y la función respiratoria.

### *IV-B. Detección del sueño y de la conducción*

Para detectar el sueño se recogen los datos de las siguientes variables fisiológicas: frecuencia cardíaca, estrés, tensión arterial y oxígeno en sangre [14].

La frecuencia cardiaca en reposo es el número de veces que el corazón late por minuto cuando se está en reposo. Una frecuencia cardiaca en reposo baja es normalmente sinónimo de una buena salud cardiovascular. El ejercicio aeróbico puede ayudar a reducir la frecuencia cardiaca en reposo con el tiempo. La temperatura, la posición del cuerpo, la actividad reciente o el estado emocional son algunos de los factores que pueden afectar a la frecuencia cardiaca. En la aplicación *Samsung Health* los datos de la frecuencia cardiaca en reposo se basan en estimaciones realizadas en habitantes de Estados Unidos.

El estrés se mide con determinados marcadores biológicos. Cuanto mayor sea el número de mediciones realizadas mayor precisión tendrán los datos recabados del estrés. El tabaco, el alcohol, la cafeína y los medicamentos pueden afectar a las mediciones del nivel de estrés. Cuando se utiliza la función de nivel de estrés, los relojes utilizan los datos de la frecuencia cardíaca, como las pulsaciones por minuto, para determinar el intervalo entre cada latido. La menor variabilidad entre latidos equivale a niveles de estrés más altos, mientras que un aumento de la variabilidad indica menos estrés.

La presión arterial se mide mediante un sensor óptico de frecuencia cardíaca conocido como sensor PPG. Vigilar la tensión arterial es muy importante para la salud. Si la presión arterial está dentro del rango normal, es un buen indicio de que tienes un corazón sano. Pero tener la presión arterial alta, también conocida como hipertensión, puede aumentar significativamente el riesgo de enfermedades cerebrales, renales y cardíacas, incluidas las apoplejías y las enfermedades coronarias cuando no se manejan adecuadamente. En la aplicación se clasificarán como: pulso, tensión arterial sistólica y tensión arterial diastólica. El rango de medición para las lecturas de tensión arterial es: sistólica: 70-180 y diastólica: 40-120.

El nivel de oxígeno en sangre es un indicador de la eficacia con la que se transporta el oxígeno por el cuerpo, lo que a su vez indica si estás respirando de manera eficaz. El nivel de oxígeno en sangre, también conocido como SpO2 (saturación de oxígeno percutáneo), es la medida del porcentaje de hemoglobina que está oxigenada en los glóbulos rojos. Un

intervalo saludable es del 95 % al 100 % cuando se está en reposo. Factores como el ejercicio intenso, la cantidad de oxígeno en el aire, la altitud y diversos problemas de salud pueden dar lecturas con porcentajes más bajos [15], [16].

Todos estos datos son recogidos por el sensor *Samsung BioActive*, que es un sensor de un solo chip 3 en 1. Los sensores son: PPG, ECG y BIA .

El sensor PPG es capaz de medir la frecuencia cardíaca, el oxígeno en sangre, el nivel de estrés y la frecuencia respiratoria. Foto significa luz, Pletismo significa cambio de volumen, y Gram significa gráfico. Por tanto, PPG es un sensor de luz verde infrarroja de baja intensidad y alta precisión que se usa para detectar el volumen del flujo sanguíneo para comprender la fluctuación en la frecuencia cardíaca.

El sensor ECG se utiliza para la detección del ritmo y frecuencia cardíacos. Dado que este sensor es voluminoso, no se puede utilizar para detectar la frecuencia cardíaca cuando el cuerpo está en movimiento. De esa manera se puede medir de manera precisa y continua la frecuencia cardíaca y la variabilidad del ritmo cardíaco (HRV, *Heart Rate Variability*), incluso durante actividad física extrema [17], [18], [19].

En cuanto al funcionamiento, cuando el corazón late, los capilares se expanden y contraen según los cambios en el volumen sanguíneo. El sensor óptico de PPG, que utiliza tecnología tolerante al movimiento, emite señales de luz que se reflejan en la piel para medir de manera precisa y continua las señales débiles del flujo sanguíneo. Por lo que cuando la luz viaja a través de los tejidos biológicos, es absorbida por los huesos, los pigmentos de la piel y la sangre venosa y arterial. Dado que la sangre absorbe la luz con más fuerza que los tejidos circundantes, los sensores de PPG pueden detectar los cambios en el flujo sanguíneo como cambios en la intensidad de la luz. La señal de voltaje de PPG es proporcional a la cantidad de sangre que fluye a través de los vasos sanguíneos. Incluso pequeños cambios en el volumen de sangre pueden detectarse con este método, proporcionando una mayor precisión.

La función de electrocardiograma funciona registrando la actividad eléctrica del corazón mediante un sensor en un *Samsung Galaxy Watch* compatible. La aplicación mide la frecuencia y el ritmo cardiaco, que se clasifican como ritmo sinusal o fibrilación auricular. Un ritmo sinusal es cuando el corazón late de forma constante. Esto se produce cuando las cavidades superiores e inferiores del corazón bombean de forma sincronizada. La fibrilación auricular es cuando el corazón late a un ritmo irregular. Esto se produce cuando las cavidades superiores del corazón no bombean de forma sincronizada con las cavidades inferiores. Si no se trata, puede derivar en coágulos de sangre, apoplejías, insuficiencias cardiacas y otros problemas de salud. Si se presentan síntomas, pueden ser: latidos rápidos o palpitaciones, latido omitido, cansancio, disnea, presión o dolor torácico, desmayos o mareos [20].

El sensor BIA realiza un análisis de la composición corporal en tiempo real colocando dos dedos sobre los dos botones laterales, que actúan como electrodos, para medir la masa muscular, la masa grasa, la grasa corporal, el Índice de Masa Corporal (IMC) y el agua corporal.

Para detectar la acción de conducción se recogen los datos de los siguientes sensores: acelerómetro, giroscopio, podómetro y GPS. Para poder realizar la monitorización solamente cuando la persona se encuentre conduciendo se ejecuta la aplicación en el móvil y se comienza la monitorización cuando se realiza una conexión Bluetooth con el coche. De esta manera, se realiza un paso de autenticación de la aplicación con el vehículo y se consigue limitar el uso a través de la vinculación entre el móvil y el coche, además de contar con una mayor seguridad.

El acelerómetro mide la fuerza, dirección y gravedad de la aceleración y orientación del dispositivo.

El giroscopio mide la velocidad angular del dispositivo y, de esa forma, su posición exacta.Uniendo los datos del giroscopio con el de otros sensores como el acelerómetro, la pulsera mantiene la orientación correcta cuando te mueves, además de poder diferenciar qué tipo de movimiento estás realizando.

El podómetro cuenta los pasos que da la persona que lleva el reloj. También se hace uso del GPS para obtener una información mucho más precisa. Ya que se va a poder medir la distancia recorrida, el tiempo que ha tardado y la cantidad exacta de pasos que se han tenido que dar para poder recorrer dicha distancia en cuestión.

El GPS es un sistema de posicionamiento global que permite determinar la posición de alguien o algo en precisas coordenadas de latitud y longitud en cualquier punto del planeta en tiempo real. El receptor GPS recoge datos de diferentes satélites para calcular tu posición como un conjunto de coordenadas. Esto permite rastrear el trayecto y la distancia. En el caso específico del deporte, un pulsómetro con GPS te ayudará a saber tu velocidad de desplazamiento, el ritmo y la distancia recorrida cuando haces ejercicio físico.

## V. Diseño y funcionalidades de la aplicación

En esta sección se definen objetivo, idea y diseño de la aplicación propuesta. Además se describen las variables de los sensores de la plataforma *Android Developers* para entender cómo se recogen los datos de las variables fisiológicas. Por último, se comenta el análisis de los datos para determinar si un conductor se está quedando dormido o no.

### *V-A. Objetivo*

En la Figura 4 se observa el diagrama de caso de uso de la aplicación, en el cual se describe el proceso con el que se extraen los datos de los sensores y variables del reloj inteligente. Luego, los datos son analizados para determinar si el conductor presenta síntomas de somnolencia. Si no los presenta y continúa conduciendo, se sigue monitorizando. En el caso de que sí los presente, se envía una alerta en forma de vibración al reloj para despertar al conductor. Finalmente, si el conductor continúa conduciendo, se siguen monitorizando las variables fisiológicas con los sensores.

### *V-B. Datos de los sensores*

La plataforma *Android Developers* tiene una clase pública *Sensor* cuyo objetivo es definir las variables de los sensores para poder obtener los datos del reloj inteligente. Existen varios sensores que te permiten supervisar el movimiento de un dispositivo: los sensores vectoriales de rotación, de gravedad, de aceleración lineal, de movimiento significativo, de contador de pasos y de detector de pasos se basan en hardware o en software, y los sensores del acelerómetro y del giroscopio siempre están basados en hardware. Las variables

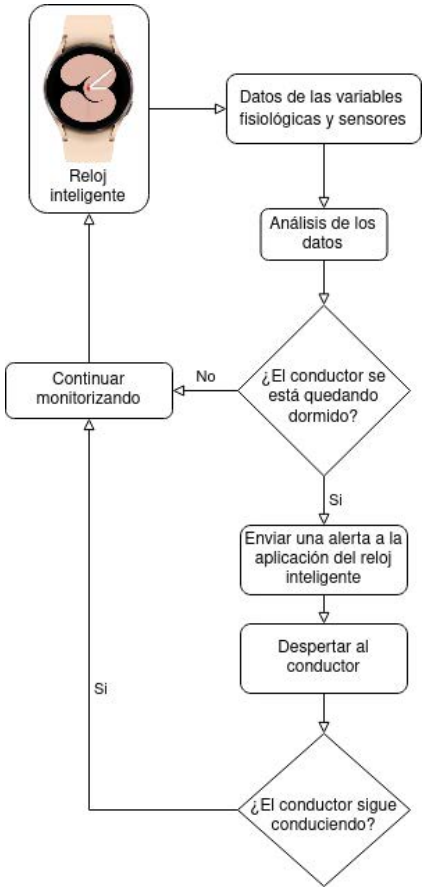



Figura 4. Diseño de la aplicación

más importantes de los sensores se dividen en paquetes para almacenar las clases. En el paquete *android.hardware* las variables más importantes son: *type accelerometer*, *type gyroscope*, *type heart rate*, *type heart beat* y *type step counter*. Todas estas variables tienen dos tipos de datos *string* o *int*. El paquete *android.location* cuenta con la clase *Location* para representar la localización geográfica en la variable *geographic location*. El paquete *android.os* contiene la clase *Vibrator* para producir la alarma en el reloj. Estos son algunos de los ejemplos de clases y variables utilizadas en este trabajo.

### *V-C. Análisis de los datos*

Los datos se pueden analizar con algoritmos que utilicen dos tipos de aprendizaje: supervisado y no supervisado. El aprendizaje supervisado supone que partimos de un conjunto de datos etiquetado previamente, es decir, conocemos el valor del atributo objetivo para el conjunto de datos que disponemos. El aprendizaje no supervisado parte de datos no etiquetados previamente.

En este trabajo inicialmente se parte de la base de que no se tienen conocimientos previos de los datos de una persona, y por tanto en ese caso se utiliza un algoritmo no supervisado. Además, se considera el caso de medición de los datos de interés en estado de reposo para poder utilizar un algoritmo supervisado.

En los entornos de tiempo real, los lenguajes tradicionales para el desarrollo de sistemas basados en reglas no son adecuados debido a la dificultad que implica su análisis temporal. Asimismo, los programas de software basados en datos en lugar de reglas de programación que codifican conjuntos específicos de instrucciones permite entrenar la computadora con grandes cantidades de datos mediante algoritmos que le confieren capacidad de aprender e identificar relaciones o patrones complejos que contribuyen a tomar decisiones precisas. De igual manera se aplican algoritmos de aprendizaje profundo en multitud de diagnósticos con EMG, EEG y ECG. Por ello, para esta tarea se ha decidido utilizar la minería de flujos de datos, ya que es capaz de encontrar los datos esenciales eliminando el ruido. Además, también analiza la llegada de los datos en flujos continuos procesando los datos en tiempo real.

## VI. Seguridad de la aplicación

Al proteger la seguridad de la aplicación, se mejora la confianza de los usuarios en ella. Por tanto, en su desarrollo se han intentado seguir buenas prácticas recomendadas de seguridad en la implementación de software.

En primer lugar se ha intentado garantizar que la comunicación sea segura, protegiendo los datos que intercambia la aplicación con otras aplicaciones y sitios web, mejorando la estabilidad de la comunicación. Esto se consigue usando *intents* implícitos y proveedores de contenido no exportado (mostrando un selector de aplicaciones, aplicando permisos basados en firmas, e inhabilitando el acceso a los proveedores de contenido de la aplicación), solicitando credenciales antes de mostrar información sensible (mediante PIN, contraseña, patrón o credencial biométrica, como reconocimiento facial o huella dactilar), aplicando medidas de seguridad de red (usando tráfico TLS, agregando una configuración de seguridad de red, y creando un administrador de confianza propio) y usando objetos *WebView* cuidadosamente (mediante canales de mensaje HTML).

En segundo lugar, se ha cuidado que la petición de los permisos sea la adecuada, usando *intents* para diferir permisos, y compartiendo datos de manera segura entre aplicaciones.

En tercer lugar, se ha prestado atención al almacenamiento de datos, guardando los datos privados en almacenamiento interno, almacenando solo datos no sensibles en archivos de caché, y usando *SharedPreferences* en el modo privado.

En cuarto lugar, se ha vigilado que los servicios y las dependencias estén actualizados, revisando el proveedor de seguridad de los Servicios de Google Play y actualizando todas las dependencias de la aplicación. Asimismo, el uso de *SafetyNet* proporciona un conjunto de servicios y APIs que ayudan a proteger la aplicación contra amenazas de seguridad, lo que incluye la manipulación del dispositivo, URL incorrectas, aplicaciones potencialmente dañinas y usuarios falsos. Cuenta con el sistema *Android Keystore* que protege el material de claves contra usos no autorizados. Esto evita la extracción del material de claves de los procesos de la aplicación y del dispositivo Android en su totalidad a fin de reducir el uso no autorizado de claves fuera del dispositivo. Además, permite que las aplicaciones especifiquen usos autorizados de sus claves y aplica estas restricciones fuera de los procesos de las aplicaciones para reducir el uso no autorizado de material de claves en el dispositivo.

Por último, para la detección de la acción de conducción se realiza la autenticación de la aplicación con el vehículo para incrementar el nivel de seguridad.

### *VI-A. Administración de claves*

Para la administración de claves se utilizan dos conceptos. Por una parte, se usa un conjunto de claves para cifrar archivos o datos de preferencias compartidas, que se almacenan en *SharedPreferences*. Por otra parte, se usa una clave principal (master), que cifra todos los conjuntos de claves, y que se almacena utilizando el sistema del almacén de claves de Android. La biblioteca de seguridad también incluye dos clases para proporcionar datos más seguros en reposo. Por una parte, la clase *EncryptedFile* se usa para proporcionar operaciones seguras de lectura y escritura desde transmisiones de archivos, mediante cifrado autenticado con datos asociados (AEAD, Authenticated Encryption with Associated Data). Por otra parte, la clase *EncryptedSharedPreferences* se usa para cifrar automáticamente claves y valores mediante una combinación de dos esquemas: primero se cifran las claves mediante un algoritmo determinista, y luego se cifran los valores con *AES-256 GCM* de una manera no determinista.

### *VI-B. Algoritmos criptográficos*

La plataforma permite elegir diferentes algoritmos para cada clase. Para cifrados se recomienda el *AES* en modo *CBC* o *GCM* con claves de 256 bits (como *AES/GCM/NoPadding*), para *MessageDigest* la familia *SHA-2*, para Mac el *HMAC* de la familia *SHA* y para firma la familia *SHA-2* con *ECDSA* (como *SHA256withECDSA*).

Se puede escoger la ejecución de diferentes operaciones criptográficas al leer o escribir un archivo, cifrar un mensaje, generar un resumen del mensaje y generar o verificar una firma digital. Concretamente, existen multitud de algoritmos disponibles compatibles con Android, tales como: *DH, DSA, AES, BLOWFISH, ChaCha20, DES, DESede, EC, GCM, PKCS12PBE, X.509, ECDH, MD5, la familia SHA*, etc. Además, los algoritmos de cifrado *AES, AES_128, AES_256, ARC4, BLOWFISH, ChaCha20, DES, DESede y RSA* permiten escoger entre los distintos modos (*CBC, ECB, GCM*, etc.), y en el caso de que se quiera, también se permite escoger entre diferentes rellenos para el algoritmo escogido.

## VII. Conclusiones y trabajos futuros

En este trabajo se ha diseñado una aplicación que recopila, a través de los sensores de un reloj inteligente, diversos parámetros de algunas de las variables fisiológicas más relevantes que permiten concluir de forma temprana si una persona se está quedando dormida al volante para, en ese caso, generar una alerta que despierta al sujeto con objeto de evitar posibles accidentes viales. Además, en la implementación se han seguido las buenas prácticas recomendadas de programación de código seguro para proteger los datos sensibles manejados por la aplicación.

Actualmente se está investigando la posibilidad de utilizar el sensor BIA con el fin de usar la impedancia para detectar el estado del tono muscular, así como los cambios en el mismo, ya que este disminuye cuando una persona se está durmiendo. Otra posible mejora es el uso del sensor del giroscopio para tratar de detectar el movimiento de caída de la muñeca en el coche debido a la somnolencia. Asimismo, para poder adaptar las alertas a todo tipo de personas se definirán dos alarmas diferentes: la vibración del reloj y un estímulo sensorial auditivo con sonido.

## Agradecimientos

Esta investigación ha sido posible gracias al Acuerdo entre Nokia y la Universidad de La Laguna para el proyecto IMMINENCE financiado por el CDTI, y a la Cátedra Institucional de Ciberseguridad Binter - Universidad de La Laguna.

## Referencias

[1] Dirección General de Tráfico (2022): ”Conducir con sueño o cansancio”. https://www.dgt.es/muevete-con-seguridad/evita-conductas-de-riesgo/Conducir-con-sueno-o-cansancio.

[2] Navarrete, R.I., Aguirre, M.S. (2013): ”Cambios Fisiológicos en el Sueño”, *Revista Ecuatoriana de Neurología*, 22(1-3).

[3] Becerril, E.J.L., Juárez, E.T. (2021): ”Una red neuronal para la detección de somnolencia en conductores”, *Revista Digital Universitaria (RDU)*, 22(6).

[4] Vicente, J., Laguna, P., Bartra, A., Bailón, R. (2016): ”Drowsiness detection using heart rate variability”, *Med Biol Eng Comput*, 54, 927–937.

[5] Dinges, D.F., Grace, R. (1998): ”PERCLOS: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance”, US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006.

[6] Li, G., Lee, B.L., Chung, W.Y. (2015): ”Smartwatch-based wearable EEG system for driver drowsiness detection”, *IEEE Sensors Journal*, 15(12), 7169-7180.

[7] Lee, B.L., Lee, B.G., Chung, W.Y. (2016): ”Standalone wearable driver drowsiness detection system in a smartwatch”, *IEEE Sensors journal*, 16(13), 5444-5451.

[8] Pratama, B.G., Ardiyanto, I., Adji, T.B. (2017): ”A review on driver drowsiness based on image, bio-signal, and driver behavior”, *IEEE International Conference on Science and Technology-Computer (ICST)*, 70-75.

[9] Leng, L.B., Giin, L.B., Chung, W.Y. (2015): ”Wearable driver drowsiness detection system based on biomedical and motion sensors”, *IEEE Sensors*, 1-4.

[10] Budak, U., Bajaj, V., Akbulut, Y., Atila, O., Sengur, A. (2019): ”An effective hybrid model for EEG-based drowsiness detection”, *IEEE sensors journal*, 19(17), 7624-7631.

[11] Lee, B., Lee, B., Chung, W. (2015): ”Wristband-Type Driver Vigilance Monitoring System Using Smartwatch”, *IEEE Sensors Journal*, 15(10), 5624-5633.

[12] Li, G., Lee, B.L., Chung, W.Y. (2015): ”Smartwatch-Based Wearable EEG System for Driver Drowsiness Detection”, *IEEE Sensors Journal*, 15(12), 7169-7180.

[13] Ramzan, M., Khan, H.U., Awan, S.M., Ismail, A., Ilyas, M., Mahmood, A. (2019): ”A Survey on State-of-the-Art Drowsiness Detection Techniques”, *IEEE Access*, 7, 61904-61919.

[14] Purnamasari, P.D. , Hazmi A.Z. (2018): ”Heart Beat Based Drowsiness Detection System for Driver”, *International Seminar on Application for Technology of Information and Communication*, 585-590.

[15] Tateno, S., Guan, X., Cao, R., Qu, Z. (2018): ”Development of Drowsiness Detection System Based on Respiration Changes Using Heart Rate Monitoring”, *57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 1664-1669.

[16] Jo, S. H., Kim, J. M., Kim, D. K.: ”Heart Rate Change While Drowsy Driving”, en *Journal of Korean medical science*, 34(8), e56, 2019.

[17] Fujiwara, K. et al (2019): ”Heart Rate Variability-Based Driver Drowsiness Detection and Its Validation With EEG”, *IEEE Transactions on Biomedical Engineering*, 66(6), 1769-1778.

[18] Vicente, J., Laguna, P., Bartra, A., Bailón, R. (2016): ”Drowsiness detection using heart rate variability”, *Medical & biological engineering & computing*, 54(6), 927-37. Epub 2016 Jan 16. PMID: 26780463.

[19] Buendia, R., Forcolin, F., Karlsson, J., Arne Sjöqvist, B., Anund, A., Candefjord, S. (2019): ”Deriving heart rate variability indices from cardiac monitoring-An indicator of driver sleepiness”, 20(3), 249-254. PMID: 30978124.

[20] Awais, M., Badruddin, N., Drieberg, M. (1991): ”A Hybrid Approach to Detect Driver Drowsiness Utilizing Physiological Signals to Improve System Performance and Wearability”, *Sensors (Basel)*, 17(9).

# Evolución de la librería QuantumSolver para el desarrollo cuántico

Daniel Escánez-Expósito
Universidad de La Laguna
Tenerife, Spain
jdanielescanez@gmail.com

Pino Caballero-Gil
Universidad de La Laguna
Tenerife, Spain
pcaballe@ull.edu.es

Francisco Martín-Fernández
IBM Research
NY, USA
paco@ibm.com

***Resumen*—Se expone en este trabajo el estado actual del *toolset* cuántico *QuantumSolver*, desarrollado íntegramente por los autores de este artículo, bajo la licencia *opensource* MIT. Dicha herramienta incluye varios algoritmos con distintas funcionalidades, como el algoritmo de Grover, el teletransporte cuántico, el protocolo de codificación superdensa, la generación de números aleatorios, la resolución de los problemas de Deutsch-Jozsa y Bernstein-Vazirani, y el protocolo BB84 de distribución de claves cuánticas. Se describen aquí los principales detalles de la implementación del *toolset*, así como algunas conclusiones obtenidas de la investigación realizada sobre sus funcionalidades.**

***Index Terms*—Computación cuántica, Qiskit, Toolset cuántico, Grover, Teletransporte cuántico, Codificación superdensa, Números aleatorios, Deutsch-Jozsa, Bernstein-Vazirani, Criptografía cuántica, Protocolo BB84**

## I. Introducción

La computación cuántica y tecnologías relacionadas han adquirido un notable interés en los últimos años, experimentando el tema un gran auge hoy en día. Su utilidad para romper los algoritmos criptográficos actuales ha generado la ya imperiosa necesidad de actualización y sustitución de protocolos seguros para las comunicaciones. Por otra parte, dadas algunas de sus curiosas propiedades, está claro que el fomento de este modelo de computación generará importantes avances tecnológicos para el conjunto de la sociedad [1].

El desarrollo del *toolset opensource* de algoritmos cuánticos *QuantumSolver*, alojado en un repositorio público en GitHub con licencia MIT [2], persigue la abstracción y encapsulamiento sencillos de *software* cuántico con diferentes funcionalidades. Entre las librerías que han implementado total o parcialmente los algoritmos y protocolos tratados en este trabajo destacan [3] y [4], si bien ninguna de ellas contempla las mismas funcionalidades, herramientas y algoritmos de la presente propuesta. Concretamente, el principal objetivo de este trabajo es el fomento del uso de las tecnologías cuánticas, mediante la evolución y mejora del *toolset QuantumSolver* [5].

### I-A. Accesibilidad

*QuantumSolver* está dirigido tanto a usuarios totalmente ajenos a la informática que, por ejemplo, deseen obtener números aleatorios gracias a la computación cuántica; como a programadores experimentados que, por ejemplo, aspiren a contribuir en la implementación de esta librería. Debido a que la propuesta pretende satisfacer a una amplia variedad de público potencial, se permite la ejecución del software por medio de dos interfaces diferentes: Interfaz por Línea de Comandos (véase Fig. (1)); e Interfaz Web (véase Fig. (2)).

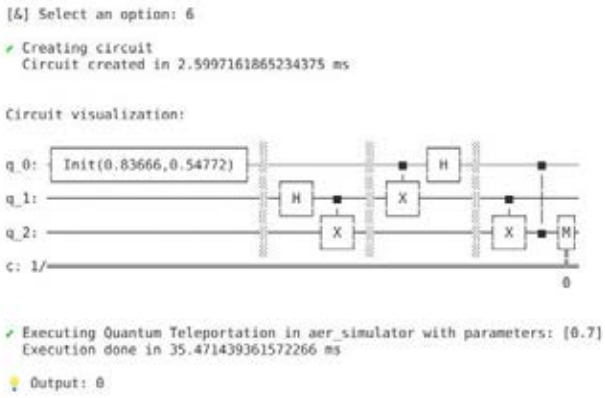



Figura 1. Interfaz por Línea de Comandos mostrando resultados del teletransporte cuántico

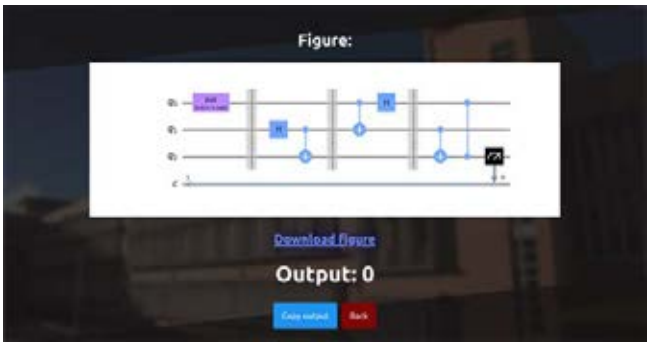



Figura 2. Interfaz Web mostrando resultados del teletransporte cuántico

Esta última cuenta con un *backend* en Python3, utilizando el *framework* Flask, y un *frontend* usando TypeScript, React, HTML5 y CSS. Esta última está más orientada al público general, y permite que cualquier usuario pueda ejecutar algoritmos cuánticos en *hardware* cuántico real, sin tener ningún tipo de experiencia previa con la programación informática.

En ambas interfaces, el programa principal de *QuantumSolver* solicita por pantalla el *API token* de IBM o permite la alternativa de ejecutar el modo invitado. En cualquier caso, como se observa en la Fig. (3) y en la Fig. (4), se despliega un menú que contiene las opciones de visualización y selección de los *backends* y algoritmos disponibles. Una vez elegido el algoritmo, también da la opción de introducir los parámetros ligados al mismo. Cuando *backend*, algoritmo y parámetros ya han sido establecidos, se despliegan dos opciones más: Ejecutar el algoritmo una única vez y obtener el resultado, además de una representación gráfica del circuito; o ejecutar

```
QuantumSolver (Guest Mode)
==========================

[1] See available Backends
[2] See available Algorithms
[3] Select Backend
        Current Backend: aer_simulator
[4] Select Algorithm
        Current Algorithm: Grover's Algorithm (2 Qubits)
[5] Select Parameters
        Current Parameters: ['10']
[6] Run Algorithm
[7] Experimental mode
[0] Exit

[&] Select an option:
```

Figura 3. Interfaz por Línea de Comandos mostrando el menú principal


Figura 4. Interfaz Web mostrando el menú principal

el algoritmo varias veces, para observar el comportamiento del mismo representado en un histograma generado. A esta última opción se le ha denominado modo experimental.

### *I-B. Responsabilidad de las entidades desarrolladas*

Gracias a *Qiskit*, el SDK de código abierto de IBM para trabajar con ordenadores cuánticos a nivel de pulsos, circuitos y módulos de aplicación [6]; se ha podido desarrollar en Python3 la librería cuántica *QuantumSolver*. Cuenta con dos componentes principales que se describen a continuación.

*QExecute* es el motor de ejecución de *QuantumSolver*. Este se encarga de la autenticación contra los servicios de IBM, que ofrecen acceso a su *hardware* (tanto *hardware* cuántico real como simuladores) por medio de un *API token* de *"IBM Quantum Experience"* [7] [8]. Además cuenta con un modo invitado para no generar la inevitable necesidad al usuario de obtener el *token* creando una cuenta en *IBM Quantum*. En este modo solo se permite ejecutar haciendo uso del simulador local *'aer_simulator'*, por lo que no se podrá utilizar el *hardware* cuántico real proporcionado por IBM. *QExecute* cuenta con métodos para la visualización del listado de los *backends* disponibles y la selección del deseado para realizar la ejecución. Además, es el componente encargado de realizar la propia ejecución de los circuitos cuánticos.

*QAlgorithmManager* es el gestor de algoritmos cuánticos de *QuantumSolver*. Se encarga de agrupar y listar todos los algoritmos disponibles, además de seleccionar el que se desea ejecutar. También es responsable de gestionar los argumentos de los diferentes algoritmos y del intercambio de información entre estos y el programa principal.

*QAlgorithm* es la entidad que se corresponde con un algoritmo cuántico. Se trata de una clase abstracta que puede servir como plantilla para añadir de manera intuitiva un nuevo algoritmo a la librería. Cualquier entidad válida derivada de esta representa un algoritmo que puede ejecutar *QuantumSolver*. Estas entidades, siguiendo la plantilla de *QAlgorithm*, contienen información relevante sobre el algoritmo: el nombre, la descripción, los parámetros, la manera en la que se debe analizar y tratar el resultado de la ejecución del circuito, y cómo se deben interpretar y comprobar los diferentes parámetros que sean introducidos como una lista de cadenas de texto. El método principal de la entidad es el de la generación parametrizada del circuito cuántico correspondiente al algoritmo.

## II. Implementación del Algoritmo de Grover

El algoritmo de Grover [9] demuestra la superior capacidad de velocidad en la búsqueda de bases de datos de un ordenador cuántico sobre un ordenador clásico. Puede acelerar cuadráticamente la resolución de un problema de búsqueda no estructurada, y además sirve como técnica general o subrutina para obtener mejoras cuadráticas en el tiempo de ejecución de una variedad de algoritmos. El procedimiento que logra esta mejora se denomina la amplificación de la amplitud.

En una lista no ordenada de $N$ elementos, clásicamente se deben consultar uno a uno hasta encontrar el buscado. Para encontrar ese elemento marcado, se tiene que comprobar una media de $N/2$ elementos, $N$ en el peor caso. Con la técnica cuántica de la amplificación de la amplitud, se logra encontrar el elemento en $\sqrt{N}$ pasos. Esto resulta en un aumento cuadrático de la velocidad, lo que supone un ahorro de tiempo considerable para encontrar elementos marcados en listas largas. Además, el algoritmo no utiliza la estructura interna de lista, lo que lo hace genérico y proporciona inmediatamente una aceleración cuántica cuadrática para un alto número de problemas clásicos.

Para implementar la base de datos [10] se puede recurrir a un oráculo $U_\omega$ descrito en notación matricial por la Ec. (1). Se debe tener en cuenta que $f(x) = 0$ si $x$ no es un elemento buscado $(x \neq w)$ y $f(x) = 1$ si $x$ es un elemento buscado $(x = w)$.

$$U_\omega = \begin{bmatrix} (-1)^{f(0)} & 0 & \cdots & 0 \\ 0 & (-1)^{f(1)} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & (-1)^{f(2^n-1)} \end{bmatrix} \tag{1}$$

Véase un ejemplo para 2 cúbits, marcando el elemento "10" en la Ec. (2).

$$U_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \leftarrow \omega = 10 \tag{2}$$

Una vez aclarado el tipo de dato para la representación de la base de datos, se procede a explicar la técnica de la amplificación de la amplitud.

1. Se establece el vector $|s\rangle = H^{\otimes n}|0\rangle^n$, dando lugar a una superposición uniforme. Es por ello que en la implementación realizada se han inicializado los $n$

cúbits y se les ha aplicado una puerta Hadamard [11] [12] a cada uno.

2. Se aplica el oráculo de reflexión $U_f$ al estado $|s\rangle$. La amplitud de $|w\rangle$ queda negativa al reflectar $|s\rangle$ sobre $|s'\rangle$. En la implementación, se consigue crear un oráculo $U_f$ aplicando una puerta *CZ* entre los dos cúbits. Además, en el cúbit contrario a aquellos cúbits que están establecidos a 0 en el elemento buscado, se debe aplicar una puerta $S$ antes y después de la *CZ*.
3. Se aplica una reflexión adicional $U_s = 2|s\rangle\langle s| - 1$ para completar la transformación. Para implementar esta reflexión, se debe aplicar una puerta Hadamard y una $Z$ a ambos cúbits. A continuación una puerta *CZ* entre ellos y finalizar con una puerta Hadamard en ambos.

La transformación realizada $U_sU_f$ (pasos 2 y 3) consigue rotar $|s\rangle$ más cerca de $|w\rangle$. Se puede denotar la aplicación de $t$ iteraciones sobre estos pasos como $|\psi_t\rangle = (U_sU_f)^t|s\rangle$. Un ejemplo de circuito generado por la implementación parametrizada, con el elemento buscado "10", se puede observar en la Fig. (5).

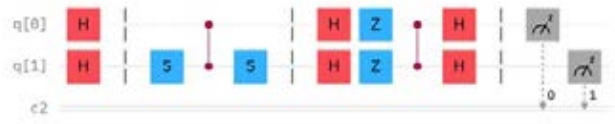



Figura 5. Ejemplo de circuito completo del algoritmo de Grover de 2 cúbits

## III. Implementación del Teletransporte Cuántico

El Teletransporte Cuántico utiliza dos bits clásicos para transmitir un cúbit. Según el teorema de no clonación [13], es imposible clonar un cúbit. Solo es posible clonar estados clásicos, no superposiciones. Al transmitir un cúbit de un emisor a un receptor, el emisor lo pierde; es por ello que se denomina Teletransporte Cuántico.

A continuación se describe la implementación realizada [14], que simula la comunicación cuántica haciendo uso de circuitos cuánticos.

En primer lugar, emisor y receptor comparten un par entrelazado de cúbits. Entonces, el emisor realiza una serie de operaciones en su cúbit y envía los resultados al receptor por medio de dos bits clásicos. Tras recibir esta información, el receptor es capaz de realizar operaciones para obtener el cúbit del emisor.

El par entrelazado se puede conseguir aplicando una puerta Hadamard a un cúbit y posteriormente hacerlo control de una *CNOT* que tiene como objetivo otro cúbit inicializado a $|0\rangle$. A continuación, el emisor procede a aplicar otra *CNOT* con control en el cúbit que desea enviar y objetivo en aquel que posee del par entrelazado. Tras esto, aplica una última puerta Hadamard al cúbit al que intervino con el control de la *CNOT* y mide ambos cúbits, enviando los resultados al receptor. Este, al recibir los dos bits clásicos, seguirá la regla vinculante a los valores obtenidos y aplicará las operaciones correspondientes a su cúbit del par entrelazado, decodificando las proyecciones pertinentes:

- $00 \rightarrow$ No hacer nada.
- $01 \rightarrow$ Aplicar una puerta $X$.
- $10 \rightarrow$ Aplicar una puerta $Z$.
- $11 \rightarrow$ Aplicar una puerta $Z$ y después una puerta $X$.

Tras hacer esto, el receptor tendrá el cúbit que el emisor quiso enviar.

Para el desarrollo del circuito se han seguido las consideraciones que compatibilizan su implementación con su ejecución en hardware real de IBM [15]. Además, para brindar una mayor interacción con el usuario, se ha parametrizado logrando la elección de la probabilidad de medir 0 en el cúbit a transmitir. Esto se puede comprobar en el modo experimental, al ejecutar el circuito un número elevado de veces, obteniendo aproximadamente el porcentaje de mediciones en 0 solicitado por el usuario. Este proceso se realiza en el módulo *initialize* presente en la Fig. (6), donde se ilustra el circuito completo implementado.

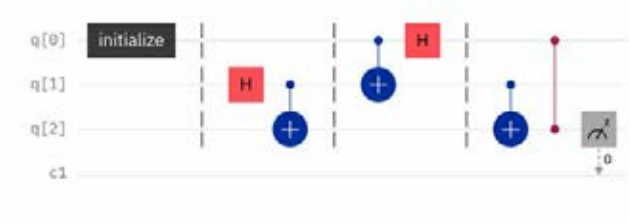



Figura 6. Circuito implementado del teletransporte cuántico

## IV. Implementación del Protocolo de Codificación Superdensa

De manera similar al Teletransporte Cuántico, el protocolo de Codificación Superdensa [16] es capaz de transmitir dos bits clásicos por medio de un único cúbit de comunicación. Los dos algoritmos son parecidos dado que, en cierta medida, el emisor y el receptor "intercambian sus equipos".

La implementación sigue los siguientes pasos:

1. Emisor ($A$) y receptor ($B$) comparten un par entrelazado de cúbits. Para generarlo, primeramente se inicializan ambos a $|0\rangle$.
$$|00\rangle = |0\rangle_A \otimes |0\rangle_B \tag{3}$$
Se aplica una puerta Hadamard al cúbit del emisor, dejándolo en estado de superposición.
$$|+0\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \tag{4}$$
Por último, se aplica una *CNOT* con control en el cúbit del emisor y objetivo en el del receptor.
$$\text{CNOT}\tfrac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{5}$$
2. Cada cúbit es enviado a su correspondiente dueño. La meta final del emisor es enviar dos bits clásicos por medio de su cúbit. Para ello se aplican las reglas ilustradas en la Tabla (I). Dependiendo del mensaje a codificar, aplica a su cúbit las puertas indicadas.
3. El receptor recibe el cúbit del emisor y le practica unas operaciones de restauración para "invertir" el proceso inicial del entrelazamiento. Estas operaciones son una *CNOT* con control en el cúbit recibido por el emisor y objetivo en el propio del receptor, y posteriormente

una puerta Hadamard al del emisor. En la Tabla (II) se ilustran los casos posibles. Al medir los cúbits obtendrá el mensaje del emisor.

Tabla I
NORMAS DE CODIFICACIÓN EN LA CODIFICACIÓN SUPERDENSA

| Mensaje a enviar | Puerta aplicada | Estado resultante |
|---|---|---|
| 00 | $I$ | $\|00\rangle + \|11\rangle$ |
| 01 | $X$ | $\|10\rangle + \|01\rangle$ |
| 10 | $Z$ | $\|00\rangle - \|11\rangle$ |
| 11 | $ZX$ | $-\|10\rangle + \|01\rangle$ |

Tabla II
CASOS DE DECODIFICACIÓN EN LA CODIFICACIÓN SUPERDENSA

| Mensaje recibido | Tras aplicar *CNOT* | Tras aplicar Hadamard |
|---|---|---|
| $\|00\rangle + \|11\rangle$ | $\|00\rangle + \|10\rangle$ | $\|00\rangle$ |
| $\|10\rangle + \|01\rangle$ | $\|11\rangle + \|01\rangle$ | $\|01\rangle$ |
| $\|00\rangle - \|11\rangle$ | $\|00\rangle - \|10\rangle$ | $\|10\rangle$ |
| $-\|10\rangle + \|01\rangle$ | $-\|11\rangle + \|01\rangle$ | $\|11\rangle$ |

Se ha implementado un algoritmo parametrizado que permite simular el protocolo de Codificación Superdensa para cualquier mensaje binario de longitud dos. Se puede observar un ejemplo de circuito generado para el mensaje "11" en la Fig. (7).


Figura 7. Ejemplo de circuito del protocolo de codificación superdensa

## V. IMPLEMENTACIÓN DE LA GENERACIÓN DE NÚMEROS ALEATORIOS

El algoritmo cuántico *QRand* implementado en *Quantum-Solver* recibe como parámetro un número natural $(n)$ y permite generar un circuito de cuya ejecución resulta un número aleatorio entre 0 y $2^n - 1$. Su funcionamiento se basa en la inicialización de $n$ cúbits, por defecto a $|0\rangle$; la aplicación de una puerta lógica Hadamard a cada uno, para generar un estado de superposición en el que se tenga la misma probabilidad de medir 0 o 1 (ver Ec. (6)); y finalmente la medición del resultado, haciendo colapsar cada cúbit en un estado aleatorio e interpretándose como un número binario. Se puede ver un ejemplo de circuito de generación de números aleatorios para 4 cúbits en la Fig. (8).

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{6}$$

## VI. IMPLEMENTACIÓN DEL ALGORITMO DE DEUTSCH-JOZSA

El algoritmo de Deutsch-Jozsa [17] es un claro ejemplo de la ventaja existente entre la ejecución de ciertos algoritmos cuánticos y la del mejor algoritmo clásico para resolver el mismo problema.

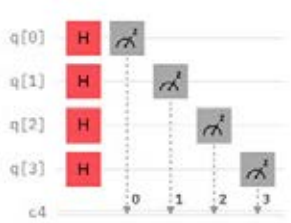



Figura 8. Circuito de generación de números aleatorios para 4 cúbits

Dada una función oculta $f$ (véase Ec. (7)), que ante una cadena binaria arbitraria devuelve 0 o 1, se debe determinar si $f$ se trata de una función constante o balanceada (se garantiza que es de alguno de estos dos tipos). Una función constante es aquella que ante cualquier entrada, devuelve siempre 0 o siempre 1. Por el contrario, una función balanceada devuelve para exactamente la mitad de las cadenas 0, y para la otra mitad 1.

$$f(\{x_0, x_1, x_2, ..., x_{n-1}\}) \in \{0,1\}, x_i \in \{0,1\} \tag{7}$$

El número de posibles cadenas binarias de entrada de longitud $n$ es $2^n$, por lo que clásicamente se podría evaluar la función hasta encontrar dos valores distintos o hasta llegar a $2^{n-1}+1$ entradas (en el peor caso). Terminado el proceso, si en las evaluaciones obtenidas se ha encontrado un solo valor distinto a los demás, se determinará $f$ balanceada y si en todas las evaluaciones ha resultado el mismo valor, se determina $f$ constante.

Utilizando la computación cuántica, es posible resolver el problema con un $100\,\%$ de confianza tras una única llamada a $f$. En este caso, la función se implementa gracias a un oráculo cuántico $U_f$ que sigue la expresión de la Ec. (8). El símbolo $\oplus$ hace referencia a la suma módulo 2.

$$U_f(|x\rangle\,|y\rangle) = |x\rangle\,|y \oplus f(x)\rangle \tag{8}$$

La implementación realizada del algoritmo cuántico de Deutsch-Jozsa [18], recibe dos parámetros: el tipo de oráculo deseado (constante o balanceado) y el número $n$ de cúbits que se utilizan como tamaño de la entrada de $f$. Una vez introducidos estos parámetros, se genera un circuito de $n+1$ cúbits. De ellos, $n$ conforman el primer registro para codificar la entrada (formada por $n$ cúbits con valor $|0\rangle$); y el restante establece el segundo registro, para la salida del oráculo cuántico (inicializado con el valor $|1\rangle$, obtenido al aplicar una puerta cuántica *X* [12] a un cúbit que por defecto tiene el valor $|0\rangle$). Además, se deberán aplicar puertas lógicas Hadamard a los $n+1$ cúbits antes y después del oráculo; excepto a la salida del mismo, que no lo precisará dado que no será medida. La Fig. (9) muestra un ejemplo de implementación del algoritmo Deutsch-Jozsa para 3 cúbits.

En lo que al oráculo $U_f$ se refiere, deberá su implementación al tipo de función $f$ a implementar. Si se opta por $f$ constante, la implementación es bastante sencilla, se ha decidido elegir al azar una de estas dos funciones:

- $\forall x, f(x) = 0$, aplicar una puerta $I$ (Identidad) al segundo registro.


Figura 9. Implementación del algoritmo Deutsch-Jozsa para 3 cúbits

- $\forall x, f(x) = 1$, aplicar una puerta $X$ (comparable con el *NOT* clásico) al segundo registro.

Para la implementación de una función $f$ balanceada, se deben utilizar $n$ puertas *CNOT* con control en cada uno de los cúbits del primer registro y con objetivo en aquel cúbit perteneciente al segundo registro.

Se pueden aplicar puertas $X$ al azar antes y después de la aplicación de las puertas *CNOT*, para que el comportamiento de este oráculo de función balanceada no sea predecible.

Un ejemplo de implementación del algoritmo Deutsch-Jozsa para 3 cúbits con oráculo transparente se puede ver en la Fig. (10).

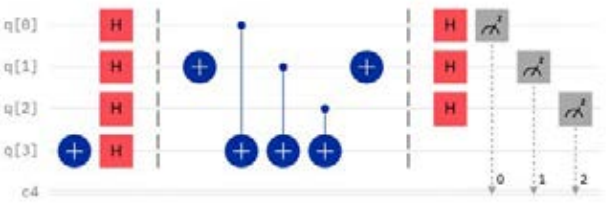



Figura 10. Algoritmo Deutsch-Jozsa para 3 cúbits con oráculo transparente

La inicialización de los cúbits se muestra en la Ec. (9), para establecer los valores indicados a los cúbits; y en la Ec. (10), tras aplicar las puertas Hadamard por primera vez.

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle \tag{9}$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle \left(|0\rangle - |1\rangle\right) \tag{10}$$

La aplicación del oráculo $U_f$ sobre los cúbits inicializados se ilustra en la Ec. (11). Se llega a la conclusión ilustrada, al considerar que $\forall x, f(x) \in \{0, 1\}$.

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle) \end{aligned} \tag{11}$$

A continuación, el segundo registro de la salida del oráculo y por tanto, su correspondiente cúbit puede ser ignorado. Se aplican $n$ puertas Hadamard a cada cúbit correspondiente con el primer registro de salida del oráculo, y el comportamiento queda reflejado en la Ec. (12). En esta, $x \cdot y = x_0y_0 \oplus x_1y_1 \oplus \ldots \oplus x_{n-1}y_{n-1}$ resultando la suma en módulo 2 del producto bit a bit.

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\sum_{y=0}^{2^n-1} (-1)^{x\cdot y}|y\rangle\right] \\ &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)}(-1)^{x\cdot y}\right] |y\rangle \end{aligned} \tag{12}$$

Al medir el primer registro, se evaluará en una cadena de tamaño $n$ formada por todos los bits a 0, si $f$ es constante; o en otro caso, si es balanceado. La probabilidad de medir la función constante, es decir, de realizar la medición de todos los bits a 0, viene determinada por la Ec. (13). Esta se toma el valor 1 con $f$ constante y 0 con $f$ balanceada.

$$p_{medir}(|0\rangle^{\otimes n}) = |\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|^2 \tag{13}$$

## VII. Implementación del Algoritmo de Bernstein-Vazirani

El algoritmo cuántico de Bernstein-Vazirani [20], se trata de un caso particular del algoritmo Deutsch-Jozsa. La implementación incluida en *QuantumSolver* recibe como parámetro una clave formada por una cadena binaria de tamaño $n$, que se utiliza para codificar un oráculo que brinda información acerca de dicha clave. Concretamente, ante una cadena candidata, la información que devuelve el oráculo es la confirmación de si el número de coincidencias de 1 entre las cadenas clave y candidata es par o impar.

$$f_s(x) = (s * x)(mod\ 2) \tag{14}$$

La Ec. (14) refleja que el oráculo realiza el producto binario entre las parejas de bits de la clave a adivinar y la cadena candidata, y luego le aplica la suma módulo 2 a los $n$ bits resultantes. Clásicamente, se podría resolver este problema con $n$ consultas al oráculo como muestra la Ec. (15).

$$\begin{cases} f_s(100\ldots0) = s_0 \\ f_s(010\ldots0) = s_1 \\ f_s(001\ldots0) = s_2 \\ \ldots \\ f_s(000\ldots1) = s_{n-1} \end{cases} \tag{15}$$

En el caso cuántico se necesita solo una consulta al oráculo, que devolverá correctamente la clave con un 100 % de probabilidad (sin contar con posibles errores de ruido generados por el *hardware*).

La implementación realizada [21] es muy similar a la del algoritmo Deutsch-Jozsa, la única diferencia reside en el oráculo. Internamente, este debe implementarse aplicando puertas *CNOT* con control en aquellos cúbits que se correspondan con los bits de la clave que estén a 1, y objetivo en la salida del oráculo.

La caracterización del oráculo, aplicado sobre la cadena $x$ candidata, se muestra en la Ec. (16).

$$|x\rangle \xrightarrow{f_s} (-1)^{s\cdot x}|x\rangle \tag{16}$$

La transformación realizada de los $n$ cúbits que codifican la entrada $|00\ldots0\rangle$ al aplicarles las puertas Hadamard de la inicialización se muestra en la Ec. (17).

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \tag{17}$$

La aplicación del oráculo cuántico se muestra en la Ec. (18).

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \xrightarrow{f_a} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x}|x\rangle \tag{18}$$

El paso final, aplicando a cada cúbit anterior una puerta Hadamard, se muestra en la Ec. (19).

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x}|x\rangle \xrightarrow{H^{\otimes n}} |a\rangle \tag{19}$$

Se observa que, al aplicar las operaciones descritas, el resultado es la clave codificada en el oráculo.

## VIII. Implementación del Protocolo BB84

Se ha desarrollado una implementación del protocolo criptográfico cuántico BB84 [22], usando la librería cuántica de *QuantumSolver*. En este caso, se ha diseñado un programa principal externo a la librería que se apoya en los módulos de la misma.

Para ello se ha implementado una entidad principal *Participant* y sus entidades derivadas: *Sender* y *Receiver*. Una instancia de la clase *Sender* puede comunicarse con otra de la clase *Receiver* asegurándose de que la comunicación es privada, gracias al protocolo implementado. Para cerciorarse de la seguridad en la comunicación, se requiere de la generación de una libreta de un solo uso (*One Time Pad*), que solo comparten el emisor y el receptor legítimo. La simulación del canal cuántico se describe mediante la utilización de circuitos cuánticos de Qiskit [23]. La única diferencia entre las entidades *Sender* y *Receiver* es que la primera tiene un método para enviar un mensaje (inicializando un circuito cuántico) y la segunda para recibirlo (añadiendo la fase de medición al circuito). La clase base *Participant* contiene los métodos para la generación y muestra de valores, ejes, claves y libretas de un solo uso, entre otros.

La implementación del protocolo (fases, entidades, condiciones y conclusiones) se puede encontrar documentada en sus respectivos ficheros de código fuente [24].

## IX. Conclusiones

La librería *QuantumSolver* para desarrollo cuántico permite, de manera sumamente accesible, la ejecución de algoritmos en *hardware* cuántico real proporcionado por IBM. También aporta una sencilla arquitectura de entidades, respaldada por una batería de pruebas unitarias (*unitary test*), que facilita enormemente la adición de nuevos algoritmos. Precisamente, el objetivo principal en cuanto a la continuación del desarrollo de la propuesta es ampliar el número de algoritmos disponibles. Gracias a la misma, se pueden implementar simulaciones más complejas, como la realizada del protocolo de criptografía cuántica BB84. Se pretende la futura implementación de los protocolos E91, B92 y del algoritmo de Shor.

## Agradecimientos

Esta investigación ha sido posible gracias a la Cátedra de Ciberseguridad Binter - Universidad de La Laguna.

## Referencias

[1] E. Gibney, "Hello quantum world! Google publishes landmark quantum supremacy claim", Nature, vol. 574, no. 7779, pp. 461-462, 2019 [Online]. Available: https://www.nature.com/articles/d41586-019-03213-z. [Accessed: 03-Jun-2022].

[2] J. D. Escánez, "QuantumSolver". [Online]. Available: https://github.com/alu0101238944/quantum-solver/. [Accessed: 03-Jun-2022].

[3] A. G. Tudorache, V. I. Manta and S. Caraiman, "Implementation of the Bernstein-Vazirani Quantum Algorithm Using the Qiskit Framework", Bulletin of the Polytechnic Institute of Iași. Electrical Engineering, Power Engineering, Electronics Section, 67(2), 31-40, 2021.

[4] A. Warke, B. K. Behera and P. K. Panigrahi, "Experimental realization of three quantum key distribution protocols", Quantum Information Processing, 19(11), 1-15, 2020.

[5] J. D. Escánez-Expósito, P. Caballero-Gil and F. Martín-Fernández, "QuantumSolver: A quantum tool-set for developers", International Conference on Security And Management. Springer Nature. 2022.

[6] IBM, "Qiskit". [Online]. Available: https://qiskit.org/. [Accessed: 03-Jun-2022].

[7] IBM, "IBM Quantum", May-2016. [Online]. Available: https://quantum-computing.ibm.com/. [Accessed: 03-Jun-2022].

[8] IBM, "User account - IBM Quantum", May-2016. [Online]. Available: https://quantum-computing.ibm.com/composer/docs/iqx/manage/account/#account-overview. [Accessed: 03-Jun-2022].

[9] L. K. Grover, "A fast quantum mechanical algorithm for database search", Proceedings of the twenty-eighth annual ACM Symposium on Theory Of Computing, 1996.

[10] IBM, "Grover's Algorithm" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/grover.html. [Accessed: 03-Jun-2022].

[11] IBM, "Learn Quantum Computation using Qiskit" [Online]. Available: https://qiskit.org/textbook/preface.html. [Accessed: 03-Jun-2022].

[12] IBM, "Single qubit Gates" [Online]. Available: https://qiskit.org/textbook/ch-states/single-qubit-gates.html. [Accessed: 03-Jun-2022].

[13] W. Wootters and W. Zurek, "A single quantum cannot be cloned", Nature, vol. 299, no. 5886, pp. 802-803, 1982.

[14] IBM, "Quantum Teleportation" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/teleportation.html. [Accessed: 03-Jun-2022].

[15] IBM, "Quantum Teleportation - 5. Teleportation on a Real Quantum Computer" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/teleportation.html#5.1-IBM-hardware-and-Deferred-Measurement-. [Accessed: 03-Jun-2022].

[16] IBM, "Superdense Coding" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/superdense-coding.html. [Accessed: 03-Jun-2022].

[17] D.Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation", Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, vol. 439, no. 1907, pp. 553-558, 1992.

[18] IBM, "Deutsch-Jozsa Algorithm" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html. [Accessed: 03-Jun-2022].

[19] IBM, "Phase Kickback" [Online]. Available: https://qiskit.org/textbook/ch-gates/phase-kickback.html. [Accessed: 03-Jun-2022].

[20] E. Bernstein and U. Vazirani, "Quantum Complexity Theory", SIAM Journal on Computing, vol. 26, no. 5, pp. 1411-1473, 1997 [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.655.1186&rep=rep1&type=pdf. [Accessed: 03-Jun-2022].

[21] IBM, "Bernstein-Vazirani Algorithm" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html. [Accessed: 03-Jun-2022].

[22] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," arXiv, 1984, doi: 10.48550/ARXIV.2003.06557. [Online]. Available: https://arxiv.org/abs/2003.06557 [Accessed: 03-Jun-2022].

[23] IBM, "Quantum Key Distribution" [Online]. Available: https://qiskit.org/textbook/ch-algorithms/quantum-key-distribution.html. [Accessed: 03-Jun-2022].

[24] J. D. Escánez, "BB84 Implementation with QuantumSolver". [Online]. Available: https://github.com/alu0101238944/quantum-solver/tree/main/src/bb84. [Accessed: 03-Jun-2022].

# Minimizing the total number of shadows in secret sharing schemes based on extended neighborhood coronas

Raúl M. Falcón
Dept. Matemática Aplicada I
Universidad de Sevilla
Avda. Reina Mercedes, 4A
Sevilla, Spain
rafalgan@us.es

N. Mohanapriya
PG and Research Dept. of Mathematics
Kongunadu Arts and Science College
Coimbatore-641 029
Tamil Nadu, India
n.mohanamaths@gmail.com

V. Aparna
PG and Research Dept. Mathematics
Kongunadu Arts and Science College
Coimbatore-641 029
Tamil Nadu, India
aparnav18794@gmail.com

***Abstract*—Every $t$-dynamic proper $n$-coloring of a graph $G$ describes a shadow allocation of any $(n, t+1)$-threshold secret sharing scheme based on $G$ so that, after just one round of communication, each participant can either reconstruct the secret, or get a distinct shadow from each neighbor. The dynamic coloring problem for $G$ consists of finding the minimum number of distinct shadows in which the secret has to split for ensuring that condition. This paper outlines the recently published authors' article *Optimal shadow allocations of secret sharing schemes arisen from the dynamic coloring of extended neighborhood coronas*, where the dynamic chromatic problem is solved for any extended neighborhood corona with center path or star. It enables one to model communication networks whose average path lengths are small even after an asymptotic growth of the center and/or outer graphs.**

***Index Terms*—Threshold secret sharing scheme, dynamic coloring, extended neighborhood corona.**

## I. Introduction

In 1979, Blakley [1] and Shamir [2] introduced independently the notion of $(n, t)$-*threshold secret sharing scheme* as a cryptographic protocol where a dealer splits a secret into $n$ pieces of information or *shadows*, which are distributed among a group of participants. Then, there is a reconstruction phase in which any authorized subgroup sharing at least $t$ distinct shadows suffices to reconstruct the secret, whereas no subgroup sharing less than $t$ shadows can do it. These schemes are particularly relevant in online storage and cloud computing, because they enforce data security among the nodes of the communication network under consideration [3].

This paper deals with threshold secret sharing schemes based on finite, simple and connected undirected graphs, where each node represents a participant of the scheme, and two nodes are adjacent if and only if there exists some proximity relationship among them so that they can cooperate to reconstruct the secret. One round of communication among nodes implies that each participant receives the shadows of her/his neighbors. In 1991, Naor and Roth [4] considered this scheme to split an arbitrary computer file into pieces of information and distribute them among a network of processors so that each node is a memory device. To reconstruct the original file, each device can access to its own memory and those ones of its neighbors. The objective is minimizing the total amount of data stored and ensuring an efficient reconstruction of the original file, even in case of networks failures or attacks [5], [6]. A comprehensive study of the network topology is necessary to design an optimal storage allocation among the nodes [7]. If no two neighbors have the same data, then any shadow allocation among the nodes of a graph on which an $(n, t)$-threshold secret sharing scheme is based constitutes an $n$-proper multicoloring of the graph. Here, each color represents a shadow of the secret. This multicoloring becomes an $n$-proper coloring, whenever each participant has exactly one shadow. We assume this last condition from here on.

In 2017, Kim an Ok [8] proposed a secret sharing scheme based on $t$-*dynamic* $n$-*colorings* of graphs [9], [10]. Given $n$ possible values (colors), such a coloring assigns one color to each vertex of a graph $G$ so that: (1) adjacent vertices have different colors, and (2) the number of different colors among the neighbors of a vertex $v$ is at least $t$, or all different if $v$ has less than $t$ neighbors. The secret sharing scheme is described as follows. Let $\{1, \ldots, n\}$ be the set of colors. Given a secret value $s$, compute shares $\{s_1, \ldots, s_n\}$ for $s$ according to a $(t+1, n)$-threshold secret sharing scheme. Each vertex colored with $i$ gets the share $s_i$. For every vertex $v$ with degree at least $t$, the secret can be recovered from the shares of $v$ and $t$ of its neighbors. Of course, there are other sets of vertices that are qualified to obtain the secret. The idea is that the secret can be recovered by sets of vertices that are close to each other in the network. The $t$-*dynamic chromatic number* $\chi_t(G)$ is the minimum number of shadows in which the secret has to split to ensure this coloring. Computing this number constitutes the $t$-*dynamic coloring problem* of the graph $G$. If $t = 1$, then it coincides with the classical chromatic number $\chi(G)$.

This paper outlines the recently published authors' article [11], which constitutes the first comprehensive study concerning Kim and Ok's secret sharing schemes. It delves into this topic by solving the dynamic chromatic problem for any extended neighborhood corona $G * H$ whose center graph $G$ is either a path or a star. The choice of both types of graphs is due to two main reasons, which are comprehensively analyzed throughout the paper. The first one is that they enable one to model complex networks whose average path lengths remain small even after an asymptotic growth of their centers and/or outer graphs. The second reason is that, if honesty is assumed by everybody, then both types of graphs enable the recon-

struction of the secret by all the participants in two rounds of communications. In the literature, even the dynamic chromatic problem has been dealt with for a wide amount of families of graphs, there exist only some partial results concerning corona products [12] and generalized corona products [13], [14]. The proofs of all the results described in this paper may be found in [11].

## II. Preliminaries

A graph $G$ is any pair formed by a set $V(G)$ of vertices and a set $E(G)$ of edges, so that every edge $vw \in E(G)$ contains two adjacent vertices $v, w \in V(G)$. The cardinalities of $V(G)$ and $E(G)$ are, respectively, the order and the size of the graph. The complete graph $K_n$ is a graph of order $n$, whose vertices are pairwise adjacent. It is a triangle if $n = 3$. A clique of a graph $G$ is any set of vertices of a complete graph within $G$. The clique number $\omega(G)$ is the largest order of any clique of $G$. The neighborhood $N_G(v)$ of a vertex $v \in V(G)$ is the set of vertices that are adjacent to $v$. Its degree $\deg_G(v)$ is the cardinality of this set. A vertex is pendant if it has degree one. The minimum and maximum vertex degrees of the graph $G$ are respectively denoted by $\delta(G)$ and $\Delta(G)$. Further, a path $P_n$, with $n > 2$, is any ordered sequence of adjacent and pairwise distinct vertices $\langle v_1, \ldots, v_n \rangle$. The star $S_n$, with $n > 2$, is a graph formed by $n$ pendant vertices and a center vertex of degree $n$.

In 1970, Frucht and Harary [15] introduced the corona product of center graph $G$, with $V(G) = \{v_1, \ldots, v_n\}$, and outer graph $H$ as the graph resulting from $G$ and $n$ copies of $H$, so that each vertex $v_i$ is joined to every vertex in the $i^{\text{th}}$ copy of $H$. Much more recently, Indulal [16] introduced the neighborhood corona $G \star H$ as the graph resulting from $G$ and $n$ copies of $H$, so that every vertex in $N_G(v_i)$ is joined to every vertex in the $i^{\text{th}}$ copy of $H$. Five years later, Adiga et al. [17] defined the extended neighborhood corona $G * H$ as the graph resulting from $G \star H$ after connecting every vertex in the $i^{th}$ copy of $H$ to every vertex of the $j^{th}$ copy of $H$, whenever $v_i v_j \in E(G)$. Figure 1 illustrates these three products.

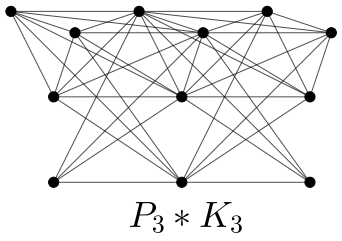



Fig. 1. Example of extended neighborhood corona.

The distance $d_G(v, w)$ between two vertices $v, w \in V(G)$ is the size of any shortest path in $G$ connecting both vertices. The average path length of the graph $G$ is

$$\ell_G := \frac{1}{|V(G)| \cdot (|V(G)| - 1)} \cdot \sum_{\substack{v,w \in V(G) \\ v \neq w}} d_G(v, w).$$

An open triplet in $G$ consists of any three vertices describing a path, but not a triangle. The clustering coefficient of the graph $G$ is

$$\mathcal{C}_G := \frac{3 \cdot T_G}{3 \cdot T_G + \tau_G},$$

where $T_G$ and $\tau_G$ denote, respectively, the number of triangles and open triplets in $G$.

A proper $k$-coloring of the graph $G$ is any map assigning $k$ distinct colors to the set $V(G)$ so that no two adjacent vertices share color. Throughout this paper, we consider the set of colors $\{0, \ldots, k-1\}$. The minimum positive integer $k$ for which this coloring exists is the chromatic number $\chi(G)$. Any such a coloring is said to be optimal. The following lemmas refer to the dynamic coloring, which has been defined in the introductory section.

*Lemma 1 ([9]):* Let $G$ be a simple finite graph and let $t$ be a positive integer. Then, $\min\{t, \Delta(G)\} + 1 \leq \chi_t(G) \leq \chi_{t+1}(G)$. Moreover, $\chi_t(G) \leq \chi_{\Delta(G)}(G)$.

*Lemma 2 ([18]):* Let $n$ and $t$ be two positive integers. Then:

a) If $n > 2$, then $\chi_t(P_n) = \begin{cases} 2, & \text{if } t = 1, \\ 3, & \text{otherwise.} \end{cases}$

b) If $n > 2$, then $\chi_t(S_n) = \min\{n, t\} + 1$.

## III. Extended neighborhood coronas

Let $G * H$ be an extended neighborhood corona of center graph $G$ and outer graph $H$, where $V(G) = \{u_1, \ldots, u_m\}$ and $V(H) = \{v_1, \ldots, v_n\}$. For each pair of positive integers $i \leq m$ and $j \leq n$, let $H^{(i)}$ denote the $i^{\text{th}}$ copy of the graph $H$, and let $v_{i,j}$ denote the copy of the vertex $v_j \in V(H)$ in $H^{(i)}$. Then,

$$\deg_{G*H}(v) =$$

$$= \begin{cases} (n+1) \cdot \deg_G(v), & \text{if } v \in V(G), \\ (n+1) \cdot \deg_G(u_i) + \deg_H(v_j), & \text{if } v = v_{i,j}. \end{cases}$$

The distance between two distinct and non-adjacent vertices $v, w \in \big(\{u_i\} \cup V(H^{(i)})\big) \times \big(\{u_j\} \cup V(H^{(j)})\big)$ is

$$d_{G*H}(v, w) = \begin{cases} 2, & \text{if } i = j, \\ d_G(u_i, u_j), & \text{otherwise.} \end{cases}$$

*Lemma 3:* The extended neighborhood corona $G * H$ satisfies the following assertions.

1) Its order is $mn + m$.
2) Its size is $(n+1)^2 \cdot |E(G)| + m \cdot |E(H)|$.
3) $\delta(G * H) = (n+1) \cdot \delta(G)$.
4) $\Delta(G * H) = (n+1) \cdot \Delta(G) + \Delta(H)$.
5) $T_{G*H} = T_G + m \cdot T_H + 2 \cdot (n+1) \cdot |E(G)| \cdot |E(H)|$.
6) Its number of open triplets is

$$\begin{aligned} \tau_{G*H} =& \tau_G + m \cdot \tau_H + (n^2 + n) \cdot \sum_{v \in V(G)} \deg_G(v)^2 + \\ &+ (n^3 + n^2 + n) \cdot \sum_{v \in V(G)} \binom{\deg_G(v)}{2} + \\ &+ 2mn \cdot |E(G)| \cdot \left( \binom{n}{2} + |E(H)| \right). \end{aligned}$$

7) Its average path length is

$$\ell_{G*H} = \frac{2 \cdot (n^2 - |E(H)|) + \ell_G \cdot (m-1)(n+1)^2}{(n+1)(mn+m-1)}.$$

Based on Lemma 3, the following result describes how the asymptotic behaviour of $\ell_{G*H}$ is equivalent to that one of $\ell_G$.

*Proposition 1:* It is verified that

1) $\lim_{m\to\infty} \ell_{G*H} = \frac{n+1}{n} \cdot \lim_{m\to\infty} \ell_G$.
2) $\frac{\ell_G\cdot(m-1)+1}{m} \leq \lim_{n\to\infty} \ell_{G*H} \leq \frac{\ell_G\cdot(m-1)+2}{m}$.
3) $\lim_{m,n\to\infty} \ell_{G*H} = \lim_{m\to\infty} \ell_G$.

Proposition 1 implies that, if the asymptotic behaviour of $\ell_G$ is unbounded, then only a growth of the outer graph is feasible. But, if $\ell_G$ has a bounded asymptotic behaviour, then an independent growth of both the center and the outer graphs is feasible. In this paper, we illustrate both cases by focusing on extended neighborhood coronas with either a center path graph $P_m$ or a center star graph $S_m$. Notice here that

$$\ell_{P_m} = \frac{m+1}{3} \qquad \text{and} \qquad \ell_{S_m} = \frac{2m}{m+1}.$$

Then, for every graph $H$, Proposition 1 implies that

$$\frac{m^2+2}{3m} \leq \lim_{n\to\infty} \ell_{P_m*H} \leq \frac{m^2+5}{3m},$$

$$\lim_{m\to\infty} \ell_{S_m*H} = \frac{2n+2}{n},$$

$$\frac{2m^2+m+1}{(m+1)^2} \leq \lim_{n\to\infty} \ell_{S_m*H} \leq \frac{2m^2+2m+2}{(m+1)^2}$$

and

$$\lim_{m,n\to\infty} \ell_{S_m*H} = 2.$$

Furthermore, concerning the asymptotic behaviour of the clustering coefficient $\mathcal{C}_{G*H}$, with $G \in \{P_m, S_m\}$ and $m > 2$, Lemma 3 implies that

$$T_{P_m*H} = m \cdot T_H + 2 \cdot (m-1)(n+1) \cdot |E(H)|,$$

$$T_{S_m*H} = (m+1) \cdot T_H + 2 \cdot m \cdot (n+1) \cdot |E(H)|,$$

$$\begin{aligned}\tau_{P_m*H} =& m - 2 + m \cdot \tau_H + (4m-6)(n^2+n) + \\ &+ (m-2)(n^3+n^2+n) + \\ &+ 2(m^2-m)n \cdot \left(\binom{n}{2} + |E(H)|\right)\end{aligned}$$

and

$$\begin{aligned}\tau_{S_m*H} =& (m+1) \cdot \tau_H + (m^2+m) \cdot (n^2+n) + \\ &+ \binom{m}{2} \cdot (n^3+n^2+n) + \\ &+ 2m^2 n \cdot \left(\binom{n}{2} + |E(H)|\right).\end{aligned}$$

Hence, $\lim_{m\to\infty} \mathcal{C}_{S_m*H} = 0$. Moreover, $\lim_{n\to\infty} \mathcal{C}_{G*H} = 0$, for any $G \in \{P_m, S_m\}$ such that $T_{G*H} \not\sim O(n^3)$. It is not the case if $T_{G*H} \sim O(n^3)$. Thus, for instance,

$$\lim_{n\to\infty} \mathcal{C}_{P_m*K_n} = \frac{4m-3}{6m^2+4m-9}$$

and

$$\lim_{n\to\infty} \mathcal{C}_{S_m*K_n} = \frac{8m+2}{15m^2-11m+8}.$$

Extended neighborhood coronas with a center path or star may, therefore, be used to model small-world networks and complex networks with small average path length.

## IV. Solving the dynamic coloring problem

Let us start this section by formulating a series of preliminary results that are useful to solve the dynamic coloring problem for any extended neighborhood corona $G * H$, with $G \in \{P_m, S_m\}$.

*Lemma 4:* Let $\mathcal{P}(G)$ be the set of pendant vertices of a graph $G$. If $\mathcal{P}(G) \neq \emptyset$, then $\chi_t(G*H)$ is lower bounded by

$$\max_{u\in\mathcal{P}(G)} \left\{ \min\{t, n+1\} + \min_{v\in N_G(u)} \{t, \deg_{G*H}(v)\} \right\}.$$

*Lemma 5:* For every positive integer $t$, let $\alpha_t = \min\left\{\lceil \frac{t}{n+1} \rceil, \Delta(G)\right\}$ and $\beta_t = \min\{n+1, \max\{t, \chi(H)\}\}$. Then,

$$\omega(G) \cdot \chi(H) \leq \chi_t(G*H) \leq \beta_t \cdot \chi_{\alpha_t}(G).$$

*Proposition 2:* If $\omega(G) = \chi(G)$, then $\chi_t(G*H) = \chi(G) \cdot \chi(H)$, for every $t \leq \chi(H)$.

Based on the previous results, we may solve the dynamic coloring problem for any extended neighborhood corona $P_m * H$, where $P_m = \langle u_1, \ldots, u_m \rangle$. Here, $m > 2$.

*Theorem 3:* For every positive integer $t$,

$$\chi_t(P_m * H) =$$

$$= \begin{cases} 2 \cdot \max\{t, \chi(H)\}, & \text{if } t \leq n+1, \\ n+t+1, & \text{if } n+1 < t < 2n+2, \\ 3n+3, & \text{otherwise.} \end{cases}$$

Figure 2 illustrates a 6-dynamic proper coloring of the extended neighborhood corona $P_4 * P_3$. It shows the distribution of shadows to get a $(10, 7)$-threshold secret sharing scheme so that each participant gets the maximum information from his/her neighbors. Each color or shadow is indicated between parentheses as a superscript above the corresponding vertex label.

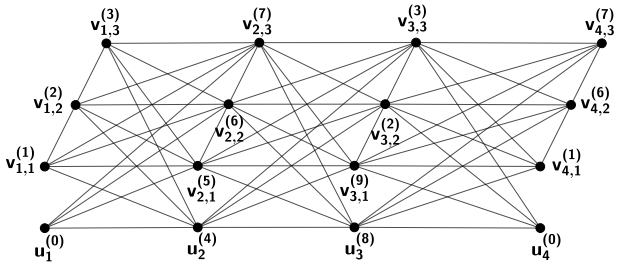



Fig. 2. Optimal 6-dynamic proper coloring of $P_4 * P_3$.

From Proposition 1, the unbounded asymptotic behaviour of $\ell_{P_m}$ implies that a small average path length of $P_m * H$ is only preserved by a dynamical growth of its outer graph. If $H$ is large enough, then Theorem 3 implies that the minimum number of shadows in which any secret must split to get a secret sharing scheme arisen from a dynamic coloring of $P_m * H$ is

$$\lim_{n\to\infty} \chi_t(P_m * H) = 2 \cdot \max\{t, \chi(H)\}.$$

We finish our study by solving the dynamic coloring problem for any extended neighborhood corona $S_m * H$, where $S_m$ has center $u_1$ and pendant vertices $u_2, \ldots, u_{m+1}$.

*Theorem 4:* Let $m > 2$ and $t$ be two positive integers. If $H$ is a graph of order $n$, then

$$\chi_t(S_m * H) =$$

$$= \begin{cases} 2 \cdot \max\{t,\, \chi(H)\}, & \text{if } t \leq n+1, \\ n+t+1, & \text{if } n+1 < t < mn+m, \\ (m+1)\cdot(n+1), & \text{otherwise.} \end{cases}$$

Figure 3 illustrates a 7-dynamic proper coloring of the extended neighborhood corona $S_3 * P_3$. It shows the distribution of shadows to get an $(11, 8)$-threshold secret sharing scheme in which each participant gets the maximum information from his/her neighbors. Again, each color or shadow is indicated between parentheses as a superscript above the corresponding vertex label.

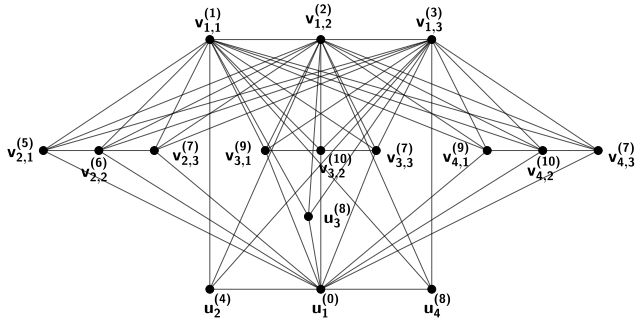



Fig. 3. Optimal 7-dynamic proper coloring of $S_3 * P_3$.

From Proposition 1, the bounded asymptotic behaviour of $\ell_{S_m}$ implies that a small average path length of $S_m * H$ is preserved by an independent dynamical growth of both its center and outer graphs. If $S_m$ is large enough, then Theorem 3 implies that the minimum number of shadows in which any secret must split to get a secret sharing scheme arisen from a dynamic coloring of $S_m * H$ is

$$\lim_{m\to\infty} \chi_t(S_m * H) = \begin{cases} 2 \cdot \max\{t,\, \chi(H)\}, & \text{if } t \leq n+1, \\ n+t+1, & \text{otherwise.} \end{cases}$$

If either the outer graph, or both the center and the outer graphs are large enough, this minimum number is

$$\lim_{n\to\infty} \chi_t(S_m * H) = \lim_{m,n\to\infty} \chi_t(S_m * H) = 2 \cdot \max\{t, \chi(H)\}.$$

## V. Conclusion and further works

In this paper, we have studied those conditions under which an extended neighborhood corona may model a small-world network or a complex network, whose average path length is small even after some dynamical growth of the graph. Depending on the asymptotic behaviour of the average path length of the center graph, this growth may be considered either in the outer graph, or, independently, in both the center and the outer graph. In order to illustrate both cases, our study has focused on those extended neighborhood coronas with a center path or star. The dynamic coloring problem has been solved for any of these graphs, which gives rise to secret sharing schemes in which each participant gets the maximum information from his/her neighbors. The minimum number of distinct shadows in which the secret must split is bounded in all of these schemes, whatever the size of the graph is. Further, we have proved that, if all the participants are honest, then two rounds of communications are enough to ensure the reconstruction of the secret by everybody. The following problem arises in a natural way. It is established as further work.

*Problem 1:* Let $G$ be a graph on which a secret sharing scheme is based on. How many rounds of communications are, at least, necessary to ensure that the secret can be reconstructed from each node in $G$?

## Acknowledgments

The authors want to express their gratitude to the anonymous referees for the comprehensive reading of the paper and their pertinent comments and suggestions, which helped improve the manuscript. Falcón's work is partially supported by the Research Project FQM-016 from Junta de Andalucía, and the Departmental Research Budget of the Department of Applied Mathematics I of the University of Seville.

## References

[1] G.R. Blakley. Safeguarding cryptographic keys. In: R. Merwin et al. (eds.), *Managing Requirements Knowledge*. International Workshop on, IEEE: New York, United States, pp. 313–317, 1979.

[2] A. Shamir. How to share a secret. *Comm. ACM*, vol. 22, pp. 612–613, 1979.

[3] V. Attasena, J. Darmont, N. Harbi. Secret sharing for cloud data security. *VLDB J.*, vol. 26, pp. 657–681, 2017.

[4] M. Naor. R.M. Roth. Optimal file sharing in distributed networks. In: C. Papadimitriou (ed.) *Proc. 32nd Annual Symposium of Foundations of Computer Science*; IEEE, NW Washington, DC, United States, pp. 515–525, 1991.

[5] M. Sardari, R. Restrepo, F. Fekri, E. Soljanin. Memory allocation in distributed storage networks. In: B. Aazhang et al. (eds.) *Proc. 2010 IEEE International Symposium on Information Theory*; IEEE, Piscataway, NJ, United States, pp. 1958–1962, 2010.

[6] X. Zhang, M. Xu, G. Da, P. Zha. Ensuring confidentiality and availability of sensitive data over a network system under cyber threats. *Reliab. Eng. Syst. Saf.*, vol. 214, paper 107697, 2021.

[7] L. Ogiela, M.R. Ogiela, H. Ko, H. Intelligent data management and security in cloud computing. *Sensors*, vol. 20, paper 3458, 2020.

[8] J. Kim, S. Ok, Dynamic choosability of triangle-free graphs and sparse random graphs. *J. Graph Theory*, vol. 87, pp. 347–355, 2017.

[9] H.J. Lai, B. Montgomery, H. Poon. Upper bounds of dynamic chromatic number. *Ars Combin.*, vol. 68, pp. 193–201, 2003.

[10] B. Montgomery. Dynamic coloring of graphs. Ph.D Thesis, West Virginia University, ProQuest LLC, Ann Arbor, MI, United States, 2001.

[11] R. M. Falcón, N. Mohanapriya, V. Aparna. Optimal shadow allocations of secret sharing schemes arisen from the dynamic coloring of extended neighborhood coronas. *Mathematics*, vol. 10, paper 2018, 13 pp., 2022.

[12] A.I. Kristiana, M.I. Utoyo, M.I., R. Alfarisi, D. Dafik, $r$-dynamic coloring of the corona product of graphs. *Discrete Math. Algorithms Appl.*, vol. 12, paper 2050019, 2020.

[13] V. Aparna, N. Mohanapriya, On $r$-dynamic coloring of neighborhood corona of path with some graphs. *J. Physics: Conf. Series*, vol. 1523, paper 012001, 2020.

[14] G. Nandini, M. Venkatachalam, R.M. Falcón. On the $r$-dynamic coloring of subdivision-edge coronas of a path. *AIMS Math.*, vol. 5, pp. 4546–4562, 2020.

[15] R. Frucht, F. Harary, On the corona of two graphs. *Aequationes Math.*, vol. 4, pp. 322–32, 1970.

[16] G. Indulal. The spectrum of neighborhood corona of graphs. *Kragujevac J. Math.*, vol. 35, pp. 493–500, 2011.

[17] C. Adiga, B.R. Rakshith, K.N. Subba Krishna. Spectra of extended neighborhood corona and extended corona of two graphs. *Electron. J. Graph Theory Appl.*, vol. 4, pp. 101–110, 2016.

[18] H. J. Lai, J. Lin, B. Montgomery, T. Shui, S. Fan. Conditional colorings of graphs. *Discrete Math.*, vol. 306, pp. 1997–2004, 2006.

# Algoritmos para códigos separadores

Marcel Fernández Muñoz
Department d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya (UPC)
marcel.fernandez@upc.edu

John Livieratos
Department of Mathematics
National & Kapodistrian
University of Athens, Greece
jlivier89@math.uoa.gr

Sebastià Martín Molleví
Departament de Matemàtiques
Universitat Politècnica de Catalunya (UPC)
sebastia.martin@upc.edu

***Resumen*—En este artículo discutimos los aspectos algorítmicos de los códigos separadores, es decir, códigos en los que dos subconjuntos cualesquiera (de tamaño fijado) de palabras tienen al menos una posición con elementos distintos. En concreto, nos centramos en el caso (no trivial) de los códigos binarios 2-separadores. En primer lugar, utilizamos el Lema Local de Lovász para obtener una cota inferior de la tasa de este tipo de códigos, y vemos que coincide con la mejor tasa conocida. Después, usamos la versión algorítmica del Lema Local de Lovász para construir códigos con dicha tasa, y discutimos sus implicaciones en relación a la complejidad computacional.**

***Index Terms*—códigos separadores, Lema Local de Lovász, demostración constructiva de Moser-Tardos**

## I. Introducción

Los códigos separadores [1] gozan de una larga tradición de estudio en las áreas de combinatoria y teoría de códigos. Este tipo de códigos han demostrado su utilidad en aplicaciones en diversos campos como el diagnóstico técnico, la construcción de funciones hash, la síntesis de autómatas o el rastreo de traidores.

Podemos representar un código como una matriz de símbolos de un alfabeto finito. La *longitud* de un código es el número de columnas de la matriz, mientras que su *tamaño* es el número de filas. La *tasa* de un código es la ratio entre (el logaritmo de) su tamaño y su longitud. Un código se llama $c$-separador si para dos conjuntos cualesquiera de como máximo $c$ filas disjuntas cada uno, hay una columna en la que los símbolos del primer conjunto son diferentes de los símbolos del segundo. Esto resulta ser un requisito muy fuerte y, aunque se han dedicado muchos esfuerzos a la obtención de cotas inferiores y superiores de las tasas de dichos códigos, las cotas conocidas no son muy ajustadas. Además, las construcciones explícitas de tales códigos son muy escasas.

Por ejemplo, para el caso de códigos binarios 2-separadores (que no es trivial), la mejor cota inferior de la tasa obtenida hasta el momento es $0{,}0642$ [1], [2], [3], mientras que la mejor cota superior es $0{,}2835$ [4]. Ya vemos que se trata de valores un tanto distantes. La cota inferior se obtiene mediante la elegante técnica de codificación aleatoria con expurgación. El inconveniente de la estrategia de codificación aleatoria es que no da ninguna pista sobre cómo obtener un código explícito que coincida con la cota inferior. Un camino alternativo a seguir para tratar de demostrar la existencia de objetos combinatorios es el Lema Local de Lovász (LLL). Este poderoso resultado fue enunciado por primera vez por Erdos y Lovász en [5]. En su forma simple y *simétrica*, el LLL da una condición necesaria sobre la posibilidad de evitar un cierto número de eventos indeseados, dadas una cotas superiores de la probabilidad de que ocurra cada evento, y del número de dependencias entre ellos.

Nuestro interés en el LLL radica en sus demostraciones algorítmicas. Moser [6] dio la primera demostración de este tipo, que se aplicó únicamente al problema de satisfacibilidad. Más tarde, Moser y Tardos [7] dieron una demostración de la versión *asimétrica* (más fuerte) del Lema en el caso general, basada en el *método entrópico* (ver [8]). Más recientemente, Giotis et al. [9], [10] aplicaron un enfoque probabilístico directo para demostrar varias formas del LLL (para una exposición analítica de las diversas formas del LLL, se puede consultar [11]). Los dos enfoques anteriores (como la mayoría de los enfoques algorítmicos del LLL), usan lo que se conoce como *marco de variables*, donde se supone que todos los eventos indeseados dependen de un cierto número de *variables aleatorias independientes*. Para enfoques algorítmicos fuera del marco de variables, véanse por ejemplo los trabajo de Harvey y Vondrak [12] y de Achlioptas y Illiopoulos [13].
**Contribuciones:** el presente artículo trata sobre códigos binarios 2-separadores.

En la Sección III, hallamos una cota inferior de dichos códigos, usando el Lema Local de Lovász al estilo de Deng et al. en [14]. En la Sección IV, mostramos construcciones explícitas de tales códigos, utilizando el trabajo de Giotis et al [9] y Kirousis y Livieratos [15].

## II. Definiciones y resultados previos

En esta sección damos algunas definiciones y resultados sobre códigos separadores, así como el enunciado del Lema Local de Lovász (LLL), que se usará a lo largo del artículo.

### *II-A. Códigos separadores*

Comenzamos con algunas definiciones básicas.
Si $\mathcal{Q}$ es un alfabeto finito de tamaño $q$, denotamos por $\mathcal{Q}^n$ al conjunto de vectores de longitud $n$ con coordenadas en $\mathcal{Q}$, a los cuales llamamos *palabras*. Un *código* $(n, M)_q$, $\mathcal{C} \subset \mathcal{Q}^n$, es un subconjunto de tamaño $M$. Sus elementos son las *palabras código*. La *distancia de Hamming* entre dos palabras código es el número de posiciones en las que difieren. La *distancia mínima* de $\mathcal{C}$, denotada por $d$, se define como la menor distancia entre dos palabras código distintas.

En teoría algebraica de códigos, $\mathcal{Q}$ suele ser el cuerpo finito de $q$ elementos, $\mathbb{F}_q$. En este caso, un código $\mathcal{C}$ es *lineal* cuando es un subespacio de $\mathbb{F}_q^n$. Un código $[n, k, d]$ es un código de longitud $n$, dimensión $k$ y distancia mínima $d$.

Sea $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^c\} \subset \mathcal{C}$, de cardinal $|U| = c$, con $\mathbf{u}^i = (u_1^i, \ldots, u_n^i)$, $1 \leq i \leq c$. Entonces $U_j = \{u_j^1, \ldots . u_j^c\}$

es el conjunto de las $j$-ésimas coordenadas de las palabras de $U$.

**Definición 1** (Sagalovich [1])**.** *Un código $\mathcal{C}$ es c-separador si para cualesquiera dos conjuntos disjuntos de palabras código, $U$ y $V$, con $|U| \leq c$, $|V| \leq c$ y $U \cap V = \emptyset$, existe al menos una coordenada $j$ tal que $U_j \cap V_j = \emptyset$.*
*Decimos entonces que la coordenada $j$ separa $U$ y $V$.*

En este artículo nos ocuparemos del caso $c = 2$.

**Definición 2.** *Sea $\mathcal{C}$ un código $(n, M)_q$ sobre $\mathcal{Q}$. La tasa $R$ del código es*

$$R = \frac{\log_q M}{n}. \tag{1}$$

Sea $R(n, c)_q$ la tasa óptima de un código $(n, M)_q$ $c$-separador. Nos interesaremos por la tasa asintótica:

$$\underline{R}_q(c) = \liminf_{n\to\infty} R_q(n, c). \tag{2}$$

El siguiente resultado nos asegura que, para el caso binario, existen códigos con tasa asintótica positiva.

**Proposición 1** (Barg et al. [3])**.** *Existen códigos c-separador binarios de longitud $n$ y tamaño $\frac{1}{2}(1 - 2^{-(2c-1)})^{-(n/(2c-1))}$, i. e.*

$$\underline{R}(n, 2)_2 \geq -\frac{\log_2(1 - 2^{-(2c-1)})}{2c - 1} - \frac{1}{n}. \tag{3}$$

Respecto a la tasa de los códigos binarios 2-separador, tenemos el siguiente corolario:

**Corolario 1** (Sagalovich [1])**.** *Existen códigos binarios 2-separador con tasa:*

$$\underline{R}_2(2) \geq 1 - \log_2(7/8) \approx 0{,}0642.$$

*II-B. Lema Local de Lovász (LLL)*

Sean $E_1, \ldots, E_m$ eventos definidos en un mismo espacio de probabilidad $\Omega$, que se consideren indeseados y que suponemos ordenados según sus índices. Consideramos un grafo (simple) $G$ con conjunto de vértices $[m] := \{1, \ldots, m\}$ y donde dos vértices $i, j \in [m]$ están conectados por una arista (no dirigida) si los eventos $E_i$ y $E_j$ son dependientes. En la literatura, estos grafos se denominan *grafos de dependencia*.

Llamamos *vecindad* de $j$ al conjunto $\Gamma_j$ de vértices adyacentes al vértice $j$ en $G$, donde suponemos que ningún vértice pertenece a su vecindad ($j \notin \Gamma_j$, $j = 1, \ldots, m$). Diremos que un evento $E_i$ tal que $i \in \Gamma_j$ es un evento de la vecindad de $E_j$. Sea también $s \geq 1$ el grado máximo de $G$ ($|\Gamma_j| \leq s$, para $j = 1, \ldots, m$) y supongamos que existe $p \in (0, 1)$ tal que $\Pr[E_j] \leq p$, $j = 1, \ldots, m$.

En su versión simple, *simétrica*, el Lema Local de Lovász proporciona una condición suficiente, dependiendo de $p$ y $s$, que permite evitar todos los eventos $E_1, \ldots, E_m$.

**Teorema 1** (Lema Local de Lovász simétrico)**.** *Supongamos que $E_1, \ldots, E_m$ son eventos cuyo grafo de dependencia tiene grado $s$ y tal que existe un $p \in (0, 1)$ con $Pr[E_j] \leq p$, $j = 1, \ldots, m$. Si*

$$ep(s + 1) \leq 1, \tag{4}$$

*entonces*

$$\Pr\left[\bigcap_{i=1}^{m} \overline{E}_i\right] > 0.$$

Nos interesa un enfoque constructivo del LLL. Por lo tanto, utilizaremos lo que se conoce como el *marco de variables*, aparecido por primera vez en un trabajo de Moser y Tardos [7]. Sean $X_i$, $i \in [t]$ variables aleatorias independientes dos a dos, definidas en el espacio de probabilidad $\Omega$ y con valores en un conjunto finito $\mathcal{Q}$. Una asignación de valores a las variables aleatorias es un vector $t$-ario $\alpha = (a_1, \ldots, a_t)$, con $a_i \in \mathcal{Q}$, $i = 1, \ldots, t$. A partir de ahora, $\Omega = \mathcal{Q}^t$.

Supongamos que los eventos $E_1, \ldots, E_m$, definidos en $\Omega$, dependen sólo de un subconjunto de las variables aleatorias, que llamamos su *alcance*. El alcance de $E_j$ se denota por $\mathrm{sc}(E_j)$ (del inglés *scope*).

El LLL proporciona una condición suficiente para la existencia de un punto en el espacio de probabilidad $\Omega$ (es decir una asignación de valores a las variables aleatorias) de modo que ninguno de los eventos ocurra. Queremos hallar esta asignación de manera eficiente. Para ello, consideramos el algoritmo 1:

**Algorithm 1**

1: Sea $\alpha$ la asignación resultante del muestreo de las variables $X_i$, $i = 1, ..., t$.
2: **while** existe un evento que ocurre bajo la asignación actual **do**
3: RESAMPLE($E_j$), donde $E_j$ es el evento con menor índice
4: **end while**
5: Devuelve la asignación actual $\alpha$.

RESAMPLE($E_j$)
1: Remuestrea las variables de sc($E_j$).
2: **while** algún evento de $\Gamma_j \cup \{j\}$ ocurre bajo la asignación actual **do**
3: RESAMPLE($E_k$), donde $E_k$ es el evento con menor índice
4: **end while**

A partir del algoritmo 1, Giotis et al. [9] demostraron el siguiente resultado:

**Teorema 2** (LLL algorítmico)**.** *Si $p$ y $s$ satisfacen*

$$\left(1 + \frac{1}{s}\right)^s ps < 1$$

*(y, por tanto, si $ep(s+1) \leq 1$), entonces existen un entero $N_0$ (que depende linealmente de $m$) y una constante $t \in (0, 1)$ (que depende de $p$ y $s$) de manera que si $N/\log N \geq N_0$, entonces la probabilidad de que el algoritmo 1 acabe en al menos $N$ rondas es inversamente exponencial en $N$.*

Del resultado anterior podemos deducir dos cosas. En primer lugar, el Teorema 2 implica la existencia de una asignación sin ningún evento indeseado, como se deduce de la línea 2, si y cuando el algoritmo termina. Además, tal asignación se obtiene en tiempo polinómico en $N$.

## III. COTA INFERIOR DE LA TASA DE UN CÓDIGO BINARIO 2-SEPARADOR

Nuestro objetivo es utilizar el Lema Local de Lovász para obtener una cota inferior en la tasa de los códigos binarios 2-separadores. La cota que obtenemos es del mismo orden de

magnitud que la cota de la Proposición 1. Sin embargo, como veremos en el siguiente apartado, la ventaja es que el uso del LLL nos permitirá construir explícitamente el código.

Sean $X_{ij}$ $1 \leq i \leq M$, $1 \leq j \leq n$, $nM$ variables aleatorias independientes con distribución de Bernoulli, donde:

$$\Pr(X_{ij} = 0) = \Pr(X_{ij} = 1) = \frac{1}{2}.$$

Sea $\Omega = \{0,1\}^{nM}$ el conjunto de todos los vectores binarios de $nM$ coordenadas. Pensaremos en $\Omega$ como el conjunto de matrices $M \times n$ con entradas binarias. Esto permite interpretar la matriz obtenida al asignar valores a las variables aleatorias $X_{ij}$, $1 \leq i \leq M$, $1 \leq j \leq n$, como un código $(n, M)_2$, es decir un código binario con $M$ palabras de longitud $n$, que denotamos por $\mathcal{C}$. Sea $U = \{\mathbf{u}^1, \mathbf{u}^2\}$ un conjunto de dos palabras distintas de $\mathcal{C}$, $\mathbf{u}^i = (u_1^i, \ldots, u_n^i)$, $i = 1, 2$, y sea

$$\mathcal{P}_{\mathcal{C}} := \{\{U, V\} \mid U \cap V = \emptyset\}$$

el conjunto de pares disjuntos de palabras distintas de $\mathcal{C}$. Para cada $\{U, V\} \in \mathcal{P}_{\mathcal{C}}$, definimos que el evento $E_{U,V}$ ocurre cuando $U, V$ *no* están separados. Hay $m = \binom{M}{2}\binom{M-2}{2}$ tales eventos, que suponemos que están ordenados arbitrariamente.

**Lema 1.** *La probabilidad del evento $E_{U,V}$ es:*

$$\Pr[E_{U,V}] = \left(\frac{7}{8}\right)^n. \tag{5}$$

*Demostración:* Consideremos los $j$-ésimos índices de $\mathbf{u}^1, \mathbf{u}^2, \mathbf{v}^1$ y $\mathbf{v}^2$. La probabilidad del evento $E^j_{U,V}$, que es el evento de que $U$ y $V$ no estén separados en la coordenada $j$-ésima, es igual a la probabilidad de que $u_j^1 \neq u_j^2$ más la probabilidad de que $u_j^1 = u_j^2 = v_j^i$, para al menos un $i \in \{1, 2\}$,

$$\Pr[E^j_{U,V}] = \frac{1}{2} + \frac{1}{2} \cdot \frac{3}{4} = \frac{7}{8}.$$

Finalmente, teniendo en cuenta que cada coordenada de una palabra código toma valores de forma independiente, obtenemos el resultado enunciado. ■

Se puede ver fácilmente que dos eventos son dependientes si tienen al menos una palabra código en común.

**Lema 2.** *El número $s$ de eventos dependientes de $E_{U,V}$ verifica*

$$s + 1 \leq \frac{7}{2}M^3. \tag{6}$$

*Demostración:* Al restar el número de eventos que no comparten una palabra código con $E_{U,V}$, obtenemos que el número de eventos dependientes de $E_{U,V}$ es igual a:

$$\begin{aligned}\binom{M}{2}\binom{M-2}{2} - \binom{M-4}{2}\binom{M-6}{2} - 1 = \\ = \frac{7}{2}M^3 - \frac{69}{2}M^2 + 121M - 149 < \frac{7}{2}M^3.\end{aligned}$$

■

A partir de los lemas anteriores y del Teorema 1, podemos enunciar el siguiente resultado:

**Teorema 3.** *Para todo $n > 0$ existe un código binario 2-separador de tamaño*

$$M \leq \sqrt[3]{\frac{2}{7e}}\left(\frac{8}{7}\right)^{n/3}. \tag{7}$$

*Demostración:* La hipótesis del LLL (Teorema 1) es $ep(s+1) \leq 1$. Si sustituimos los valores de $p$ y de la cota de $s$ obtenidos en los Lemas 1 y 2, y despejamos M en la inecuación 4, obtenemos

$$e\left(\frac{7}{8}\right)^n\left(\frac{7}{2}M^3\right) \leq 1 \Leftrightarrow M \leq \sqrt[3]{\frac{2}{7e}}\left(\frac{8}{7}\right)^{n/3}. \quad ■$$

Del Teorema 3 se deduce el siguiente resultado:

**Corolario 2.** *Existen códigos binarios 2-separadores con tasa $R \approx 0{,}0642$.*

*Demostración:* Como el código es binario, de la definición 2 tenemos que

$$\begin{aligned}R =& \frac{\log_2 M}{n} \\ =& \frac{\log_2\left(\sqrt[3]{\frac{2}{7e}}\left(\frac{8}{7}\right)^{n/3}\right)}{n} \\ =& \frac{\log_2\left(\sqrt[3]{\frac{2}{7e}}\right)}{n} + \frac{n\log_2(\frac{8}{7})}{3n} \\ =& \frac{\log_2\left(\sqrt[3]{\frac{2}{7e}}\right)}{n} + \frac{\log_2(\frac{8}{7})}{3}.\end{aligned}$$

Si hacemos el límite cuando $n$ tiende a $\infty$, obtenemos $R \approx 0{,}0642$. ■

**Observación 1.** *Nuestro problema goza de mucha simetría. En primer lugar, en virtud del Lema 1, todos los eventos tienen exactamente la misma probabilidad de ocurrir. Además, la cota de $s$ en el Lema 2 también es la misma para todos los eventos. En realidad, podría mejorarse si incluyéramos los términos de grado 1 y 2 del polinomio $\frac{7}{2}M^3 - \frac{69}{2} + 121M - 149$, pero esto no cambiaría el valor del límite cuando $n$ tiende a infinito.*

Debido a las simetrías mencionadas, vemos que el hipotético uso de la versión asimétrica del Lema (véase por ejemplo [7]) no mejoraría nuestro resultado. Cabe preguntarse si podrían aplicarse aquí otras versiones existentes del LLL, con la esperanza de obtener mejores resultados.

## IV. Construcciones explícitas

En esta sección veremos cómo obtener construcciones explícitas de códigos binarios 2-separadores con tasa positiva, para lo cual aplicaremos la versión algorítmica del LLL. Veremos que, si queremos conseguir tasa positiva, la complejidad computacional de nuestro algoritmo va a ser exponencial en la longitud del código (ver Observación 2).

Sean $t = nM$ y $m = \binom{M}{2}\binom{M-2}{2}$. Renombramos las variables aleatorias $X_{ij}$ y los eventos $E_{U,V}$: tenemos $t$ variables aleatorias $X_1, \ldots, X_t$ (reordenadas arbitrariamente) y $m$ eventos $E_1, \ldots, E_m$ (también ordenados arbitrariamente), con $p = (\frac{7}{8})^n$ y $s + 1 = \frac{7}{2}M^3$. Entonces podremos aplicar directamente el algoritmo 1 y el análisis de Giotis et al. [9] para obtener algorítmicamente los resultados de la Sección III. A continuación, destacamos brevemente algunas partes de este análisis, basadas en la demostración de [15].

**Teorema 4.** *Para cada $n > 0$, existe un algoritmo aleatorio tal que la probabilidad de que su ejecución dure al menos*

*$N$ rondas es inversamente exponencial en $N$, y que da como resultado un código 2-separador de tamaño*

$$M \leq \sqrt[3]{\frac{2}{7e}} \left(\frac{8}{7}\right)^{n/3}.$$

*Demostración:* Primero observemos que si el algoritmo 1 termina, entonces devuelve (en virtud de la línea 2) una asignación de valores a las variables aleatorias de modo que no ocurra ningún evento indeseado. Esto se traduce, como ya hemos visto, en un código 2-separador.

Una llamada *raíz* de RESAMPLE es cualquier llamada hecha desde la línea 3 del cuerpo del algoritmo, mientras que una llamada *recursiva* es una llamada hecha desde la línea 3 de la subrutina RESAMPLE. Una ronda es la duración de cualquier llamada a RESAMPLE. Puede demostrarse que cualquier evento que no ocurrió al comienzo de un RESAMPLE($E_j$), i.e. cualquier par separado de palabras código distintas, continúa siendo separado si y cuando esa llamada termina: todo evento que ocurra en cualquier momento durante RESAMPLE($E_j$) será posteriormente verificado y remuestreado por alguna subrutina RESAMPLE, llamada desde dentro de RESAMPLE($E_j$). Además, por la línea 2 del bloque principal, es sencillo ver que si y cuando RESAMPLE($E_j$) termina, $E_j$ no ocurre, aunque haya ocurrido al principio. Por lo tanto, el algoritmo progresa en cada ronda, y lo que se ha conseguido no se pierde en las siguientes rondas (para una demostración completa, véase [15]).

Dada una ejecución del algoritmo 1, construimos un bosque con raíces y etiquetado $\mathcal{F}$ (es decir, un bosque compuesto por árboles con raíz, donde los nodos están etiquetados), de la siguiente manera:

(i) Para cada llamada a RESAMPLE($E_j$), añadimos un nodo con etiqueta $E_j$.

(ii) Si se llama a RESAMPLE($E_r$) desde la línea 3 de RESAMPLE($E_j$), entonces el correspondiente nodo con etiqueta $E_r$ es un hijo del etiquetado con $E_j$.

No es difícil ver que las raíces de $\mathcal{F}$ corresponden a llamadas a RESAMPLE desde el bloque principal del algoritmo, mientras que el resto de los nodos corresponden a llamadas recursivas. Además, por la discusión anterior, tenemos que las etiquetas de las raíces son distintas dos a dos. Sucede lo mismo con las etiquetas de los hermanos. Finalmente, si un nodo con etiqueta $E_r$ es hijo de uno etiquetado con $E_j$, entonces $r \in \Gamma_j$.

Llamamos al bosque creado de la forma anterior, el *bosque testigo* de la ejecución del algoritmo. Dada una ejecución que dura $N$ pasos, su bosque testigo tiene $N$ nodos. Ordenamos los nodos del bosque de la siguiente manera: (i) los árboles y hermanos se ordenan según los índices de sus etiquetas (ii) los nodos de un árbol se ordenan en pre-orden, respetando el ordenamiento de los hermanos. Así, de cada bosque testigo $\mathcal{F}$ con $N$ nodos, podemos obtener su *secuencia de etiquetas* $(E_{j_1}, \ldots, E_{j_N})$.

Ahora, si $\mathrm{P}_N$ es la probabilidad de que el algoritmo 1 dure al menos $N$ rondas, tenemos que:

$$\mathrm{P}_N = \Pr[\text{se construya algún bosque testigo } \mathcal{F} \text{ con } N \text{ nodos}]. \quad (8)$$

Consideremos ahora el siguiente algoritmo de *validación*, que toma como entrada la secuencia de etiquetas de un bosque testigo:

**Algorithm 2** VALALG.

Input: Secuencia de etiquetas $(E_{j_1}, \ldots, E_{j_N})$ de $\mathcal{F}$
1: Muestrea las variables $X_i$, $i = 1, ..., t$.
2: **for** $i = 1, \ldots, N$ **do**
3: **if** $E_{j_i}$ ocurre durante la asignación actual **then**,
4: Resmuestrea las variables de $\mathrm{sc}(E_{j_i})$
5: **else**
6: Retorna `failure` y sale
7: **end if**
8: **end for**
9: Retorna `success`

Observemos que la salida `success` o `failure` de VALALG no tiene nada que ver con si todos los eventos se han evitado o no y, por lo tanto, con que el código sea separable.

El evento de que una ejecución del algoritmo 1 produzca un bosque testigo $\mathcal{F}$, implica el evento de que VALALG tendrá éxito cuando la entrada sea $\mathcal{F}$ (VALALG puede hacer las mismas elecciones aleatorias que el algoritmo 1). Así pues, podemos acotar la probabilidad de que el algoritmo 1 dure al menos $N$ rondas por:

$$\mathrm{P}_N \leq \sum_{|\mathcal{F}|=n} \Pr[\text{VALALG tiene éxito en la entrada } \mathcal{F}], \quad (9)$$

donde $|\mathcal{F}|$ denota el número de nodos de $\mathcal{F}$. En la desigualdad (9), sea $V_i(\mathcal{F})$ el evento de que VALALG no falle en la ronda $i$ de la entrada $\mathcal{F}$. Se puede demostrar que, dado un bosque $\mathcal{F}$ cuya secuencia de etiquetas es $(E_{j_1}, \ldots, E_{j_N})$, se cumple que:

$$\Pr[\text{VALALG tenga éxito partiendo de una entrada } \mathcal{F}] = \prod_{i=1}^{N} \Pr[V_i(\mathcal{F}) \mid \bigcap_{r=1}^{i-1} V_r(\mathcal{F})] = \prod_{i=1}^{N} \Pr[E_{j_i}] = p^N.$$

En consecuencia, para acotar la parte derecha de la desigualdad (9), necesitamos contar el número de bosques con $N$ nodos internos. Se puede demostrar que para hacer eso, podemos contar el número $f_N$ de bosques planos con $N$ nodos internos formados por $m$ árboles $(s+1)$-arios con raíz (para los detalles necesarios, ver nuevamente [15]).

Sea $t_N$ el número de árboles planos con raíz $(s+1)$-arios completos con $N$ nodos internos. Se cumple que

$$t_N = \frac{1}{sN+1}\binom{(s+1)N}{N}$$

(ver [16, Theorem 5.13]).

Usando la aproximación de Stirling, deducimos que existe una constante $A$ (dependiendo sólo de $s$) tal que:

$$t_N < A\left(\left(1+\frac{1}{s}\right)^s (s+1)\right)^N. \quad (10)$$

Finalmente, en virtud de la desigualdad (10), obtenemos:

$$f_N = \sum_{\substack{N_1+\cdots+N_m=N \\ N_1,\ldots,N_m \geq 0}} t_{N_1} \cdots t_{N_m} < (AN)^m \left(\left(1+\frac{1}{s}\right)^s (s+1)\right)^N. \quad (11)$$

Para concluir la demostración, las ecuaciones (9) y (11) implican:

$$\mathrm{P}_N < (AN)^m \left( \left(1 + \frac{1}{s}\right)^s (s+1)p \right)^N . \qquad (12)$$

■

**Observación 2.** *En la línea 2, para que el* ALGORITMO 1 *encuentre el evento con menor índice, debe revisar los aproximadamente* $2M^4$ *elementos de* $\mathcal{P}$ *y verificar si están separados. Así mismo, en la línea 2 de una llamada* RESAMPLE$(E_j)$*, el* ALGORITMO 1 *debe revisar los aproximadamente* $5M^3$ *eventos de la vecindad de* $E_j$*. Dada la cota de* $M$ *que hemos obtenido, es fácil ver que, en ambos casos, el número de eventos que deben revisarse es exponencial en* $n$*.*

## V. Conclusiones

Hemos demostrado que el Lema Local de Lovász se puede utilizar para obtener cotas de la tasa en códigos separadores, obteniendo el mismo resultado que con el uso de la técnica de codificación aleatoria con expurgación. La ventaja de nuestro método es que nos permite obtener construcciones explícitas, utilizando la versión algorítmica del LLL. Así pues, también detallamos la construcción de un código separador con tasa positiva, que resulta ser de complejidad exponencial en la longitud del código. A la vista de los resultados de la Sección III, parece difícil alejarse de tal complejidad si queremos mantener tasas positivas.

## Agradecimientos

El trabajo de Marcel Fernández ha sido financiado por TCO-RISEBLOCK (PID2019-110224RB-I00) MINECO.
El trabajo de Sebastià Martín ha sido financiado por el Ministerio de Ciencia e Innovación, PID2019-109379RB-I00.

## Referencias

[1] Y. L. Sagalovich, "Separating systems", *Problems Inform. Transmission*, vol. 30, no. 2, pp. 105–123, 1994.

[2] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE transactions on information theory*, vol. 47, no. 3, pp. 1042–1049, 2001.

[3] A. Barg, G. R. Blakley, and G. A. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Transactions on Information Theory*, vol. 49, no. 4, pp. 852–865, 2003.

[4] J. Körner and G. Simonyi, "Separating partition systems and locally different sequences," *SIAM journal on discrete mathematics*, vol. 1, no. 3, pp. 355–359, 1988.

[5] P. Erdős and L. Lovász, "Problems and results on 3-chromatic hypergraphs and some related questions," *Infinite and finite sets*, vol. 10, pp. 609–627, 1975.

[6] R. A. Moser, "A constructive proof of the Lovász local lemma," in *Proceedings 41st Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2009, pp. 343–350.

[7] R. A. Moser and G. Tardos, "A constructive proof of the general Lovász local lemma," *Journal of the ACM (JACM)*, vol. 57, no. 2, p. 11, 2010.

[8] T. Tao, "Moser's entropy compression argument," 2009, available: https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/.

[9] I. Giotis, L. Kirousis, K. I. Psaromiligkos, and D. M. Thilikos, "On the algorithmic Lovász local lemma and acyclic edge coloring," in *Proceedings of the twelfth workshop on analytic algorithmics and combinatorics*. Society for Industrial and Applied Mathematics, 2015, available: http://epubs.siam.org/doi/pdf/10.1137/1.9781611973761.2.

[10] I. Giotis, L. M. Kirousis, J. Livieratos, K. I. Psaromiligkos, and D. M. Thilikos, "Alternative proofs of the asymmetric Lovász local lemma and Shearer's lemma," in *Proceedings of the 11th International Conference on Random and Exhaustive Generation of Combinatorial Structures, GASCom*, 2018, available: http://ceur-ws.org/Vol-2113/paper15.pdf.

[11] M. Szegedy, "The Lovász local lemma–a survey," in *International Computer Science Symposium in Russia*. Springer, 2013, pp. 1–11.

[12] N. J. Harvey and J. Vondrák, "An algorithmic proof of the Lovász local lemma via resampling oracles," in *Proceedings 56th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2015, pp. 1327–1346.

[13] D. Achlioptas and F. Iliopoulos, "Random walks that find perfect objects and the Lovász local lemma," *Journal of the ACM (JACM)*, vol. 63, no. 3, p. 22, 2016.

[14] D. Deng, D. R. Stinson, and R. Wei, "The Lovász local lemma and its applications to some combinatorial arrays," *Designs, Codes and Cryptography*, vol. 32, no. 1-3, pp. 121–134, 2004.

[15] L. Kirousis and J. Livieratos, "A simple algorithmic proof of the symmetric lopsided Lovász local lemma," in *International Conference on Learning and Intelligent Optimization*. Springer, 2018, pp. 49–63.

[16] R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*. Addison-Wesley, 2013.

# Comercio de datos con servicio de muestreo gratuito

Rafael Genés-Durán
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
rafael.genes@upc.edu

Oscar Esparza
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
oscar.esparza@upc.edu

Juan Hernández-Serrano
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
j.hernandez@upc.edu

Fernando Román-García
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
fernando.roman@upc.edu

Miquel Soriano
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
miquel.soriano@upc.edu

Jose L. Muñoz-Tapia
Departamento de Ingenieria Telemática
Universitat Politècnica de Catalunya
Barcelona, España
jose.luis.munoz@upc.edu

***Resumen*—Los datos se perciben actualmente como uno de los recursos más valiosos de la industria. En este contexto, han surgido mercados de datos para facilitar el comercio de datos de manera coordinada. Para facilitar los intercambios abiertos, justos y transparentes, es necesario asegurar la confianza, tanto entre los diferentes participantes como la confianza en el ecosistema. Gracias a las emergentes tecnologías blockchain, existen nuevos mercados de datos que brindan esta confianza de manera descentralizada con nuevos servicios disruptivos como resolución automatizada de conflictos, el no repudio y contabilidad auditable. En este documento, presentamos un mercado de datos descentralizado llamado i3-MARKET que implementa dichos servicios. También describimos un servicio de muestreo de datos que permite a un consumidor y a un proveedor acordar de forma segura, la visualización de forma gratuita de una parte de los datos antes de comprometerse a la compra, de forma que pueda validar la calidad de los mismos. Finalmente, el documento analiza cómo integrar el servicio de muestreo gratuito en el ecosistema de i3-MARKET.**

***Index Terms*—Muestreo; Mercado de datos; Descentralización; Blockchain; Contratos inteligentes; Consentimiento explícito; Privacidad; Protección de datos;**

## I. Introduction

Uno de los activos más valiosos y poderosos que puede tener una empresa son sus datos [1]. Los estudios muestran que las empresas dependen cada vez más de los análisis de datos para potenciar su valor comercial [2]. Además, muchas empresas confían en datos de terceros para realizar un mejor análisis y potenciando sus resultados. En este sentido, la Unión Europea ha destacado la importancia de los datos no solo en la economía y la innovación, sino también en referencia a las políticas y legislación [3]. A pesar de que en el pasado los datos han sido considerados una materia prima que complementaba el valor de un producto, hoy en día los datos se consideran en sí mismos un producto.

Recientemente, se han creado una multitud de mercados de datos. La mayoría de estos mercados son plataformas centralizadas que facilitan la comercialización de datos y que conectan a los proveedores de datos con potenciales consumidores [4]. Los mercados de datos centralizados, sin embargo, presentan riesgos ya que el almacenamiento y la operación centralizada plantea mucho poder y una necesidad de confianza absoluta en unas pocas manos. En este contexto, ha surgido un nuevo conjunto de ecosistemas descentralizados con el propósito de crear intercambios de datos descentralizados, justos, confiables y transparentes sin la necesidad de tener que confiar completamente en actores centralizados. La principal tecnología para construir estos mercados de datos descentralizados es la tecnología blockchain, que permite una operación descentralizada, por lo que el poder no está en manos de un actor o administración.

Uno de los principales desafíos que enfrenta un mercado de datos descentralizado es cómo crear confianza entre los proveedores de datos y los potenciales clientes. Los proveedores de datos quieren que les paguen por lo que ofrecen, mientras que los consumidores de datos quieren saber exactamente por lo que están pagando antes de pagarlo.

Este documento analiza cómo integrar un servicio de muestreo gratuito en un mercado de datos descentralizado. Un servicio de muestreo gratuito permite que un consumidor y un proveedor acuerden de forma segura una porción de un conjunto de datos que el consumidor puede consultar antes de comprometerse a la compra. El objetivo de este tipo de servicio es aumentar la confianza del consumidor en el valor de los datos a la venta.

En este articulo, el servicio de muestreo gratuito considerado se denomina DEFS [5], intercambio de datos con muestreo gratuito y el mercado de datos considerado es i3-MARKET [6]. El proyecto i3-MARKET está financiado por la Unión Europea y su objetivo es implementar un backplane de código abierto que sea inteligente, interoperable, integrador y facilmente desplegable. Este sistema permitirá la federación de mercados y espacios de datos actualmente emergentes, pero aún aislados, y también garantizará una competencia justa para todas las partes interesadas. Con la integración del servicio DEFS en el ecosistea de i3-MARKET, se permitirá a cada mercado participante, la posibilidad de ofrecer dicho muestreo gratuito para fomentar la confianza de los clientes respecto a los proveedores.

El resto del documento está organizado de la siguiente

manera: La Sección II proporciona los antecedentes necesarios para comprender las tecnologías descentralizadas y en específico, la blockchain. La Sección III describe los diferentes aspectos y detalles de la integración del servicio de muestreo gratuito DEFS en el proyecto i3-MARKET, utilizando la blockchain como intermediaria. Finalmente, la Sección IV presenta las conclusiones.

## II. Background

Distributed Ledger Technologies (DLT) se refiere a los protocolos tecnológicos que construyen, operan y comparten un libro de cuentas o ledger entre varios participantes. Un ledger contiene un diario de transacciones, manteniendo el estado de cada parte. En una DLT, el ledger se comparte, lo que significa que tiene muchas réplicas consistentes distribuidas sobre los nodos de la red. A pesar de que las DLT son una tecnología distribuida que no tiene una administración central para almacenar y manejar los datos, la DLT es capaz de mantener réplicas consistentes del estado de las cuentas. Para hacerlo, utiliza algoritmos de consenso que definen algunas reglas para generar consenso entre los participantes desconfiados.

La tecnología principal para construir un ledger público es una red blockchain. En una red blockchain, los usuarios pueden ejecutar su propio nodo para enviar transacciones, o pueden usar nodos disponibles de otros participantes. Como se mencionó anteriormente, gracias a los algoritmos de consenso, todos los nodos de la blockchain ven el estado global que resulta de la ejecución de todas las transacciones en el orden predefinido [7]. Sin embargo, en lugar de enviar cada transacción de forma independiente, lo que aumentaría la complejidad del algoritmo de consenso, los nodos agrupan varias transacciones en bloques, y luego el consenso se aplica para ordenarlos [8]. Cada bloque está encadenado con el anterior mediante la inclusión del hash del bloque anterior. Como resultado, una blockchain crea una secuencia única de transacciones ordenadas y agrupadas en bloques que se vinculan mediante criptografía. La integridad de los datos incluidos en la cadena de bloques está asegurada porque, una vez que se acepta una transacción en un bloque y éste es publicado en la red, todos los nodos conocen estas transacciones y no será factible manipular las transacciones una vez que se publican en la blockchain [9].

Los usuarios pueden tener una o más cuentas. Las cuentas se identifican a través de un identificador público, generalmente derivado de una clave pública utilizando una función hash. Se pueden crear nuevas cuentas de blockchain simplemente generando un par de claves asimétricas y derivar el identificador de cuenta a partir de la clave pública. En general, los identificadores de cuenta no están directamente vinculados con ningún dato del usuario, por lo que pueden ser considerados identificadores pseudo-anónimos. Las transacciones llevan el identificador de la cuenta de origen y destino, y son digitalmente firmadas con la clave privada de la cuenta de origen.

Las blockchain se clasifican principalmente como públicas o privadas. Una blockchain pública (o sin permisonado) está abierta para todos, por lo que cada nodo es capaz de participar en el algoritmo de consenso, y leer/enviar transacciones. Los ledger públicos suelen utilizar una criptomoneda para recompensar a los nodos justos. De esta forma, se favorece el correcto úso de la red y se evita en gran medida los intentos de modificar el estado de la red de forma fraudulenta. Las blockchains públicas pueden ser muy resistentes a la censura, ya que es un desafío difícil comportarse maliciosamente cuando hay muchas partes aleatorias con las mismas responsabilidades.

Alternativamente, las blockchains permisionadas, también llamadas privadas, han sido propuestas principalmente para uso comercial. En este tipo de ledgers, los usuarios necesitan permiso explícito para participar en el consenso de la red, es decir, en el orden de las transacciones. El protocolo y las transacciones son privados y solo están disponibles para los participantes que han sido autorizados para unirse a la red, lo que implica cierta privacidad. En una blockchain privada, las cuentas generalmente se crean utilizando una Autoridad de Certificación (CA) y los usuarios están previamente identificados, por lo que es más fácil de lograr el cumplimiento de regulaciones como Know Your Customer [10] (KYC) para prevenir el lavado de dinero o el Reglamento General de Protección de Datos [11] (GDPR) para la seguridad y privacidad de los usuarios.

Muchas blockchains, públicas y permisionadas, no solo permiten realizar transacciones regulares que modifican los saldos de criptomonedas en el ledger, sino que también tienen la capacidad de almacenar y ejecutar programas públicos y auditables llamados contratos inteligentes o smart contracts. Una vez que se implementa un contrato inteligente en la red, su código se replica en cada nodo, por lo que este programa tiene la misma disponibilidad e integridad que el resto de transacciones regulares. Los contratos inteligentes se pueden utilizar para ejecutar una lógica de negocio predefinida, ya que consisten en funciones que permiten la ejecución de flujos de trabajo automatizados y auditables. Pueden actualizar su propio estado, almacenar variables y (en ciertas blockchains) instanciar otros contratos inteligentes para generan nuevas transacciones.

La operaciones en contratos inteligentes que se ejecutan en una red pública están garantizadas por miles de nodos en todo el mundo, por lo que las funcionalidades de los contratos inteligentes no pueden ser censuradas o paradas [12]. Algunas ventajas de implementar la lógica empresarial en blockchain son que la lógica es pública, auditable, inmutable e inviolable, lo que garantiza que la ejecución siempre será como se define. Por lo tanto, los contratos inteligentes se pueden utilizar para hacer cumplir los términos de un acuerdo entre las partes sin necesidad de intermediarios [13].

Ethereum [14] es la tecnología blockchain más popular en términos del uso de contratos inteligentes y se puede utilizar para generar cadenas de bloques tanto públicas como privadas. De hecho, Ethereum es la plataforma elegida por muchos desarrolladores para implementar aplicaciones usando blockchain [15]. En el ecosistema Ethereum, uno de los principales proyectos es Hyperledger Besu [16].

La blockchain puede considerarse una tecnología disruptiva para los mercados de datos, principalmente debido a sus propiedades inherentes de seguridad, confianza, transparencia e integridad. Un mercado de datos que use blockchain se beneficiaría de la monetización de los intercambios de datos,

la contabilidad de la plataforma o la capacidad de resolver disputas, entre otros.

## III. Proposal

En esta sección, discutimos cómo integrar el servicio de muestreo DEFS dentro del ecosistema i3-MARKET.

### *III-A. i3-Market*

i3-MARKET es un MARKETplace Inteligente, Interoperable, Integrador y facilmente desplegable que permite la federación de datos existentes, pero aún aislados, entre diferentes espacios y mercados. Su visión se basa en el diseño centrado en la descentralización, es decir, la decisión basada en el consenso y la auditabilidad; su objetivo principal es construir una competencia leal y confianza entre las diferentes partes interesadas con el fin de fomentar un ecosistema europeo de comercio de datos digitales.

La tecnología i3-MARKET genera confianza al permitir que todas las partes interesadas participen en igualdad de condiciones. Con tal objetivo, se basa en gran medida en el uso de contratos inteligentes para proporcionar certezas sobre las acciones de los stakeholders y para regular las interacciones requeridas; todo por el cumplimiento del marco legal de la privacidad y la protección de datos de la UE.

i3-MARKET está diseñado para admitir dinero fiat y el token de i3-MARKET, que se construye como un token EIP-1155 [17]. Dado que generar confianza en el sistema es clave, se ha desarrollado un sistema de facturación confiable y auditable diseñado para ambos tipos de pagos. Este sistema previene las siguientes situaciones entre dos pares cualesquiera de un intercambio de datos, a saber, proveedor y consumidor:

- Negar que haya ocurrido un intercambio de datos determinado.
- Afirmar que ocurrió un intercambio de bloques de datos, lo que de hecho, no sucedió.

Como resultado, los proveedores no podrán facturar a un consumidor por un bloque de datos no intercambiado; y los consumidores no podrán rechazar o cancelar un pago por un bloque de datos que fue canjeado con éxito.

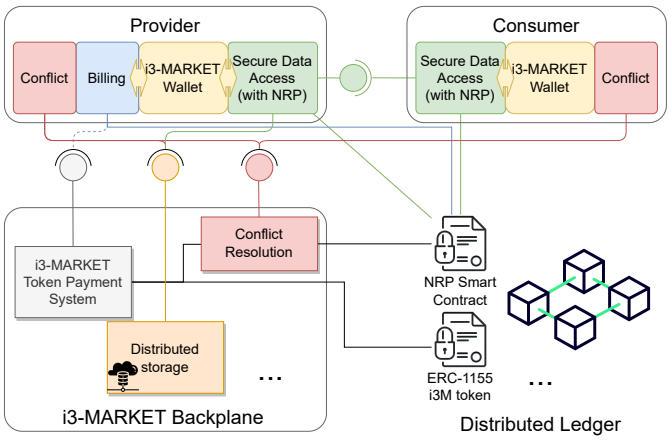



Figura 1. Intercambio de datos i3-MARKET

Para que esto suceda, cada bloque de datos debe intercambiarse utilizando el llamado protocolo de no repudio [18] (NRP), que es la base para activar pagos y resolución de eventuales conflictos.

Las interacciones entre los diferentes stakeholders y componentes de i3-MARKET con respecto al intercambio de datos se muestra en la Figura 1. El NRP, que está respaldado con un contrato inteligente que asume el papel de un ledger confiable, es ejecutado para cada bloque de datos intercambiado, creando pruebas criptográficamente verificables que junto con los datos contabilizados por el libro de cuentas, se pueden utilizar más tarde para probar con éxito la integridad del intercambio de datos. Para mayor comodidad, además del almacenamiento local, el proveedor y el consumidor pueden almacenar las pruebas en el almacenamiento distribuido i3-MARKET, que es seguro y confiable.

El enfoque actual de i3-MARKET es independiente del sistema de pago, dado que la facturación se puede automatizar en base a la verificación de las pruebas NRP y los datos contabilizados en el NRP contrato inteligente, sin importar si se trata de un pago con dinero fiat o basado en tokens. Obviamente, con el primero, las pruebas de NRP solo hacen que la facturación sea confiable en el sentido de que, al ser comprobables e inmutables por naturaleza, son válidos para soportar eventuales disputas del pago en juicio, que son manejados por el sistema de Resolución de Conflictos [19], que puede verificar las pruebas y el ledger (a través del contrato inteligente NRP). Sin embargo, los pagos de i3-MARKET también se pueden realizar con tokens. En este último caso, la actual implementación requiere la interacción con el sistema de pago de tokens i3-MARKET para activar los pagos después de una ejecución exitosa (probada) de NRP.

### *III-B. Servicio de muestreos*

Un servicio gratuito de muestreo de datos está destinado a lograr las siguientes propiedades:

1. **Evaluación de muestras de datos**. Los consumidores obtienen una parte aleatoria del conjunto de datos antes de realizar el pago. El protocolo tolera que cualquier actor pueda manipular el fragmento seleccionado evitando procedimientos injustos.
2. **Garantías de pago**. Los consumidores no pueden recuperar los datos sin realizar el pago y el proveedor no puede retirar el dinero antes de revelar los datos. Por lo tanto, los consumidores tienen acceso a los datos si y solo si el proveedor tiene acceso al pago.
3. **Económicamente rentable**. A pesar de las altas tarifas que últimamente tiene cada blockchain pública, el protocolo es optimizado por diseño para reducir los datos computados y almacenados en el contrato inteligente. Específicamente, los datos involucrados en blockchain son independientes del tamaño de la conjunto de datos. Así, tanto la cantidad de datos almacenados, computados y el número de interacciones con el contrato inteligente son constantes.
4. **No repudio**. El uso de contratos inteligentes implica el uso de transacciones que se registran en una vía pública. Entonces, la descentralización impone que las partes involucradas en la venta no puedan negar su participación una vez realizado el canje.
5. **Animación**. Cada procedimiento en el protocolo está incrustado en tiempos de espera para garantizar que

cada contrato inteligente tiene un final de vida, incluso si una de las partes abandona el proceso por adelantado.

DEFS [5] es un servicio gratuito de muestreo de datos que cumple con las propiedades anteriores.

### *III-C. Integración de DEFS en el ecosistema i3-Market*

En DEFS [5], un contrato inteligente actúa como intermediario durante el procedimiento de pago, asegurando a los proveedores que recibirán el pago por los datos intercambiados.

Además del proceso de pago, DEFS es diseñado con la capacidad de proporcionar muestras aleatorias del conjunto de datos a los consumidores, para que puedan inferir si vale la pena pagar por el conjunto completo de datos, lo que aumenta la confianza del lado del consumidor. Para lograr esto, DEFS utiliza diferentes árboles de Merkle [20]. Un árbol de Merkle es una estructura de datos en la que cada hoja del árbol contiene la información criptográfica, el hash, de un bloque de datos. Luego, cada par de hojas se combinan para generar un nuevo nodo con el hash resultante. Este procedimiento se repite sobre el árbol hasta obtener un único hash: la raíz de Merkle. En el árbol, cada hoja contiene el hash de un fragmento de datos y cada nodo que no es una hora, contiene el hash de los hashes concatenados de sus hijos [21]. La figura 2 muestra un ejemplo de un árbol de Merkle con 4 hojas.

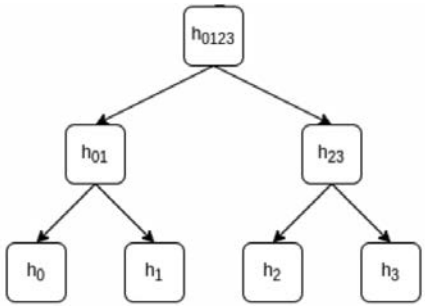



Figura 2. Árbol de Merkle con 4 hojas.

Como se puede ver en la figura 2, para mostrar que cierto valor está almacenado en una hoja del árbol, uno debe proporcionar una prueba de Merkle que consiste en la lista de nodos que una hoja requiere para calcular la raíz de todo el árbol.

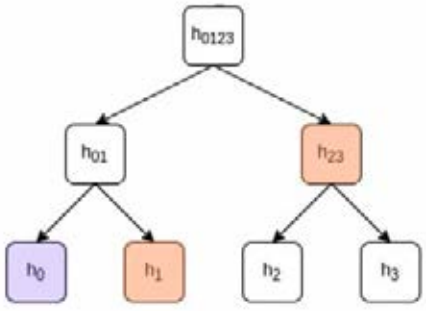



Figura 3. Ejemplo de una prueba de Merkle.

Por ejemplo, una prueba que muestre que $h_0$ está almacenado en el árbol hash de la Figura 3, consistiría en los nodos $h_1$, $h_{23}$, $h_{0123}$.

Tener en cuenta que, con $h_0$ y la prueba de Merkle, cualquiera puede validar la raíz del árbol. Si la raíz coincide con $h_{0123}$, la prueba valida la pertenencia de los datos $h_0$ en el árbol.

En DEFS, el árbol de Merkle permite vincular un conjunto de datos a la raíz del árbol, permitiendo una verificación eficiente y segura de la consistencia y el contenido de los fragmentos de datos. A continuación, describimos brevemente el protocolo y luego, discutimos su integración con el ecosistema i3-MARKET.

Para empezar, el protocolo asume que los proveedores de datos anuncian sus datos al público utilizando medios fuera de la blockchain. En este caso, podemos usar el motor semántico de i3-MARKET para anunciar los datos y sus metadatos asociados para propiciar coincidencias entre consumidores y proveedores. Luego, cuando un consumidor está interesado en un conjunto de datos en particular, se comunica con el proveedor, quien inicia el protocolo DEFS para realizar el intercambio de datos y el pago. DEFS consta de tres fases diferentes:

1. **Preparación del protocolo**: en esta fase inicial, el proveedor prepara no solo los datos a intercambiar, sino también todos los parámetros y material criptográfico necesarios para demostrar que el intercambio de datos es seguro y privado. Más específicamente, el proveedor:
   - Divide el conjunto de datos completo en porciones. Estas porciones se eligen al azar del conjunto de datos (no consecutivamente).
   - Genera una semilla para generar las claves criptográficas simétricas.
   - Utiliza estas k-claves para crear un árbol de Merkle, cuya raíz se puede utilizar para comprobar la exactitud de este material criptográfico.
   - Cifra una permutación aleatoria de las porciones de datos con las claves, obteniendo una versión encriptada y aleatoria de todo el conjunto de datos.
   - Crea otro árbol hash de Merkle usando los hash de estos criptogramas como hojas, cuya root se puede utilizar para verificar la corrección de los criptogramas generados.
   - Habilita una nueva transacción en el componente DEFS de i3-MARKET que desencadena un nuevo contrato inteligente que contiene ciertos parámetros públicos y ese contrato inteligente actúa como intermediario durante el resto del protocolo.

   Si el consumidor tiene interés en obtener el conjunto de datos, el protocolo continúa como sigue:
   - El consumidor recibe el conjunto de datos completo encriptado sobre los datos seguros acceso pero no puede ser descifrado en ese mismo momento.
   - El consumidor consulta el contrato inteligente para obtener la raíz del árbol de criptogramas y verifica que todos los criptogramas pertenecen a este árbol.

   En este punto, todas las entidades (consumidor, proveedor y contrato inteligente) están listas para comenzar la fase de ejecución del protocolo, en la que el consumidor

tendrá acceso a la totalidad conjunto de datos y realizar el pago.

2. **Ejecución del protocolo**: en esta fase, el consumidor podrá obtener muestras del conjunto de datos (gratis) para evaluar si vale la pena pagar, y si es así, lo hará obtener el conjunto de datos y al proveedor se le pagará:
   - El consumidor elegirá algunos trozos para ser revelados y notificará al proveedor sobre el backplane.
   - El proveedor publicará las claves para esas muestras, por lo que el consumidor puede evaluar la calidad del conjunto de datos.
   - Si el consumidor no está convencido, el protocolo termina aquí. Sin embargo, si decide que vale la pena pagar el conjunto de datos, éste realizará una transacción al contrato con el pago en forma de tokens de i3-MARKET. Cada actor obtiene estos tokens en su billetera.
   - El componente requiere que el proveedor publique la semilla (que revelará todas las claves de cifrado) en el contrato inteligente.
   - Si el consumidor puede descifrar correctamente el conjunto de datos, después de un tiempo de espera, el contrato inteligente enviará los tokens a la billetera del proveedor y el termina el protocolo.
   - Si el consumidor puede probar que hubo problemas con el procedimiento anterior, se inicia la fase de resolución de conflictos para obtener la devolución. Para activar este procedimiento de resolución de conflictos, el consumidor no necesita acceder al componente de resolución de conflictos en el backplane. En cambio, el contrato inteligente en sí mismo puede gestionar estos conflictos.

   La siguiente fase sólo será necesaria en caso de que el consumidor considere que es engañado.
3. **Resolución de conflictos***: esta fase es opcional, solo se realiza si el consumidor detecta un mal comportamiento del proveedor. Los siguientes puntos muestran los casos que pueden terminar con un reembolso si ella puede demostrar este mal comportamiento:
   - Las claves no se generan correctamente.
   - Los criptogramas no tienen el formato adecuado.

   Como resultado, el componente DEFS puede permitir eliminar la necesidad de un administrador de contratos inteligentes y permitir pagos directos con el token i3-MARKET entre proveedores y consumidores.

En resumen, el protocolo DEFS en i3-MARKET reemplazará al NRP actual solo en el caso de criptografía los tokens se utilizan para los pagos. Para este objetivo, DEFS debe integrarse en las bibliotecas actuales que manejan datos seguros. Dado que el ecosistema de software i3-MARKET se basa actualmente en JavaScript, la integración de DEFS requiere generar módulos JS apropiados que funcionen tanto en node.js como en navegadores. Además, dado que el proveedor y el consumidor tendrían que interactuar con la blockchain, las bibliotecas deben estar conectadas de forma segura a i3-MARKET Wallet. La figura 4 muestra las interacciones entre las diferentes partes interesadas y componentes.

El servicio DEFS proporciona algunos beneficios con respecto a los servicios actualmente disponibles en el ecosistema

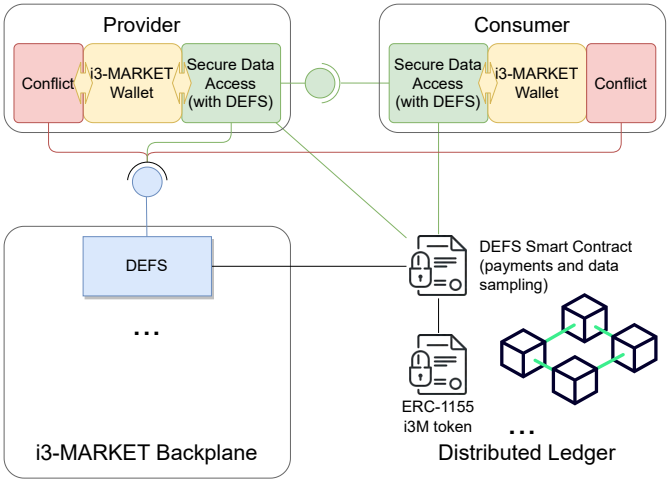



Figura 4. i3-MARKET con DEFS

i3-MARKET. En particular:

1. **Servicio de muestreo**: DEFS permite la posibilidad de obtener una muestra de los datos ofrecidos antes de comprometer un pago, lo que asegura que el conjunto de datos vale el dinero ofrecido.
2. **Pago automático**: DEFS puede administrar el proceso de pago con el token i3-MARKET de forma automatizada y resolver cualquier conflicto que pueda surgir en el intercambio de datos entre consumidores y proveedores.
3. **Eficiencia**: Actualmente, el servicio de acceso seguro a datos de i3-MARKET encripta cada bloque de datos con una clave diferente que se deriva de una semilla diferente. Esta semilla se publica en el contrato después del pago, lo que significa que en la implementación actual de i3-MARKET hay muchas transacciones para almacenar estos valores. Por el contrario, el servicio DEFS genera las claves de cifrado para todo el conjunto de datos a partir de una sola semilla y esto reduce el número de transacciones a la cadena de bloques a solo una.

## IV. Conclusion

Han surgido mercados de datos para facilitar su comercialización. En este artículo, presentamos un mercado de datos descentralizado llamado i3-MARKET que implementa servicios de nueva generación como resolución automatizada de conflictos, no repudio y contabilidad auditable. También describimos un nuevo servicio de muestreo gratuito de datos y cómo integrar este servicio en el ecosistema actual de i3-MARKET. Finalmente, presentamos los beneficios de este servicio que incluyen aumentar la confianza del consumidor sobre los datos que va a comprar, pago automatizado y aumento de la eficiencia. El servicio DEFS presentado solo permite pagar con tokens, pero no con fiat. Como trabajo futuro, estamos planeando dividir este servicio en dos. Uno que gestiona el servicio de muestras y otro que se ocupa de los pagos. De esta forma, el servicio de muestreo podrá funcionar en el ecosistema de i3-MARKET con cualquier tipo de pago.

## Agradecimientos

Esta investigación ha sido financiada por el Programa de Investigación e Innovación H2020 de la Unión Europea,

proyecto i3Market (H2020-ICT-2019-2 número de subvención 871754) y el Ministerio de Ciencia y Educación de España, proyecto TCO-RISEBLOCK (PID2019-110224RB-I00).

## REFERENCIAS


[1] S. Jossen, "The World's Most Valuable Resource is No Longer Oil, But Data," *The Economist*, pp. 1–8, 2017. [Online]. Available: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data

[2] H. Richter and P. R. Slowinski, "The Data Sharing Economy: On the Emergence of New Intermediaries," *IIC International Review of Intellectual Property and Competition Law*, vol. 50, no. 1, pp. 4–29, jan 2019.

[3] E. Comission, "Shaping europe's digital future."

[4] M. Spiekermann, "Data Marketplaces: Trends and Monetisation of Data Goods," *Intereconomics*, vol. 54, no. 4, pp. 208–216, 7 2019.

[5] R. Genés-Durán, J. Hernández-Serrano, O. Esparza, M. Bellés-Muñoz, and J. Muñoz, "Defs - data exchange with free sample protocol," *Electronics (Switzerland)*, vol. 10, no. 12, pp. 1455:1–1455:34, 06 2021.

[6] "Intelligent, Interoperable, Integrative and deployable open source MARKETplace with trusted and secure software tools for incentivising the industry data economy - i3-MARKET. H2020-ICT-2019-2 871754." [Online]. Available: https://www.i3-market.eu/

[7] A. M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. O'Reilly Media, Inc., 2014.

[8] I. Bashir, *Mastering blockchain*. Packt Publishing Ltd, 2017.

[9] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv preprint arXiv:1906.11078*, 2019.

[10] "The joint money laundering steering group," [Accessed: 01-May-2022]. [Online]. Available: https://www.jmlsg.org.uk/guidance/current-guidance

[11] Council of the European Union European Parliament, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," 4 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32016R0679

[12] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50 759–50 779, 2019.

[13] J. Liu and Z. Liu, "A survey on security verification of blockchain smart contracts," *IEEE Access*, vol. 7, pp. 77 894–77 904, 2019.

[14] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project White Paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[15] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–43, 2020.

[16] "Hyperledger besu," [Accessed: 01-May-2022]. [Online]. Available: https://github.com/hyperledger/besu

[17] P. C. J. T. E. B. R. S. Witek Radomski, Andrew Cooke, "Eip-1155: Multi token standard," [Accessed: 04-May-2022]. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1155

[18] J. Hernández-Serrano. i3-market non-repudiation library. [Online]. Available: https://github.com/i3-Market-V2-Public-Repository/SP3-SCGBSSW-CR-NonRepudiationLibrary

[19] ——. i3-market conflict-resolver service. [Online]. Available: https://github.com/i3-Market-V2-Public-Repository/SP3-SCGBSSW-CR-ConflictResolverService

[20] R. C. Merkle, "A certified digital signature," in *Advances in Cryptology — CRYPTO' 89 Proceedings*, G. Brassard, Ed. New York, NY: Springer New York, 1990, pp. 218–238.

[21] F. Haider, "Compact sparse merkle trees," Cryptology ePrint Archive, Report 2018/955, 2018, https://eprint.iacr.org/2018/955.

# Ataques por correlación: Posibilidad de éxito en comunicaciones inalámbricas

Ana Isabel Gómez
Universidad Rey Juan Carlos
Calle Tulipán, s/n.Madrid
ana.gomez.perez@urjc.es

Domingo Gómez-Pérez
Universidad de Cantabria
Avenida de los Castros s/n, Santander
domingo.gomez@unican.es

Andrew Tirkel
Scientific Technologies
Cecil Street, 8, Melbourne
atirkel@bigpond.net.au

***Resumen*—Las comunicaciones inalámbricas se pueden considerar un estándar de facto para la conexión de múltiples dispositivos. La naturaleza física de estas comunicaciones presenta un mayor riesgo de interceptación o secuestro, por lo que se deben preveer los posibles ataques a su seguridad. La intención original de las técnicas de espectro ensanchado era garantizar cierta inmunidad a la interceptación, por ejemplo en la telefonía móvil 3G. En trabajos previos en la literatura se detectó que, cuando se usan técnicas de espectro ensanchado de secuencia directa (DSSS), las secuencias comúnmente utilizadas son susceptibles de ataques de correlación de orden superior, sin explorarse totalmente cuál es la posibilidad de este tipo de ataques. Nuestro trabajo toma como punto de partida las aseveraciones realizadas sobre la seguridad de DSSS y reduce la complejidad de los ataques a órdenes de correlación inferior. Finalmente completamos nuestro estudio con las limitaciones físicas presentes para un atacante y la viabilidad técnica de la construcción de un interceptor que opere con información limitada, así como cuál es su aplicación a las comunicaciones inalámbricas.**

***Index Terms*—Ataques por correlación, técnicas de espectro expandido, Familias de secuencias**

## I. Introducción

Las comunicaciones inalámbricas se están convirtiendo en un estándar de facto para la comunicación entre personas, así como entre máquinas y otros dispositivos. La popularidad de la Internet de las cosas (IOT), las redes de sensores y el control autónomo de vehículos ha aumentado la demanda de interconectividad de un número creciente de dispositivos a altas velocidades de datos. Debido a estas demandas, las tecnologías de comunicaciones inalámbricas celulares y no celulares 5G se han adaptado para ser una opción competitiva para ofrecer estos servicios. Aun así los enlaces de espectro ensanchado dedicados también son empleados por las fuerzas de seguridad, las instituciones financieras o incluso algunas redes de sensores y comunicaciones máquina a máquina.

La seguridad es siempre un problema importante para las comunicaciones inalámbricas. Hay muchas amenazas como la interceptación, la interferencia, el ataque del hombre en el medio, el robo de identidad, etc. Los ataques pueden producirse en varios niveles y capas de la pila de protocolos. Un ejemplo reciente de ataque a los protocolos 4G/5G es Torpedo [1] que afecta a la capa de datos. Se trata de un ataque oportunista que explota el protocolo de paginación utilizado para notificar una llamada entrante o un mensaje sms a un dispositivo de usuario en la celda. Si el ataque Torpedo tiene éxito, se revela información sobre la señal y la ubicación del dispositivo y puede utilizarse para crear un ataque no ciego a la capa física. Sin embargo, este tipo de ataques sólo pueden producirse si la señal es interceptada y puede ser decodificada a través de la capa física. Por estos motivos, lo ideal es que la seguridad de la capa física garantice el secreto de la transmisión, impida la detección de la ubicación y ofrezca una protección contra las interferencias, así como una baja probabilidad de detección. En la capa de datos las principales técnicas provienen de la criptografía, mientras que en la capa física se complementa con la técnicas de estenografía, que se basa en hacer la información imperceptible para todo el mundo excepto para el destinatario. El objetivo es autentificar u ocultar la propia transmisión y, en general, puede aplicarse en diferentes partes de la capa física en la subcapa de radiofrecuencia (RF) o en el nivel de banda base.

Por otro lado, la correlación de orden superior y el análisis espectral de orden superior se han utilizado durante mucho tiempo en la óptica, la acústica y el procesamiento de señales en general, véase [2]. Estos resultados han encontrado también aplicación para mostrar que las secuencias binarias utilizadas en las comunicaciones de espectro expandido pueden exponer información a un posible atacante si se analizan sus correlaciones de orden superior. El primer trabajo en esta dirección fue propuesto por Warner et al. [3], [4], mostrando que las familias de secuencias clásicas utilizadas en CDMA como los códigos Gold y ≪small Kasami≫ son detectables cuando se realiza un ataque ciego que haga uso de la correlación triple. En un trabajo posterior, Adams [5] mostró que las secuencias maximales y los códigos Gold podrían ser detectados en señales interceptadas mediante la observación de patrones en los picos de la correlación de orden triple, sin ruido, y en correlaciones de orden superior ($>3$) cuando la relación señal ruido (SNR) es alta. Los ataques descritos por Adams [5], Gouda [6], [7] y otros [8], [9], [10] requieren que se elimine la portadora y no tienen en cuenta efectos como el ruido aditivo, interferencia, propagación multicamino, etc. Trabajos más recientes [11] tienen en cuenta la propagación pero no la recuperación de la señal a partir de eliminación de la portadora. La única ingeniería inversa documentada con éxito de los códigos de propagación fue lograda por Gao [12] que presentó una búsqueda de fuerza bruta para encontrar los

polinomios de recursión de los códigos Gold truncados de longitud 8184 en las señales interceptadas desde un satélite Galileo. También se recuperaron con éxito otros códigos de las señales del E6. Además, el grupo [13], [14] recuperó los códigos secretos de propagación en el satélite de navegación chino BeiDou. La mayor parte del esfuerzo se invirtió en la adquisición y extracción de la portadora y los datos de navegación, mientras que la ingeniería inversa de los códigos fue la tarea más sencilla.

En este trabajo se presenta una generalización de este enfoque utilizando ataques por correlación de orden superior para estudiar la seguridad de los esquemas basados en DSSS. Nos centramos en la conocida capa física de la telefonía móvil 3G como prueba de concepto para construir un receptor de interceptación ciega, aunque nuestros resultados son aplicables a DSSS en general. Para ello examinamos cuáles son los requisitos para la eliminación exitosa de la portadora y la complejidad computacional necesaria para realizar un ataque por correlación que recupere los códigos empleados en la comunicación.

El trabajo tiene la siguiente organización, la sección II hace una breve introducción a capa física en redes de comunicaciones 3G y resume la complejidad de un ataque de correlación de orden superior para las secuencias conocidas. La sección III trata la construcción de un receptor de interceptación ciega. Por último, la sección IV discute las implicaciones a la seguridad de la capa física de las comunicaciones inalámbricas y el trabajo futuro.

## II. Ataques por Correlación a la capa física

Las técnicas de espectro ensanchado se consideran robustas contra interferencias intencionadas y ofrecen una baja probabilidad de interceptación y detección. Los usuarios comparten un canal común mediante acceso múltiple por división de código (CDMA), haciendo uso de un conjunto de claves precompartidas que permiten generar secuencias de ruido pseudoaleatorio (PN). En las comunicaciones de Radio Frecuencia (RF) pueden encontrarse tres implementaciones distintas de la modulación.

En primer lugar, la portadora de RF puede ser modulada en tiempo, frecuencia o fase por una secuencia pseudoaleatoria. Un receptor legítimo necesita disponer de esta secuencia pseudoaleatoria para eliminar la portadora de RF recibida y recuperar la información. Por esto, se puede recuperar la señal incluso cuando la señal está por debajo del nivel de ruido, siempre que la ganancia de procesamiento sea suficiente.

En segundo lugar, la portadora de radiofrecuencia modulada por la señal puede transmitirse como en el caso anterior y luego "camuflarse" añadiendo otras portadoras coherentes que utilicen también secuencias con baja correlación cruzada con la secuencia de propagación de la portadora de la señal. Para recuperar la información, el receptor previsto debe conocer todas estas secuencias. Esto proporciona la máxima seguridad, ya que para un interceptor, la señal siempre está por debajo del ruido, independientemente de la proximidad al transmisor.

Por último, a la señal portadora se le puede añadir una señal auxiliar para autentificar la transmisión. Esta señal añadida sirve como marca de agua, que también puede ser cifrada para mayor seguridad. Esto no reduce la probabilidad de interceptación, pero puede evitar que un atacante suplante o tome el control del canal, lo que podría tener efectos devastadores en aplicaciones como las redes de sensores.

Los métodos descritos anteriormente pueden aplicarse de forma independiente y simultánea si se requiere una mayor seguridad. En este trabajo limitamos el análisis al primer método.

En el caso de una red 3G se utiliza espectro ensanchado bifásico de secuencia directa a nivel de RF con un código Gold de longitud 63 para el canal de bajada (móvil a base). También tiene dispersión del espectro de la señal de audio que puede considerarse como dispersión en banda base utilizando segmentos cortos de una secuencia maximal. Por esto, para convertir una señal de RF de espectro ensanchado a banda base, el receptor debe tener la frecuencia portadora correcta como oscilador local de referencia y un código de dispersión. La señal de RF en el receptor tiene un umbral de detección de -8 dB con respecto al ruido.

Un enlace seguro suele funcionar con la potencia mínima suficiente para garantizar una SNR de 8-10 dB después de la dispersión, de modo que el posible interceptor también se encuentra en una desventaja equivalente a la ganancia de procesamiento del código de dispersión. Esta ganancia de procesamiento es de 18 dB para los códigos Gold de longitud 63. Los enlaces seguros pueden utilizar códigos mucho más largos, por lo que, en general, el interceptor puede tener una desventaja total de hasta 100 dB en comparación con los usuarios legítimos. De alguna manera, el interceptor necesita compensar parte de esta desventaja beneficiándose de la ganancia de procesamiento sin conocer la secuencia asignada. Trabajos anteriores [15], [5] superan este obstáculo utilizando correlaciones de orden superior para encontrar picos de altura completa no triviales que permitan al interceptor recuperar toda la ganancia de procesamiento. Para la modulación de la secuencia, el polinomio que genera la secuencia asignada puede obtenerse a partir de la ubicación de un pico o picos de tercer orden. A continuación, el interceptor puede generar una réplica local de la secuencia y recuperar la señal utilizando el procesamiento tradicional de correlación de segundo orden. A partir de este punto el interceptor también puede intentar interferir al usuario legítimo o falsear el enlace de comunicación.

### *II-A. Análisis de la correlación*

El enfoque tradicional para estudiar las correlaciones de orden superior es trabajar con vectores de elementos en los cuerpos finitos. Para representar la secuencia PN, sea $\mathbb{F}_2$ el cuerpo finito con 2 elementos. Entonces una secuencia binaria $(s_n)$ de periodo $T$ es una lista finita de elementos en $\mathbb{F}_2$. Asumimos $(s_n)$ que se puede extender periódicamente para todos los índices, i.e. $s_{n+T} = s_n$.

En la mayoría de aplicaciones las secuencias utilizan modulación bifase, en consecuencia solo las secuencias de $\{-1, 1\}$ son consideradas, que pueden construirse a partir de cualquier

secuencia $(s_n)$ del siguiente modo:

$$S_n = (-1)^{s_n}.$$

La correlación de orden $k$ en el caso periódico fue introducida en [16] para analizar las propiedades aleatorias de varias secuencias binarias conocidas. La definición formal es:

$$\theta_k(s_n) = \max_D \left| \sum_{n=0}^{T-1} (-1)^{s_{n+d_1}+s_{n+d_2}+\cdots+s_{n+d_k}} \right|,$$

donde $D = (d_1, \cdots, d_k)$ y $0 \leq d_1 < \ldots < d_k < T$. Una secuencia binaria $s_n$ tiene un pico completo en la correlación periódica de orden $k$ si $\theta_k(s_n) = T$.

Para una secuencia $(s_n)$ de periodo $T$ en $\mathbb{F}_2$, decimos que tiene complejidad lineal $L$ si existe $c_0, \ldots c_{L-1} \in \mathbb{F}_2$ tal que

$$s_{n+L} = \sum_{j=0}^{L-1} c_j s_{n+j}, \quad n \geq 0, \quad c_0, \ldots, c_{L-1} \in \mathbb{F}_2.$$

La secuencia puede ser generada por una recursión lineal de orden $L$, siendo este valor $L$ mínimo.

*Teorema 1:* Dada una secuencia binaria $(s_n)$ de periodo $T$ con complejidad lineal $L$, la secuencia tiene un pico completo en la correlación de orden $2k$ siempre que se cumpla que

$$2^L \leq \binom{T}{k}. \tag{1}$$

*Proof:* Dada la recursión lineal mínima que genera la secuencia $(s_n)$, hay únicamente $2^L$ secuencias diferentes de longitud $T$ que pueden ser generadas por la misma recursión lineal.

Por otro lado, cualquier secuencia $(y_n)$ definida como

$$y_n = \sum_{j=1}^{k} s_{n+d_j}, \quad d_1, \ldots, d_k \leq T. \tag{2}$$

puede ser generada por la misma recursión lineal. Hay $\binom{T}{k}$ combinaciones para elegir $d_1, \ldots, d_k$ donde $d_i < d_j$ if $i < j$. Por claridad, un desplazamiento cíclico de una secuencia está definido con los mismos elementos desplazados por una constate y periódicamente reemplazados. Si se cumple la Ec. (1), entonces existen dos conjuntos diferentes de desplazamientos cíclicos $d_1, \ldots, d_k$ y $e_1, \ldots, e_k$, entonces tenemos que

$$\sum_{j=1}^{k} s_{n+d_j} = \sum_{j=1}^{k} s_{n+e_j} \implies \sum_{j=1}^{k} (s_{n+d_j} - s_{n+e_j}) = 0, \tag{3}$$

implicando que hay un pico completo en la correlación de orden $2k$. ■

En este caso se puede aplicar para el estudio de las construcciones más comunes usadas en CDMA, ver Tabla 7.1 en [17]. Las construcciones de Kasami y Gold son conjuntos de secuencias binarias cuya complejidad lineal es $3n/2$ y $2n$, respectivamente [18]. Un resumen de los resultados en las principales construcciones se puede encontrar en la tabla II-A con los órdenes de correlación necesarios para detectar un pico completo en la secuencia.

Tabla I
CÓDIGOS EN CDMA CON PERIODOS, COMPLEJIDAD LINEAL Y PICOS COMPLETOS EN LA CORRELACIÓN DE ORDEN $k$

| Construcción | Periodo | L | k |
|---|---|---|---|
| m-sequences | $2^n - 1$ | $n$ | 3 |
| Small Kasami | $2^n - 1$ | $3n/2$ | 4 |
| Gold codes | $2^n - 1$ | $2n$ | 6 |
| Large Kasami | $2^n - 1$ | $5n/2$ | 6 |

En resultados previos [5] recobrar la secuencia utilizando la información de la correlación requería que el orden fuera $k = 9$. Para ese caso, la complejidad computacional era de $T^8$ y los resultados estaban mal condicionados respecto al ruido y los errores.

Esto lleva a una conjetura plausible: La mayoría de las secuencias de longitud no prima usadas en CDMA exhiben picos no triviales en correlaciones de bajo orden $k$. Muchas secuencias son particularmente vulnerables dependiendo de su traza o complejidad lineal [19]. La existencia de estos picos en la correlación constituye un riesgo para la seguridad, ya que un atacante que intercepte la señal RF puede recuperar estos picos sobre el ruido y deducir los ≪shifts≫ asociados para recuperar la secuencia por ingeniería inversa.

## III. IMPLEMENTACIÓN PRÁCTICA

Una búsqueda ciega de una señal DSSS (Direct Sequence Spread Spectrum) que recibe una antena es comparable en complejidad a la búsqueda SETI (Search for Extraterrestrial Intelligence). Esto es debido a que un atacante no cuenta con conocimiento a priori de los siguientes parámetros:

- Frecuencia, fase y polarización de la portadora RF.
- Frecuencias de reloj y temporización
- Longitud, alfabeto y polinomio de recursión de la secuencia utilizada.
- Datos transmitidos y su temporización

En contraste, un receptor legítimo conoce todos estos datos, así como la dirección aproximada de recepción de la señal. El atacante no tiene conocimiento de la tasa de muestreo mínima requerida o de la duración necesaria de la interceptación. Por esto el ataque puede fracasar debido a violaciones del criterio de Nyquist, o debido a una interceptación con duración insuficiente. El uso de la máxima frecuencia de muestreo y duración posible no garantiza el éxito del ataque o puede estar más allá de las capacidades del hardware.

Un problema adicional que se suele encontrar es que la señal de RF interceptada suele estar muy por debajo del ruido del canal, por lo que su detección puede requerir una búsqueda exhaustiva. La presencia de una señal solo puede determinarse una vez que se adivina la secuencia de propagación y su temporización y reloj. La señal también puede verse afectada por el desvanecimiento, la propagación multitrayecto o las interferencias de otros usuarios. Si el transmisor o el receptor son móviles, la dirección de llegada puede variar con el tiempo y la señal está sujeta a un desplazamiento Doppler desconocido.

Por tanto un ataque a la señal RF puede tener el objetivo de interceptar la señal RF y de decodificar los datos, sin tener que decodificar la secuencia. Esto es relativamente una tarea sencilla pero esto representa un amenaza limitada para el receptor legítimo. Siguiendo esta idea, Warner [4], [20] propuso el método de la triple correlación aplicado a la señal RF, pudiendo recuperar la portadora para el caso de secuencias maximales. Posteriormente Adams et al. [21], [5] ampliaron este análisis a secuencias maximales, códigos Gold o ≪small Kasami≫ empleados en las señales en presencia de ruido aditivo gaussiano (WGN). En general el ataque de triple correlación se encuentra limitado a las dos primeras secuencias, mientras que un ataque Berlekamp-Massey requiere un relación señal a ruido alta o errores de bit insignificantes.

### *III-A. Receptor de interceptación*

El ataque de ingeniería inversa a una transmisión CDMA de secuencia directa bifásica implica utilizar un proceso de autocorrelación triple. Por esto el receptor es complejo, como se ilustra en la figura 1. Este receptor recupera la frecuencia aproximada de la portadora utilizando un ≪squaring loop≫ [22] con las mejoras requeridas [23], [24] y una DFT (Discrete Fourier Transform) y la fase exacta de la portadora utilizando un bucle digital. La señal de RF se convierte a banda base utilizando esta portadora recuperada y esta señal de banda base se analiza para determinar el periodo del chip [25], la temporización [26] y el periodo del símbolo. A continuación, se realiza una triple correlación y se utiliza la ubicación de los picos de altura completa para recuperar la secuencia de dispersión y con esto la señal de banda base.
Dado que la triple correlación se realiza sobre la portadora de radiofrecuencia modulada recibida, una implementación digital del correlacionador triple está limitada por la velocidad de muestreo del convertidor analógico-digital. En consecuencia, este método sólo puede utilizarse para señales inferiores a 1 Ghz aproximadamente. Para frecuencias por encima de eso, hay dos opciones:

- Correlador analógico utilizando mezcladores de celdas Gilbert y líneas de retardo analógicas como dispositivos SAW.
- Convertir la señal a menos de 1 GHz y realizar la triple correlación digitalmente.

El primer método está limitado a menos de 5 Ghz. El segundo método supone que la frecuencia de la portadora se conoce entorno a un 1 GHz. En este escenario, hay que tener en cuenta que los mezcladores y las celdas Gilbert están diseñados fundamentalmente para dos señales y los dispositivos SAW son conocidos por su distorsión y la necesidad de pre y post amplificadores y circuitos de adaptación.

La figura 1 hace uso de la autocorrelación de tercer orden para la recuperación de la secuencia de dispersión y, por consiguiente, está limitada a secuencias maximales. En el caso de los códigos Gold y otras secuencias utilizadas en CDMA, el requisito de orden de correlación es aún mayor, como se ilustra en la Tabla II-A. Esto añade una considerable complejidad al diseño y aumenta el efecto del ruido en la correlación. Además, para estos códigos multiusuario, la ubicación de los picos revela la familia pero no el código exacto. Éste se puede encontrar mediante la correlación cruzada de la secuencia de banda base recibida con todos los códigos posibles de esta familia. Se trata de una correlación de segundo orden, pero supone un paso adicional en el proceso de recuperación que añade una sobrecarga de cálculo y un posible retraso temporal.

Una posible mejora de lo anterior es la inclusión de la detección de la dirección de llegada. En [27], [27] se exploran dos antenas omnidireccionales separadas físicamente para interceptar una señal de espectro ensanchado que se encuentra bajo el ruido. El ruido que llega a las dos antenas no está correlacionado y las antenas están separadas por $d$, que es de unos pocos metros. Si $d < 1/B$, donde $B$ es el ancho de banda de la señal de espectro ensanchado, esta señal muestra un pico de correlación de segundo orden cuando la señal temprana se retrasa para sincronizarla con la señal posterior. Esto permite recuperar parámetros como el periodo de símbolo de la señal o la longitud de la secuencia entre otros, siguiendo un método patentado en [27].

Sin embargo, hay problemas que deben resolverse antes de su aplicación. El más importante es que el correlacionador acústico-óptico es caro, no es muy preciso y requiere un hardware específico. La tecnología FPGA de ≪RF on chip≫ puede ofrecer una alternativa atractiva. Hay que tener en cuenta los efectos de la multitrayectoria, el desvanecimiento y las interferencias de banda ancha y estrecha, aunque esto puede reducirse mediante filtros espaciales o temporales. Por último, el problema de la detección de la dirección de llegada puede considerarse tridimensional, lo que aumenta el coste, el tamaño y la complejidad del receptor de intercepción. Se pueden realizar mejoras mediante el filtrado espacial utilizando una antena de alta directividad de barrido mecánico o electrónico para restringir o filtrar el ruido y las interferencias entrantes, lo que puede reducir la necesidad de filtrado en el dominio del tiempo, que es costoso desde el punto de vista computacional.

## IV. Conclusiones

Los sistemas CDMA de espectro ensanchado de secuencia directa son vulnerables a un ataque de correlación de orden superior y análisis espectral de orden superior. Sin embargo, nuestros resultados muestran que, aunque se puede construir un correlador de tercer orden a un precio razonable, su uso para atacar una señal inalámbrica como se ha descrito anteriormente [15], [3] tiene un impacto limitado.

Por el contrario, las secuencias de salto de frecuencia y de salto de tiempo se puede demostrar que están a salvo de un ataque de correlación de orden superior analizando su función de ≪auto-hit≫. Las secuencias de salto de tiempo tienen un alto grado de dispersión y suelen tener un límite de ≪auto-hit≫ de 2. Dado que los valores de ≪auto-hit≫ se registran como una operación XOR en valores de chip de $\{0, 1\}$ entre la secuencia original y un desplazamiento , realizar este proceso con más desplazamientos sólo puede reducir el número de

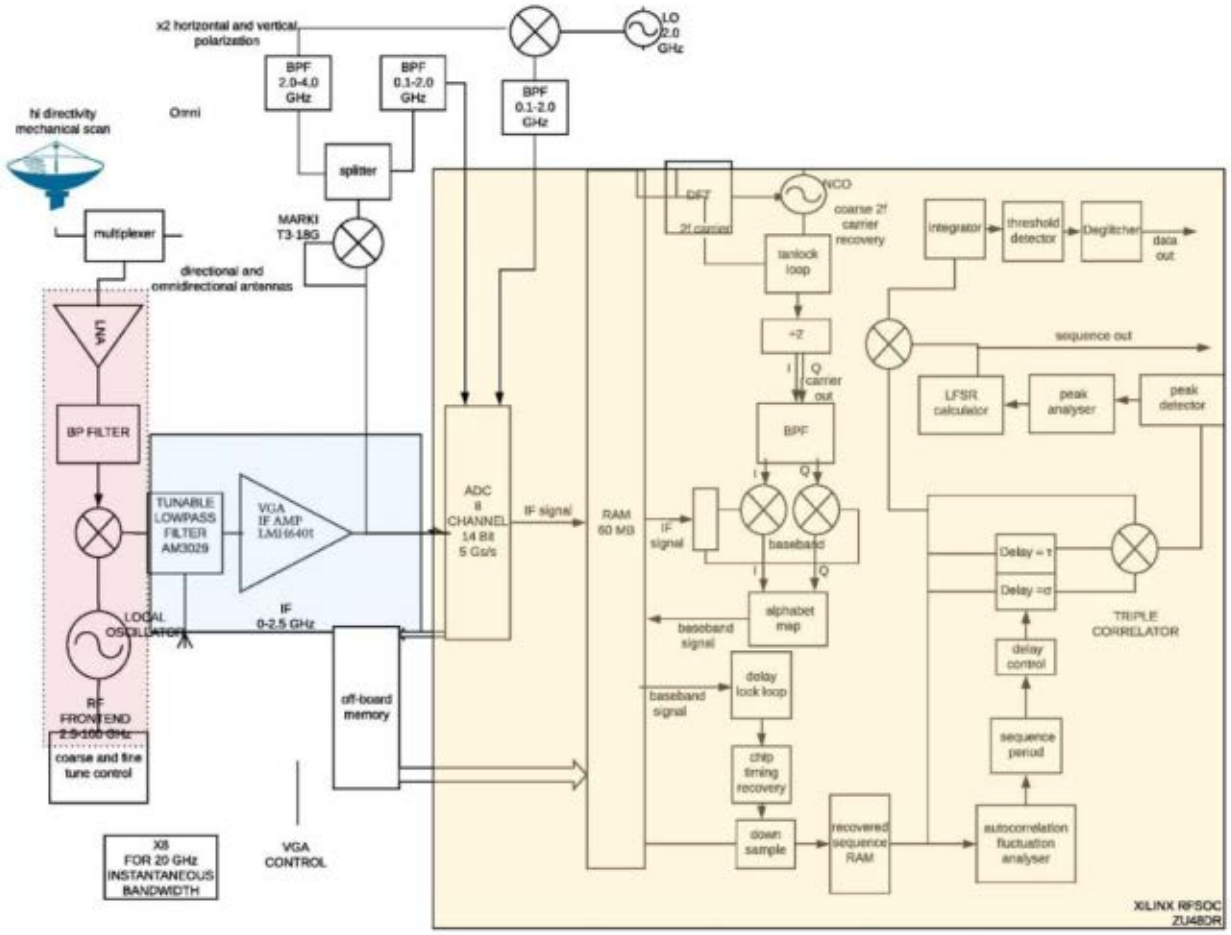



Figura 1. Receptor de interceptación basado en un correlador de orden 3.

≪auto-hits≫ por debajo de 2, por lo tanto, las autocorrelaciones de orden superior de las secuencias de salto de tiempo no pueden exhibir picos de altura completa. Las secuencias de salto de frecuencia pueden interpretarse como matrices bidimensionales en las que una dimensión es la frecuencia y la otra el tiempo. Dado que las secuencias de saltos de frecuencia para acceso múltiple también tienen un límite de ≪auto-hits≫ de 2, se puede hacer un razonamiento similar al anterior.

Las secuencias CDMA de secuencia directa no son dispersas, pero sus operaciones implican la multiplicación aritmética de los símbolos $\{+1, -1\}$, contribuyendo cada bit de la secuencia con un +1 o - 1 a una puntuación de autocorrelación de orden superior. La probabilidad de que todos los resultados de la multiplicación tengan la misma paridad aumenta con el orden de la autocorrelación. De ahí que los sistemas de secuencia directa sean los únicos propensos a los ataques de correlación de orden superior.

Los trabajos futuros se extenderán a otras familias de secuencias propuestas tanto en CDMA o otras como secuencias Glyn, Power Kasami e Hyperovals [28]. Por otro lado se considerarán los efectos de la multitrayectoria, el desvanecimiento, la interferencia deliberada o la interferencia de otros usuarios para diseñar el receptor de interceptación. Finalmente, los sistemas no lineales pueden analizarse utilizando las correlaciones de orden superior, ya que el uso de secuencias binarias pseudoaleatorias como secuencias maximales como fuente de excitación para estudiar la no linealidad puede generar problemas de dispersión [29]. Por esto, estas aplicaciones pueden beneficiarse de la elección de secuencias de excitación libres de comportamiento patológico en su autocorrelación de orden superior.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto ≪Secuencias y Curvas en Criptografía≫ (PID2019-110633GB-I00/ AEI / 10.13039/501100011033).

## REFERENCIAS


[1] S. R. Hussain, M. Echeverria, O. Chowdhury, N. Li, and E. Bertino, "Privacy attacks to the 4g and 5g cellular paging protocols using side channel information." in *NDSS*, vol. 19, 2019, pp. 24–27.

[2] D. R. Brillinger, "An introduction to polyspectra," *The Annals of mathematical statistics*, pp. 1351–1374, 1965.

[3] E. Warner, B. Mulgrew, and P. Grant, "Triple correlation analysis of m-sequences," *Electronics Letters*, vol. 29, no. 20, pp. 1755–1756, 1993.

[4] ——, "Triple correlation analysis of binary sequences for codeword detection," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, no. 5, pp. 297–302, 1994.

[5] E. R. Adams, "Identification of pseudo-random sequences in ds/ss intercepts by higher-order statistics," Cranfield Univ (United Kingdom). Royal Military College of Science, Tech. Rep., 2004.

[6] M. Gouda, "High immunity triple channelized correlation receiver," in *2009 First International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE, 2009, pp. 409–413.

[7] M. Gouda, A. El-Hennawy, and A. E. Mohamed, "Detection of gold codes using higher-order statistics," in *2011 First International Conference on Informatics and Computational Intelligence*. IEEE, 2011, pp. 361–364.

[8] J.-W. Jang and Y.-S. Kim, "Low probability of intercept property of binary sidelnikov sequences," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2015, pp. 733–735.

[9] J.-W. Jang and D.-W. Lim, "Large low probability of intercept properties of the quaternary sequence with optimal correlation property constructed by legendre sequences," *Cryptography and Communications*, vol. 8, no. 4, pp. 593–604, 2016.

[10] G. Burel and C. Bouder, "Blind estimation of the pseudo-random sequence of a direct sequence spread spectrum signal," in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No. 00CH37155)*, vol. 2. IEEE, 2000, pp. 967–970.

[11] H. Mirzadeh Sarcheshmeh, H. Khaleghi Bizaki, and S. Alizadeh, "Pn sequence blind estimation in multiuser ds-cdma systems with multipath channels based on successive subspace scheme," *International Journal of Communication Systems*, vol. 31, no. 12, p. e3591, 2018.

[12] G. X. Gao, J. Spilker, T. Walter, P. Enge, and A. R. Pratt, "Code generation scheme and property analysis of broadcast galileo l1 and e6 signals," in *ION GNSS*, 2006, pp. 1526–1534.

[13] G. Gao, A. Chen, S. Lo, D. Lorenzo, and P. Enge, "Bgnss over china: The compass meo satellite codes,[inside gnss, vol. 2, no. 5," *Jul./Aug*, pp. 36–43, 2007.

[14] G. X. Gao, A. Chen, S. Lo, D. De Lorenzo, T. Walter, and P. Enge, "Compass-m1 broadcast codes in e2, e5b, and e6 frequency bands," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 599–612, 2009.

[15] B. M. Sadler, "Triple-correlation direct sequence receiver," in *[1993 Proceedings] IEEE Signal Processing Workshop on Higher-Order Statistics*. IEEE, 1993, pp. 323–326.

[16] C. Mauduit and A. Sárközy, "On finite pseudorandom binary sequences i: Measure of pseudorandomness, the legendre symbol," *Acta Arithmetica*, vol. 82, no. 4, pp. 365–377, 1997.

[17] V. P. Ipatov, *Spread spectrum and CDMA: principles and applications*. John Wiley & Sons, 2005.

[18] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proceedings of the IEEE*, vol. 68, no. 5, pp. 593–619, 1980.

[19] Z. Chen, A. I. Gómez, D. Gómez-Pérez, and A. Tirkel, "Correlation measure, linear complexity and maximum order complexity for families of binary sequences," *Finite Fields and Their Applications*, vol. 78, p. 101977, 2022.

[20] E. Warner, B. Mulgrew, and P. Grant, "Robust spreading code detection in multipath," *Electronics Letters*, vol. 30, no. 20, pp. 1648–1649, 1994.

[21] E. R. Adams, M. Gouda, and P. C. Hill, "Detection and characterisation of ds/ss signals using higher-order correlation," in *Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications*, vol. 1. IEEE, 1996, pp. 27–31.

[22] J. Garus, R. Studański, R. Was, and R. J. Katulski, "The analysis of the ds cdma transmission—studies in the environment of real signals," in *2011 34th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2011, pp. 461–464.

[23] Z. Qiu, H. Peng, and T. Li, "A blind despreading and demodulation method for qpsk-dsss signal with unknown carrier offset based on matrix subspace analysis," *IEEE Access*, vol. 7, pp. 125 700–125 710, 2019.

[24] A. A. Nasir, S. Durrani, H. Mehrpouyan, S. D. Blostein, and R. A. Kennedy, "Timing and carrier synchronization in wireless communication systems: a survey and classification of research in the last 5 years," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 180, 2016.

[25] R. De Gaudenzi, M. Luise, and R. Viola, "A digital chip timing recovery loop for band-limited direct-sequence spread-spectrum signals," *IEEE Transactions on Communications*, vol. 41, no. 11, pp. 1760–1769, 1993.

[26] F. S. Khodadad, F. Ganji, A. Safaei, and F. S. Khodadad, "A robust pn length estimation in down link low-snr ds-cdma multipath channels," in *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*, vol. 2. IEEE, 2010, pp. 951–955.

[27] A. W. Houghton and C. D. Reeve, "Detection of spread spectrum signals," Sep. 21 1999, uS Patent 5,955,993.

[28] S. W. Golomb and G. Gong, *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

[29] J.-y. Hu, G. Yan, and T. Wang, "Identifying odd/even-order binary kernel slices for a nonlinear system using inverse repeat m-sequences," *Computational and mathematical methods in medicine*, vol. 2015, 2015.

# Comparative analysis of lattice-based post-quantum cryptosystems

Miguel Ángel González de la Torre
Institute of Physical and Information Technologies (ITEFI)
Spanish National Research Council (CSIC)
ma.gonzalez@csic.es

Luis Hernández Encinas
Institute of Physical and Information Technologies (ITEFI)
Spanish National Research Council (CSIC)
luis@iec.csic.es

José Ignacio Sánchez García
Institute of Physical and Information Technologies (ITEFI)
Spanish National Research Council (CSIC)
nacho.sanchez@iec.csic.es

***Abstract*—In recent times, lattice-based cryptographic schemes have been presented as the most promising encryption schemes against quantum computation attacks. Currently, the National Institute of Standards and Technology (NIST) is in the process of setting new post-quantum standards in two categories: key encapsulation mechanisms and digital signatures. During the different phases of the standardization process the proposals presented have been deeply studied in several ways. Their performance has been, after the security of the algorithms, an important feature to be taken into account during the evaluation process. Each submission to the NIST call includes a performance analysis and the NIST reported about the comparison among the candidates. In this communication, we focus our work on how the different lattice-based algorithms perform their implementations. Moreover, we compare each algorithm to the others by running some tests in the same device, to obtain a result as fair as possible.**

***Index Terms*—Lattice-based cryptographic schemes, Performance of implementations, Post-Quantum Cryptography NIST call, FrodoKEM, Kyber, SABER, NTRU**

## I. Introduction

It is well known that current encryption algorithms are in jeopardy because of the development of quantum computation. In fact, there are algorithms, like Shor's algorithm [1], which could break the main asymmetric encryptions if a quantum computer with enough computation power were developed. Moreover, symmetric encryption also face the same threat. From the results by Grover [2], it was thought that the key size should be doubled to obtain the same security level than now, but quantum algorithms which use Simon's subroutines [3] were published in [4] and in [5] which present a higher level of danger for the symmetric encryption.

The 2016 NIST Post-Quantum Cryptography (PQC) call is intended to set the future encryption and digital signature standards. During the selection process the submitted algorithms have been subjected to an intensive scrutiny and, in the third round, there were only four finalists in the Public Key Encryption/Key Encapsulation Mechanism (PKE/KEM) category and three in the signature scheme category. Three of the four PKE/KEM finalists in this round are lattice-based schemes. In fact, in July 7, 2022, the NIST published a first set of selected algorithms in both categories [6]. They are the following: CRYSTALS-KYBER for PKE/KEM and CRYSTALS-DILITHIUM, Falcon, and SPHINCS+ for digital signature algorithms. As one can see, the PKE/KEM selected by now is a lattice-based algorithm.

The main objective of this work is to analyze the differences among the three finalists (Kyber, SABER, and NTRU) and one of the alternatives (Frodo) of the third round, all of them lattice-based candidates. All the algorithms studied achieve INDistinguishability under Adaptive Chosen Ciphertext Attack (IND-CCA) semantic security and present parameter sets targeting the brute force security levels established by NIST. However, in the performance analysis the results are not equal, since the mathematical operations or structural design of each algorithm differs considerably. Our work focuses in locating and studying the structural or mathematical differences that lead to the disparity in performance. To reach this goal we study the algorithms and run the code of each one, all of them in the same environment.

The rest of this communication is organized as follows. Section II contents the problems defined on lattices over which the security of the four considered algorithms reduces; moreover, the four algorithms studied are presented in this section. It is explained how each algorithm works and the operations involved. In Section III we run the tests presented for each algorithm and after an analysis and comparison, we show our results.

## II. Lattice-based problems and algorithms

The Learning With Errors (LWE) problem is parameterized by a positive integer $n$, a prime number $q$, an error distribution $\chi$ over $\mathbb{Z}_q$, and the number of samples $m$.

Let consider a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, $m$ vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$ uniformly chosen and $m$ error integers $e$ sampled from an error distribution $\chi$. The search-LWE$_{n,q,\chi,m}$ problem consists on finding the secret $\mathbf{s}$ having access to $\mathbf{a}_i$, and $b_i = \langle \mathbf{a}_i, s\rangle + e \pmod q$. The decision-LWE$_{n,q,\chi,m}$ problem consists on differentiating the secret $\mathbf{s}$ used to compute the $b_i = \langle \mathbf{a_i}, \mathbf{s}\rangle + e \pmod q$ from a randomly chosen $\mathbf{s}$.

Let $R$ be a lattice with a ring structure, then the Ring Learning With Errors (RLWE) problem is the LWE problem defined as before in $R$, instead of $\mathbb{Z}_q^n$. The underlying ring structure of $R$ allows the public key to be smaller and moreover, the computation to be simpler. Nevertheless, it is considered that this fact also supposes a higher vulnerability

to the attacks that can exploit this structure. In general, the ring considered is $R = R_q = \mathbb{Z}_q[x]/(\phi_n(x))$, where $\phi_n(x)$ is a cyclotomic polynomial. In addition, the Module Learning With Errors (MLWE) problem is also similar to the one defined previously, but in this case, instead of considering a ring structure, a module structure is defined. Finally, a modification of the MLWE problem can be considered in order to obtain the Module Learning With Rounding (MLWR) problem. The nature of the LWE problems implies adding an error, if this is not done, the problem will be easily resolvable. The MLWR variant of the problem instead of sampling from the error distribution adds a deterministic error in the process of key generation and encryption and the reduces to a second smaller module to reach the shared secret.

During the course of the post-quantum standardization process by the NIST, lattice-based cryptography has established as one, if not the best, alternative to current cryptography. In the third round of the NIST post-quantum standardization process, there are four PKE/KEM finalists and three of them are lattice-based algorithms. These algorithms are CRYSTALS-Kyber, SABER, and NTRU. In our study we have included these three proposals together with Frodo, which is not a finalist in the NIST call, but instead it is considered as alternative. Frodo is a latticed-based algorithm, that is only considered as an alternative because its computational cost is higher than the others. One of our objectives is, in fact, to determine how big the difference between Frodo and the other lattice-based algorithms is.

Since the notation used to denote each algorithm of each scheme may be confusing, here is schematically explained the general case. We will denote a PKE as the set PKE $= \{KeyGen, Enc, Dec, M, C, S\}$, where *KeyGen*, *Enc*, and *Dec* are, respectively the key generation, the encryption, and decryption algorithms; and $M$ is the set of messages, $C$ the set of cryptograms, and $S$ the set where the randomness of the algorithms is chosen from. A KEM is defined as the set KEM $= \{KG, Encaps, Decaps\}$, where *KG*, *Encaps*, and *Decaps* are the key generation, the encapsulation, and the decapsulation algorithms, respectively. Moreover, we will consider three hash functions $H_1$, $H_2$ and $H_3$.

Each proposal to the NIST process studied in this survey consists in two schemes: a PKE and a KEM. Each proposal and scheme will be denote as `NameScheme`, where `Name` denotes the name of the algorithm and `Scheme` $\in \{$PKE, KEM$\}$. Moreover, to refer to a specific algorithm for a given `Scheme`, we will use the notation `NameScheme.Algorithm`, where `Algorithm` $\in \{KeyGen, Enc, Dec\}$ if `Scheme` = PKE and `Algorithm` $\in \{KG, Encaps, Decaps\}$ if `Scheme` = KEM.

The submissions studied base their PKE security in a lattice-problem (this means that it can be semantically proved that breaking the PKE is at least as difficult as solving the lattice problem). After designing a PKE, a variant of the Fujisaki-Okamoto (FO) transformation is applied to the PKE, resulting in a IND-CCA secure KEM. Frodo, Kyber, and SABER apply (in the third round submission) the $\text{FO}^{\not\perp'}$ version of this transformation, while NTRU uses the $\text{U}^{\not\perp}_m$ transformation (see [7] and [8]). Table I shows the way in which the KEM algorithms (*KG*, *Encaps*, and *Decaps*) are constructed from the PKE algorithms (*KeyGen*, *Enc*, and *Dec*) and the considered hash functions. This Table I shows the scheme of the $\text{FO}^{\not\perp'}$ transformation used in Frodo, Kyber, and SABER, where

$$\text{FO}^{\not\perp'}[\text{PKE}, H_1, H_2, H_3] = \text{KEM}.$$

The hash functions are defined as follows (see Table I):

$$H_1 \colon \{0,1\}^{pk_{length}} \to \{0,1\}^*,$$
$$H_2 \colon \{0,1\}^{c_{length}+k_{length}} \to \{0,1\}^{K_{length}},$$
$$H_3 \colon \{0,1\}^{*+m_{length}} \to \{0,1\}^{r_{length}+k_{length}}.$$

TABLE I
$\text{FO}^{\not\perp'}$ TRANSFORMATION

| $KG$ |
|---|
| $(pk, sk') \leftarrow KeyGen$ |
| $phk = H_1(pk)$ |
| $s \leftarrow_R S$ |
| $sk = (sk, s, pk, phk)$ |
| **return** $(pk, sk)$ |
| $Encaps(pk)$ |
| $m \leftarrow_R M$ |
| $(r, k) \leftarrow H_3(phk \| m)$ |
| $c = Enc(pk, m, r)$ |
| $K = H_2(k, c)$ |
| **return** $(K, c)$ |
| $Decaps(sk, c)$ |
| $m' \leftarrow Dec(sk', c)$ |
| $(r', k') \leftarrow H_3(phk \| m')$ |
| If $c \neq Enc(pk, m', r')$ |
| **return** $H_2(s, c)$ |
| else **return** |
| $K = H_2(k', c)$ |

For all proposals, the difference between the transformations from a performance standpoint is minimal. Moreover, these transformations do not add any mathematical operation that may be considered costly; instead the KEM performance is directly based on that of the PKE and the times the *KeyGen*, *Enc*, and *Dec* algorithms are called. None of the processes, apart from the PKE algorithms, supposes a huge computational cost. The PKE of each algorithm supposes most of the computational cost of the respective KEM.

### A. Frodo

FrodoKEM is a lattice-based KEM designed with a conservative perspective [9]. As other lattice-based algorithms in the NIST PQC call, Frodo applies the FO-transformation to form an IND-CCA secure KEM from an INC-CPA (INDistinguishability under Chosen Plaintext Attack) PKE. FrodoPKE bases its security in the classic LWE problem. This problem is well known, whereas RLWE or MLWE problems might suffer attacks that use the algebraic structure of the lattice; however, today, there is no attack known for these lattice-based problems that can not be applied to LWE. The design of Frodo is considered conservative and prioritizes security over all the other traits. Alleging that the computational cost is superior to the other algorithms, the NIST considered Frodo as an alternative in the third round and not as a finalist. The Bundesamt für Sicherheit in der Informationstechnik (BSI) maintains its recommendation of Frodo as a PQC

mechanism with a high security margin against future attacks. BSI considers that Frodo has not been included among the finalists of the third round of the NIST PQC call due to efficiency considerations, but there are currently no doubts about its security [10].

As commented before, Frodo submission consists in two schemes FrodoPKE, which is used as an internal subroutine, and FrodoKEM, the main public key algorithm. The scheme that beholds the mathematical operations is FrodoPKE, while FrodoKEM is constructed following the scheme in Table I. Since Frodo is based in the original LWE problem, the operations that FrodoPKE executes are addition and multiplication of matrices with integer entries. These operations are expensive from a performance standpoint when the dimension of the matrix is big. There are three parameter sets proposed in the Frodo submitted to the third round [9], targeting the levels 1, 3, and 5 of security. These are denoted respectively as Frodo-640, Frodo-976, and Frodo-1344, where the number denotes the value of $n$, i.e., the dimension of the lattice $\mathbb{Z}_q^n$ on each case.

### B. CRYSTALS-Kyber

CRYSTALS-Kyber is one of the most promising algorithms among the finalists of the NIST call [11]. In fact, it is the only proposal selected by NIST after the third round [6] in the category of PKE/KEM. In the report of the second round [12] of this call, it was stated that both Kyber and SABER were favorites to become the lattice-based standard. Kyber is based in the MLWE problem, with a structure very similar to Frodo (much more similar than NTRU and slightly more similar than SABER). Kyber uses, as Frodo, the $\text{FO}^{\not\perp'}$ transformation to design a IND-CPA KEM, for which there are three parameter sets proposed in [11]. Even if the structure is similar, the operations performed in the execution of KyberKEM scheme are very different from the matrix multiplication of Frodo. Kyber submission makes use of a Number-Theoretic Transform (NTT) for fast polynomial multiplication, with good results.

### C. SABER

SABER's biggest difference with FrodoKEM and Kyber is that it uses a variant of the LWE problem called MLWR [13]. This version of the LWE problem provides bandwidth advantages without reducing the security of the scheme. In the second round status report from NIST [14], it was stated that the only mild concern with this algorithms was that none of the known security reductions from MLWE to MLWR can be applied to SABER. Despite this, SABER was regarded as one of the favorites, since it has a very good performance and, as the other algorithms, provides IND-CCA security.

### D. NTRU

NTRU is a lattice-based proposal based in the NTRU-problem, whose decisional version reduces to the search RLWE problem. This problem is known even before the NIST standardization process started. The current submission of NTRU is the merger of NTRUEncrypt and NTRU-HRSS-KEM algorithms, also incorporating the results published in [8]. NTRU is slightly behind in performance compared with the two other finalists, mostly because of the key generation algorithm. However, as FrodoKEM, NTRU has been known for a longer time which gives confidence in the security that it provides.

The NTRU submission [15] presents four parameter sets and evaluates the security categories of these sets both in a non-local model and a local model. In the second round status report of the NIST call [12] it was considered the non-local model was equivalent to the CoreSVP model used by other lattice-based algorithms. For this reason, we will consider the NTRU parameter sets for its security in this model. This model implies that the lowest security parameter set of NTRU does not reach the level 1 of security, the two following parameter sets target security 1 and, lastly the highest parameter set targets level 3.

### E. Structural resemblance

Since the four presented lattice-based algorithms base their security in lattice problems; in fact, three of them are based in similar versions of the same problem, it is easy to suppose that there will be resemblance between the algorithms. The algorithms Frodo, Kyber, and SABER share a similar structure, while NTRU differs considerably. Following the LWE structure, the PKE of these three algorithms samples a matrix, $A$, and two error vectors, $s$ and $e$ (in the case of Frodo both are matrices), it calculates $b = As + e$, and sets $pk = (A, b)$ (or $seed_A$ instead) and $sk = (s)$. Clearly, in each case the parameters that affect the mentioned values are different, adapted to the particularities of the algorithm. The encryption also runs similarly: first the three error vectors $s'$, $e'$, $e''$ (again matrices for Frodo) are sampled and then $b' = s'A + e'$ and $v = s'b^T + e''$ are calculated. SABER encryption is different in the way the error is chosen, but the operations presented are the same. We point out this fact because the involved operations suppose the highest computational cost of the respective algorithm, so it is important to keep these operations in mind while analyzing the performance in the following section.

## III. Algorithm comparison and results

The presented lattice-based KEMs reach IND-CCA semantic security and the underlying lattice problem provides them resistance to quantum brute force attacks. Also, every algorithm presented in its submission at least three sets of parameters, targeting the brute force security levels 1, 3 and 5 indicated by NIST [14]. The only exception is NTRU, that, as commented before, does not reach the level 5 completely. After the security criterion stated by NIST, the second one was the performance criterion. The parameter sets presented in the submissions modify, principally, the dimension of the lattice to reach each level of security (other parameters are adjusted depending on the algorithm to ensure security). For compatibility among all proposals considered, in our study we choose the parameters sets that target security 3.

Since the performance of each proposal was determined following the criterion and architecture that each developer group considered appropriate; in our work, in order to have a fair comparison, we have considered the available tests in the given code by using the same architecture for all of them. In particular, we measure the running time (in cycles) and the size of the keys, plaintext (that is, the shared secret, since all

the algorithms are KEMs) and ciphertext (in bytes) of each algorithm.

To compare the performance, in cycles, of the algorithms, we run the code provided by each developer team. We study the performance of the KEM presented in each submission, instead of only the PKE, since the KEMs are the IND-CCA secure algorithms and they are intended to be used on a hybrid scheme. In fact, in our study we determine the performance of the KEM on each case, by analyzing the PKE key generation, and the encryption and decryption processes used in each KEM. Moreover, we have used the public software versions of the algorithms, provided by each developer team on their website and GitHub repository. The code is in C and prepared to be executed in Linux.

For our first run of the test we used a virtual machine. The characteristics of this virtual machine are: UBUNTU 18.04.LTS 64 bits, RAM: 4 GB, acceleration VT-x/AMD-V. Host: Windows 10 PRO 21H1-19043.1645 Intel(R) Core(TM) i7-4790 CPU.

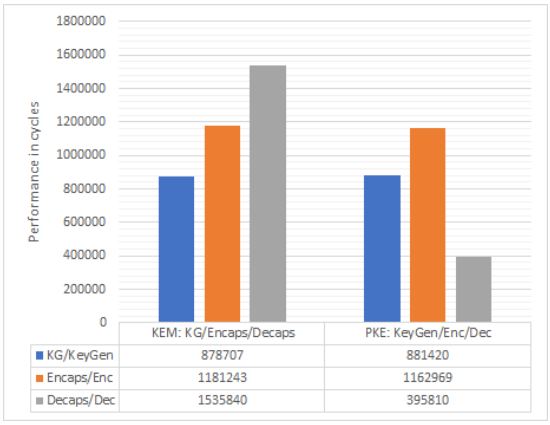



Fig. 1. Our results on KyberKEM and KyberPKE performance

As an example, Figure 1 shows our results of KyberKEM and KyberPKE performance. It can be appreciated that the difference between the two algorithms is minimal except for the decryption and decapsulation algorithms. The reason is that Kyber (and also SABER and FrodoKEM) uses re-encryption. Re-encryption means that the decapsulation receives a ciphertext $c$ and calculates KyberPKE.Dec$(sk, c) = m'$ (in the case of SABER or Frodo, they apply the corresponding decryption algorithm); the plaintext obtained $m'$ is encrypted again to check if the resulting ciphertext is $c$ or a failed one. Table I describes the transformation applied in Kyber to form KyberKEM and how re-encryption is used in the decapsulation algorithm. NTRU does not apply this mechanism in the KEM construction, due to two facts: the decryption algorithm checks if both the received ciphertext and the obtained plaintext (and derived randomness) are all valid (i.e. belong to the set of possible ciphertext, plaintext or randomness), and because NTRUPKE is perfectly correct (i.e. the probability of error in the encryption is 0).

### A. Key size

Table II contains our results, for each security level, of the sizes (in bytes) of public keys ($pk$), secret key ($sk$), ciphertext ($c$) for the algorithms Frodo, Kyber, SABER and NTRU. This data has been obtained from each algorithm respective submission to the third round of the NIST call. For the secret key we just consider the size of the secret information.

TABLE II
PUBLIC AND SECRET KEYS, AND CIPHERTEXT SIZE (BYTES) COMPARISON FOR SOME LATTICE-BASED ALGORITHMS

| Data | Sec. Level | Frodo | Kyber | SABER | NTRU |
|---|---|---|---|---|---|
| $pk$ | 1 | 9616 | 800 | 672 | 1138/930 |
| | 3 | 15632 | 1184 | 992 | 1230 |
| | 5 | 21520 | 1568 | 1312 | - |
| $sk$ | 1 | 10272 | 1138 | 864 | 1450/1234 |
| | 3 | 15664 | 1184 | 1280 | 1590 |
| | 5 | 21568 | 1568 | 1696 | - |
| $c$ | 1 | 9720 | 768 | 736 | 1138/930 |
| | 3 | 15744 | 1088 | 1088 | 1230 |
| | 5 | 21632 | 1568 | 1472 | - |

It is important to fix a definition for the secret key, since there are differences among the four algorithms. In our work, we consider that the secret key is the secret information that the decapsulation algorithm uses to obtain the shared secret. In the lattice-based algorithms studied such secret information is composed by the secret key of the underlying PKE and a random string $s$ (see Table I) used to define the shared secret in the case of an error. Frodo, Kyber, and SABER submissions define the secret key of the KEM as the secret information commented before, but they add the public key and the hash of the public key. The reason why they include this data in the secret information is because in the decapsulation they use re-encryption to check if the decrypted plaintext corresponds to the received ciphertext. However, since the public key and its hash are public, in fact the public key is generated and distributed by the party that executes the decapsulation, we considered not including those in our analysis. For a better understanding of the previous data, in Figure 2 we present a graphic representation of the key and ciphertext sizes (in bytes) for the parameter sets that target security level 3.


Fig. 2. Sizes of key pairs and ciphertext of lattice-based algorithms

The difference in the size of keys and ciphertexts between Frodo and the other algorithms is notable. If, in particular, we analyze the case of the public key (for example) between SABER and Frodo, we can observe that both algorithms define the public key as $(seed_A, b)$ where $A$ is a $n \times n$ matrix and $b$ (denoted by $B$ in the Frodo submission) is obtained doing the operation $b = As + e$, being both $s$ and $e$ error vectors sampled by an error distribution. In the case of

SABER, since the lattice chosen is a polynomial module, $b$, $s$, and $e$ are polynomials that can be expressed as vectors. In Frodo, $B$, $S$ and $E$ are matrices, then $B$ is the public key and $S$ as part of the secret key supposes a much larger amount of memory.

Another important issue while comparing these algorithms is the size of the shared secret obtained (see Table III). Only Frodo presents variance in the number of secure bytes, having less bytes than the other algorithms in the parameter sets that target the security levels 1 and 3.

TABLE III
SIZES OF THE SHARED SECRET (IN BYTES)

| Target Security | Frodo | Kyber | SABER | NTRU |
|---|---|---|---|---|
| 1 | 16 | 32 | 32 | 32 |
| 3 | 24 | 32 | 32 | 32 |
| 5 | 32 | 32 | 32 | - |

## B. Performace

The motivation of our analysis of the performance comes from the fact that the results provided on each algorithm submission are obtained using different architectures. The results we have considered for Frodo as reference were computed by using a 3.4GHz Intel Core i7-6700 (Skylake), in the case of Kyber from an Intel Core i7-4770K (Haswell), the results of SABER were computed on an Intel Xeon E3-1220 v3 (3100 MHz) Hiphop from Supercop, and for NTRU are considered the results that the developers obtained from an Intel Core i7-4770K (Haswell). These results are represented in Figure 3, in which we sampled the performance of the parameter sets of each algorithm that target the security level 3.

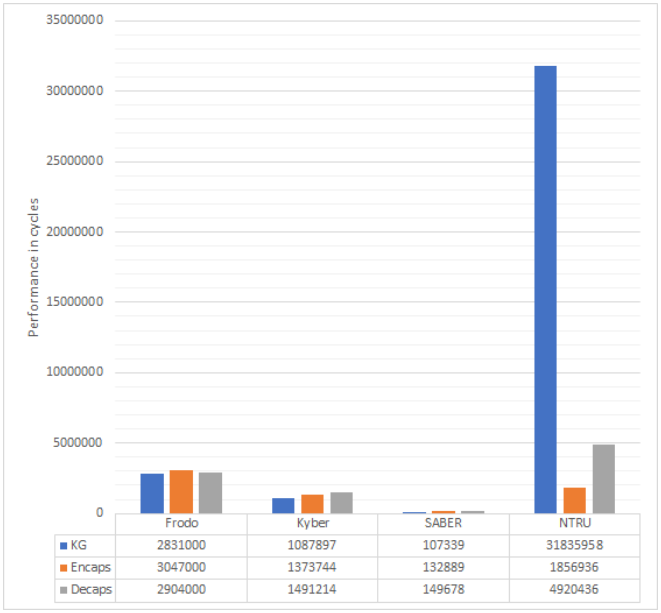



Fig. 3. Performance (cycles) results from the algorithms submissions

Due to the disparity in the architectures used (although Kyber and NTRU use a similar one), our work in this section consists on executing the test of performance provided by each submission with the same architecture, to be able to reach a better understanding of the difference in performance between the schemes. Figure 4 shows the results obtained in our analysis about the performance of the four proposals.

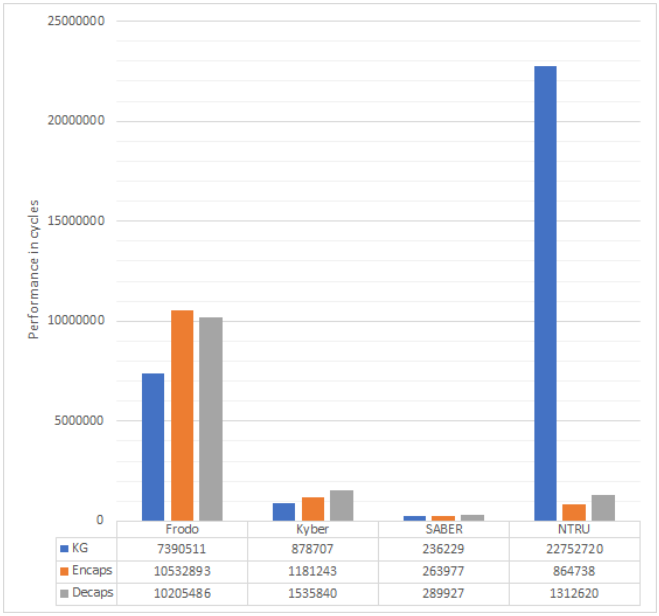



Fig. 4. Performance (cycles) of the lattice-based algorithms

Our results differ from the results provided on each scheme supporting documentation, although we expect that to some extend due to the change of architecture. Also, we used the most up to date version of the code for each scheme and this may have affected the results. Even so, the profile of each scheme in Figure 4 is quite similar to the presented in Figure 3. For example, in our execution, Frodo suffered an increase in the number of cycles measured in all three algorithms, but the profiles of the data that represent the *KG*, *Encaps* and *Decaps* algorithms, in Figures 3–4, do not change radically but remain considerably similar.

Since the computer we used is not dedicated to the environment running the code, we expected a overall worse performance in the virtual machine computation, in comparison with the results provided in the submissions. Our results in Kyber are the closest of all schemes to the originals provided in [11]. In the case of SABER, an increment in the number of cycles of the $100\%$ can be appreciated, but even then the profile of the results is the same as the results considered from [13] represented in Figure 3. Finally, for NTRU our results are better than the provided in [15], for all algorithms and, similarly to the other cases, the relation in performance between the algorithms of NTRU remains close between the submission results and ours.

Our analysis confirms, as the NIST report on lattice-based algorithms does, that all three finalist algorithms Kyber, SABER, and NTRU have a better overall performance than Frodo (except the KeyGen algorithm for NTRU).

We can still make more conclusions based on the results obtained. In Section II-E we pointed out that Frodo, Kyber, and SABER present a very similar structure. In fact, the main difference between the algorithms is the underlying lattice-based problem and the further consequences of these choices. For example, Kyber and SABER lattice dimension takes values 2, 3 and 4 in the three parameter sets presented

on each submission. This means that the matrix $A$ used in both algorithms has dimensions $2 \times 2$, $3 \times 3$ and $4 \times 4$ respectively (the higher the dimension the better the security is). In the case of Frodo, the matrix $A$ has dimensions $n \times n$ and $n$ takes the values $640$, $976$ and $1344$. Since the lattice chosen by Kyber and SABER is a polynomial module, the matrix elements are polynomials, which allows to carry as much information as a matrix in $\mathbb{Z}_q^{n \times n}$, as in the Frodo design. Multiplication of high-dimension matrices (and also, the matrix transposition that is also needed in the encryption of Frodo) is considered a high-cost operation. As explained in [16], Frodo used originally the traditional algorithm to multiply matrices, multiplying a row with a column term by term and then adding the results. This algorithm runs in $O(n^3)$, where $n$ is the matrix dimension.

Bos et al. presented in [16] an alternative algorithm for the matrix multiplication that improves considerably the overall performance of the algorithm (a 20% improvement). This algorithm was considered in a recent update of the Frodo submission, although it is not mentioned in the third round version.

NTRUPKE is very different compared to the other algorithms, so it is difficult to establish a comparison between the mathematical operations that take place underneath. However, the test that we run for NTRU (as in the other cases is the test proposed with the original code) gives the performance of some functions that the PKE uses during the different algorithms. The key generation algorithm calculates the modular inverse of two vectors. These are the operations that suppose a computational cost much higher in the key generation, in comparison with the other algorithms. These results can lead to different conclusions. First of all, it is undoubtedly a drawback to have this kind of performance in one algorithm. However, the key generation algorithm must be executed once, to generate a public key pair, and then these keys can be used to perform various key exchanges. This particularity of the key generation algorithms makes the computational cost of NTRU a less serious issue, since NTRU reaches IND-CCA security to use each key pair more than once. Still, in some implementations, it may be convenient to use what is called ephemeral keys, which means to use a new key pair of the KEM for each key exchange, and in this kind of applications we would not recommend NTRU prior to the other algorithms.

**Acknowledgments**. This work was supported in part by the R+D+i grant P2QProMeTe (PID2020-112586RB-I00), funded by MCIN/AEI/10.13039/501100011033, and in part by the R+D+i grant ORACLE (PCI2020-120691-2), funded by MCIN/AEI/10.1303 9/501100011033, and European Union "NextGenerationEU/PRTR", and in part by the EU Horizon 2020 research and innovation programme, project SPIRS (Grant Agreement No. 952622).

The authors want to express their gratitude to the reviewers for their valuable comments, which have helped to improve this manuscript.

## REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999, https://doi.org/10.1137/S0036144598347011.

[2] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Physical Review Letters*, vol. 79, no. 2, pp. 325–328, 1997, https://doi.org/10.1103/PhysRevLett.79.325.

[3] D. Simon, "On the power of quantum computation," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1474–1483, 1997, https://doi.org/10.1137/S0097539796298637.

[4] X. Bonnetain, A. Hosoyamada, M. Naya-Plasencia, Y. Sasaki, and A. Schrottenloher, "Quantum attacks without superposition queries: the offline Simon's algorithm," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT 2019, Lecture Notes Comput. Sci.*, vol. 11291, 2019, pp. 552–583, https://doi.org/10.1007/978-3-030-34578-5_20.

[5] ——, "Quantum attacks without superposition queries: the offline Simon's algorithm," *arXiv*, 2020, https://arxiv.org/abs/2002.12439.

[6] NIST, "Post-quantum cryptography. selected algorithms 2022," On-line publication, 2022, https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[7] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in *Proc. 15th International Conference Theory of Cryptography TCC'2017, Lecture Notes Comput. Sci.*, vol. 10677, 2017, pp. 341–371, https://doi.org/10.1007/978-3-319-70500-2_12.

[8] T. Saito, K. Xagawa, and T. Yamakawa, "Tightly-secure key-encapsulation mechanism in the quantum random oracle model," in *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 2000, Lecture Notes Comput. Sci.*, vol. 10822, 2018, pp. 520–551, https://doi.org/10.1007/978-3-319-78372-7_17.

[9] E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila, "FrodoKEM learning with errors key encapsulation (Round 3 Submission)," Online publication, 2021, https://frodokem.org/#spec.

[10] BSI, *Cryptographic Mechanisms: Recommendations and Key Lengths, version 2022-01*, Bundesamt für Sicherheit in der Informationstechnik, BSI TR-02102-1, 2022/01/28, 2022, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf.

[11] R. Avanzi, J. Bos, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber," Online publication, 2020, https://pq-crystals.org/.

[12] NIST, "Status report on the second round of the NIST post-quantum cryptography standardization process," U.S. Department of Commerce, Report NISTIR 8309, Tech. Rep., 2020, https://doi.org/10.6028/NIST.IR.8309.

[13] A. Basso, J. M. B. Mera, J.-P. D'Anvers, A. Karmakar, S. Sinha, M. V. Beirendonck, and F. Vercauteren, "SABER: Mod-LWR based KEM (Round 3 Submission)," Online publication, 2020, https://www.esat.kuleuven.be/cosic/pqcrypto/saber/.

[14] NIST, "PQC standardization process: Third round candidate announcement," Online publication, 2020, https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement.

[15] C. Chen, O. Danba, J. Hoffstein, A. Hulsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang, "NTRU," Online publication, 2020, https://ntru.org/.

[16] J. W. Bos, M. Ofner, J. Renes, T. Schneider, and C. van Vredendaal, "The matrix reloaded: Multiplication strategies in FrodoKEM," Cryptology ePrint Archive, Report 2021/711, 2021, https://ia.cr/2021/711.

# La Publicación de Trayectorias: un Estudio sobre la Protección de la Privacidad

Patricia Guerra-Balboa
KASTEL Security Research Labs
Instituto Tecnológico de Karlsruhe
patricia.balboa@kit.edu

Àlex Miranda-Pascual
Dept. Ingeniería Telemática
Universidad Politécnica de Cataluña
alex.miranda.pascual@upc.edu

Javier Parra-Arnau
Dept. Ingeniería Telemática
Universidad Politécnica de Cataluña
javier.parra@upc.edu

Jordi Forné
Dept. Ingeniería Telemática
Universidad Politécnica de Cataluña
jordi.forne@upc.edu

Thorsten Strufe
KASTEL Security Research Labs
Instituto Tecnológico de Karlsruhe
thorsten.strufe@kit.edu

***Resumen*—El análisis de las trayectorias encierra numerosas promesas, desde mejoras en la gestión del tráfico hasta recomendaciones de ruta, o incluso en el desarrollo de infraestructuras. Sin embargo, conocer los lugares en los que uno ha estado es extremadamente invasivo. Por ello, surge la necesidad de anonimizar bases de datos de trayectorias, preservando las estadísticas globales útiles para el análisis, mientras que la información específica y privada de los individuos permanece inaccesible.**

**En este trabajo analizamos el estado del arte en la publicación de trayectorias con garantías de privacidad, revisando nociones, mecanismos y métricas de utilidad. De este análisis concluimos limitaciones de las propuestas actuales y teniendo en cuenta tanto los problemas de privacidad como los de utilidad, esbozamos oportunidades de investigación para el desarrollo de mecanismos eficaces bajo una protección específica y rigurosa.**

***Index Terms*—privacidad de trayectorias, anonimización, nociones sintácticas y semánticas, utilidad, privacidad diferencial**

## I. Introducción

Día a día, el valor e interés de los datos de trayectoria se vuelven más notables, no solo en nuestras vidas, sino también entre las empresas de análisis de datos. Al mismo tiempo, la capacidad de los dispositivos personales (como los *smartphones*) y de los sistemas de navegación para recoger, procesar y analizar con precisión estos datos está creciendo a un ritmo nunca visto, gracias a los recientes avances tecnológicos. La gestión del tráfico, la planificación urbanística, el diseño de sistemas de transporte, la predicción de rutas o la seguridad pública son solo algunas de las muchas aplicaciones que se benefician del análisis de trayectorias [1].

A pesar del bien económico y social que supone este análisis, las tensiones relativas a los riesgos para la privacidad son cada vez mayores [2], [3].

Las trayectorias son secuencias de coordenadas espaciotemporales (localizaciones y tiempos). Dada la cantidad de información almacenada en ellas, las trayectorias suponen un gran riesgo de privacidad. Por ejemplo, delatan fácilmente cuándo y durante cuánto tiempo un individuo desarrolla una actividad o visita un lugar, lo que permite a un atacante inferir circunstancias y tendencias que afectan a aspectos privados de su vida, como su estado de salud, sus creencias religiosas, sus relaciones sociales, o sus preferencias políticas o sexuales.

Por otra parte, anonimizar las trayectorias no es tarea fácil, como observaremos en las siguientes secciones. Métricas y técnicas bien conocidas en el campo de la privacidad de datos, como el $k$-anonimato [4] o la privacidad $\epsilon$-diferencial ($\epsilon$-DP, por sus siglas en inglés) [5], no son aplicables de forma inmediata a estos conjuntos de datos secuenciales y de gran dimensión, y las garantías de privacidad que prometen en el campo a menudo son poco claras.

Asimismo, la singularidad de los desplazamientos humanos hace que, con poco conocimiento previo sobre los objetivos (como su lugar de residencia o trabajo), los adversarios puedan mejorar sus ataques contra los algoritmos de protección [6], [7]. Además, se pueden reconstruir las trayectorias originales utilizando mapas de carreteras, límites de velocidad o modelos simples de correlación espaciotemporal incluso tras aplicar algunos procesos de anonimización, como ofuscaciones. En este contexto, las investigaciones demuestran que conocer solo cuatro puntos espaciotemporales a baja resolución es suficiente para identificar de forma única al $95\,\%$ de los individuos de una base de datos de escala nacional [8].

Además, las nuevas trayectorias podrían seguir siendo "semánticamente" idénticas, de manera que la información sensible del usuario siguiese estando expuesta. Por ejemplo, tras añadir ruido a las coordenadas del usuario encontramos que las nuevas coordenadas siguen estando dentro del mismo *parking* de un centro comercial, luego, pese a la modificación numérica, la semántica sigue idéntica después de la anonimización, con lo que no se ha proporcionado ninguna protección eficaz.

Por último, numerosas aplicaciones de análisis de datos de trayectorias requieren de la publicación continuada y secuencial de datos, como el control y gestión del tráfico a tiempo real. Sin embargo, garantizar la privacidad en este escenario es una tarea desafiante. Los métodos de anonimización sintáctica no pueden ofrecer privacidad en un escenario de actualizaciones y republicaciones de la base de datos, ya que, aunque cada publicación sea, por ejemplo, $k$-anónima, el acceso al historial de publicaciones permite contrastar y romper la $k$-anonimidad. La privacidad diferencial, aunque goza de la propiedad de composición y preserva (hasta cierto punto) la garantía de privacidad después de actualizaciones

repetidas de datos, viene lamentablemente al coste de una degradación significativa de la utilidad de los datos [9], [10].

En este artículo, examinamos el estado del arte sobre la publicación de trayectorias con garantías de privacidad y utilidad. Nuestro análisis de la tecnología de anonimización actual abarca las nociones sintácticas y semánticas de privacidad, y se organiza en métricas de privacidad, utilidad, y mecanismos de anonimización. A partir de este análisis, establecemos varios retos, derivados de las ideas y también de las limitaciones de las propuestas existentes en la anonimización de trayectorias, identificando oportunidades para futuras investigaciones.

El resto del artículo se organiza de la siguiente manera. En primer lugar, se presenta el estado del arte de la anonimización de datos de trayectorias. A continuación, se exponen las limitaciones y los problemas identificados en nuestro análisis de la literatura. Por último, se analizan oportunidades y soluciones, y se formulan algunas observaciones finales.

## II. Trayectorias y Bases de Datos

Hay diversos tipos de trayectorias. Las más sencillas consisten en una secuencia ordenada de puntos espaciotemporales: $T = (x_1, y_1, t_1) \to \cdots \to (x_n, y_n, t_n)$. Existen representaciones más complejas denominadas *trayectorias semánticas*, en las que se considera adicionalmente la dimensión categórica. En estas, cada punto es un *punto de interés* (PDI), es decir, coordenadas dotadas de un significado semántico, como un nombre o una descripción, y posiblemente otra información como el número de visitantes u horarios de apertura.

Las *bases de datos* de trayectorias consisten en múltiples trayectorias de diferentes individuos (u *objetos en movimiento*) sobre una región común. Sin embargo, existen notables diferencias entre ellas. Algunas bases de datos consisten en trayectorias de igual longitud, algunas de las cuales, además, están recogidas periódicamente (es decir, cada trayectoria tiene un punto cada $t$ tiempo) [11]; mientras que otras son menos regulares, con puntos que solo aparecen cuando el usuario llega (o permanece) en un lugar notable [12].

## III. Midiendo la Privacidad y Utilidad

**Métricas de privacidad.** El objetivo del *control de divulgación estadístico* (SDC, por sus siglas en inglés) es permitir la extracción de estadísticas globales útiles sobre toda la población, pero evitando que se pueda aprender nueva información sobre algún usuario en particular. Existen dos familias conocidas de nociones de privacidad en este campo [13]: las nociones *sintácticas* y *semánticas* [14].

En el caso sintáctico, los representantes clásicos son el $k$-*anonimato* [4] y sus extensiones (como $l$-*diversidad* [15] y $t$-*cercanía* [16]). Se han hecho varios intentos de adaptar estas nociones para los datos de trayectoria. Por ejemplo, se dice que un conjunto de datos satisface $(k, \delta)$-*anonimato* [17] si, para cualquier trayectoria, existen $k-1$ otras trayectorias tales que en cada paso de tiempo las localizaciones correspondientes no están a más de $\delta/2$ de distancia. Asimismo, un conjunto de datos es $k^m$-*anónimo* [18] si cada subtrayectoria de longitud como máximo $m$ está contenida en al menos $k$ trayectorias diferentes. Otras nociones, como el $k^{\tau,\epsilon}$-*anonimato* [19] o la $(K, C)_L$-*privacidad* [20], consideran y acotan la información adicional que se le permite aprender al atacante.

En el caso semántico, la privacidad diferencial (DP) [5] es probablemente la noción más conocida. Bajo un marco matemático formal, esta noción establece una cota superior $\epsilon$ sobre la posibilidad de éxito de un atacante que intenta inferir la información real de un usuario a partir del output del mecanismo. Formalmente, un mecanismo aleatorio $\mathcal{M}$ es $\epsilon$-*diferencialmente privado* ($\epsilon$-DP) [5] si para todo par de bases de datos vecinas $D, D'$ (i.e., que difieren en una única entrada) y todo $\mathcal{S} \subseteq \mathrm{Range}(\mathcal{M})$,

$$\mathrm{P}\{\mathcal{M}(D) \in \mathcal{S}\} \leq \mathrm{e}^{\epsilon} \cdot \mathrm{P}\{\mathcal{M}(D') \in \mathcal{S}\}. \qquad \text{(III.1)}$$

DP es una mera garantía matemática sin ninguna semántica asociada, por lo tanto, es importante especificar exactamente qué información está protegida por ella. La noción original de DP (*user-level*) pretende proteger la existencia completa de los registros de un usuario en una base de datos, es decir, toda contribución de un usuario es indetectable viendo el *output* de un mecanismo DP.

Un gran número de variantes de DP adaptan el concepto *vecindad* de bases de datos: Este concepto determina lo que se considera una única entrada en la base de datos, y, por tanto, que es lo que quedará protegido bajo un mecanismo DP. El *nivel de granularidad* se refiere a esta definición de vecindad en una noción de privacidad. El primer nivel aparece con la privacidad *event-level* [21], [22]. En el mundo de los datos de trayectorias, la privacidad *user-level* protege todo el historial de la trayectoria de cualquier usuario, mientras que la *event-level* protege cada punto espaciotemporal (es decir, un evento). De este modo, el atacante puede saber si un usuario pertenece a la base de datos (ya que cambiar todos los datos de su trayectoria produce un efecto no acotado por $\epsilon$), pero no debería ser capaz de inferir si en un momento determinado estuvo en unas coordenadas o no.

Kellaris *et al.* [23] introducen un punto medio entre de ambas nociones, la privacidad $w$-*event*, que, en cambio, protege $w$-ventanas de eventos secuenciales (véase también la Fig. 1). Esta noción se convierte en privacidad *event-level* con $w = 1$ y *user-level* cuando $w$ es la longitud máxima de una trayectoria en la base de datos. De esta forma, el atacante puede seguir infiriendo la presencia de un usuario en la base de datos, pero es incapaz de determinar si una secuencia de $w$ localizaciones pertenece a su trayectoria o no.

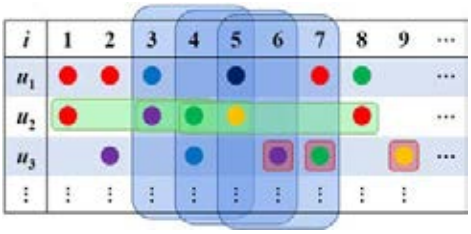



Figura 1. Un ejemplo de una base de datos no recogidas periódicamente, donde los puntos de colores representan diferentes localizaciones. Los recuadros redondeados representan el alcance de la privacidad *event-level* (rojo), $w$-*event* (azul) y $\ell$-*trajectory* (verde), para $w = \ell = 3$. Observe que los cuadros azules ($w$-ventanas) siempre abarcan $w$ intervalos de tiempo, independientemente de cuántos puntos incluyan, y que los cuadros verdes ($\ell$-trayectorias) incluyen siempre $\ell$ puntos, independientemente del número de intervalos de tiempo que abarcan.

Como mejora de la privacidad $w$-*event*, Cao y Yoshikawa [12] adaptan la noción de $\epsilon$-DP específicamente para

trayectorias no recogidas periódicamente en el tiempo, obteniendo la privacidad $\ell$-*trajectory* (véase la Fig. 1). Aquí se cambia la noción de $w$-ventana por la de $\ell$-trajectoria, siendo esta una secuencia de $\ell$ localizaciones consecutivas visitadas por un usuario.

**Métricas de utilidad.** Se han propuesto diversas métricas para cuantificar la utilidad de las trayectorias anónimas. Dado que las técnicas de anonimización siempre llevan asociadas una pérdida de utilidad, uno de los principales objetivos de los mecanismos es minimizar esta pérdida al máximo.

Una forma de medir la utilidad en el número o la proporción de datos que quedan inalterados tras el saneamiento. La *preservación de localizaciones* [24] es un buen ejemplo: Alta utilidad se preserva cuando las trayectorias saneadas incluyen las localizaciones presentes en las trayectorias originales, y no falsas. Asimismo, se puede medir la utilidad como la minimización del número de localizaciones no descartadas.

No obstante, normalmente, mover las coordenadas de la trayectoria unos pocos metros no altera la utilidad. Por lo tanto, la preservación también puede definirse en función de un factor de similitud. Una forma popular de cuantificar este factor es mediante el uso de *métricas de similitud*, funciones que cuantifican la diferencia entre dos trayectorias. Estas incluyen, por ejemplo, las medidas clásicas como la distancia euclidiana [17] o de Hausdorff [11], [25], [26], o EDR [27], entre otras. Por otro lado, en [28] se considera una medida que hace una media geométrica de las distancias espacial, temporal y categórica; teniendo así en cuenta cada una de las dimensiones de las trayectorias semánticas.

Otras métricas menores se basan en la preservación de propiedades específicas, como son la *preservación de la longitud de trayectorias* [29], [30], *secuencias frecuentes* [31] y *lugares más visitados* [29], [28]. Esta información se obtiene mediante funciones de consulta $q$, y, por lo tanto, se puede medir la preservación usando una *función de error para consultas* (Ec. III.2) [17], [31], [32], [33], [30], que compara los datos anonimizados $D'$ frente a los datos originales $D$:

$$\text{error}(q) = \frac{|q(D) - q(D')|}{\text{máx}\{q(D), b\}}, \qquad \text{(III.2)}$$

donde $b$ es una cota usada para funciones de consulta extremadamente selectivas.

Finalmente, una última categoría de métricas de utilidad se basaría en asegurar resultados realistas, es decir, que eviten localizaciones consecutivas inalcanzables en el tiempo dado o en lugares geoespacialmente incoherentes [24], [28].

## IV. Mecanismos: Logrando Privacidad

En esta sección, examinamos los mecanismos de anonimización de trayectorias más relevantes que cumplen las nociones de privacidad mencionadas. En primer lugar, describimos los mecanismos que ofrecen garantías sintácticas, para después abarcar los que garantizan las nociones semánticas.

**Privacidad sintáctica.** Existen tres técnicas principales para proporcionar privacidad sintáctica [34]: *supresión*, la eliminación de aquellas localizaciones, o trayectorias enteras, que presentan un riesgo de reidentificación; *generalización*, que hace que los registros sean indistinguibles de otros reduciendo la precisión de las trayectorias o agrupando los datos en grupos más grandes; y el *enmascaramiento (perturbativo)*, que comprende una multitud de técnicas incluyendo la *perturbación* de los datos, basada en la adición de ruido; el *clustering* o *microagregación* de localizaciones; y la *generación de trayectorias falsas*, entre otras muchas. La gran mayoría de las tecnologías de anonimización combina varias de estas técnicas. A continuación, describimos sucintamente los trabajos más relevantes.

El primer mecanismo que utiliza $k$-anonimato para abordar la anonimización de trayectorias es *Never Walk Alone* (NWA) [17]. Consiste en un algoritmo voraz que agrupa las trayectorias en *clusters* y luego realiza una translación espacial (enmascaramiento) para lograr $(k, \delta)$-anonimato. También suprime las localizaciones atípicas. Los mismos autores introducen posteriormente variaciones mejoradas como W4M [27].

Otros enfoques basados en el enmascaramiento incluyen un método perturbador descrito en [24]. Este método agrupa las trayectorias con microagregación y luego permuta las localizaciones usando la generalización de atributos sensibles y la supresión local. Poulis *et al.* [18] proponen métodos en los que las localizaciones más cercanas se fusionan en pares hasta que se satisface $k^m$-anonimato. Otros métodos que agrupan y suprimen trayectorias incluyen TOPF [35].

Otra combinación popular es la de las técnicas de supresión con las de generalización: En [36], puntos específicos son eliminados o se sustituyen por regiones en una cuadrícula de celdas, obteniendo trayectorias generalizadas. De forma similar, GLOVE [37] elimina primero todas las localizaciones periféricas, y luego emplea una generalización no uniforme, de modo que cada punto se somete a una reducción mínima para garantizar $k$-anonimato. Basándose en este, Tu *et al.* [38] introducen los primeros mecanismos que satisfacen $k$-anonimato, $l$-diversidad y $t$-cercanía al mismo tiempo. Otros métodos usando esta combinación aparecen en [39], donde las localizaciones se clasifican en áreas y luego se agrupan.

Por último, Chen *et al.* [20] definen un método basado en la supresión local, que elimina solo algunas instancias del conjunto de datos para garantizar $(K, C)_L$-privacidad, preservando al mismo tiempo los puntos espaciotemporales y las secuencias frecuentes.

**Privacidad semántica.** A continuación, examinamos los algoritmos de anonimización que buscan publicar bases de datos de trayectorias con garantías de privacidad diferencial.

Los *conteos ruidosos* [32], [31] es un enfoque común basado en añadir ruido al conteo de las trayectorias o secuencias de las mismas. El ejemplo fundamental es [32], que construye un árbol que almacena toda la base de datos. En cada nivel $n$ del árbol se almacenan los conteos de las secuencias de $n$ localizaciones ($n$-gramas) que se obtienen recorriendo el árbol desde la raíz hasta cada uno de los nodos de dicho nivel. Estos conteos se alteran con ruido a partir de la distribución de Laplace (incluidos los conteos que originalmente eran cero), obteniendo así DP. Además, en [31], se propone un método para generar datos sintéticos a partir de los $n$-gramas publicados.

Los *mecanismos basados en clustering* constituyen otro enfoque usado en la privacidad de trayectorias [11], [26], [20]. La idea es fusionar localizaciones de diferentes trayectorias en cada tiempo siguiendo una partición probabilística basada en el mecanismo exponencial.

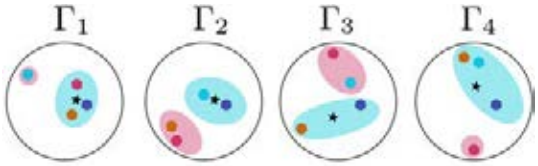



Figura 2. Ejemplo de la anonimización mediante técnicas de *clustering*. Las diferentes trayectorias se representan en diferentes colores, con puntos que corresponden a las localizaciones físicas en cada paso de tiempo. Las áreas coloreadas representan los *clusters* escogidos, y las estrellas denotan sus respectivos centroides. En este ejemplo, las trayectorias tienen longitud $|T| = 4$ y la partición seleccionada contiene $m = 2$ subconjuntos.

Más concretamente, los autores proponen una función de puntuación para medir las distancias entre las trayectorias que cruzan las localizaciones en cada instante de tiempo. Utilizando el mecanismo exponencial y esta función, se elige una de las $m$-particiones candidatas de $\Gamma_i$, el universo de localizaciones en el tiempo $t_i$. Luego, todas las localizaciones de cada subconjunto se agrupan y se sustituyen por su correspondiente centroide (véase la Fig. 2). Las nuevas trayectorias se construyen reconectando los centroides correspondientes, con su recuento atribuido siguiendo el mecanismo de Laplace.

Por último, Cunnigham *et al.* [28] introducen un mecanismo que *perturba trayectorias semánticas* y satisface $\epsilon$-DP local ($\epsilon$-LDP) [40]. Los autores también encuentran una forma de implementar información pública en el mecanismo para mejorar su utilidad sin afectar al valor $\epsilon$. Utilizan esta información pública para dividir el conjunto de todos los PDI en regiones espacio-tempo-categóricas, de manera que cada una de ellas contenga un cierto número de PDI. Esencialmente, el mecanismo puede dividirse en cuatro partes: en primer lugar, se generaliza cada localización en la región correspondiente; se dividen estas nuevas trayectorias en $n$-gramas que luego son perturbadas con el mecanismo exponencial para garantizar $\epsilon$-LDP; a continuación, la trayectoria es reconstruida minimizando una función de distancia definida sobre las tres dimensiones; y, por último, el mecanismo vuelve al dominio inicial eligiendo aleatoriamente una localización, tiempo y categoría concreta en cada región, asegurándose de que las localizaciones consecutivas de una trayectoria son alcanzables en el tiempo correspondiente.

## V. Limitaciones

En esta sección exploramos las limitaciones presentes en el ámbito de la anonimización de trayectorias.

Como punto principal, debido a la gran complejidad de los datos de trayectoria, no existe ningún mecanismo de anonimización que funcione considerablemente mejor que todos los otros [41]. Además, hay una falta de consenso a la hora de evaluar la privacidad y utilidad de los métodos propuestos. Esto desemboca en dos grandes problemas: primero, dificultad en la comparación de mecanismos; y segundo, evaluaciones sesgadas, ya que diferentes propuestas evalúan sus mecanismos respecto a sus propias métricas, dando así una falsa intuición sobre la protección o utilidad que realmente proporcionan.

En los siguientes apartados exploramos las limitaciones en privacidad y utilidad que presentan los mecanismos. Como veremos, muchas de las limitaciones aparecen a consecuencia de las propiedades de las trayectorias, como son su alta dimensión y correlación, o su naturaleza dispersa.

### *V-A. Limitaciones en las Garantías de Privacidad*

Aunque las nociones sintácticas pueden, en general, proporcionar mejores datos de utilidad que DP, son susceptibles de sufrir varios ataques bien conocidos (e.g., *ataques de contraste* o *ataques de vinculación de atributos*). Esto, junto con el hecho de que no son componibles [42], limita la aplicación de la tecnología sintáctica para proteger los datos de trayectoria en contextos dinámicos. Otro problema común en los métodos sintácticos es considerar solamente la dimensión espacial de las trayectorias, como hacen en [17], [39], [35]. Estos son susceptibles de sufrir ataques en las otras dimensiones, ya que estas contienen aún información sensible no protegida.

En el resto de la sección, revisamos las propuestas que se basan en la privacidad semántica. Aunque DP se presentó como una fuerte garantía de privacidad, la noción exacta que proporciona a veces no es clara, como se explica en [43]. Además, al igual que las nociones sintácticas [20], varios trabajos [44], [45], [46] han establecido la debilidad de esta noción cuando las correlaciones entre los atributos en la base de datos son notorias. Desgraciadamente, en los datos de trayectoria, existe un alto grado de correlación dado por su naturaleza (leyes físicas de los movimientos, restricciones de velocidad de las carreteras, comportamiento humano habitual, etc.) [8] y las relaciones sociales de los individuos. En consecuencia, se puede producir una violación de la privacidad, aunque se apliquen mecanismos de DP. Esto se debe a que la noción de DP asume una muestra aleatoria simple i.i.d. como base de datos de entrada. En el caso de trayectorias reales, esta hipótesis no se sostiene, por ello, las garantías totales de DP no pueden ser garantizadas.

Un análisis [12] de las nociones de DP adaptadas a los datos de las trayectorias sugiere que la privacidad *event-level* no es segura. La correlación entre las localizaciones cercanas en el tiempo pueden usarse para atacar fácilmente esta noción. Así mismo, cualquier localización visitada más de una vez estará desprotegida.

En [12] también afirman que la privacidad $w$-*event* falla porque las trayectorias de los usuarios son dispersas y no están periódicamente recogidas, por lo que podrían no caber en la ventana. Aunque, tanto la privacidad $w$-*event* como la $\ell$-*trajectory* son más robustas que la *event-level* en términos de ataques de correlación, los atributos de un usuario que excedan el marco de la ventana quedarán desprotegidos, por ejemplo, los puntos inicial y final de una trayectoria.

Si nos centramos en propuestas específicas, surge un problema en métodos basados en *clustering* cuando consideramos el modelo de autocorrelación de las trayectorias. Un atacante con un buen modelo de correlación puede descartar la mayoría de uniones falsas de centroides, recuperando así las conexiones originales.

### *V-B. Limitaciones en las Garantías de Utilidad*

Esta sección describe varias limitaciones que afectan a la utilidad de los datos. A continuación se exponen las limitaciones generales y, después, las más específicas relacionadas con las métricas y las metodologías.

**Problemas generales.** En primer lugar, destacamos algunas cuestiones generales que son inherentes a la naturaleza de los datos de las trayectorias. Los mayores desafíos en términos de utilidad aparecen debido al carácter único de las trayectorias, como la diversidad o alta dimensión de este tipo de datos.

Las trayectorias pueden ser muy dispares. Este suceso afecta a las nociones sintácticas como el $k$-anonimato. Los conjuntos de datos con trayectorias dispersas o cortas (alta unicidad) suponen un gran reto, ya que las trayectorias pueden tener poco solapamiento, lo que conlleva una inevitable pérdida utilidad, pues se necesita una mayor modificación de los datos para conseguir un grupo indistinguible. Del mismo modo, para las nociones semánticas (DP), la diversidad produce sensibilidades elevadas y la necesidad de añadir más ruido para lograr el mismo nivel de protección.

El factor de realismo también toma un papel importante para medir la utilidad. Los mecanismos perturbadores puede crear trayectorias imposibles, con localizaciones inalcanzables o incoherentes. Por ejemplo, los métodos basados en *clustering* [11], [26], [20] pueden dar lugar a nuevas localizaciones que podrían ser ilógicas, como coordenadas sobre edificios o ríos. Más generalmente, Gramaglia *et al.* [19] afirma que uno no puede basarse en datos aleatorios, perturbados o sintéticos para preservar la veracidad de los datos, ya que la adición de datos ficticios introduce sesgos imprevisibles en los datos saneados.

Además, los métodos como los de generalización podrían ser ineficientes para bases de datos de alta dimensión, debido a la *maldición de la dimensionalidad* [47].

**Métricas.** Tenemos unos cuantos problemas relacionados con la elección de las métricas de utilidad. Medidas de similitud como la distancia euclidiana o de Hausdorff, usadas en [17], [11], [25], [26], no tienen en consideración la coordenada temporal. Por lo tanto, dos trayectorias recorriendo la misma ruta, pero en tiempos diferentes, serán consideradas iguales según estas medidas, lo que claramente escode información y puede limitar su uso. Por ejemplo, no podrían usarse en la predicción de atascos, ya que ignoran factores como el flujo de tráfico.

Adicionalmente, métricas menores, como la *preservación de la longitud* o *de los lugares más visitados*, no deberían usarse en solitario, puesto que pueden devolver buenos resultados para ciertos mecanismos que no preserven los demás aspectos de las trayectorias, como las localizaciones, la forma o el tiempo.

Por último, se debe tener cuidado a la hora de elegir parámetros en algunas métricas, como por ejemplo en la *preservación de secuencias frecuentes* o en la fórmula de error de *queries* de conteos. Si se toma la preservación de secuencias de longitud $K$, con $K$ grande, su evaluación de la utilidad no va a ser representativa.

**Metodologías.** A continuación, informamos de las deficiencias en las metodologías del estado del arte en términos de utilidad.

Ni los métodos de *clustering* [11], [26], [20] ni los de *conteos ruidosos* [32], [31] manejan, analizan o protegen la dimensión temporal de las trayectorias. Perdiendo por ende, utilidad en numerosas aplicaciones como es la predicción de atascos y generando patrones extraños como la eliminación de paradas. Asimismo, ambas metodologías generan trayectorias irreales debido a la substitución por el centroide, en el caso de *clustering*, y a la generación de conteos positivos de secuencias que inicialmente no existían, en los *conteos ruidosos*.

Otro problema es la escasa utilidad que pueden ofrecer las aproximaciones de *conteos ruidosos* [32], [31], resultante de asumir implícitamente que las trayectorias contienen un gran número de prefijos y $n$-gramas comunes. Dado que el proceso añade ruidos a los conteos reales, si los conteos son pequeños, el ruido añadido a cada uno será más grande, con consecuencias fatales en términos de utilidad. Desgraciadamente, las bases de datos reales no siguen esta condición con mucha frecuencia (es decir, no podemos asumir que habrá muchos $n$-gramas comunes). Además, a causa del coste computacional, estos requieren bases de datos significativamente pequeñas, dificultando aún más su aplicación para bases de datos reales.

## VI. Oportunidades

En esta sección, esbozamos posibles líneas de investigación futuras que pueden superar algunas de las deficiencias identificadas en las secciones anteriores. Dadas las limitaciones técnicas de las nociones sintácticas para proteger los datos dinámicos y su debilidad frente a ataques de contraste, consideramos conveniente centrarse en la tecnología de anonimización que ofrece garantías semánticas.

En términos de garantías de privacidad, el principal problema de la aplicación de DP en las trayectorias es que la correlación de datos puede violar el nivel de privacidad prometido. Una posibilidad para hacer frente a esto es adaptar nociones alternativas (basadas en la idea original de DP). Existen ciertas tentativas al respecto, como la propuesta en [46], pero estas no son completamente concluyentes. Se requiere más trabajo para el caso concreto del modelo de correlación presente en las trayectorias.

También es interesante elaborar un nivel de granularidad adecuado para este contexto. *User-level* es un nivel de protección robusto, aunque, no obstante, proteger una participación en la base de datos no siempre es necesario, ya que no suele ser muy sensible (conocer nuestra participación en la base de datos revela tan solo que tenemos un coche y vivimos en una ciudad o país), mientras que la posibilidad de conseguir una utilidad real de mis datos es escasa. Aunque tiene sentido relajar esta noción, ninguna de las granularidades presentadas consigue proteger robustamente los atributos y en ningún caso la identidad. Por ello, una granularidad que proteja todos mis atributos y tenga en cuenta las correlaciones de mi modelo sería deseable.

Por lo general, en el tratamiento de datos, los métodos de *clustering* son una buena opción. Respecto esto, estamos interesados en un algoritmo que tenga en cuenta el tiempo, y creemos que la agrupación de trayectorias enteras o subtrayectorias de las mismas, en lugar de en cada instante de tiempo, podría ser más prolífera en términos de utilidad. Esto también reduciría problemas como las incoherencias temporales de las trayectorias resultantes. Además, métodos de *clustering* que no solo dependen de la dimensión espacial, sino también de la temporal, podrían proporcionar una mayor utilidad, y ayudar a resolver ciertos inconvenientes como la eliminación de paradas.

Para terminar, nos gustaría mencionar ciertas métricas de utilidad que, dependiendo de las necesidades, podrían ser más eficaces para evaluar futuros algoritmos y resultados. Medir la similitud entre la base de datos original y anonimizada puede ser una buena opción si se usa medidas como EDR [48] o la propuesta en [28], que consideran el tiempo. Otras métricas como la *preservación de secuencias frecuentes* pueden ser útiles como métricas secundarias. Por otro lado, es conveniente implementar módulos de post procesado que aseguren un realismo de los datos. Cunningham *et al.* [28] evita la publicación de trayectorias imposibles, detectando cuando dos localizaciones consecutivas están demasiado apartadas para llegar de una a otra en el tiempo dado y corrigiéndolas. Claramente, se podrían incorporar algoritmos análogos a cualquier mecanismo de anonimización para garantizar que todas las trayectorias sean realistas.

## VII. Conclusiones

En la primera parte de este artículo se ha analizado los avances actuales en la anonimización de los datos de trayectoria. Hemos examinado cómo se representan estos datos y qué aspectos pueden capturar; y hemos revisado las métricas y los mecanismos de anonimización más relevantes que proporcionan tanto protección sintáctica como semántica. Esta disección de la tecnología actual nos ha permitido profundizar en las limitaciones de las soluciones actuales, en cuanto a las garantías de privacidad prometidas, y la utilidad que queda tras la anonimización. Esta segunda parte de nuestro trabajo ha identificado más específicamente los impedimentos técnicos, por los cuales una parte importante de la tecnología examinada puede no proteger eficazmente la privacidad de los individuos y/o preservar la mayor parte de la utilidad de los datos de la trayectoria.

## Agradecimientos

Javier Parra Arnau es beneficiario de una beca de investigación Alexander von Humboldt. Este trabajo también ha recibido el apoyo de la Fundación "la Caixa" (código de beca LCF/BQ/PR20/11770009), del programa H2020 de la Unión Europea (acuerdo de subvención Marie Skłodowska-Curie n.º 847648), del Gobierno de España en el marco del proyecto "COMPROMISE" (PID2020-113795RB-C31/AEI/10.13039/501100011033), y del proyecto BMBF "PROPOLIS" (16KIS1393K). Los autores del KIT cuentan con el apoyo de KASTEL Security Research Labs (Tema 46.23 de la Asociación Helmholtz) y de la Estrategia de Excelencia de Alemania (EXC 2050/1 'CeTI').

## Referencias

[1] S. P. Gangadharan, "How can big data be used for social good?" *The Guardian*, 2013.
[2] B. Tarnoff, "Big data for the people: It's time to take it back from our tech overlords," *The Guardian*, 2018.
[3] S. Ovide, "Just collect less data, period." *New York Times*, 2020.
[4] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: $k$-Anonymity and its enforcement through generalization and suppression," SRI Int., Tech. Rep., 1998.
[5] C. Dwork, "Differential privacy," in *ICALP*, 2006.
[6] C. Dai *et al.*, "CenEEGs: Valid EEG selection for classification," *TKDD*, 2020.
[7] Y. Yang *et al.*, "TAD: A trajectory clustering algorithm based on spatial-temporal density analysis," *ESA*, 2020.
[8] Y.-A. De Montjoye *et al.*, "Unique in the crowd: The privacy bounds of human mobility," *Sci. Rep.*, 2013.
[9] J. Bambauer, K. Muralidhar, and R. Sarathy, "Fool's gold: An illustrated critique of differential privacy," UArizona, Tech. Rep., 2013.
[10] M. Fredrikson *et al.*, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *USENIX Secur.*, 2014.
[11] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *INFOCOM*, 2015.
[12] Y. Cao and M. Yoshikawa, "Differentially private real-time data release over infinite trajectory streams," in *MDM*, 2015.
[13] A. Hundepool *et al.*, *Statistical Disclosure Control*. Wiley, 2012.
[14] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," *TDP*, 2013.
[15] A. Machanavajjhala *et al.*, "$l$-diversity: Privacy beyond $k$-anonymity," *TKDD*, 2007.
[16] N. Li, T. Li, and S. Venkatasubramanian, "$t$-Closeness: Privacy beyond $k$-anonymity and $l$-diversity," in *ICDE*, 2007.
[17] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," *ICDE*, pp. 376–385, 2008.
[18] G. Poulis *et al.*, "Apriori-based algorithms for $k^m$-anonymizing trajectory data," *TDP*, 2014.
[19] M. Gramaglia *et al.*, "Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories," *INFOCOM*, 2017.
[20] R. Chen *et al.*, "Privacy-preserving trajectory data publishing by local suppression," *Inf. Sci.*, 2013.
[21] C. Dwork, "Differential privacy: A survey of results," in *TAMC*, 2008.
[22] C. Dwork, "Differential privacy in new settings," in *SODA*, 2010.
[23] G. Kellaris *et al.*, "Differentially private event sequences over infinite streams," *VLDB Endow.*, 2014.
[24] J. Domingo-Ferrer and R. Trujillo-Rasua, "Microaggregation- and permutation-based anonymization of movement data," *Inf. Sci.*, 2012.
[25] S. Chen *et al.*, "RNN-DP: A new differential privacy scheme base on recurrent neural network for dynamic trajectory privacy protection," *JNCA*, 2020.
[26] M. Li *et al.*, "Achieving differential privacy of trajectory data publishing in participatory sensing," *Inf. Sci.*, 2017.
[27] O. Abul, F. Bonchi, and M. Nanni, "Anonymization of moving objects databases by clustering and perturbation," *IEEE IS*, 2010.
[28] T. Cunningham *et al.*, "Real-world trajectory sharing with local differential privacy," *arXiv preprint*, 2021.
[29] M. Luca *et al.*, "A survey on deep learning for human mobility," *arXiv preprint*, 2020.
[30] M. E. Gursoy, V. Rajasekar, and L. Liu, "Utility-optimized synthesis of differentially private location traces," in *TPS-ISA*, 2020.
[31] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length $n$-grams," in *CCS*, 2012.
[32] R. Chen, B. Fung, and B. C. Desai, "Differentially private trajectory data publication," *arXiv preprint*, 2011.
[33] W. Wang *et al.*, "Travel trajectory frequent pattern mining based on differential privacy protection," *JWCMC*, 2021.
[34] T. T. Portela, F. Vicenzi, and V. Bogorny, "Trajectory data privacy: Research challenges and opportunities," in *GEOINFO*, 2019.
[35] Y. Dong and D. Pi, "Novel privacy-preserving algorithm based on frequent path for trajectory data publishing," *KBS*, 2018.
[36] M. Nergiz *et al.*, "Towards trajectory anonymization: A generalization-based approach," *TDP*, 2009.
[37] M. Gramaglia *et al.*, "GLOVE: Towards privacy-preserving publishing of record-level-truthful mobile phone trajectories," *TDS*, 2021.
[38] Z. Tu *et al.*, "Protecting trajectory from semantic attack considering $k$-anonymity, $l$-diversity, and $t$-closeness," *TNSM*, 2019.
[39] A. Monreale *et al.*, "C-safety: A framework for the anonymization of semantic trajectories," *TDP*, 2011.
[40] S. P. Kasiviswanathan *et al.*, "What can we learn privately?" *SICOMP*, 2011.
[41] M. Fiore *et al.*, "Privacy in trajectory micro-data publishing: A survey," *TDP*, 2020.
[42] J. Soria-Comas and J. Domingo-Ferrer, "Big data privacy: Challenges to privacy principles and models," *JDSE*, 2016.
[43] J. Lee and C. Clifton, "How much is enough? choosing $\varepsilon$ for differential privacy," in *ISC*, 2011.
[44] Y. Cao *et al.*, "Quantifying differential privacy under temporal correlations," in *ICDE*, 2017.
[45] H. Wang *et al.*, "Why current differential privacy schemes are inapplicable for correlated data publishing?" *WWW*, 2021.
[46] B. Yang, I. Sato, and H. Nakagawa, "Bayesian differential privacy on correlated data," in *MOD*, 2015.
[47] C. C. Aggarwal, "On $k$-anonymity and the curse of dimensionality," in *VLDB Endow.*, 2005.
[48] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *SIGMOD*, 2005.

# The role of Artificial Intelligence in Digital Twin's Cybersecurity

Mohammad Hossein Homaei
*Escuela Politécnica*
*Universidad de Extremadura*
Caceres, Spain
mhomaein@alumnos.unex.es

Andrés Caro Lindo
*Escuela Politécnica*
*Universidad de Extremadura*
Caceres, Spain
andresc@unex.es

José Carlos Sancho Núñez
*Escuela Politécnica*
*Universidad de Extremadura*
Caceres, Spain
jcsanchon@unex.es

Óscar Mogollón Gutiérrez
*Escuela Politécnica*
*Universidad de Extremadura*
Caceres, Spain
oscarmg@unex.es

Javier Alonso Díaz
*Escuela Politécnica*
*Universidad de Extremadura*
Caceres, Spain
javieralonso@unex.es

***Abstract*——The Digital Twin will be the logical step in providing products and services in the future. From a digital twin, insights can be gained into performance and processes. Its role is expanding day by day in various fields, including industry, agriculture, healthcare, smart city applications, astronomy, etc. In the context of digital twins, IoT and artificial intelligence are two of the main components that can meet the system's needs in modelling and communication infrastructure. Cyber security requirements for digital twins have grown, and numerous new challenges appear in this field every day. In this study, the challenges, attacks, threats, and artificial intelligence solutions in the cybersecurity sector of digital twins have been analysed. Also, this paper will serve as a reference point and guideline for cybersecurity researchers and digital twins professionals, particularly those with a technical interest in intelligent computing or AI.**

***Index Terms*—Digital Twins, Cybersecurity, Artificial Intelligence, Internet of Things**

## I. Introduction

The Digital Twin (DT) is a new concept of technology couple of the last few years. DTs are used by several industries and systems, including manufacturing, construction, health care, aerospace, transportation, and smart cities [1]. DT technologies are expected to increase and become widely used in the coming decades. Using DT technologies makes it possible to create a virtual duplicate of our natural system and review activities, interactions, and outcomes of various decisions made within real-world systems. By utilising DT technologies, the industry improves productivity, efficiency, and availability, leading to increased quality. The IoT is one of the most critical parts of this concept. With the growing trend of technology and the increasing applications of the IoT and artificial intelligence, new applications are emerging in this field every day. Due to the vastness of this network, the challenges and risks of implementing cybersecurity models in the real world are many.

With advanced technologies, the smart world will become a reality where all the physical objects will be equipped with embedded computing and communication capabilities. Monitoring the process via the Internet was once a challenging task, but today, the infrastructure is growing steadily, and some standards are releasing which help stabilise the communication. Industry 4.0 will require extensive research on Cyber-Physical Systems (CPS) to bridge the physical and virtual worlds. This concept states that if production systems are smart, they will be able to function more efficiently [2], [3]. Data acquisition has become relatively more straightforward than in previous decades due to the affordability and availability of sensors and actuators. However, the lack of secure platforms is also one of its challenges.

There are good reasons why the cyber security issues of digital twins have not been sufficiently explored. The most important of these reasons can be that DTs are considered critical systems because they participate in automation processes, and working with them is a bit difficult due to the issue's sensitivity. Second, digital twins contain parts of intellectual property that represent a digital copy of the physical world, so most private collections in the world, to protect their business secrets, allow cyber security experts access to digital twins of the process or production line or monitoring products or they do not provide their services. These two aspects of subject sensitivity and copyright or protection and ownership of data and process are desirable to cybercriminals who are trying to disrupt or harm the business of a group or organisation. Most of these applications include basic infrastructures and non-operating governmental or private defences, whose damage can endanger the security of a country [1], [3].

In addition, a cybercriminal may harm DT not only from the physical environment but also from the digital space to take control of its underlying infrastructure and production assets. Obviously, the attack surface is very different because the DT paradigm is to connect the two worlds through communication systems, technologies, and algorithms [1], [3].

Smart cities use DTs to determine the optimal way to maintain critical assets by eliminating guesswork. DT platforms are ideal for leveraging the IoT to boost enterprise services and platforms. Despite its features and benefits, the DT is vulnerable to cyber-attacks due to multiple attack levels and novelty, lack of standardisation and security requirements, and several reasons [3]. There are several cyber-attacks in

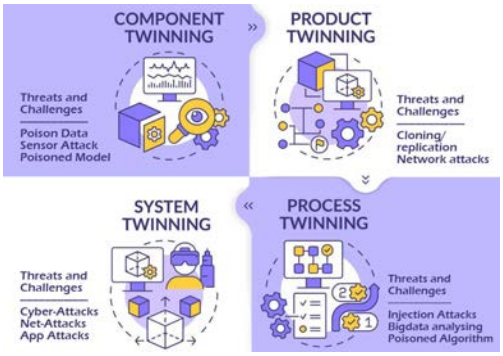



Fig. 1. Digital Twins and Cybersecurity challenges

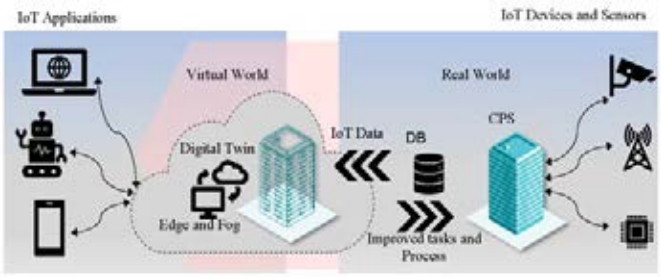



Fig. 2. The relationship between DT, CPS and IoT

the DT ecosystem, and the type of attack will depend on the cybercriminal's goals. Our study has addressed some of the challenges that cyber-security DT presents with artificial intelligence. Figure 1 shows DT's components and their cyber-security issues [2], [4].

Today, Artificial Intelligence (AI) applications have reached the field of protecting IoT and DT systems from attackers; because it is impossible to detect human or attacker abnormal behaviour in the high number of network records that may indicate an attack [5]. However, in the case of the DT, cybercriminals only have to find one vulnerability. In contrast, cybersecurity experts have to protect multiple targets. Eventually, it has led to cybercriminals developing AI to detect and thwart complex algorithms that detect abnormal activity. The following section discusses other research that has been conducted on the cybersecurity challenges and requirements for DT systems. The third part describes a category of attacks and solutions. The fourth section is dedicated to concluding and recommending other studies.

## II. RELATED WORK

This section reviews the DT and IoT environment studies and explores their security issues and recent approaches, including artificial intelligence. Like all innovations, DT design, application, and use are limited by structural limitations, infrastructure, and even social acceptance. A physical-cyber system's biggest challenge is creating a digital interface that supports the capabilities such as interoperability, trust, stability, reliability, and predictability. Nevertheless, from a technical point of view, data security and human performance, data quality improvement, latency, real-time simulation, large-scale data fusion and aggregation, intelligent data analysis and analysis, forecasting capacity, transparency and generalisation of technologies in various fields of application are the most critical challenges of DTs [6].

Different viewpoints exist regarding the relationship between Digital Twins and other concepts such as Cyber-Physical Systems (CPS), simulation and modelling and the Internet of Things (IoT) [7]. Despite their closeness, these concepts differ by their nature in their concept, core elements, and applications. Figure 2 shows the logical affinity of the elements.

The scope of digital twin cybersecurity and the search results in the Scopus and Wos databases have limited the number of review articles in this field. As shown in Table I, only seven review articles have been published in this field in the last year.

TABLE I
NUMBER OF PAPERS PUBLISHED IN CS AND DT SUBJECT

| Scopus and Web of science papers | *2012 - 2021* | *2021 - 2022* |
|---|---|---|
| Cyber Security | 30,366 | 2,194 |
| Digital Twins | 9,500 | 1,675 |
| CS & DT | 122 | 32 |
| CS & DT survey | 16 | 7 |

Recent review articles have often dealt with cybersecurity and digital twins, while artificial intelligence tools, directly and indirectly play an important role in securing these technologies. The context of this article is the use of artificial intelligence and machine learning as a double-edged sword. There has been little research in this area, as indicated by Table II.

TABLE II
SECURITY GOALS IN DIGITAL TWINS AND IOT.
✓: INCLUDED *X*: NOT INCLUDED

| Survey Papers 2021-2022 | *CS* | *DT* | *AI* |
|---|---|---|---|
| Alcaraz et al. [1] | ✓ | ✓ | *X* |
| Lv et al. [2] | ✓ | ✓ | *X* |
| Shi et al. [3] | ✓ | ✓ | *X* |
| Qian et al. [4] | ✓ | ✓ | *X* |
| Malika et al. [5] | *X* | ✓ | ✓ |
| Faleiro et al. [6] | ✓ | ✓ | *X* |
| Al-Turjman et al. [7] | ✓ | ✓ | *X* |

### A. Security Challenges in DTs

As the digital revolution spreads, many personal and commercial devices become "smart". In DT's networks, security and privacy may fail. The dynamic nature of IoT connectivity presents a set of security challenges. Here are some examples [6], [7]:

- Network scale: In the DT/IoT, billions of smart devices are interconnected on demand and logic, combined with the sheer volume, speed, and structure of real-world data.
- Heterogeneity: The DT/IoT intends to connect many heterogeneous devices to implement advanced applications to improve the quality of human life. As a result, IoT devices come in various shapes and sizes, resulting in diverse hardware and software designs. The local policy area also adds to the heterogeneity.
- Connection: The DT/IoT provides the link between devices and the information they send and receive. Therefore, DT/IoT networks must be available anywhere, anytime, and be able to communicate with other entities under predetermined standards and protocols.

- Mobility and dynamism: Network reconfiguration must be dynamic and adaptable since IoT devices are constantly added and removed.
- Vulnerability: DT systems are vulnerable to various types of attacks, such as cookie theft, cross-site scripting, structured query language injection, session hijacking, and even distributed denial of service.

### B. DT's security goals and threats

Cybersecurity literature does not provide a consensus regarding the essential security goals in DT and IoT infrastructure. Several terms and definitions are overlapping, e.g., Authentication can sometimes be used for Identification since both are necessary for each other. There are different definitions of security goals, so this paper is not elaborated on fully. Table III outlines the DT and IoT security goals based on existing literature [7], [8].

TABLE III
SECURITY GOALS IN DIGITAL TWINS AND IOT.
✓: INCLUDED *X*: NOT INCLUDED

| | Layer | | |
|---|---|---|---|
| **Security goals** | ***Sensing*** | ***Network*** | ***Application*** |
| Authorisation | ✓ | ✓ | ✓ |
| Authentication | ✓ | ✓ | ✓ |
| Availability | ✓ | ✓ | ✓ |
| Identification | ✓ | ✓ | ✓ |
| Integrity | *X* | ✓ | ✓ |
| Freshness | ✓ | ✓ | *X* |
| Confidentiality | ✓ | ✓ | ✓ |
| Privacy | *X* | ✓ | ✓ |
| Non-repudiation | ✓ | ✓ | ✓ |

Cybersecurity threats are vast and have various countermeasures to mitigate risks. The DT platform can detect cyber threats such as sensor attacks, spoof-node attacks, hardware manipulation attacks, energy manipulation attacks, sniffing, DDoS, sensitive data leakage, and fault tolerance [8]–[11] (Figure 3).

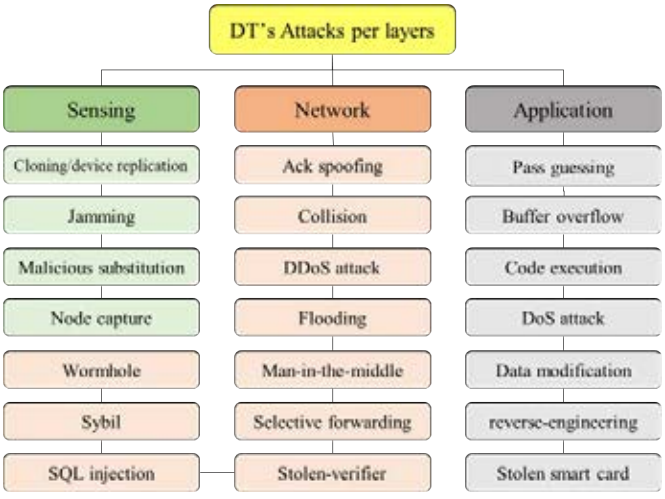



Fig. 3. Famous attacks on DT and IoT platforms

## III. AI AS A DUALISM

AI and ML tools, like any technology, have both positive and negative aspects; in other words, they are cutting both ways [12]. It is critical to understand what the tool's utility is in Cybersecurity. Experts use it to improve security, and cyber-attackers gain illegal access to systems. The following are some popular ML approaches that help to prevent or detect malicious activities.

### A. Machine Learning

There are two main kinds of ML: supervised and unsupervised learning. Supervised learning takes training data that has been designated as malicious or legitimate by humans and then inputs that data into an algorithm to create a model that compares "classes" of data it is analysing. Unsupervised learning does not require training data or manual labelling. By contrast, the algorithm groups similar data into classes and then categorises them by the degree of coherence within one class and modularity between classes [13], [14]. Depending on the nature of the problem following algorithms may be used under any of the previously mentioned approaches.

*1) Naïve Bayes:* For Cybersecurity, Naïve Bayes (NB) is a popular algorithm that tries to classify data based on the underlying Bayesian theorem, which holds that anomalous activities generally result from multiple events rather than one attack. In supervised learning, NB analyses each activity to determine its risk of being abnormal once trained and generated. Also, ML algorithms can create the other models discussed in the following [13], [14].

*2) Support Vector Machines:* The support vector machines (SVM)'s technique is primarily used in Cybersecurity to analyse Internet traffic patterns and classify them into HTTP, FTP, and SMTP components. Since SVMs are supervised ML techniques, they are often used in applications where attacks can be simulated, such as generating training data from network traffic generated during penetration testing.

*3) K-Nearest Neighbour:* K-Nearest Neighbour (k-NN) is used for intrusion detection since it can rapidly adapt to new traffic patterns to detect previously unknown, even zero-day attacks. Cybersecurity experts are also using k-NN to detect attacks in real-time. This method has been used to detect false data injection attacks [15].

*4) Decision Trees:* The decision tree builds rules based on data samples used for training [15]. A description of the traffic is generated iteratively (often simply "attack" or "normal") using iterative division. This approach detects DoS attacks by analysing traffic's size, flow, and duration in Cybersecurity. The traffic may be low in quantity, but if it continues for a long time, it is likely to be an attack and classified as such.

### B. Artificial Neural Networks

Artificial neural networks (ANNs) are a technique that uses the way neurons interact in the brain to interpret and pass information [16]. ANNs are powerful mathematical models because they can adjust their models to new information [17]. In Cybersecurity, AI is unique and is rapidly growing, and besides being costly and resource-intensive, AI may not be feasible to protect small systems. Businesses with large networks may find these solutions helpful, especially if they are thinking about introducing IoT devices or have already done so. Cybersecurity with AI would also benefit the massive systems one would find in a smart city. AI would be capable of giving rapid response times, which are crucial in systems like

traffic management [16]. Moreover, many AI cybersecurity measures detect or stop attacks in progress rather than prevent attacks in the first place, which means that other preventative security measures should be instituted.

### C. Federated Learning Model

Federated Learning (FL) and Federated Cybersecurity approachs enable communication and collaboration at different levels to provide security solutions to DT and IoT applications. Privacy and security may be compromised due to the traditional exchange of data/information between classes and/or within groups. Therefore, FL offers exchanging data/information security and privacy alternatives to the conventional model. With Federated Cybersecurity, FL's ability to collaborate and exchange information at any level can make DT and IoT networks safer and more secure. An example of FL is to predict pressure differences as air passes through gradually clogged filters, thereby assisting the IoT sensor in predicting when the filter needs to be changed. Users could easily poison the algorithm by manipulating their filters. Nevertheless, only a few projects have attempted to create a federated security model utilising multiple global models [14], [18].

## IV. ATTACKS AND COUNTER BY AI AND ML

Cybercriminals have begun to employ malicious AI to aid attacks, often so that intrusion detection algorithms of the IoT can be thwarted or beneficial AI can work against them. This leads to new challenges. The following methods are used by cyber attackers to breach intelligence-based systems.

### A. Automated Detection of Vulnerabilities

Vulnerabilities in a system can be discovered using ML. Security professionals can use this technology to identify vulnerabilities that need to be patched intelligently, but attackers can also utilise this technology to find and exploit vulnerabilities. Increasing use of technology, especially technologies with low-security standards such as IoT devices, has increased vulnerabilities that attackers can control, including zero-day vulnerabilities. Attackers often use AI to find and exploit vulnerabilities much more quickly than developers can fix them. These detection tools are also available to developers, but note that developers have a disadvantage when dealing with security; they need to identify and fix each potential vulnerability, while an attacker only needs to find one, making automatic detection an essential tool for attackers [19].

### B. Fuzzing Technique

The fuzzing technique or symbolic execution analyses input variables by setting the absolute value to a symbol instead of an exact number. Fuzzing has proven to be a valuable method for finding automated vulnerabilities since specially crafted inputs are fed to programs to trigger vulnerabilities and crash the system [19].

### C. Infiltration Attacks

An infiltration attack occurs when a cyber-attacker alters the input of an AI system in a way that causes it to malfunction or produce incorrect results. A typical infiltration attack involves adding an attack pattern to the input, such as taping a stop sign to confuse self-driving cars or adding small amounts of noise to an image invisible to the human eye to confuse AI [20]. Note that an infiltration attack does not require compromise of the AI's algorithm or security to be carried out. Only the input that the attacker wants to compromise the output must be altered. Most sophisticated attacks, however, are invisible to the human eye. By modifying a small part of the image, the attacker can mislead the algorithm. In contrast, imperceptible attacks are invisible to the human eye. A small amount of digital dust, which is not visible to a human eye but significant enough to affect an AI's output, can be included in this category.

Attacks are usually digital or physical, with few cyber-attacks that combine both. In many physical attacks, the attack pattern should be more prominent rather than imperceptible since physical objects must be digitised for processing and, in the process, lose some finer detail. However, some cyber-attacks can still be strenuous to detect even with data loss. In contrast to physical attacks, digital attacks target digital inputs such as images, videos, audio recordings, and files. Due to the digital nature of the inputs, no detail is lost during the digitising process. In this way, attackers can make exact attacks more invisible to the human eye than physical attacks. Therefore, digital attacks are not necessarily hidden. In some cases, strange patterns on Photoshop glasses applied to celebrities may cause the AI to recognise the image as another person, even though the person is still the same.

Moreover, placing a piece of tape on a stop sign in a specific way could cause an algorithm not to recognise it or even classify it as a green light. On a larger scale, this could affect traffic pattern detectors in smart cities if the car does not obey the stop sign. Therefore, a noise-based input attack could also cause smart assistants to malfunction and execute unintended commands [16], [21].

### D. False data Injection and Poisoning

There are many similarities between data poisoning attacks and intake attacks [19] . On the other hand, data poisoning aims to alter inputs over a long time. The alteration of inputs flaws the AI that analyses the data; Since AI is still being trained before being deployed, data poisoning generally occurs during training. AI can learn to fail when given specific inputs selected by the attacker; e.g. The enemy military may poison AI so that it does not recognise certain aircraft types, such as drones, if the military uses AI to detect aircraft. For instance, in predictive maintenance systems, data poisoning is used on AIs who constantly learn and analyse data to come up with and adjust predictions. An AI can be poisoned in three main ways [20].

- Poisoned Dataset: Data poisoning is the method that most directly impacts an AI's understanding. Any flaws within these datasets will subsequently affect the AI's performance when poisoned datasets contain inaccurate or mislabelled information [22].
- Poisoned Algorithm: The algorithm poisoning attack exploits an AI's learning algorithms. The method of attack is prevalent in federated learning, a technique of training ML while keeping the individual's privacy in mind.

- Poisoned Model: The last type of poisoning replaces a legitimate model with an already poisoned one prepared ahead of time; all an attacker needs to do is access the file that stores the model and returns it. It is also possible to alter the equations and data within the trained model file [22]. Despite double-checking the trained model, this method can still be dangerous.

## V. CONCLUSIONS

This study, inspired by the growing importance of Cybersecurity in the digital twin's applications, is presented in this paper. This research aimed to demonstrate how AI can play a significant role in intelligent decision-making and designing intelligent and automated cybersecurity systems. This paper has reviewed security intelligence models that thoughtfully combine AI-based methods such as ML and DL to address cybersecurity issues. Using such AI-based models can help solve problems alerting, from malware analysis to detecting phishing attacks and malicious code, briefly discussed in this paper. As discussed in this paper, the concept of AI-based security intelligence modelling in DT and IoT platforms can aid the cybersecurity computing process with an actionable and intelligent approach. Finally, It seems that this paper will serve as a reference point and guideline for cybersecurity researchers and DTs professionals, particularly those with a technical interest in intelligent computing or AI. A secure DT-based platform for precision agriculture will be presented in a future work.

## ACKNOWLEDGEMENTS

This work was supported by the Junta de Extremadura (European Regional Development Fund), Consejeria de Economiıa, Ciencia y Agenda Digital, under Project GR21099.

## REFERENCES

[1] C. Alcaraz and J. Lopez, "Digital Twin: A Comprehensive Survey of Security Threats," *IEEE Communications Surveys & Tutorials*, 2022, https://doi.org/10.1109/comst.2022.3171465DOI: 10.1109/COMST.2022.3171465.

[2] Z. Lv, D. Chen, H. Feng, A. K. Singh, W. Wei, and H. Lv, "Computational Intelligence in Security of Digital Twins Big Graphic Data in Cyber-Physical Systems of Smart Cities," *ACM Transactions on Management Information Systems*, Apr. 2022, https://doi.org/10.1145/3522760DOI: 10.1145/3522760.

[3] L. Shi, S. Krishnan and S. Wen, "Study Cybersecurity of Cyber Physical System in the Virtual Environment: A Survey and New Direction," *Australasian Computer Science Week 2022*, pp. 46-55. 2022 https://dl.acm.org/doi/10.1145/3511616.3513098DOI: 10.1109/COMST.2022.3171465.

[4] C. Qian, X. Liu, C. Ripley, M. Qian, F. Liang, and W. Yu, "Digital Twin—Cyber Replica of Physical Things: Architecture, Applications and Future Research Directions," *Future. Internet*, vol. 14, no. 2, p. 64, Feb. 2022, https://doi.org/10.3390/fi14020064DOI: 10.3390/fi14020064.

[5] H. Malik, G. Chaudharyb, and S. Srivastavac, "Digital Transformation through Advances in Artificial Intelligence and Machine Learning," *Journal of Intelligent & Fuzzy Systems*, 42(2), 2022, pp. 615–622, https://doi.org/10.3233/jifs-189787DOI: 10.3233/JIFS-189787.

[6] R. Faleiro, L. Pan, S. R. Pokhrel, and R Doss, "Digital Twin for Cybersecurity: Towards Enhancing Cyber Resilience," *International Conference on Broadband Communications, Networks and Systems*, pp. 57-76. Springer, Cham, 2021, https://doi.org/10.1007/978-3-030-93479-8_4DOI: 10.1007/978-3-030-93479-8_4.

[7] F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, "An overview of security and privacy in smart cities' IoT communications," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3677, Mar. 2022,https://doi.org/10.1002/ett.3677DOI: 10.1002/ett.3677.

[8] K. Alshammari, T. Beach, and Y. Rezgui, "Cybersecurity for digital twins in the built environment: current research and future directions," *Journal of Information Technology in Construction*, vol. 26, pp. 159–173, Apr. 2021, https://doi.org/10.36680/j.itcon.2021.010DOI: 10.36680/j.itcon.2021.01.

[9] C. Herwig, R. Pörtner, and J. Möller, Digital Twins, vol. 177. Cham: Springer International Publishing, 2021, https://doi.org/10.1007/978-3-030-71656-1DOI: 10.1007/978-3-030-71656-1.

[10] D. Holmes, M. Papathanasaki, L. Maglaras, M. A. Ferrag, S. Nepal, and H. Janicke, "Digital Twins and Cyber Security – solution or challenge?," *in 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, Sep. 2021, pp. 1–8, https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566277DOI: 10.1109/SEEDA-CECNSM53056.2021.9566277.

[11] J. C. Olivares-Rojas, E. Reyes-Archundia, J. A. Gutierrez-Gnecchi, I. Molina-Moreno, J. Cerda-Jacobo, and A. Mendez-Patino, "Towards Cybersecurity of the Smart Grid using Digital Twins," *IEEE Internet Computing*, pp. 1–1, 2021, https://doi.org/10.1109/MIC.2021.3063674DOI: 10.1109/MIC.2021.3063674.

[12] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, Jan. 2022, https://doi.org/10.1109/MCOM.001.21143DOI: 10.1109/MCOM.001.21143.

[13] R. Majeed, N. A. Abdullah, M. Faheem Mushtaq, M. Umer, and M. Nappi, "Intelligent Cyber-Security System for IoT-Aided Drones Using Voting Classifier," *Electronics (Basel)*, vol. 10, no. 23, p. 2926, Nov. 2021, https://doi.org/10.3390/electronics10232926DOI: 10.3390/electronics10232926.

[14] M. Alazab, S. P. RM, P. M, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated Learning for Cybersecurity: Concepts, Challenges, and Future Directions," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3501–3509, May 2022, https://doi.org/10.1109/TII.2021.3119038DOI: 10.1109/TII.2021.3119038.

[15] Z. Zhang, F. Wen, Z. Sun, X. Guo, T. He, and C. Lee, "Artificial Intelligence-Enabled Sensing Technologies in the 5G/Internet of Things Era: From Virtual Reality/Augmented Reality to the Digital Twin," *Adv. Intell. Syst.*, p. 2100228, Mar. 2022,https://doi.org/10.1002/aisy.202100228DOI: 10.1002/aisy.202100228.

[16] Z. Zhang et al., "Artificial intelligence in cyber security: research advances, challenges, and opportunities," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1029–1053, Feb. 2022,https://doi.org/10.1007/s10462-021-09976-0DOI: 10.1007/s10462-021-09976-0.

[17] C. M. Paredes, D. Martínez-Castro, V. Ibarra-Junquera, and A. González-Potes, "Detection and Isolation of DoS and Integrity Cyber Attacks in Cyber-Physical Systems with a Neural Network-Based Architecture," *Electronics (Basel)*, vol. 10, no. 18, p. 2238, Sep. 2021, https://doi.org/10.3390/electronics10182238 DOI: 10.3390/electronics10182238.

[18] B. Ghimire and D. B. Rawat, "Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things," *IEEE Internet Things J.*, pp. 1–1, 2022, https://doi.org/10.1109/jiot.2022.3150363DOI: 10.1109/JIOT.2022.3150363.

[19] L. Yu, H. Wang, L. Li, and H. He, "Towards Automated Detection of Higher-Order Command Injection Vulnerabilities in IoT Devices," *International Journal of Digital Crime and Forensics*, vol. 13, no. 6, pp. 1–14, Nov. 2021, https://doi.org/10.4018/IJDCF.286755DOI: 10.4018/IJDCF.286755.

[20] W. Lalouani, M. Younis, M. Ebrahimabadi, and N. Karimi, "Countering Modeling Attacks in PUF-based IoT Security Solutions," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 18, no. 3, pp. 1–28, Jul. 2022, https://doi.org/10.1145/3491221DOI: 10.1145/3491221.

[21] Z. Lv, Y. Li, H. Feng, and H. Lv, "Deep Learning for Security in Digital Twins of Cooperative Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021 https://doi.org/10.1109/TITS.2021.3113779 DOI: 10.1109/TITS.2021.3113779.

[22] Z. Zhang, R. Deng, P. Cheng, and Q. Wei, "On Feasibility of Coordinated Time-Delay and False Data Injection Attacks on Cyber-Physical Systems," *IEEE Internet of Things Journal*, pp. 1–1, 2021, https://doi.org/10.1109/JIOT.2021.3118065DOI: 10.1109/JIOT.2021.3118065.

# Esquema promocional sobre blockchain

Amador Jaume Barceló
Dpto.de Matemáticas e Informática.
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
amador.jaume@uib.es

M. Francisca Hinarejos
Dpto.de Matemáticas e Informática.
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
xisca.hinarejos@uib.es

Josep-Lluís Ferrer-Gomila
Dpto.de Matemáticas e Informática.
Universitat de les Illes Balears
Edif. Anselm Turmeda (UIB)
Cra. Valldemossa km 7.5
jlferrer@uib.es

***Resumen*—Los esquemas promocionales (basados en puntos, cupones electrónicos, etc.) son estrategias de marketing que tienen como finalidad atraer a nuevos clientes mediante descuentos u otras ofertas atractivas. Su implementación en el mundo digital está basado en modelos centralizados en el comerciante o, en el caso de varios comerciantes, en una entidad de confianza. En estos modelos, la entidad central es el punto sensible del sistema, que en el caso de los esquemas promocionales, puede suponer problemas de seguridad, como la falsificación o el doble canjeo de los puntos o de los cupones. La blockchain, una tecnología emergente que proporciona un entorno descentralizado, transparente e inmutable, parece ser una buena alternativa a los modelos centralizados tradicionales. En este artículo, se presenta un esquema promocional multi-comerciante que utiliza la blockchain para garantizar el intercambio equitativo entre las diferentes partes.**

***Index Terms*—esquema promocional, cupones electrónicos, puntos promocionales, blockchain, intercambio equitativo**

## I. Introducción

Los esquemas promocionales buscan atraer a nuevos clientes, mediante descuentos u otras ofertas atractivas, para que adquieran determinados productos o servicios y, de esta forma, incrementar las ventas. Entre los diversos esquemas promocionales, cabe reseñar los que utilizan puntos o cupones electrónicos. En estos esquemas, los clientes pueden obtener puntos (o cupones) adquiriendo determinados productos o realizando acciones que beneficien al comerciante, por ejemplo, llevando nuevos clientes. Los puntos (o cupones) obtenidos pueden canjearse para acceder a descuentos o recompensas en el mismo comerciante o en otro comerciante que los admita. En el segundo caso, el comerciante que acepta puntos (o cupones) canjeados puede tener derecho a recibir una compensación económica por las recompensas o descuentos que ha concedido.

Hacer uso de esquemas promocionales basados en puntos o cupones es muy eficaz para atraer clientes y fomentar el incremento del volumen de ventas. *Juniper Research* indica que recibir puntos o cupones puede desencadenar compras que inicialmente no se tenía intención de realizar [1]. Sin embargo, el tipo de beneficios percibidos por los clientes puede reducir la utilización de los puntos o cupones recibidos de los comerciantes. Según *ResearchAndMarkets* [2], el 54 % de los clientes en EEUU deja de realizar compras en determinados comercios si la recompensa obtenida no es relevante para ellos, o el número de puntos necesarios para obtener una recompensa atractiva es demasiado elevado.

De esta manera, estos esquemas promocionales pueden considerarse un mecanismo de pago alternativo, por lo que presentan los mismos problemas de seguridad que el dinero electrónico, como la falsificación, el doble canjeo o la negativa de haber realizado o recibido una transferencia. Un ejemplo de esto último sería cuando un cliente reclama falsamente haber canjeado puntos (o cupones) para obtener una recompensa, o cuando un comerciante asegura falsamente no haber recibido los puntos de un cliente para no entregarle la recompensa. Aparte de los problemas similares al dinero electrónico, los puntos (o cupones) pueden presentar restricciones adicionales, como por ejemplo la prohibición de que se transfieran entre clientes. Adicionalmente, a medida que estos programas se expanden e incorporan más comerciantes con intereses diversos, es necesario definir procesos seguros de afiliación con el objetivo de garantizar que todo participante sólo pueda realizar aquellas acciones para las que esté autorizado. Asimismo, son imprescindibles procesos seguros de desafiliación de los comerciantes que abandonan el esquema promocional.

El diseño e implementación de estos esquemas promocionales se ha basado tradicionalmente en modelos centralizados, en los que una entidad central garantiza los requisitos de seguridad y asegura el cumplimiento de las reglas por parte de todos los participantes. Sin embargo, para poder ejercer correctamente dicha función, esta entidad central debe ser robusta, imparcial, inalterable e incorruptible. Estas propiedades son muy difíciles de garantizar para una única entidad.

Una de las herramientas más prometedoras para solucionar los problemas de los modelos centralizados es la blockchain. Esta tecnología emergente ofrece un entorno descentralizado, transparente e inmutable. La blockchain permite mantener un registro de todas las transacciones que se realizan sobre esta red, pudiendo servir como prueba en caso de conflicto. Algunas blockchain permiten el despliegue de *smart contracts*, que son programas inalterables que pueden, entre otras operaciones, consultar y actualizar datos de este registro, manteniendo siempre el estado anterior de estos datos. Por lo tanto, los *smart contracts* parecen un recurso interesante para implementar esquemas promocionales, de manera que puedan encargarse de gestionar los puntos (o cupones): validar y ejecutar la obtención y canje de puntos, mantener un registro tanto de los puntos (o cupones) asociados a la cuenta de su propietario como de todas las transferencias de puntos (o cupones) ejecutadas.

En la bibliografía encontramos diferentes propuestas de esquemas promocionales basados en la tecnología blockchain [3], [4]. Algunas propuestas requieren de una entidad centralizada que realiza las transacciones en nombre del resto de participantes [3], [5]. Pramanik et al. [6] proponen una

solución en la que sólo se tiene en cuenta la transferencia de puntos entre clientes, sin definir cómo se obtienen, canjean, etc. Otras propuestas definen el esquema a alto nivel, sin una especificación técnica ni una evaluación de la propuesta [7], [8], [9]. En definitiva, no se ha encontrado en la bibliografía ninguna propuesta que cumpla los requisitos definidos en la sección II.

En este artículo, se presenta un protocolo para un esquema promocional en un escenario multi-comerciante que utiliza la blockchain para garantizar el intercambio equitativo de puntos. Además, el protocolo puede permitir la transferencia de puntos entre clientes para aumentar la utilidad percibida por estos. El protocolo asegura la validez de todos los puntos transferidos y la generación de puntos únicamente por parte de entidades autorizadas. Además, se incluyen medidas para desincentivar ciertas prácticas que perjudican a los comerciantes que participan en el esquema promocional, como la emisión masiva de puntos. Si bien el protocolo que se presenta en este artículo está orientado a puntos, también podría aplicarse a *fungible coupons*, cupones indistinguibles entre sí y todos con el mismo valor.

El resto del artículo está estructurado de la siguiente manera. En la sección II, se da una visión general del escenario que se contempla. En la sección III, se definen las especificaciones del protocolo tanto para el escenario en el que se permite la transferencia de puntos entre clientes como el escenario en el que su transferencia no está permitida. En la sección IV, se analizan los requisitos de seguridad y la viabilidad en términos de costes económicos de la propuesta presentada. Finalmente, se incluyen las conclusiones en la sección V.

## II. Escenario: entidades y requisitos del sistema

El escenario que se plantea en este artículo es un esquema promocional por puntos multi-comerciante en el que los clientes pueden obtener y canjear puntos en diferentes comerciantes, independientemente del comerciante en el que se obtuvieron los puntos. Además, se podrá permitir que los clientes puedan compartir sus puntos para aumentar su utilización del sistema.

Antes de entregar puntos a los clientes, los comerciantes deben generar puntos dentro del sistema, depositando una cantidad económica equivalente en un fondo común. Los comerciantes únicamente podrán recuperar este depósito cuando reciban puntos canjeados por parte de clientes. De esta forma, se incentiva la entrega y aceptación equilibrada de puntos entre todos los comerciantes, evitando que haya comerciantes que entreguen cantidades masivas de puntos a clientes. El valor económico equivalente de un punto será establecido por el consorcio.

Una vez ejecutado el proceso de generación de puntos, los comerciantes pueden entregar puntos a aquellos clientes que compren determinados productos o realicen acciones que beneficien al comerciante. Los clientes que reciben puntos pueden canjearlos en cualquiera de los establecimientos pertenecientes a algún comerciante del consorcio, así como, si está permitido, transferir puntos a otro cliente. En todos los procesos de transferencia de puntos (comerciante-cliente, cliente-comerciante o cliente-cliente), debe garantizarse que una entidad solamente pueda transferir los puntos que posea en ese momento.

Los comerciantes necesitan estar autorizados para participar en el esquema promocional. El proceso de afiliación es competencia del consorcio de comerciantes. En cuanto al proceso de desafiliación, este puede ser por iniciativa del propio comerciante o por decisión del consorcio. El hecho de desafiliarse del consorcio no le da derecho al comerciante a recuperar dinero que pudiera tener en depósito (este dinero solo debe poder rescatarlo por la compensación de puntos canjeados por los clientes en su comercio). En ningún caso, la desafiliación de un comerciante debe provocar pérdidas de puntos a los clientes que hubieran recibido puntos de dicho comerciante.

Otra propiedad que se debe cumplir es el anonimato de los clientes a lo largo de todos los procesos. Esto implica que ninguna entidad pueda identificar a los clientes que han recibido o enviado puntos dentro del esquema, aunque los comerciantes o los otros clientes con los que intercambien puntos pudieran conocer su identidad. En cuanto al anonimato de los comerciantes, no es prioritario y no forma parte de los requisitos de este escenario, ya que en este tipo de esquemas, los propios comerciantes publicitan su participación para atraer al mayor número de clientes posible.

## III. Especificación del programa promocional

En esta sección se detallan las especificaciones del protocolo que contempla dos escenarios posibles: impedir la transferencia de puntos entre clientes, o permitir su transferencia. La figura 1 presenta una visión general de todas las entidades y procesos presentes en este escenario. Todos los procesos requieren de la ejecución de funciones del *smart contract*, que se realizan mediante transacciones *on-ledger*. Adicionalmente, los procesos de afiliación, desafiliación, entrega, transferencia y canjeo requieren una negociación previa mediante el intercambio de mensajes que se realizará *off-ledger* entre las dos entidades correspondientes. La tabla I define los parámetros utilizados en la especificación de los distintos procesos.

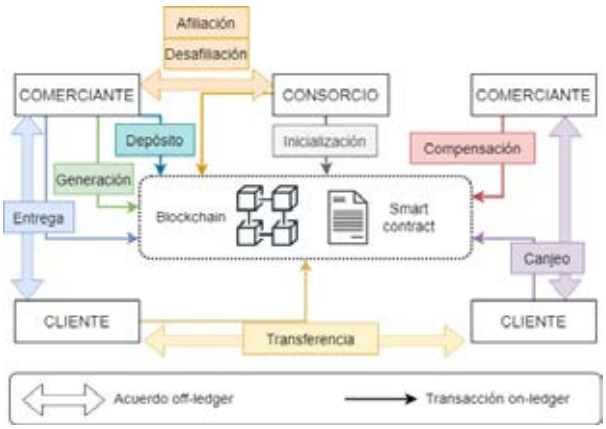



Figura 1. Visión general del escenario con puntos transferibles entre clientes

Esta solución requiere de las siguientes entidades:

- *Consorcio*. Es la entidad que representa a la asociación de todos los comerciantes que participan en el esquema promocional. Se encarga del despliegue del *smart contract*, la afiliación de nuevos comerciantes y la desafiliación forzosa de un comerciante en caso de conflicto. Se considera una entidad de confianza que

Tabla I
DEFINICIÓN DE PARÁMETROS.

| Parámetro | Definición |
|---|---|
| $M$ | Consorcio |
| $m_i$ | Comerciante del consorcio |
| $c_i$ | Cliente del sistema |
| $SC$ | Smart contract |
| $@x$ | Dirección de blockchain de la entidad x |

va a actuar siempre en beneficio de la asociación de comerciantes y del esquema promocional. Para ejercer sus funciones, necesita una dirección de blockchain $@M$.

- *Comerciantes*. Pueden solicitar la generación de puntos depositando la cantidad correspondiente en el fondo común, aceptar puntos canjeados por parte de clientes y recibir una compensación económica por los puntos aceptados. Cada comerciante necesita una dirección de blockchain $@m_i$ (a la que tendrá asociada una *wallet* con dinero), y estar autorizado para emitir puntos por el consorcio de comerciantes.
- *Clientes*. Pueden recibir puntos de comerciantes, canjear puntos por descuentos o recompensas en cualquiera de los comerciantes del consorcio y, si el sistema lo autoriza, transferir puntos a otro cliente. Cada cliente necesita una dirección de blockchain $@c_j$ la cual se recomienda que sea utilizada únicamente para el esquema promocional.
- *Smart Contract* (*SC*). Es el programa que permite la gestión segura de los puntos por parte de las diferentes entidades autorizadas.

El *smart contract* debe mantener la siguiente estructura de datos:

- *monetaryValue* (*VM*). Es el valor económico equivalente a un punto.
- *consortium*. Indica la dirección de blockchain del consorcio, obtenida en el momento del despliegue del *smart contract*.
- *merchants*. Es la lista de comerciantes que pueden emitir puntos. Los comerciantes desafiliados dejan de formar parte de esta lista desde el momento de su desafiliación.
- *pointsBalances*. Indica los puntos que tiene cada una de las entidades, sean clientes o comerciantes.
- *moneyBalances*. Indica la cantidad monetaria depositada por uno de los comerciantes $m_i$ en el fondo común, pero que aún no ha recuperado a través del proceso de compensación. Se corresponde con la diferencia entre el equivalente económico de los puntos generados y el de los puntos compensados por cada comerciante $m_i$.

En las siguientes subsecciones, se detallan los procesos que forman parte de la solución propuesta.

### *III-A. Inicialización*

El consorcio despliega el *smart contract* en la blockchain, estableciendo el parámetro $monetaryValue$. En este mismo proceso, el *smart contract* debe registrar la dirección $@M$ como dirección del consorcio para controlar la ejecución de determinadas funciones que únicamente puedan ser realizadas por el consorcio.

### *III-B. Afiliación*

El comerciante $m_i$ desea afiliarse para emitir puntos, por lo que realiza una petición al consorcio $M$, dándole a conocer su dirección $@m_i$.

$$m_i \rightarrow M : [@m_i]$$

El consorcio, en caso de aceptar su petición, ejecuta la función *affiliateMerchant* del *smart contract*.

$$M \rightarrow SC : affiliateMerchant[@m_i]$$

El *smart contract* comprueba que $@M$ es la dirección del consorcio, registrada en el proceso de inicialización. Tras la correcta validación, añade $@m_i$ a $merchants$, la lista de comerciantes activos en el sistema.

### *III-C. Desafiliación*

El proceso de desafiliación del comerciante $m_i$ puede realizarse por el propio comerciante o por el consorcio. Debe ejecutarse la función *disaffiliateMerchant*, pasándole como parámetro $@m_i$.

$$m_i/M \rightarrow SC : disaffiliateMerchant[@m_i]$$

El *smart contract* comprueba cuál de los dos casos es, simplemente validando que la dirección $@m_i$ sea la misma que la del comerciante que ejecuta la función o, en caso contrario, que la dirección del que ejecuta la función sea $@M$. A continuación elimina $@m_i$ de la lista $merchants$. A partir de este momento $m_i$ ya no puede ejecutar ninguna de las funciones que están a disposición de los comerciantes afiliados.

### *III-D. Depósito*

El comerciante $m_i$ que desea generar $n$ puntos primero debe depositar la cantidad económica equivalente a estos $n$ puntos ($VM \cdot n$ unidades monetarias). Esta cantidad económica deberá ser abonada desde su wallet y será depositada en el *smart contract* $@SC$.

$$m_i \rightarrow SC : deposit[@m_i, @SC, n]$$

Una vez depositada la cantidad económica correspondiente, el *smart contract* actualiza el registro de dinero asociado al comerciante $m_i$ en $moneyBalances$.

### *III-E. Generación de puntos*

El comerciante $m_i$ solicita la generación de $n$ puntos ejecutando la función *generatePoints* del *smart contract*.

$$m_i \rightarrow SC : generatePoints[@m_i, n]$$

El *smart contract* comprueba que $@m_i$ es una dirección de comerciante y que haya depositado previamente la cantidad económica equivalente de los puntos que desea generar (*VM* $\cdot$ $n$ unidades monetarias). Si se cumplen ambas condiciones, actualiza los puntos de $m_i$ en $pointsBalances$.

### III-F. Entrega, Canjeo y Transferencia de puntos

Las operaciones que hay que ejecutar en el *smart contract* para los procesos de entrega, canjeo y transferencia de puntos cambian ligeramente dependiendo de si los puntos son transferibles entre clientes o no lo son.

Cuando los puntos son transferibles, puede utilizarse una misma función para todas las operaciones de entrega, canjeo y transferencia de puntos que deben ejecutarse en el *smart contract*. Esta función se ha denominado *transferPoints*. A continuación, se presentan los parámetros que recibe esta función en cada caso:

- Entrega de puntos del comerciante $m_i$ al cliente $c_j$:

$$m_i \rightarrow SC : transferPoints[@m_i, @c_j, n]$$

- Canjeo de puntos por parte del cliente $c_j$ en el comerciante $m_i$:

$$c_j \rightarrow SC : transferPoints[@c_j, @m_i, n]$$

- Transferencia de puntos entre dos clientes $c_j$ y $c_k$:

$$c_j \rightarrow SC : transferPoints[@c_j, @c_k, n]$$

En cualquiera de las situaciones, el *smart contract* comprueba que la entidad que desea entregar, canjear o transferir puntos dispone de suficientes puntos en $pointsBalances$ y, únicamente en caso afirmativo, actualiza los puntos de ambas entidades en $pointsBalances$.

Cuando los puntos no son transferibles entre clientes, los procesos de entrega y canjeo requieren cada uno de una función distinta en el *smart contract*. Al estar prohibida la transferencia entre clientes, no es necesario implementar una función al respecto, sino que debe garantizarse que las funciones correspondientes a los procesos de entrega y canjeo no puedan ser utilizadas para tal fin:

- Entrega de puntos del comerciante $m_i$ al cliente $c_j$. Antes de actualizar los puntos de $m_i$ y $c_j$ en $pointsBalances$, el *smart contract* debe verificar que $@m_i$ se halla en la lista $merchants$.

$$m_i \rightarrow SC : deliverPoints[@c_j, n]$$

- Canjeo de puntos por parte del cliente $c_j$ en el comerciante $m_i$. Antes de actualizar los puntos de $c_j$ y $m_i$ en $pointsBalances$, el *smart contract* debe verificar que $@m_i$ se halla en la lista $merchants$ con el fin de evitar que dos clientes se transfieran puntos entre sí utilizando esta función.

$$c_j \rightarrow SC : redeemPoints[@m_i, n]$$

### III-G. Compensación

El comerciante $m_i$ recupera el dinero equivalente a $n$ puntos ejecutando la función *withdraw* del *smart contract*.

$$m_i \rightarrow SC : withdraw[@m_i, n]$$

El *smart contract* comprueba que $@m_i$ está en la lista $merchants$, tiene suficientes puntos en $pointsBalances$ y tiene depositado suficiente dinero en el fondo común ($VM \cdot n \leq moneyBalances$ de $m_i$). Si todo es correcto, actualiza los puntos de $m_i$ en $pointsBalances$ y su registro de dinero depositado en $moneyBalances$. Finalmente, deposita el valor económico equivalente a los $n$ puntos compensados en la *wallet* de $m_i$.

## IV. Análisis y Evaluación

### IV-A. Análisis de la seguridad

En esta sección, se analiza la seguridad de la propuesta presentada en la sección III. En primer lugar, se garantiza el no repudio, debido a que la blockchain registra, en la cadena de bloques, la ejecución de las funciones del *smart contract* (utilizando las transacciones). Cada transacción va firmada por la entidad que la origina (un comerciante, un cliente o el consorcio), la cual es verificada por cada nodo de la red. A su vez, los nodos generan bloques firmados que incluyen las transacciones validadas hasta ese momento. Por ejemplo, en Ethereum, la dirección asociada a cada participante se genera a partir de su clave pública, por lo que es posible validar la firma de cada uno de ellos a partir de la dirección de envío de cada transacción.

Cada entidad tiene asociada a su dirección de blockchain su balance de puntos. El *smart contract*, en el momento de ejecutar cualquier operación que requiera el uso de puntos (entrega, canjeo o transferencia), extrae la dirección de la entidad que ejecuta el proceso, la cual permite verificar si fue firmada por la entidad que creó la transacción. Por lo tanto, un atacante no puede utilizar los puntos de otra entidad, a menos que posea la clave privada asociada a la dirección de dicha entidad. Además, como el *smart contract* gestiona el balance de puntos (incrementando y decrementando los balances correspondientes), una entidad no puede dar los mismos puntos a dos o más entidades. Por lo tanto, se garantiza la protección contra el doble uso de puntos.

El sistema proporciona protección contra la falsificación de puntos. Como la ejecución de todas las funciones de entrega, canjeo y transferencia están registradas en la blockchain y la gestión del balance de puntos está controlada por el *smart contract*, no se pueden generar puntos utilizando otro proceso que no sea el de generación de puntos. En este proceso, el *smart contract* comprueba que la dirección que ejecuta el proceso esté dada de alta en el registro de comerciantes controlado por el mismo *smart contract*, y que ha depositado la cantidad económica equivalente a los puntos que desea generar.

Se garantiza el anonimato del cliente, ya que estos no se han de registrar en el sistema, y solo se conoce su dirección en la blockchain. Se recomienda que cada cliente cree una cuenta nueva para operar en el sistema, y así evitar que pueda ser identificado a partir de los posibles datos que tenga asociado el cliente a dicha dirección.

El *smart contract* no permite la generación, entrega o compensación de puntos a comerciantes desafiliados. El *smart contract* comprueba, antes de ejecutar cualquier función a disposición de los comerciantes afiliados, si la dirección utilizada para ejecutar el proceso se encuentra en el registro de comerciantes afiliados al sistema. De esta manera, se garantiza que los comerciantes desafiliados (o no pertenecientes al consorcio) no puedan ejecutar ninguno de estos procesos.

Más allá de los requisitos de seguridad, se garantiza el equilibrio entre puntos generados y compensados por parte de cada comerciante. Por un lado, un comerciante debe efectuar un depósito de dinero en el fondo común para generar puntos, por lo que se evita la generación masiva de puntos dentro del sistema. Por otro lado, un comerciante únicamente puede

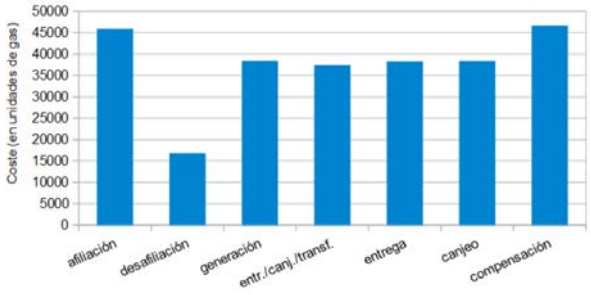



Figura 2. Coste (en unidades de gas) de cada función

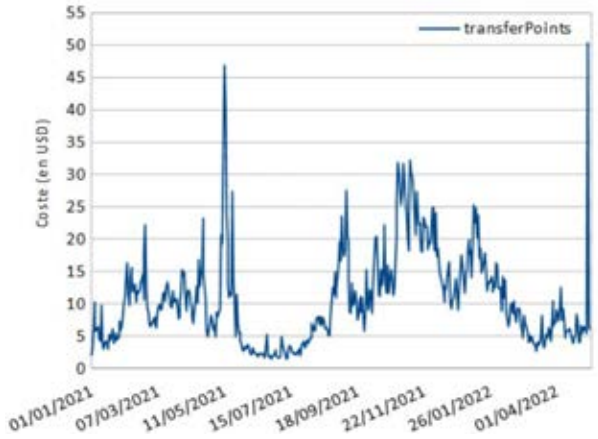



Figura 3. Coste (en USD) de transacción de la función *transferPoints* a lo largo del tiempo en Ethereum

recuperar del fondo común la cuantía equivalente al valor económico de los puntos que posea. Aunque un comerciante puede generar puntos y después ejecutar el proceso de compensación para volver a recuperar su dinero, no obtendría ningún beneficio en el proceso y el resto de comerciantes no se vería perjudicado por este comportamiento.

### *IV-B. Evaluación del coste*

La propuesta del esquema promocional puede ser implementada en diferentes plataformas. Es conveniente estudiar cada escenario de implantación para seleccionar la plataforma blockchain que mejor se adapte a cada escenario. Las blockchain públicas permiten ofrecer una mayor confianza, al no requerir de ninguna entidad centralizada. Ethereum es una blockchain pública de segunda generación, la segunda en el ecosistema blockchain en cuanto a capitalización y la más utilizada para el despliegue de *smart contracts*. Por lo tanto, en esta sección se evalúa la viabilidad en coste económico de nuestra propuesta utilizando Ethereum.

En Ethereum, la ejecución de transacciones tiene asociado un coste económico que sirve como incentivo para que los mineros validen y publiquen la transacción en un nuevo bloque. Este coste se paga en ethers. Debido a la alta volatilidad de esta criptomoneda, Ethereum utiliza una medida conocida como gas para fijar dicho coste. Para obtener una estimación del coste (en unidades de gas) de la ejecución de las diferentes funciones, se ha desplegado el *smart contract* sobre Ganache utilizando Truffle [10].

La figura 2 muestra el coste (en unidades de gas) de la ejecución de todas las funciones definidas. El coste de entr./canj./transf. representa el coste de ejecutar la función *transferPoints*, que contenía las operaciones para realizar la entrega, el canjeo y la transferencia entre clientes, cuando esta última se permite.

El coste final de ejecución en Ethereum se obtiene del producto entre el coste (en unidades de gas) y el precio por unidad de gas. Teniendo en cuenta la alta volatilidad de las criptomonedas, para evaluar los costes en dinero fiat, se considera el período de tiempo desde el 1 de enero de 2021 hasta el 4 de mayo de 2022. Primero se ha obtenido el coste medio diario de transacción por unidad de gas y la tasa de cambio diaria entre ether y dólar a lo largo de este período, a partir de los datos proporcionados por [11]. La figura 3 muestra la evolución del coste (en USD) de ejecución de la función *transferPoints*. Las demás funciones siguen una tendencia muy parecida.

Tabla II
ESTADÍSTICAS DE COSTES DE EJECUCIÓN (USD) SOBRE ETHEREUM.

| | Máximo | Mínimo | Media | SD |
|---|---|---|---|---|
| Despliegue Transf. | 1525,2502 | 38,4533 | 323,1979 | 219,4109 |
| Despliegue NoTransf. | 1614,6920 | 40,7082 | 342,1505 | 232,2773 |
| Afiliación | 61,6275 | 1,5537 | 13,0588 | 8,8653 |
| Desafiliación | 22,7460 | 0,5735 | 4,8198 | 3,2721 |
| Generación | 51,5702 | 1,3001 | 10,9276 | 7,4185 |
| Entr./Canj./Transf. | 50,1740 | 1,2649 | 10,6318 | 7,2177 |
| Entrega | 51,3757 | 1,2952 | 10,8864 | 7,3905 |
| Canjeo | 51,5246 | 1,2990 | 10,9180 | 7,4119 |
| Compensación | 62,7340 | 1,5816 | 13,2932 | 9,0244 |

La tabla II muestra los parámetros estadísticos máximo, mínimo, media y desviación típica de la ejecución de las distintas funciones a lo largo del período de análisis. Además, incluye los parámetros estadísticos del coste de despliegue del *smart contract* para el escenario con puntos transferibles entre clientes ($DespliegueTransf.$) y el del *smart contract* para el de puntos no transferibles ($DespliegueNoTransf.$).

Los resultados reflejan que los costes son muy elevados, aunque todo dependerá del coste/beneficio en escenarios concretos.

Una posible solución para reducir los costes es utilizar una *Rollup* [13], es decir, una solución de segunda capa sobre Ethereum que permite realizar transacciones de manera más rápida, y a un menor coste económico, reduciendo la cantidad de transacciones y datos que se guardan en la red Ethereum. Para analizar el efecto de la utilización de una solución de este tipo en nuestra propuesta, se ha seleccionado Polygon [12], con la que se ha seguido el mismo análisis realizado con Ethereum. En la figura 4 y en la tabla III se muestran los resultados de este análisis.

Como se puede apreciar, la disminución de los costes económicos, utilizando Polygon, es considerable, siendo superior al 99 %. Pero para evaluar la aplicabilidad de la propuesta en diferentes escenarios y utilizando este tipo de redes blockchain, debe tenerse en cuenta el coste de ofrecer ese servicio frente al beneficio económico de cada una de las transferencias de puntos, es decir, el valor del descuento o recompensa que obtendrán los clientes. Consideremos el caso de una transferencia entre dos clientes, en el que el coste por transferencia es de $0,06 (en el peor caso evaluado), por

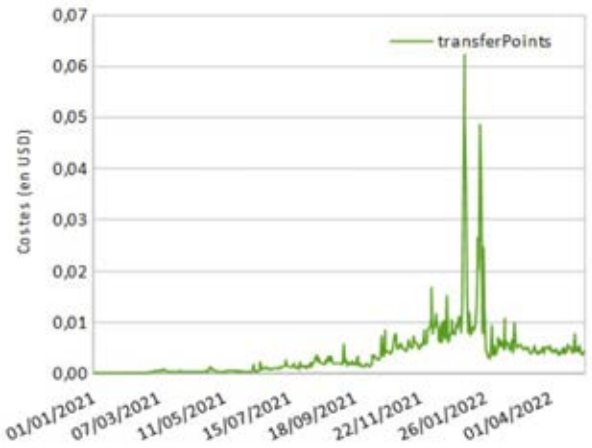



Figura 4. Coste (en USD) de transacción de la función *transferPoints* a lo largo del tiempo en Polygon

Tabla III
ESTADÍSTICAS DE COSTES DE EJECUCIÓN (USD) SOBRE POLYGON.

| | Máximo | Mínimo | Media | SD |
|---|---|---|---|---|
| Despliegue Transf. | 1,8932 | $< 0,0001$ | 0,1094 | 0,1801 |
| Despliegue No Transf. | 2,0042 | $< 0,0001$ | 0,1158 | 0,1906 |
| Afiliación | 0,0765 | $< 0,0001$ | 0,0044 | 0,0073 |
| Desafiliación | 0,0282 | $< 0,0001$ | 0,0016 | 0,0027 |
| Generación | 0,0640 | $< 0,0001$ | 0,0037 | 0,0061 |
| Entr./Canj./Transf. | 0,0623 | $< 0,0001$ | 0,0036 | 0,0059 |
| Entrega | 0,0638 | $< 0,0001$ | 0,0037 | 0,0061 |
| Canjeo | 0,0640 | $< 0,0001$ | 0,0037 | 0,0061 |
| Compensación | 0,0779 | $< 0,0001$ | 0,0045 | 0,0074 |

lo que el valor económico total de la transferencia debe ser lo suficientemente elevado como para compensar este coste. Por ejemplo, en el caso del programa de puntos (Avios [14]) de Iberia, en el momento de escribir este artículo, se pueden canjear 9000 Avios por un viaje. Si bien se pueden obtener puntos por otros medios, también se pueden comprar lotes de Avios (por ejemplo, 8000 Avios por 164 €), siendo despreciable el coste de $0,06. Consideremos ahora un escenario en el que el valor de los puntos es más reducido, como en el caso de Starbucks [15], en cuyo programa de fidelización se ganan 1 o 2 puntos por cada $1 gastado, y se pueden canjear 25 puntos por un café. En este último caso, el coste de $0,06 supone entre el 3 % y el 6 % del valor económico de cada transferencia, el cual podría considerarse bajo y tenido en cuenta a la hora de calcular el valor de los puntos en el sistema.

## V. CONCLUSIONES

Ha quedado demostrado que es posible implementar un esquema promocional por puntos utilizando *smart contracts*, aprovechando así las propiedades que ofrece la blockchain para garantizar los requisitos de seguridad necesarios. De esta forma, se ha logrado desarrollar una propuesta en la que se puede tanto impedir como permitir la transferencia de puntos entre clientes.

En cuanto a la viabilidad económica, aunque la implementación en una blockchain pública como Ethereum podría no ser viable para diferentes escenarios, hemos comprobado que es posible utilizar soluciones de segunda capa que permiten aumentar la velocidad en la que se validan las transacciones y reducir el coste económico de las mismas, como se ha mostrado con la utilización de Polygon.

Como trabajo futuro, se analizará la utilización de otras soluciones blockchain, como las ZK-Rollup, o el uso de una blockchain permisionada para proporcionar privacidad en la gestión de puntos del consorcio.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Economia, Industria y Competitividad (MINECO), la Agencia Estatal de Investigacion (AEI) y European Regional Development Funds (ERDF) bajo los proyectos BlobSec PID2021-122394OB-I00 y FeltiCHAIN RTI2018-097763-B-I00 (MINECO/AEI/ERDF, EU).

## REFERENCIAS

[1] RetailMeNot. https://www.retailmenot.com/ (accedido el 8 de mayo de 2022).
[2] ResearchAndMarkets. https://www.researchandmarkets.com (accedido el 8 de mayo de 2022).
[3] Chia-Hung Liao, Ya-Wen Teng, and Shyan-Ming Yuan.: "Blockchain-Based Cross-Organizational Integrated Platform for Issuing and Redeeming Reward Poin", en *In Proceedings of the Tenth International Symposium on Information and Communication Technology (SoICT 2019)*, pp. 407–411, 2019.
[4] O. Sönmeztürk, T. Ayav and Y. M. Erten: "Loyalty program using blockchain", en *2020 IEEE International Conference on blockchain (blockchain)*, pp. 509–516, 2020.
[5] Bülbül, Şeref and İnce, Gökhan: "Blockchain-based framework for customer loyalty program", en *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 342–346, 2018.
[6] Pramanik, Bijon Kumar and Rahman, AZM and Li, Mei: "Blockchain-based reward point exchange systems", en *2020 IEEE International Conference on Blockchain (Blockchain)*, pp. 509–516, 2020.
[7] Nguyen, Cong T and Hoang, Dinh Thai and Nguyen, Diep N and Pham, Hoang-Anh and Tuong, Nguyen Huynh and Dutkiewicz, Eryk: "Blockchain-based Secure Platform for Coalition Loyalty Program Management", en *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2021.
[8] Jaewon Choi: "Modeling the intergrated customer loyalty program on blockchain technology by using credit card", en *International Journal on Future Revolution in Computer Science & Communication Engineering*, pp. 388–391, 2018.
[9] Sönmeztürk, Osman and Ayav, Tolga and Erten, Yusuf M: "Loyalty program using blockchain", en *2020 IEEE International Conference on Blockchain (Blockchain)*, pp. 509–516, 2020.
[10] "Truffle Suite". Truffle Blockchain Group. https://trufflesuite.com/ (accedido el 9 de mayo de 2022).
[11] "Ethereum Average Gas Price Chart". Etherscan. https://etherscan.io/chart/gasprice (accedido el 9 de mayo de 2022)
[12] "Polygon PoS Chain Average Gas Price Char". Polygonscan. https://polygonscan.com/chart/gasprice (accedido el 9 de mayo de 2022)
[13] "An Incomplete Guide to Rollups". Vitalik Buterin's website. https://vitalik.ca/general/2021/01/05/rollup.html (accedido el 8 de mayo de 2022).
[14] "Qué son los Avios". Iberia. https://www.iberia.com/es/iberiaplus/como-funciona/ (accedido el 8 de mayo de 2022).
[15] "Starbucks Rewards". Starbucks Coffee Company. https://www.starbucks.com/rewards (accedido el 8 de mayo de 2022).

# Gotta Catch 'em All: Aggregating CVSS Scores

Ángel Longueira-Romero
*Industrial Cybersecurity*
*Ikerlan Technology Research Centre (BRTA)*
Arrasate/Mondragón, Spain
alongueira@ikerlan.es

Jose Luis Flores
*Industrial Cybersecurity*
*Ikerlan Technology Research Centre (BRTA)*
Arrasate/Mondragón, Spain
jlflores@ikerlan.es

Rosa Iglesias
*Industrial Cybersecurity*
*Ikerlan Technology Research Centre (BRTA)*
Arrasate/Mondragón, Spain
riglesias@ikerlan.es

Iñaki Garitano
*Dept. of Electronics and Computing*
*Mondragon Unibertsitatea*
Arrasate/Mondragón, Spain
igaritano@mondragon.edu

***Abstract*—Security metrics are not standardized, but international proposals such as the Common Vulnerability Scoring System (CVSS) for quantifying the severity of known vulnerabilities are widely used. Many CVSS aggregation mechanisms have been proposed in the literature. Nevertheless, factors related to the context of the System Under Test (SUT) are not taken into account in the aggregation process; vulnerabilities that in theory affect the SUT, but are not exploitable in reality. We propose a CVSS aggregation algorithm that integrates information about the functionality disruption of the SUT, exploitation difficulty, existence of exploits, and the context where the SUT operates. The aggregation algorithm was applied to OpenPLC V3, showing that it is capable of filtering out vulnerabilities that cannot be exploited in the real conditions of deployment of the particular system. Finally, because of the nature of the proposed algorithm, the result can be interpreted in the same way as a normal CVSS.**

***Index Terms*—CVSS, security metrics, aggregation, attack graphs, vulnerabilities.**

## I. Introduction

System security quantification is not an easy task [1]. There exist both a lack of consensus and standardization around security metrics [2], [3], [4], [5], [6], [7], [8]. For this reason, research efforts keep aiming to unify this field [9].

Among these efforts, the Common Vulnerability Scoring System (CVSS) is a widely extended standard for vulnerability quantification [10]. CVSS is a public framework that provides a standardized method for assigning quantitative values to security vulnerabilities according to their severity. A CVSS score is a decimal number in the range [0, 10][1] [11].

The CVSS is aimed to quantify the severity of vulnerabilities in individual and specific software items, however the majority of systems are actually a composition of simpler isolated items with different interdependencies. This situation highlights one of the biggest problems related to security quantification [12], the difficulty to really measure the global security state of a composite system. To do so, it would be necessary to aggregate each individual CVSS value into a global one in a consistent and coherent way.

The official CVSS documentation does not propose any kind of aggregation mechanism, and nowadays, there is no standardized method [13]. In addition to this, previous research works do not usually integrate contextual or interdependency information about the vulnerabilities to update the CVSS. This means that aspects such as whether affected functionalities, the environment of deployment, or the existence of exploits are usually neglected.

Context is a critical aspect to integrate in the aggregation process. This can be illustrated using a device implementing multiple functionalities as an example. To perform those functionalities, usually it will contain assets that implement those functionalities. But depending on the context where the device is deployed, some of its functionalities might not be needed. So the assets implementing unused functionalities would be disabled, and therefore, their vulnerabilities could not be exploited. It can also be the case that the asset implementing a functionality is simply inaccessible, so it could not also be exploited.

This research proposes a novel aggregation algorithm for a set of CVSS values[2]. This approach is based on the Extended Dependency Graphs (EDGs) proposed by Longueira-Romero *et al.* [14]. Because EDGs are capable of modeling dependencies, this algorithm can also be applied to computer networks. Our proposal is capable of selecting the most relevant CVSS to be aggregated, taking into account four different context-related properties of the System Under Test (SUT):

1) Functionality disruption.
2) Exploitation difficulty.
3) Existence of exploits, and their development state.
4) Context of deployment.

This approach increases the granularity of the CVSS base, environment and temporal metrics, where not every possible

[1]The latest version at the time this paper was written is version 3.1.

[2]The Python code implementing the aggregation algorithm is available at GitHub https://github.com/aaalongueira/CVSS_Aggregation.

value in the scale $[0, 10]$ is achievable, or the result of changing the value of a submetric has almost no effect on the final CVSS [13], [15]. Moreover, our proposal is capable of detecting which branch in the EDG is contributing the most (more critical) to the final score.

This paper is organized as follows: We review existing aggregation methods in Section II. Our proposal is explained in Section III, and tested in a use case in Section IV. Finally, Section V contains the conclusions and future work of this research.

## II. Related Work

Nowadays, there is no widely-accepted method to aggregate CVSS values for software composition. All of them can be classified into one of the following categories [16], [17]: (1) Arithmetic Aggregation, (2) Attack Graph-based Aggregation, and (3) Bayesian Network-based Aggregation.

### A. Arithmetic Aggregation

This method uses arithmetic operations to aggregate the values [18], [19], [20], [21]. Common examples of this approach are taking the maximum of the CVSS values, their arithmetic mean, or a combination of them. For example, Heyman *et al.* [18], proposed an algorithm to aggregate CVSS values in dependency graph that is based on taking the maximum value in each case, according to certain conditions.

Although their simplicity makes them suitable for initial approximations, their results can be biased in two ways:

1) **Exploitable by quantity:** When a system poses several vulnerabilities that by their own are not critical and cannot be exploited, they can sum up to an aggregated value of a high impact vulnerability (overfitting). This can happen when multiple simple mechanisms are combined as the aggregation algorithm.
2) **Exploitable by criticality:** When there exist a critical vulnerability, the whole system will be usually classified as critical. Nevertheless, that vulnerability might not be exploitable, nor being affecting the functionality of the system. This is specially common when using the maximum as the aggregation algorithm.

### B. Attack Graph-based Aggregation

This approach models the relationships between vulnerabilities using attack graphs, converting CVSS scores into probabilities [22], [23], [24], [25], [26], [27]. In this way, both the CVSS value and the place of the vulnerability in the whole graph are taken into account.

Cheng *et al.* in [16] proposed a graph-based aggregation method that uses the underlying metrics of CVSS, where the dependency relationships between vulnerabilities are usually visible. As the center of the aggregation algorithm, they use the product of the CVSS used as probabilities, also known as the join probability of both vulnerability.

The main drawback with these approaches is that the relationship between individual vulnerabilities cannot be obtained straightforwardly from existing databases. This means that establishing a relation between two vulnerabilities implies that they can be chained during an attack, which is not always obvious. Moreover, factors such as exploitability of the vulnerabilities, or existing exploits are not taken into account.

### C. Bayesian Network-based Aggregation

Going a step further, these methods integrate the conditional relationship between vulnerabilities, modeling them using Bayesian networks [28], [29], [30]. Poolsappasit *et al.* [29] proposed a CVSS aggregation framework using Bayesian networks. They used the Bayesian probability factorization formula as the aggregation mechanism:

$$p(x) = \prod_{i=0} p(x_v | x_{pa(v)})$$

Bayesian network-based approaches have to deal with establishing the relationships between the vulnerabilities, but also with the calculation of conditional probabilities, that have to be usually estimated. As the previous ones, these techniques do not integrate information about how functionality if affected by existing vulnerabilities, or the possibility to actually exploit them.

## III. Proposed Approach for Metric Aggregation

In this paper, we propose a CVSS aggregation algorithm inspired by the risk propagation formula [31] described in MAGERIT [32], [33]. First, we describe the corrections factors involved in our proposal. Then, the aggregation formula is introduced. Finally, the algorithm and the interpretation of the results is explained in detail.

### A. Correction Factors

The proposed aggregation algorithm integrates correction factors to adapt the formula described in MAGERIT. These correction factors apply individually for each CVSS, except for the average and summarized factors. Correction factors are summarized in Table I.

1) *Functionality factor ($\rho$):* This correction factor represents whether any functionality of the systems is affected by its vulnerabilities. It is represented by a binary value, being $0$ when no functionality is affected, and $1$ when any of them is affected. For example, a cryptographic library with a vulnerability in SHA1. If the SUT does not make use of SHA1 in any way, the vulnerability would not be exploitable, and could be removed from the analysis ($\rho = 0$).
2) *Deepness Factor ($\beta$):* This factor represents the difficulty of chained exploitation of each vulnerability. It is represented by a value between $[0, 1]$ inversely proportional to the amount of assets to compromise in order to exploit vulnerability. Vulnerabilities close to the entry point will account more for the final aggregation, whereas those that are far away will account less. In this approach, linear interpolation is proposed to calculate the weight of each layer, because of its simplicity. Nevertheless, different interpolations could be used according to the

TABLE I: Correction factors proposed for adapting the Bayesian sum proposed in MAGERIT.

| CORRECTION FACTOR | DESCRIPTION | AUTOMAtED |
|---|---|---|
| Functionality factor ($\rho$) | Binary value indicating whether a vulnerability affects or not the functionality of the SUT. | ■ |
| Deepness factor ($\beta$) | Value between $[0, 1]$ proportional to the position of the affected asset in the EDG of the SUT. | ■ |
| Context factor ($\gamma$) | Binary value indicating vulnerability exploitability in the real and particular conditions of the SUT. | □ |
| Exploit factor ($\mu$) | Existence of a public exploit, proportional to its state of development: Not defined ($\mu = 0.5$), Theoretical ($\mu = 1.25$), Proof-Of-Concept ($\mu = 1.5$), Functional ($\mu = 1.75$), and Automated ($\mu = 2$). | ■ |
| Summarized factor ($\lambda$) | This factor summarizes the effect of all the above ones, $\lambda = \rho\beta\gamma\mu\sigma$. | ■ |
| Average factor ($\sigma$) | Function that adjust the value of the sum to avoid its rapid evolution to 10. | ■ |

criticality of the system. Fig. 1 shows the corresponding $\beta$ for a four-layer system.

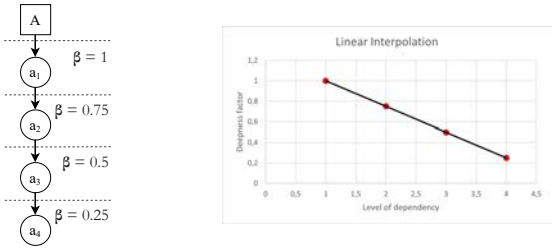



Fig. 1: Calculation of the deepness factor for a four-layer of dependency example.

3) *Context factor ($\gamma$):* This factor considers whether the exploitation of a vulnerability is actually possible in the real scenario where the system is deployed. It is represented by a binary value, where 0 indicated that it is not possible, and 1, that it is possible. It is calculated comparing the attack vector of the CVSS with the real conditions where the device is deployed. For example, this can happen when a vulnerability with a high CVSS score needs physical access to be exploited, but in reality the device is physically isolated. To reflect this, the CVSS should be updated, lowering the resulting value [16]. This factor aims to complement the existing submetrics in the temporal and the environment metrics of the CVSS. Both the temporal and the environmental scores lack of an "isolated" value for the attack vector.
4) *Exploit factor ($\mu$):* This factor accounts for the existence of a public exploit for a given vulnerability, being proportional to its state of development. The temporal score of the CVSS already implements this feature, but the CVSS values are not updated in practice [15]. Moreover, taking into account the temporal score has almost no effect as opposed to using the raw initial base score. This means that a CVSS just considering the base score is higher than a CVSS considering an exploit code maturity of "functional exploit exists". To solve this issue, we introduce the following values for the exploit factor: Not defined ($\mu = 0.5$), Theoretical ($\mu = 1.25$), Proof-Of-Concept ($\mu = 1.5$), Functional ($\mu = 1.75$), and Automated ($\mu = 2$). These values are equivalent to the scale defined in the CVSS Specification Document [10].
5) *Summarized factor ($\lambda$):* The $\lambda$ factor accounts for the effect of all the factors above:

$$\lambda = \rho\beta\gamma\mu \tag{1}$$

6) *Average factor ($\sigma$):* This factor defines the behavior of the aggregation function. It can be chosen as needed (*e.g.*, the arithmetic or harmonic mean), but taking into account all the values to be added.

### B. Aggregation Formula

The aggregation function is defined as:

$$\Gamma(\overrightarrow{V}) = 10 - \frac{1}{\sigma} f(\overrightarrow{V}) \tag{2}$$

Where $\overrightarrow{V}$ is a vector $(cvss_0, cvss_1, \ldots, cvss_n)$ with all the corrected CVSS values to be added, $cvss$, being $n$ the last value to be added. $f(\overrightarrow{V}) = a_n$ is defined as the following recursive function:

$$a_n = 10\left[1 - \left(1 - \frac{\lambda_{a_{n-1}}}{10} a_{n-1}\right) \cdot \left(1 - \frac{\lambda_{cvss_n}}{10} cvss_n\right)\right] \tag{3}$$

Where the base case is defined as:

$$a_0 = \lambda_{cvss_0} cvss_0 \tag{4}$$

### C. Algorithm

The proposed aggregation algorithm is divided into the following steps (see Fig. 2):

1) Calculation of the correction factors for each CVSS,
2) Calculation of the summarized factor for each CVSS,
3) Calculation of the corrected CVSS values,
4) Calculation of the average correction function, and
5) Aggregation.

Notice that the dependency graph of the SUT, the vulnerabilities associated to each element of the dependency graph, and their CVSS value are needed.

*1) Correction factors for each CVSS:* The first step obtains the values of each correction factor for each CVSS:

1) **Functionality factor ($\rho$):** This factor is obtained using the description provided in the corresponding CVE of each CVSS. The description provides enough information to decide whether the functionality of the system is affected.
2) **Context factor ($\gamma$):** This factor is obtained by comparing the value of the Attack Vector (AV) submetric of the CVSS, with the real environment of deployment of the SUT.

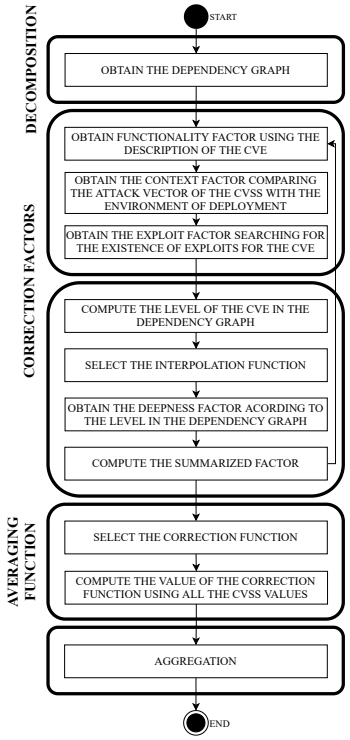



Fig. 2: Flowchart showing the main steps of the aggregation algorithm for each CVSS.

3) **Exploit factor ($\mu$):** To obtain this factor, public databases have to be queried to find any potential exploit for each vulnerability.
4) **Deepness factor ($\beta$):** For any given CVSS, its value is obtained according to the deepness in the exploit chain for the SUT [14].

*2) Summarized factor for each CVSS:* The summarized factor, $\lambda$, is obtained by multiplying all the corrections factors obtained in the previous step, following Equation 1.

*3) Corrected CVSS values:* The corrected CVSS values are obtained by multiplying each CVSS by its corresponding summarized factor, $\lambda$. At this point, it is necessary to check for overflows, because the exploitation factor generated corrected CVSS values higher than 10. Values higher than 10 are set to 10 at this stage.

*4) Correction function:* At this point, it is necessary to choose an averaging function. Choosing one function over the other will cause the aggregation result to grow slower or faster toward 10 in each addition. In this case, and for the sake of clarity, we chose the arithmetic mean, but any other kind of mean (*e.g., harmonic mean*) could be used according to each scenario.

*5) Aggregation:* Finally, the aggregated value is computed using Equation 2.

### D. Interpretation of the result

The advantage of this method is that the result can be interpreted in the same way that a normal CVSS would be interpreted. This is because of the correction factors in Equation 2, that only let the algorithm return high values when vulnerabilities with high CVSS values are exploitable in reality ($\lambda$ is close to 1). This mechanism ensures that multiple aggregated low CVSS values do not result in a critical score just because there are a large number of them.

## IV. Use Case

To test the potential of our proposal, we analyzed Version 3 of OpenPLC project, obtaining a CVSS aggregated value for its vulnerabilities using the proposed algorithm.

OpenPLC is the first functional open source Programmable Logic Controller (PLC), both in software and hardware [34]. It was mainly created for research purposes, because it provides its entire source code [35], [36]. The current version of the project is OpenPLC V3 [37].

### A. Use Case Scenario

For this use case, we are going to make the next assumptions:

- The system executing OpenPLC V3 is deployed in an isolated network.
- The system running OpenPLC V3 is physically isolated.
- The attacker is an insider without access to the systems.
- The reference point for the deepness factor will be the `webserver.py` in Fig. 3.

### B. Structure of OpenPLC

The first step was to obtain the inner structure of OpenPLC V3 using the Extended Dependency Graph (EDG) proposed in [14]. To simplify the obtained graph, we only represented the shortest path to each node, so the worst case scenario (more accessible from the outside) is considered. The result is shown in Fig. 3.

### C. Calculation of the Correcting Factors

OpenPLC V3 has five vulnerabilities: two vulnerabilities affecting `libgcc_s`, and three vulnerabilities affecting `libc`. Table II shows each vulnerability in more detail.

From these data, it is possible to obtain all corrections factors for each vulnerability, as follows (Table II summarizes the results):

*1) Functionality Factor ($\rho$)r:* This factor is obtained from the analysis of the description of each CVE. From these data, we have to decide whether the functionality of OpenPLC V3 is affected (“1”) or not (“0”).

*2) Deepness Factor ($\beta$)b:* By taking a look at Fig. 3, it can be seen that the maximum deepness level is four. So the possible values for the deepness factor are the ones shown in Fig. 1. More precisely, vulnerabilities CVE-2019-15847 and CVE-2018-12886 have a deepness factor of 0.25, because they are at level four. By contrast, vulnerabilities CVE-2017-18269, CVE-2018-11236, and CVE-2018-11237 have a deepness factor of 0.5, because they are at level three.

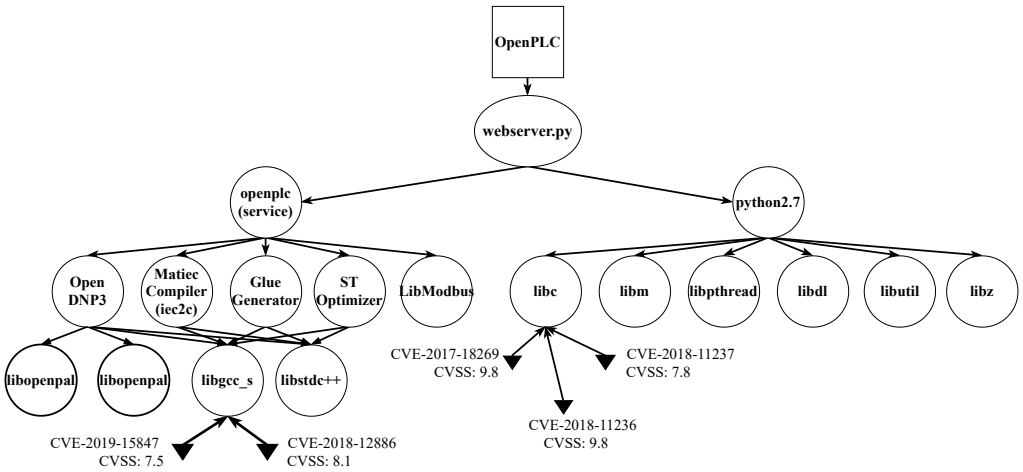



Fig. 3: Extended Dependency Graph of OpenPLC V3. Circles represent individual assets, black triangles are the vulnerabilities associated to each asset, and the square represent the entry point to the system, or root node of dependency.

TABLE II: Vulnerabilities present in OpenPLC V3. For each one, the CVSS is shown, together with their associated Attack Vector (AV), and their correction factors.

| CVE | CVSS | Attack Vector | Functionality ($\rho$) | Deepness ($\beta$) | Context ($\gamma$) | Exploit ($\mu$) | Summarized ($\lambda$) | Corrected CVSS |
|---|---|---|---|---|---|---|---|---|
| CVE-2017-18269 | 9.8 | Network | 1 | 0.5 | 1 | 1.25 | 0.625 | 6.125 |
| CVE-2018-11236 | 9.8 | Network | 0 | 0.5 | 1 | 0 | 0 | 0 |
| CVE-2018-11237 | 7.8 | Local | 1 | 0.5 | 0 | 1.25 | 0 | 0 |
| CVE-2018-12886 | 8.1 | Network | 1 | 0.25 | 1 | 1.25 | 0.313 | 2.530 |
| CVE-2019-15847 | 7.5 | Network | 1 | 0.25 | 1 | 1.25 | 0.313 | 2.344 |

*3) Context Factor* $(\gamma)g$*:* From the initial assumptions, insiders can only exploit the existing vulnerabilities from the local network. This means that every vulnerability that has an attack vector of "network" (N) can be exploited, thus CVE-2017-18269, CVE-2018-11236, CVE-2018-12886, and CVE-2019-15847 are exploitable by the attacker. Vulnerabilities whose attack vector is "local" (L) cannot be exploited, because physical access is needed. Therefore, CVE-2018-11237 cannot be exploited.

*4) Exploit Factor* $(\mu)m$*:* Public databases have to be queried to find existing exploits for each vulnerability. According to their state of development, a different value is assigned.

*5) Summarized Factor* $(\lambda)l$*:* The summarized factor for each vulnerability is obtained as the product of the previous factors, as shown in Equation 1. At this step, by taking a look at the resulting values of $\lambda$, it is possible to know which CVSS will contribute to the final aggregation and in which percentage ($\lambda > 0$), and which ones will not contribute at all ($\lambda = 0$).

*6) Average Factor* $(\sigma)s$*:* Finally, we obtained the average factor by calculating the arithmetic mean of all the initial CVSS values: $\sigma = 8.6$.

### *D. Aggregation*

The previous step before the aggregation is obtaining the corrected CVSS value for each initial CVSS. This is done by multiplying each CVSS by their corresponding summarized value ($\lambda$). The corrected values are shown in Table II.

Finally, the aggregation is performed using the corrected CVSS values. The aggregation is an iterative process that takes the first two values to be added, and adds them using Equation 2. Then, this result is added to the third value to be added, and so on, until there are no more values.

For OpenPLC V3, this process returns a final aggregated value of 9.1. Without the correction factors, the result would be 10. Nevertheless, taking into account features such as the exploitability of the vulnerabilities, the context of the SUT, or its functionalities, we can select the most important CVSS values to be aggregated. With such process, the total amount of CVSS values to be added is simplified. This also helps to simplify potential attack paths.

This result was obtained aggregating three of the five CVSS values present in OpenPLC V3. The associated CVSS for CVE-2018-11236 and CVE-2018-11237 were not taking into account for the aggregation, because they do not affect to any functionality of the system, Moreover, CVE-2018-11237 cannot be exploited in the conditions described in the use case.

CVE-2017-18269 (with an associated CVSS of 9.8) is the vulnerability with the highest value for $\lambda$. Therefore, it is going to contribute the most to the final aggregated value. CVE-2018-12886 and CVE-2019-15847 follow with a CVSS of 8.1 and 7.5 respectively. As it is shown, the selected vulnerabilities have a high CVSS, so it is expected that the aggregated value would be also high. This is reflected in the obtained result of 9.1.

Finally, it is worth highlighting that the final result is lower

than the highest CVSS value present in OpenPLC V3. This difference is due to the effect of the correction factors: as the CVE-2017-18269 is further away from the entry point of the system (in layer 3), its real CVSS value in lower.

## V. Conclusions and Future Work

In this research work, we proposed a new aggregation algorithm for CVSS values. The proposed approach integrates correction factors to select the most relevant CVSS values to be added based on contextual information. For each vulnerability, we check for:

1) Functionality disruption.
2) Exploitation difficulty.
3) Existence of exploits, and their development state.
4) Context of deployment.

We assigned a different correction factor to each one of the previous properties to further ponder the initial CVSS value and adjust it to the real context where the system is operating.

The proposed aggregation algorithm was applied to OpenPLC V3 in a use case. Two of the existing vulnerabilities were filtered out by the algorithm, as they cannot be exploited in the described context of OpenPLC V3. The rest of the vulnerabilities were aggregated, and the result (9.1) was indeed lower than the highest CVSS present in the system (9.8). This shows that the CVSS for each vulnerability was correctly adjusted to the real context of deployment of OpenPLC V3.

As future work, we plan to perform the aggregation at the submetric level of the CVSS, instead of using the base metric value, giving more granular values for each factor.

## Acknowledgements

Iñaki Garitano is a member of the Intelligent Systems for Industrial Systems research group at Mondragon Unibertsitatea (IT1676-22), supported by the Department of Education, Universities and Research of the Basque Government. This work was partially supported by the *Ayudas Cervera para Centros Tecnológicos* grant of the Spanish Center for the Development of Industrial Technology (CDTI) under the project EGIDA (CER-20191012), and by the Basque Country Government under the ELKARTEK program, project REMEDY - Real Time Control And Embedded Security (KK-2021/00091).

## References

[1] S. Pfleeger and R. Cunningham, "Why measuring security is hard," *IEEE Security Privacy*, July 2010.
[2] S. M. Bellovin, "On the brittleness of software and the infeasibility of security metrics," *IEEE Security & Privacy*, vol. 4, no. 4, pp. 96–96, 2006.
[3] A. Atzeni and A. Lioy, "Why to adopt a security metric? a brief survey," *Advances in Information Security*, vol. 23, pp. 1 – 12, 2006.
[4] V. Verendel, "Quantified security is a weak hypothesis: A critical survey of results and assumptions," in *Proceedings of the 2009 Workshop on New Security Paradigms Workshop*, NSPW '09, (New York, NY, USA), pp. 37–50, ACM, 2009.
[5] S. Stolfo, S. M. Bellovin, and D. Evans, "Measuring security," *IEEE Security Privacy*, May 2011.
[6] W. H. Sanders, "Quantitative security metrics: Unattainable holy grail or a vital breakthrough within our reach?," *IEEE Security Privacy*, vol. 12, no. 2, pp. 67–69, 2014.
[7] M. Rudolph and R. Schwarz, "A critical survey of security indicator approaches," in *2012 Seventh International Conference on Availability, Reliability and Security*, pp. 291–300, Aug 2012.
[8] S. Sentilles, E. Papatheocharous, and F. Ciccozzi, "What do we know about software security evaluation? a preliminary study," in *QuA-SoQ@APSEC*, 2018.
[9] D. G. Ángel Longueira-Romero, Rosa Iglesias and I. Garitano, "How to Quantify the Security Level of Embedded Systems? A Taxonomy of Security Metrics," in *Proceedings of the 2020 18th IEEE International Conference on Industrial Informatics (INDIN)*, 2020.
[10] FIRST - global Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System (CVSS)." https://www.first.org/cvss/v3-1/, 2021-02-03.
[11] National Institute for Standards and Technology (NIST), "National Vulnerability Database NVD — Vulnerabilities." https://nvd.nist.gov/vuln/search, 2021-02-03.
[12] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," vol. 49, dec 2016.
[13] H. Howland, "Cvss: Ubiquitous and broken," *Digital Threats*, sep 2021.
[14] A. Longueira-Romero, R. Iglesias, J. L. Flores, and I. Garitano, "A novel model for vulnerability analysis through enhanced directed graphs and quantitative metrics," *Sensors*, vol. 22, no. 6, 2022.
[15] J. Spring, E. Hatleback, A. Householder, A. Manion, and D. Shick, "Time to change the cvss?," *IEEE Security & Privacy*, vol. 19, no. 2, pp. 74–78, 2021.
[16] P. Cheng, L. Wang, S. Jajodia, and A. Singhal, "Aggregating cvss base scores for semantics-rich network security metrics," in *2012 IEEE 31st Symposium on Reliable Distributed Systems*, pp. 31–40, 2012.
[17] B. Kordy, L. Piètre-Cambacédès, and P. Schweitzer, "Dag-based attack and defense modeling: Don't miss the forest for the attack trees," *Computer Science Review*, vol. 13-14, pp. 1–38, 2014.
[18] T. Heyman, R. Scandariato, C. Huygens, and W. Joosen, "Using security patterns to combine security metrics," in *2008 Third International Conference on Availability, Reliability and Security*, pp. 1156–1163, 2008.
[19] Z. Song, Y. Wang, P. Zong, Z. Ren, and D. Qi, "An empirical study of comparison of code metric aggregation methods–on embedded software," in *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 114–119, 2019.
[20] M. Walkowski, M. Krakowiak, M. Jaroszewski, J. Oko, and S. Sujecki, "Automatic cvss-based vulnerability prioritization and response with context information," in *2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, 2021.
[21] C. Fruhwirth and T. Mannisto, "Improving cvss-based vulnerability prioritization and response with context information," in *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 535–544, 2009.
[22] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *Data and Applications Security XXII* (V. Atluri, ed.), (Berlin, Heidelberg), pp. 283–296, Springer Berlin Heidelberg, 2008.
[23] S. Zhang, X. Ou, A. Singhal, and J. Homer, "An empirical study of a vulnerability metric aggregation method," in *Mission Assurance and Critical Infrastructure Protection*, 2011 World Congress in Computer Science, 2011-08-18 00:08:00 2011.
[24] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 75–85, 2012.
[25] M. Zhang, L. Wang, S. Jajodia, and A. Singhal, "Network attack surface: Lifting the concept of attack surface to the network level for evaluating networks' resilience against zero-day attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 310–324, 2021.
[26] J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. R. Rajagopalan, and A. Singhal, "Aggregating vulnerability metrics in enterprise networks using attack graphs," *Journal of Computer Security*, vol. 21, no. 4, pp. 561–597, 2013.
[27] L. Gallon and J.-J. Bascou, "Cvss attack graphs," in *2011 Seventh International Conference on Signal Image Technology Internet-Based Systems*, pp. 24–31, 2011.
[28] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," in *Proceedings of the 4th ACM Workshop on Quality of Protection*, QoP '08, (New York, NY, USA), p. 23–30, Association for Computing Machinery, 2008.

[29] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.
[30] P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy, "Using bayesian networks for cyber security analysis," in *2010 IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, pp. 211–220, 2010.
[31] M. A. Amutio, J. Candau, and J. A. Mañas, "MAGERIT V3.0. Methodology for Information Systems Risk Analysis and Management. Book III - Technical Guide," National Standard, Ministry of Finance and Public Administration, Madrid, Spain, 2012.
[32] M. A. Amutio, J. Candau, and J. A. Mañas, "MAGERIT V3.0. Methodology for Information Systems Risk Analysis and Management. Book I - The Method," National Standard, Ministry of Finance and Public Administration, Madrid, Spain, 2014.
[33] A. Syalim, Y. Hori, and K. Sakurai, "Comparison of risk analysis methods: Mehari, magerit, nist800-30 and microsoft's security management guide," in *2009 International Conference on Availability, Reliability and Security*, pp. 726–731, 2009.
[34] Thiago Alves, "OpenPLC Project." https://www.openplcproject.com/.
[35] T. R. Alves, M. Buratto, F. M. de Souza, and T. V. Rodrigues, "Openplc: An open source alternative to automation," in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 585–589, 2014.
[36] T. Alves and T. Morris, "Openplc: An iec 61,131–3 compliant open source industrial controller for cyber security research," *Computers & Security*, vol. 78, pp. 364–379, 2018.
[37] Thiago Alves, "OpenPLC V3." https://github.com/thiagoralves/OpenPLC_v3, 2021-05-15.

# Computación segura multiparte coútil para cálculo de funciones arbitrarias

Jesús A. Manjón
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
UNESCO Chair in Data Privacy
Avda. Països Catalans 26, Tarragona
jesus.manjon@urv.cat

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
UNESCO Chair in Data Privacy
Avda. Països Catalans 26, Tarragona
josep.domingo@urv.cat

***Resumen*—En la computación segura multiparte (MPC), varias entidades llevan a cabo conjuntamente un cálculo sobre sus respectivos valores de entrada manteniéndolos privados en todo momento. Aunque la MPC existe desde 1980, solamente el desarrollo reciente de compiladores de propósito general ha permitido el cálculo de funciones arbitrarias. Aun así, el uso de esta clase de compiladores requiere un esfuerzo y una habilidad notables por parte de los programadores: el código original se traduce a circuitos booleanos o aritméticos, lo que impone restricciones en el uso de iteraciones y llamadas recursivas, con las consiguientes dificultades de implementación. Para salvar este escollo, presentamos un sistema que permite la ejecución de un protocolo MPC sobre una función arbitraria expresada en código ordinario sin restricciones. Nuestro sistema evita la vinculación entre cada uno de los participantes y sus correspondientes entradas y salidas. Nuestro método se basa en un canal coútil anónimo y un mecanismo de reputación descentralizada, hace un uso limitado de criptografía y no necesita que los participantes sean semihonestos: basta con que sean racionales y actúen en interés propio, lo que incluye a participantes racionalmente maliciosos (que atacan el sistema si ello les beneficia). Nuestros experimentos demuestran que las reputaciones capturan correctamente el comportamiento de los participantes y aseguran la obtención de resultados correctos a los participantes con alta reputación.**

***Index Terms*—Computación multiparte, coutilidad, privacidad, computación por pares**

## I. Introducción

En la computación segura multiparte (MPC), varias entidades llevan a cabo conjuntamente un cálculo sobre sus respectivos valores de entrada manteniéndolos privados en todo momento. Desde la aparición de los primeros protocolos en los años 1980s ([1], [2], [3], [4]) y hasta hace poco, sólo existían protocolos MPC prácticos para ciertas aplicaciones específicas. De hecho, la aparición de compiladores de propósito general para calcular funciones arbitrarias es un avance reciente (véase [5] para un estado del arte de esta clase de compiladores). La mayoría de compiladores traducen el código de la computación a realizar a un circuito booleano o aritmético y emplean primitivas criptográficas como la compartición de secretos, la transferencia olvidadiza o los circuitos distorsionados (*garbled circuits*), entre otras, para generar el protocolo MPC.

Como se comenta en [5], el uso de los compiladores más avanzados requiere un gran esfuerzo y habilidad de programación, debido a las limitaciones inherentes de la representación en forma de circuito, principalmente en lo que respecta a las iteraciones y llamadas recursivas (que deben ser acotadas, pues hay que "aplanarlas").

En este artículo proponemos un protocolo MPC con las siguientes características:

- Es de propósito general, en el sentido de que funciona para cualquier tipo de cálculo, sin importar cuán complicado sea;
- Usa criptografía solamente para garantizar la autenticación y la confidencialidad de las comunicaciones;
- Supone la existencia de una comunidad de pares (P2P), en la que cada participante acumula una reputación de manera descentralizada;
- Se basa en la utilización de un canal anónimo coútil que garantiza que las entradas y los resultados, aun siendo visibles para algunos participantes, no pueden vincularse inequívocamente a los participantes a quienes corresponden;
- Proporciona resultados correctos y exactos siempre y cuando los participantes sean racionales, es decir, movidos por su propio interés. Esto no excluye a participantes racionales maliciosos, que podrían convertirse en atacantes si ello les beneficiase. Es decir, el modelo racional que adoptamos és más general que el modelo semihonesto pero menos que el modelo malicioso.

En la sección II presentamos el concepto de coutilidad y el sistema de reputaciones coútil utilizado. En la sección III detallamos los protocolos que conforman el sistema. La sección IV introduce un análisis de coutilidad. La sección V presenta nuestro trabajo experimental. Finalmente, en la sección VI resumimos las conclusiones del artículo.

## II. Coutilidad y reputación descentralizada coútil

En términos de teoría de juegos, un protocolo es autoimpuesto (*self-enforcing*) si, en cada paso del protocolo, ejecutar el siguiente paso sin desviarse es un equilibrio del subjuego restante. Por otra parte, un protocolo $\Pi$ es coútil [6] si y sólo si cumple las tres condiciones siguientes: (i) $\Pi$ es autoimpuesto; (ii) la utilidad conseguida por un agente que participa en el protocolo es más grande que la que conseguiría sin participar; (iii) no existe otro protocolo $\Pi'$ que aporte una utilidad mayor al conjunto de agentes ni a un agente concreto.

La primera condición asegura que, si un agente decide participar en el protocolo, no se desvíe en su ejecución. La segunda condición es necesaria para asegurar que participar en el protocolo es atractivo para todos los agentes. Finalmente, la tercera puede ser reformulada en términos de teoría de juegos diciendo que el protocolo es Pareto-óptimo.

En [7] se presenta una adaptación coútil del protocolo de gestión descentralizada de reputaciones EigenTrust [8]. Más allá de su co-utilidad, este protocolo posee las siguientes características: descentralización, pseudonimato de los participantes, coste computacional moderado, adecuada gestión de los nuevos participantes (para disuadir que participantes con baja reputación salgan y entren de nuevo al sistema con nuevas identidades) y resistencia a las manipulaciones de la reputación (como la autopromoción o la difamación). En nuestro artículo usaremos una versión simplificada de dicha gestión descentralizada de reputaciones.

## III. Sistema coútil para computación multiparte

### *III-A. Participantes y modelo de seguridad*

Los participantes en nuestro sistema son nodos en una red P2P que pueden asumir uno o más de los siguientes roles:

- Los *clientes* son los participantes que deciden llevar a cabo una computación conjunta, y que aportan valores de entrada privados y obtienen resultados privados. Como el objetivo principal de la MPC es obtener resultados correctos, suponemos que los clientes aportarán siempre entradas correctas.
- Los *operarios* llevan a cabo los cálculos requeridos por los clientes.
- Los *mensajeros* reciben mensajes de los clientes y los reenvían a otros mensajeros, o bien directamente a los operarios.
- Los *gestores de responsabilidad* son los encargados de gestionar la reputación de los otros nodos de la red. A cada nodo $P$ (sea cliente, operario o mensajero) se le asignan de manera pseudoaleatoria $M$ gestores de responsabilidad, evitando de esta manera que un nodo $P$ pueda seleccionar a sus gestores de responsabilidad.

En nuestro sistema no se pueden vincular de manera inequívoca los valores de entrada ni los resultados a sus correspondientes clientes, aunque es cierto que tanto entradas como resultados pueden ser visibles para otros participantes. Nuestra garantía de privacidad se basa en dicha invinculabilidad (*unlinkability*) más que en la confidencialidad. Por ello, *si las entradas o los resultados son tales que no es aceptable revelarlos ni tan sólo de forma no vinculable o tales que sus mismos valores o formatos los hacen vinculables a ciertos clientes, entonces nuestro sistema no debe usarse.*

Suponemos que los participantes son *racionales*: siempre y cuando se les proporcionen los incentivos necesarios, los participantes asumirán y cumplirán correctamente sus roles asignados en el protocolo, aunque se mostrarán interesados en conocer las entradas y los resultados de clientes específicos. Por tanto, hemos diseñado los protocolos de manera que los participantes racionales no tengan incentivos reales para desviarse en su ejecución. Sin embargo, puede haber una minoría de participantes maliciosos que se comporten de manera *irracional* y que se desvíen de los protocolos aunque ello les perjudique también a ellos.

### *III-B. Requisitos*

El objetivo principal de un cliente racional es obtener un resultado correcto de una computación conjunta manteniendo tanto sus valores de entrada como el resultado privados ante el resto de participantes. Para llevar a cabo este objetivo, los protocolos han sido diseñados de manera que sea imprescindible para el cliente tener una alta reputación. Por tanto, el incentivo que tiene un participante para cooperar en el funcionamiento del sistema es incrementar su reputación, para así poder convertirse en un cliente exitoso.

Para que la reputación sea efectiva, debe cumplir los siguientes requisitos:

- *Recompensa*. Si un operario lleva a cabo una computación correcta para un cliente, su reputación debe aumentar.
- *Castigo*. Si un operario lleva a cabo de manera incorrecta una computación para un cliente, su reputación debe disminuir. Igualmente, si un cliente se desvía de los protocolos, su reputación debe disminuir.
- *Recompensa probabilística*. Un participante que actúa de mensajero debe tener una cierta probabilidad de ver aumentada su reputación.
- *Utilidad*. Tener una reputación alta debe ser atractivo para todos los participantes. Concretamente, cuanto más alta sea su reputación, más fácil debe ser para un cliente conseguir un resultado correcto a la vez que preserva su privacidad.

### *III-C. Computación multiparte de próposito general en el modelo racional: protocolo principal*

Para facilitar la comprensión de nuestros protocolos, describimos brevemente como funcionaría una versión básica de los mismos. Unos cuantos nodos de la red P2P deciden participar en una computación conjunta y toman el papel de clientes. Cada cliente $P$ selecciona de manera secreta un operario $P_w$ (que no conoce la identidad de $P$). Todos los operarios reciben las entradas de todos los clientes a través de un canal coútil anónimo que garantiza que los valores no puedan ser vinculados a sus correspondientes clientes. Una vez que todas las entradas han llegado a todos los operarios, cada cliente $P$ envía a su operario $P_w$ la computación que ha de llevar a cabo y una clave que será usada por $P_w$ para enviar el resultado de vuelta a $P$ a través de un canal anónimo inverso. La idea de utilizar un canal anónimo en MPC fue por primera vez propuesta en [9]; la novedad en nuestro artículo reside en que el canal anónimo no depende de una entidad centralizada y es racionalmente sostenible.

En el modelo racional los participantes pueden desviarse de su comportamiento esperado si no se les incentiva correctamente. Por ejemplo, un operario podría devolver un resultado aleatorio sin llevar a cabo realmente una computación que podría ser costosa o un mensajero podría no reenviar los mensajes recibidos, con el objetivo en ambos casos de ahorrar sus propios recursos. Para superar este problema, añadimos dos mecanismos: (i) redundancia para detectar una computación incorrecta; (ii) reputación descentralizada para recompensar las computaciones y los reenvíos de mensaje correctos, y para

castigar el mal comportamiento. Las reputaciones de todos los participantes son *públicas*.

**Protocol 1:** MPC RACIONAL DE PROPÓSITO GENERAL

1 Los clientes $ID_1, \ldots, ID_m$ entre un conjunto de $n$ participantes (donde $m, n$ son públicos y $4 \leq m \leq n$) se conocen mutuamente y deciden llevar a cabo una computación conjunta $(O_1, O_2, \ldots, O_m) = C(I_1, I_2, \ldots, I_m)$, donde la entrada $I_i$ y el resultado $O_i$ han de ser privados para $ID_i$;
2 **for** $i = 1$ **to** $m$ **en paralelo do**
3 El cliente $ID_i$ usa un pseudónimo $P_i$;
4 $P_i$ "poda" el código $C$ y se queda con la parte $C_i$ que calcula $O_i$, es decir, $O_i = C_i(I_1, I_2, \ldots, I_m)$;
5 $P_i$ selecciona secretamente como operarios a $r$ nodos $P_{i_1}, \ldots, P_{i_r}$ escogidos aleatoriamente entre los $\kappa_i > r$ nodos con la reputación más cercana a $g_i$ y los recién llegados;
6 **for** $l = 1$ **to** $n$ **do**
7 $P_i$ ejecuta C-FWD-CH$(P_i, PK_l(I_i||nonce_i), PK_l(nil), P_l)$;
8 **for** $k = 1$ **to** $r$ **do**
9 $P_i$ ejecuta $O_{i,k} =$ C-FWD-CH$(P_i, PK_{i_k}(K_i), PK_{i_k}(C_i), P_{i_k})$;
10 **forall** *gestores de responsabilidad* $AM$ *de* $P_i$ **do**
11 **if** $P_i$ *no muestra a* $AM$ *el recibo de recompensa recibido del primer mensajero* **then**
12 $AM$ asigna una reputación local $\ell_{AM,i} = 0$ a $P_i$;
13 Sea $O_i$ el valor más frecuente en $\{O_{i,1}, \ldots, O_{i,r}\}$;
14 **for** $k = 1$ **to** $r$ **do**
15 **if** $O_{i,k} = O_i$ *AND* $O_i \neq nil$ **then**
16 $P_i$ asigna $\ell_{ii_k} = 1$;
17 **else**
18 $P_i$ asigna $\ell_{ii_k} = 0$;
19 Ejecutar el protocolo que actualiza las reputaciones globales públicas $g_i$ de cada nodo $P_i$.

El protocolo 1 es una versión detallada del protocolo explicado anteriormente extendida con redundancia (cada cliente escoge a $r$ operarios diferentes) y reputación descentralizada. Cada cliente $P$ envía sus entradas cifradas a los operarios $P_w$ a través del canal anónimo ejecutando el protocolo 2. Posteriormente les envía por el mismo canal la computación $C_i$ que han de llevar a cabo y la clave $K$ con la que cifrar el resultado, y queda a la espera. Tras recibir los resultados de todos los operarios, el cliente demuestra a sus gestores de responsabilidad que ha recompensado al primer mensajero, compara estos resultados entre sí y considera el mayoritario como el resultado final. Finalmente, recompensa o castiga a los operarios según el resultado recibido.

La idea general de los diferentes protocolos es que un nodo sólo interactuará con otros nodos de similar reputación ($g_i$). Por ejemplo, un operario sólo aceptará trabajar para un cliente cuya reputación sea superior o muy cercana a la suya, o un mensajero sólo aceptará reenviar un mensaje si éste proviene de otro nodo con una reputación superior o similar a la suya. De esta manera, los nodos con reputaciones bajas (producto de malas computaciones o mal comportamiento) acaban arrinconados sin posibilidad de interacción con otros nodos, y por tanto, sin posibilidad de recibir resultados correctos.

*Nota 1 (Los recién llegados):* Los nodos que entran en el sistema con posterioridad a su puesta en marcha lo hacen con una reputación global $g = 0$. Si no lleváramos a cabo ninguna acción especial, estos nodos no serían nunca seleccionados por los nodos honestos para ser sus operarios porque la diferencia entre las reputaciones de unos y otros es demasiado grande. Tan sólo podrían computar para los clientes maliciosos, que conseguirían computaciones correctas gracias a los recién llegados honestos que entran en el sistema. Por ello, cuando un nodo entra en el sistema, durante un número determinado de iteraciones se le permite ser escogido por los clientes como operario sea cual sea la diferencia entre reputaciones. De esta forma, el nodo podrá ganar reputación siempre y cuando ejecute las computaciones correctamente. No obstante, como contrapartida y para evitar que los nodos maliciosos encuentren ventajas en salir del sistema y volver a entrar con un pseudónimo diferente (blanqueo), un recién llegado no podrá ejercer él mismo como cliente y no obtendrá ningún tipo de resultado.

*Nota 2 (El parámetro $\kappa_i$):* El parámetro $\kappa_i$ es secreto para cada cliente $P_i$. Si $\kappa_i \gg r$, entonces los operarios son escogidos entre un conjunto de nodos más grande, lo que hace más difícil para los operarios saber qué cliente les ha escogido. Sin embargo, por otro lado, escoger un valor de $\kappa_i$ demasiado alto puede ser arriesgado para $P_i$ porque podría significar que los operarios escogidos tienen una reputación demasiado elevada (y rechazarán trabajar para $P_i$) o demasiado baja (lo que significaría que no son completamente fiables).

*Nota 3 (Compartir operarios):* Una manera de reducir la computación y la comunicación del protocolo 1 al aplicar redundancia sería que los clientes compartiesen operarios. Por ejemplo, si cada cliente propusiera un operario, un grupo de $m$ operarios estaría disponible para todos los clientes; para conseguir un nivel de redundancia similar, el protocolo 1 requeriría escoger a $m^2$ operarios en total. Sin embargo, compartir operarios tiene varias desventajas. Por un lado, todos los operarios llevarían a cabo las mismas computaciones $C$ para todos los clientes. Por otro lado, los clientes se verían forzados a confiar en unos operarios que han sido seleccionados por otros clientes. Esto es particularmente desaconsejable si las reputaciones de los clientes son muy heterogéneas: los clientes con alta reputación pueden escoger a operarios con reputaciones más altas (y por tanto, más fiables) que los propuestos por clientes con reputaciones más modestas.

*Nota 4 (Enviar a todos):* En la línea 7 del protocolo 1 se puede apreciar que los valores de entrada son enviados a todos los participantes de la red. Esto es necesario para que todos los operarios reciban los valores de entrada de todos los clientes, sin obligar a los mismos a compartir entre ellos los operarios que ha escogido cada uno. De hecho, un nodo no sabe que es operario hasta que recibe el mensaje con la computación que ha de llevar a cabo (línea 9).

*III-D. Protocolo para el envío de mensajes a través del canal coútil anónimo*

El protocolo 2 corresponde al funcionamiento del canal coútil anónimo. Si un nodo recibe un mensaje, lo primero que hace es comprobar si se lo ha enviado un nodo con una reputación superior o similar a la suya (líneas 21, si el nodo es un mensajero, y 28, si el nodo es el operario destinatario del mensaje). Si la reputación del remitente era demasiado baja, el nodo descarta el mensaje. Si en cambio acepta el mensaje y no es el operario a quien iba dirigido, debe decidir si enviarlo directamente al mismo o reenviarlo a otro mensajero intermedio (línea 19). Si decide reenviarlo, lo hará a un nodo con una reputación superior o similar a la suya (ver nota 6).

Los operarios encargados de las computaciones irán acumulando las entradas de los diferentes clientes (línea 6), y ejecutarán la computación requerida cuando hayan recibido $m$ entradas y $C_i$ (línea 15). El resultado, cifrado con la clave $K$ que les había enviado el cliente previamente, lo enviarán de vuelta al cliente a través del protocolo 3.

*Nota 5 (El parámetro de flexibilidad $\delta$):* En el protocolo 2, un nodo $P_j$ no descarta un mensaje recibido de un nodo $P_i$, siempre y cuando la reputación de $P_i$ ($g_i$) sea al menos $g_j - \delta$, siendo $\delta$ una pequeña cantidad de reputación que introduce flexibilidad en la interacción. Un valor $\delta$ muy elevado no sería aceptable desde un punto de vista racional: los nodos de alta reputación tienen poco que ganar aceptando mensajes o computando para nodos con una reputación muy inferior a la suya (p.e. si estos nodos de baja reputación son clientes, podrían no recompensarles posteriormente por la computación efectuada).

*Nota 6 (La función* SELECT*):* En el protocolo 2, la función $P_t = \text{SELECT}(g_s, g_d)$ es ejecutada por un nodo $P_s$ para seleccionar a otro nodo $P_t$ como mensajero, con el fin de que reenvíe el mensaje hacia el operario $P_d$. Hay diversas maneras de llevar a cabo esta selección. Sin embargo, la opción racional para $P_s$ es escoger a un mensajero $P_t$ con una reputación suficiente para que $P_d$ no descarte el mensaje si $P_t$ se lo envía directamente. Por tanto, si la reputación de $P_s$ es $g_s \geq g_d - \delta$, $P_s$ puede seleccionar aleatoriamente cualquier nodo con una reputación entre $[g_d - \delta, g_s + \delta]$: cualquier mensajero $P_t$ con una reputación de al menos $g_d - \delta$ no correrá el riesgo de ver su mensaje descartado por el operario $P_d$. En cambio, otros mensajeros $P_t$ con reputaciones superiores a $g_s + \delta$ descartarían directamente los mensajes de $P_s$. Por otro lado, si $g_s < g_d - \delta$, $P_s$ ha de escoger el nodo con la máxima reputación posible que no exceda de $g_s + \delta$, porque cualquier otro nodo con un reputación superior descartaría directamente su mensaje.

*III-E. Protocolo para el envío de resultados a través del canal inverso*

Una vez se ha llevado a cabo una computación, un operario envía el resultado de vuelta a través del protocolo 3 (C-REV-CH), ya que, al no conocer la identidad del cliente, no puede utilizar el canal anónimo. Se recorre el camino inverso hasta que el resultado llega al cliente $P_{prev}$ (que es quien puede descifrar el resultado al conocer la clave $K$) y éste recompensa al primer mensajero $P_s$, que a su vez le entrega al cliente el recibo que ha de mostrar a sus gestores de responsabilidad en la línea 11 del protocolo 1.

**Protocol 2:** C-FWD-CH($P_s, msg, Ecomp, P_d$)

1 Parámetro $p \in [0, 1]$;
2 **if** $P_s = P_d$ **then**
3  $P_d$ descifra $comp = SK_d(Ecomp)$;
4  **if** $comp = nil$ **then**
5  $P_d$ descifra $I||nonce = SK_d(msg)$;
6  **if** $P_d$ *no ha recibido previamente* $nonce$ **then** $P_d$ añade $I$ a $Ilist$;
7  **else**
8  **if** $comp =$ *"refuse"* **then**
9  $P_d$ asigna $out := nil$;
10  **else**
11  $P_d$ espera hasta que $Ilist$ contiene $m$ valores de entrada diferentes;
12  $P_d$ computa $out := comp(Ilist)$;
13  $P_d$ vacía $Ilist$;
14  $P_d$ recupera $K = SK_d(msg)$;
15  $P_d$ ejecuta C-REV-CH($P_d, E_K(out), Ecomp, P_{prev}$);
16 **else**
17  **if** $P_s$ *es el origen de* $msg$ **then** $p_{forward} = 1$ **else** $p_{forward} = p$;
18  $P_s$ computa $decision =$ Bernoulli($p_{forward}$);
19  **if** *decision* $= 1$ **then**
20  $P_s$ envía $(msg, Ecomp)$ a $P_t = \text{SELECT}(g_s, g_d)$;
21  **if** *la reputación de* $P_s$ *es al menos* $g_t - \delta$ **then**
22  $P_t$ ejecuta C-FWD-CH($P_t, msg, Ecomp, P_d$);
23  **else**
24  $P_t$ descarta $(msg, Ecomp)$;
25  **else**
26  $P_s$ envía directamente $(msg, Ecomp)$ a $P_d$;
27  $P_d$ descifra $comp = SK_d(Ecomp)$;
28  **if** $comp \neq nil$ *y la reputación de* $P_s$ *es menor que* $g_d - \delta$ **then**
29  $P_d$ asigna $Ecomp := PK_d$("refuse");
30  $P_d$ ejecuta C-FWD-CH($P_d, msg, Ecomp, P_d$);

**Protocol 3:** C-REV-CH($P_s, E_K(out), Ecomp, P_{prev}$)

1 $P_s$ envía $(E_K(out), Ecomp)$ a $P_{prev}$;
2 **if** $P_{prev}$ *conoce la clave* $K$ **then**
3  $P_{prev}$ descifra $out$;
4  $P_{prev}$ envía $S_{prev}$("$P_{prev}$ recompensa a $P_s$") a $P_s$;
5  $P_s$ reenvía $S_{prev}$("$P_{prev}$ recompensa a $P_s$") a los AM de $P_{prev}$;
6  Los AM de $P_{prev}$ ponen $\ell_{prev,s} = 1$;
7  $P_s$ devuelve un recibo firmado $S_s$("$P_s$ acusa recibo de recompensa de $P_{prev}$") a $P_{prev}$;
8 **else**
9  Sea $P_{prev2}$ el nodo del cual $P_{prev}$ recibió el mensaje con la computación $Ecomp$;
10  $P_{prev}$ ejecuta C-REV-CH($P_{prev}, E_K(out), Ecomp, P_{prev2}$).

*Nota 7 (Recompensa únicamente para el primer mensajero):* Como hemos comentado, en el protocolo C-REV-CH solamente se recompensa al primer mensajero. El protocolo que actualiza las reputaciones globales requiere que las reputaciones locales sean normalizadas previamente (ver [8]). Por tanto, si todos los mensajeros fueran recompensados por el cliente, el incremento de reputación de cada uno de los mensajeros sería mucho menor. Un mensajero que recibiera un mensaje $msg$ lo enviaría siempre directamente a su destino final con el objetivo de minimizar el número de mensajeros para $msg$. Por ello, en la práctica, solo existiría un mensajero, que automáticamente sabría que el nodo que le ha enviado $msg$ es el cliente, lo que rompería el anonimato del canal.

## IV. Análisis de co-utilidad

En la presente sección argumentamos por qué el sistema formado por los protocolos presentados anteriormente es coútil, es decir, por qué será ejecutado sin desviaciones por participantes racionales:

- El objetivo de los clientes es ejecutar computaciones conjuntas y obtener sus correspondientes resultados correctos, a la vez que mantienen privados tanto sus valores de entrada como sus resultados. Por esta razón, podemos suponer que los clientes llevarán a cabo correctamente sus tareas en los protocolos 1, 2 y 3. Si un cliente no recompensara al primer mensajero, sería penalizado con una reducción de su reputación, lo que le haría más difícil obtener resultados correctos en sucesivas computaciones. También tiene sentido que los clientes escojan a un recién llegado como uno de sus $r$ operarios redundantes: cuantos más nodos honestos participen, mayor resistencia a ataques tendrá el sistema. La redundancia de los operarios minimiza el riesgo en caso de que el recién llegado sea malicioso.
- Los mensajeros son esenciales para la ejecución del canal anónimo en los protocolos C-FWD-CH y C-REV-CH. Su incentivo es ser recompensados en el protocolo 3 como primer mensajero (*a priori* no saben si son o no son el primer mensajero) con un incremento de su reputación.
- De los operarios se espera que lleven a cabo correctamente la computación requerida en C-FWD-CH. Posteriormente, en C-REV-CH se espera que inicien el recorrido inverso del trayecto para devolver el resultado. Su incentivo es ser recompensados en el protocolo 1 si el resultado que devuelven es el mayoritario entre los devueltos por el conjunto de operarios de ese cliente.
- Los gestores de responsabilidad tiene un rol importante en los protocolos 1 y 3, y en el protocolo de cálculo de las reputaciones globales. En la descripción del modelo de seguridad (Sección III-A) suponíamos que los nodos era racionales, aunque no excluíamos la posibilidad de ataques racionales. Dado que los nodos interactuarán en sucesivas iteraciones, el interés racional de los gestores de responsabilidad será favorecer a los nodos que han llevado a cabo computaciones correctas, ya que ellos mismos pueden ser clientes en futuras rondas de computación. Por el otro lado, el hecho de que los $M$ gestores de responsabilidad de un nodo sean escogidos de manera pseudoaleatoria dificulta el conflicto de interés y facilita la gestión honesta de la reputación de los nodos.

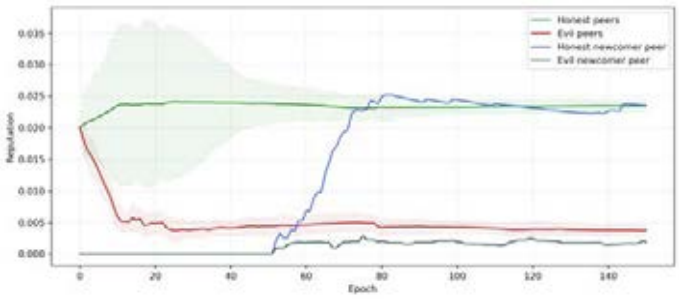



Figura 1. Evolución de las reputaciones de las diferentes clases de nodos en un escenario con el 20 % de nodos maliciosos

## V. Resultados experimentales

En esta sección mostramos los resultados de los experimentos llevados a cabo con el sistema coútil propuesto. Consideramos *nodos honestos* (los que llevan a cabo todos los pasos de los protocolos correctamente) y *nodos maliciosos* (los que no).

*Resultado esperado.* Si nuestro sistema está bien diseñado, los nodos honestos deberían obtener una mayor proporción de resultados correctos que los nodos maliciosos. Además, la reputación de cada nodo debería estar muy correlacionada con su comportamiento y con la proporción de computaciones correctas que obtiene. La razón es que es probable que los nodos que llevan a cabo computaciones incorrectas como operarios acaben teniendo una baja reputación, y para un nodo con baja reputación es complicado encontrar (como cliente) a operarios o mensajeros fiables. Asimismo, un nodo honesto recién llegado debería tener a medio plazo una proporción de computaciones correctas parecida a la del resto de nodos honestos. En cambio, si el recién llegado se comportara de manera maliciosa, debería mantener una proporción baja de computaciones correctas, como el resto de nodos maliciosos de la red. De esta manera, un nodo malicioso con baja reputación no conseguiría ninguna ventaja si decide dejar el sistema y volver a entrar con un pseudónimo diferente.

*Configuración experimental.* Construimos una red P2P con $n = 50$ nodos y dejamos que evolucionase durante 150 iteraciones; en cada iteración se llevó a cabo una computación conjunta que implicó a $n = 10$ clientes escogidos aleatoriamente entre los 50 nodos. Cada cliente aportó un número como valor de entrada privado para la computación, que consistía simplemente en devolver la posición de ese valor en la ordenación de todos los valores de los clientes. En este caso, un nodo malicioso retornaba siempre un valor aleatorio en lugar del resultado real de la computación (con lo que mantenía un 10 % de posibilidades de acertar el resultado correcto). Definimos $r = 3$ (tres operarios redundantes para cada cliente), otorgamos una reputación inicial de $g = 1/n = 0{,}02$ a cada cliente, fijamos inicialmente $\delta = g * 0{,}75 = 0{,}015$ y un período para considerar a un nodo como recién llegado de 25 iteraciones.

En el experimento, el 20 % de los nodos eran nodos maliciosos y además introdujimos 2 nodos recién llegados

al sistema en la iteración 50, uno honesto y otro malicioso. La figura 1 muestra la evolución de la reputación de cada clase de nodos durante las 150 iteraciones. Las líneas indican la reputación media de los nodos de un tipo determinado, mientras que la región sombreada muestra la desviación estándar. Podemos observar como ya en las primeras iteraciones, mientras el comportamiento de los nodos está siendo modelado, la reputación de los nodos maliciosos va decreciendo rápidamente. Alrededor de la novena iteración ya hay una clara diferencia entre las reputaciones de los nodos honestos y las de los maliciosos. A partir de la iteración 50 aparece también la evolución tanto del recién llegado honesto (azul) como del malicioso (gris). Como durante las primeras 25 iteraciones tras su aparición, el resto de nodos les pueden escoger como operarios a pesar de no tener la reputación adecuada, el comportamiento de ambos nodos es modelado convenientemente. La reputación del recién llegado honesto va aumentando hasta situarse al nivel del resto de nodos honestos, mientras que la del recién llegado malicioso se queda al nivel del resto de nodos maliciosos de la red. Recordemos que, durante esta fase, los recién llegados sólo pueden ejercer de operarios y mensajeros, no de clientes, por lo que un nodo malicioso no obtendría ninguna ventaja si decide salir y entrar de nuevo al sistema con otro pseudónimo. Evitamos de este modo ataques de blanqueo.

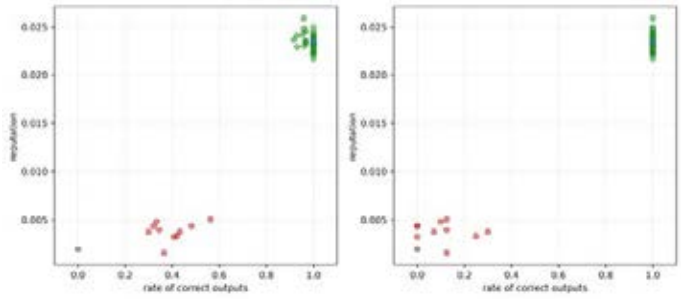



Figura 2. Proporción de computaciones correctas en función de la reputación de los diferentes nodos, con un 20 % de nodos maliciosos. A la izquierda, en las 150 iteraciones; a la derecha, en las últimas 50.

En el gráfico izquierdo de la figura 2 vemos la proporción de computaciones correctas que han obtenido tanto los nodos honestos (en verde), como los maliciosos (rojo) y los recién llegados (en azul el honesto y en gris el malicioso) desde la primera iteración, en un experimento con la configuración explicada anteriormente. Podemos observar que la gran mayoría de las computaciones que han solicitado los nodos honestos han sido correctas. En cambio, los nodos maliciosos han recibido aproximadamente la mitad de sus computaciones de manera correcta, lo que cumple con lo esperado. De hecho, como se puede ver en el gráfico derecho de la figura 2, la reputación es mucho más decisiva en las últimas 50 iteraciones. En estas iteraciones, todas las computaciones solicitadas por los clientes honestos (con alta reputación) fueron correctas, lo que contrasta con la proporción de 10 % de computaciones correctas entre las solicitadas por los nodos maliciosos (con baja reputación). Por tanto, en cuanto el sistema se estabiliza, los nodos honestos obtienen siempre resultados correctos.

## VI. Conclusiones

En este artículo hemos presentado un sistema P2P para computación multiparte que, con un uso muy moderado de la criptografía, asegura que no podrán vincularse inequívocamente los valores de entrada y los resultados con las partes correspondientes. La ventaja principal de este sistema es que es de propósito general y no hace uso de circuitos, por lo que funciona para cualquier tipo de cálculo expresado en código de alto nivel (sin restricciones de iteraciones o llamadas recursivas).

Como trabajo futuro prevemos ampliar el análisis de coutilidad y llevar a cabo más experimentos que nos permitan afinar la configuración inicial de los parámetros del sistema. La investigación futura se centrará en intentar reducir el coste de comunicaciones.

## Agradecimientos

Agradecemos las subvenciones de los organismos siguientes: Comisión Europea (proyectos H2020-871042 “SoBigData++” y H2020-101006879 “MobiDataLab”), Generalitat de Catalunya (Premio ICREA Acadèmia al segundo autor) y MCIN/AEI /10.13039/501100011033 /FEDER, UE (proyectos RTI2018-095094-B-C21 y PID2021-123637NB-I00). Los autores pertenecen a la Cátedra UNESCO de Privacidad de datos, pero las opiniones en este artículo son suyas y no necesariamente compartidas por UNESCO o los organismos financiadores.

## Referencias

[1] A. C. Yao: “Protocols for secure computations”, en *23rd Annual Symposium on Foundations of Computer Science - SFCS 1982*, pp. 160-164. IEEE, 1982.

[2] O. Goldreich, S. Micali, A. Wigderson: “How to play any mental game or a completeness theorem for protocols with honest majority”, en *19th Annual ACM Symposium on the Theory of Computing - STOC 1987*, pp. 218-229. ACM, 1987.

[3] M. Ben-Or, S. Goldwasser, A. Wigderson: “Completeness theorems for non-cryptographic fault-tolerant distributed computation”, en *20th Annual ACM Symposium on the Theory of Computing - STOC 1988*, pp. 1-10. ACM, 1988.

[4] D. Chaum, C. Crépeau, I. Damgård: “Multiparty unconditionally secure protocols”, en *20th Annual ACM Symposium on the Theory of Computing - STOC 1988*, pp. 11-19. ACM, 1988.

[5] M. Hastings, B. Hemenway, D. Noble, S. Zdancewic: “SoK: General purpose compilers for secure multiparty computation”, en *IEEE Symposium on Security and Privacy - SP 2019*, pp. 1220-1237. IEEE, 2019.

[6] J. Domingo-Ferrer, S. Martínez, D. Sánchez, J. Soria-Comas: “Co-utility: self-enforcing protocols for the mutual benefit of participants”, en *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 148-158. 2017.

[7] J. Domingo-Ferrer, O. Farràs, S. Martínez, D. Sánchez and J. Soria-Comas: “Self-enforcing protocols via co-utile reputation management”, en *Information Sciences*, vol. 367-368, pp. 159-175. 2016.

[8] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina. “The EigenTrust algorithm for reputation management in P2P networks”, en *12th International Conference on World Wide Web*, pp. 640-651. ACM, 2003.

[9] Y. Ishai, E. Kushilevitz, R. Ostrovsky, A. Sahai: “Cryptography from anonymity”, en *47th Annual IEEE Symposium on Foundations of Computer Science - FOCS 2006*, pp. 239-248. IEEE, 2006.

# Two Decoding Algorithms in Group Codes

Consuelo Martínez López
Universidad de Oviedo
cmartinez@uniovi.es

Fabián Ricardo Molina Gómez
Universidad de Oviedo
molinafabian@uniovi.es

***Abstract*—Error-correcting codes based cryptography is postulated as one of the most promising post-quantum cryptographic methods. The first proposal in this direction was made by McEliece. Several alternatives to a McEliece-type cryptographic scheme have been proposed, but without significant success. We think that group codes, that is, linear codes that can be identified with (two-sided) ideals of a group algebra $\mathbb{K}G$, could be used with this aim. Here, assuming that $\mathbb{K}G$ is semisimple, we use its decomposition as the sum of two ideals (one of them the group code) generated by central idempotents to design two decoding algorithms.**

***Index Terms*—Linear codes, Group, Group algebra, Semisimple algebra, Decoding.**

## I. Introduction

Coding Theory has applications in Cryptography, since it is possible to design cryptosystems based on error-correcting codes that, in principle, are resistant to quantum computing. In fact, error-correcting code-based cryptography is postulated as one of the most promising post-quantum computing cryptographic methods. In this type of schemes, a "good" error correcting code is used as public key of the scheme.

The first system in this line was proposed in [1] by R. J. McEliece in 1978. His proposal uses Goppa codes, hidden under a given linear code, as public key. To decrypt the message, one needs to know how to decode using this linear code. McEliece's proposal remains unbroken so far. Although the encryption and decryption processes are simple, the main problem of this scheme is the large size of the public keys.

Successive proposals to improve the original one suggested the replacement of Goppa codes by more structured codes ([2], [3], [4]). However all of them have been broken ([5], [6]). Another alternative to reduce the size of McEliece cryptosystem is the use of codes having a non-trivial permutation automorphism group ([7], [8]). Nevertheless, the knowledge of the permutation automorphism group of a code allows some attacks by reducing the degrees and the number of variables in the algebraic system to be solved. So, these proposals have also been totally or partially broken in [9].

In this paper we consider group codes. These codes are linked to (two-sided) ideals of the group algebra $\mathbb{K}G$, where $G$ is a finite group of order $n$ and $n$ is the length of the code. Assuming a fixed order for elements in $G$, we fix $\mathfrak{B} = G$ the $\mathbb{K}-$basis of $\mathbb{K}G$. Hence, we can also write the elements of $\mathbb{K}G$ as $n-$sequences of elements of the field $\mathbb{K}$. Let'us remember that a $(n, k, d)-$linear code $\mathfrak{C}$ is a $G-$code, if there is an isomorphism of $\mathbb{K}-$vector spaces $\phi : \mathbb{K}^n \to \mathbb{K}G$, satisfying that $\phi(\mathfrak{C})$ is a two-sided ideal of $\mathbb{K}G$. A linear code $\mathfrak{C}$ is called (abelian) group code, if there exists a (abelian) group $G$ such that $\mathfrak{C}$ is an $G-$code.

Linear codes that are (abelian) group codes were characterized by Bernal et al. ([10]) in terms of their permutation automorphism group. Notice that a group code can be realized as $G-$code for different groups and possibly some of then are abelian and some are non-abelian groups. In [11] the question of the existence of non-abelian group codes, that is, group codes that can not be realized as abelian group codes, was addressed. Authors proved that the existence of a non-abelian $G-$code is possible only if the length of the code is at least 24 and proved that it happens for $G = \mathcal{S}_4$ in a semisimple case. Later, in [12], the existence of such codes in the non-semisimple case was also addressed, constructing non-abelian $\mathcal{S}_4-$codes over $\mathbb{K} = \mathbb{F}_2$ and $\mathbb{K} = \mathbb{F}_3$. In the same paper, authors proved the existence of a non-abelian group code that is optimal in the sense that it has length 24, dimension 6 and minimal distance 10. Let us notice that 10 is the maximal distance of a binary linear code of length 24 and dimension 6 and it can not be reached in any abelian group code. Finally in [13], authors proved that there are $G-$codes over $\mathbb{F}_p$ that are non-abelian group codes for every prime $p \geq 3$. In this way they can conclude that there are non-abelian group codes of length 24 for every finite field.

The fact that we can use abelian and non-abelian groups to construct group codes and the difficulty to distinguish group codes among linear codes are, a priori, good properties to use these codes in the design of a McEliece type cryptosystem. But it is essential to have an efficient decoding algorithm. A permutation decoding algorithm for some abelian group codes in the semisimple case was partially proposed in [14]. Then in [15], authors suggest an algorithm that can be seen as a variant of the classical syndrome decoding algorithm for one-error correction. In this work we propose two general decoding algorithms for any group codes in the semisimple case.

*Remark*. The decoding algorithms explained here do not work neither for left (or right) group codes nor for group codes in the non-semisimple case. The reason is that the main property that we use is that for every ideal $I$ of $\mathbb{K}G$ (identified

with a $G-$code over $\mathbb{K}$), there exists an ideal $I^+$ such that $\mathbb{K}G = I \oplus I^+$. *Thus, $x \in I$ if and only if $xy = 0$ for all $y \in I^+$*. This fact is not fulfilled for left (right) ideals and does not work in the non-semisimple case.

## II. Preliminaries

From now, $G$ will denote a finite group of order $n$ and $\mathbb{K} = \mathbb{F}_q$ will be a finite field of characteristic $p > 0$ that does not divide to $n$. This is equivalent to say that the group algebra $\mathbb{K}G$ is semisimple (Maschke's Theorem). Thus, $\mathbb{K}G$ can be written as the sum of $s$ minimal two-sided ideals (see [16]). That is,

$$\mathbb{K}G = \langle e_1 \rangle \oplus \cdots \oplus \langle e_s \rangle. \tag{1}$$

The minimal two-sided ideals $\langle e_i \rangle$ are generated by primitive central idempotents $e_i$ and are called simple components of $\mathbb{K}G$. Furthermore, every two-sided ideal of $\mathbb{K}G$ is generated by a central idempotent and is a direct sum of some simple components of $\mathbb{K}G$.

So we will suppose from now on that $\mathfrak{C} = \langle e_0 \rangle$ where $e_0$ is the sum of some central primitive idempotents. Hence, $\mathfrak{C}$ is a group code. Let us denote $n$, $k$ and $d$, respectively, the length, the dimension and the minimal distance of $\mathfrak{C}$. Also we denote by $\mathfrak{C}^+ = \langle e_0^+ \rangle$ the direct summand of $\mathfrak{C}$ where $e_0^+ = 1 - e_0$ and whose dimension is $n-k$. A codeword $\underline{c} \in \mathfrak{C}$ has the form $\underline{c} = ze_0$, for some $z \in \mathbb{K}G$ and therefore, an element $\underline{c} \in \mathbb{K}G$ is a codeword, if and only if, $\underline{c}e_0^+ = 0$. This is important because it allows to obtain the control conditions. If $\underline{r}$ is the received word, then $\underline{r} = \underline{c} + \underline{e}$ where $\underline{c}$ is the sent codeword and $\underline{e}$ is the error produced during the transmission process. The syndrome of $\underline{r}$ is defined as $S = \underline{r}e_0^+$. Consequently,

$$S = (\underline{c} + \underline{e})e_0^+ = \underline{e}e_0^+ \tag{2}$$

and therefore, decoding by minimal distance is equivalent to find a word $X \in \mathbb{K}G$ that has weight $w$ less than or equal to the correcting capacity $t$ and that is solution of $Xe_0^+ = S$. Notice that there are no errors ($\underline{e} = 0$) if and only if $S = 0$.

**Theorem 1.** Let $\mathfrak{C}$ be a group code that corrects up to $t$ errors and $\underline{r}$ a received word with syndrome $S = \underline{r}e_0^+$. If there exists one element that of weight $w \leq t$ and that is a solution of the system $Xe_0^+ = S$, then it is unique.

*Proof:* Note that, if $\underline{e}, \underline{e}' \in \mathbb{K}G$ are distinct solutions of the mentioned system having weights $w, w' \leq t$, respectively, then $\underline{e}e_0^+ = \underline{e}'e_0^+ = S$. So $(\underline{e} - \underline{e}')e_0^+ = 0$ and consequently, $\underline{e} - \underline{e}' \neq 0$ is an element of $\mathfrak{C}$. This is a contradiction since the minimal weight of $\mathfrak{C}$ is $d \geq 2t + 1$ and $\underline{e} - \underline{e}'$ has weight less than or equal to $w + w' \leq 2t$. ∎

## III. Framework

The first algorithm that we present is inspired in the well known syndrome decoding algorithm for linear codes. So to start decoding, previously we compute the products $g_i e_0^+$ for all $i \in \{1, \ldots, n\}$ and so we denote $C_{g_i} \in M_{n\times 1}(\mathbb{K})$ the column vector of coefficients of $g_i e_0^+$ with respect to the fixed basis on $G$ (This precalculation has a complexity of the order $\mathcal{O}(n^2)$).

**Theorem 2.** If $b < d$ and $g_{i_1}, \ldots, g_{i_b}$ are distinct elements of $G$, then the matrix

$$\mathcal{C}(g_{i_1}, \ldots, g_{i_b}) := \begin{pmatrix} C_{g_{i_1}} & \cdots & C_{g_{i_b}} \end{pmatrix} \in M_{n\times b}(\mathbb{K}),$$

has rank $b$.

*Proof:* If $\mathcal{C}(g_{i_1}, \ldots, g_{i_b})$ has rank less than $b$, then there are elements $\nu_1, \ldots, \nu_b \in \mathbb{K}$ not all of them equal to zero such that $\nu_1 C_{g_{i_1}} + \cdots + \nu_b C_{g_{i_b}} = 0$. Hence, $(\nu_1 g_{i_1} + \cdots + \nu_b g_{i_b})e_0^+ = 0$ and therefore $x = \nu_1 g_{i_1} + \cdots + \nu_b g_{i_b} \in \mathfrak{C}$. This fact is a contradiction since $x \neq 0$ has weight less than or equal to $b$ and the minimal distance of $\mathfrak{C}$ is $d$. ∎

When a word $\underline{r}$ is received, the algorithm compute the syndrome $S = \underline{r}e_0^+$ and we define the column vector $B$ such that $B^T = S$. The idea of this decoding algorithm is to find $\underline{e} = \alpha_1 g_{i_1} + \cdots + \alpha_w g_{i_w}$ with weight $w \leq t$ (that is, $g_{i_1}, \ldots, g_{i_w}$ are distinct elements of $G$ and $\alpha_1, \ldots, \alpha_w$ are non zero elements of $\mathbb{K}$) such that $S = \underline{e}e_0^+$.

Note that, we don't know, a priori, neither the number of errors nor their positions. To work easily, the algorithm looks for a subset of $t$ elements ($t-$set) of $G$ that contains error positions. So it looks for an element $\underline{u} = \beta_1 g_{i_1} + \cdots + \beta_t g_{i_t}$ such that $\beta_i = \alpha_i$ if $g_{i_j}$ is an error position and $\beta_i = 0$ otherwise.

That is, the algorithm looks for an element $\underline{u}$ such that its coordinates are a solution of the (linear) system

$$X_1 C_{g_{i_1}} + \cdots + X_t C_{g_{i_t}} = B. \tag{3}$$

By theorem Theorem 1, we know that any system like this one either has a unique solution or does not have solutions. So we are looking for a $t-$set $\{g_{i_1}, \ldots, g_{i_t}\}$ such that the system (3) is determinate compatible. This is equivalent to the matrix $\mathcal{C}(g_{i_1}, \ldots, g_{i_t})$ and the extended matrix

$$\mathcal{M}(g_{i_1}, \ldots, g_{i_t}) := \left( \begin{array}{ccc|c} C_{g_{i_1}} & \cdots & C_{g_{i_t}} & B \end{array} \right),$$

have both rank equal to $t$.

*Remark.* Notice that the number of elements $\alpha_1, \ldots, \alpha_w$ that are non zero in the unique solution of the system (3) indicates the number of errors produced during the transmission and $\alpha_1, \ldots, \alpha_w$ give the magnitudes of the errors.

*Syndrome Decoding Algorithm (Algorithm 1)*

*Step 1:* Compute all syndromes. If $S = 0$, then there are no errors and the algorithm ends. Otherwise, it continues to

*Step 2:* Select randomly $\{g_{i_1}, \ldots, g_{i_t}\}$ a $t-$set of $G$. Consider $\mathcal{M}(g_{i_1}, \ldots, g_{i_t})$ and compute its rank.

a. If the rank is $t$, find the unique solution $\alpha_1, \ldots, \alpha_t \in \mathbb{K}$ of the system (3). If $\alpha_{j_1}, \ldots, \alpha_{j_w}$ are the non zero elements in the above solution, then the error is $\underline{e} = \alpha_{j_1} g_{i_{j_1}} + \cdots + \alpha_{j_w} g_{i_{j_w}}$ and the algorithm ends.
b. Otherwise, the $t-$set is discarded and another $t-$set of $G$ is considered randomly and Step 2 is repeated with it.

The algorithm finishes when a $t-$set $\{g_{i_1}, \ldots, g_{i_t}\}$ satisfies

$$\text{Rank}(\mathcal{M}(g_{i_1}, \ldots, g_{i_t})) = t, \tag{4}$$

what allows us to find the error or when all $t-$sets of $G$ have been checked and none satisfies (4). In the last case, we conclude that the number of errors is greater than $t$ and the group code can not correct them for $\mathfrak{r}$.

**Observation 1.** It is important to mention that, if the error has weight $w = t$, then the $t-$set of $G$ is unique and when the error has weight $w < t$, then there are many $t-$sets of $G$ that contain the $w$ error positions and all these $t-$sets generate the same solution for the error. For this reason, the algorithm will end when it finds the first $t-$set of $G$ that allows to find the error. That is, when $w \leq t$, we get a unique expression for the error. Also, the syndrome decoding algorithm has a complexity of the order

$$\mathcal{O}\left(n^3 \times \binom{n}{t}\right).$$

**Example 1.** Let $G = \mathcal{S}_4$ and $\mathbb{K} = \mathbb{F}_5$. We fix

$$\begin{array}{ll}
g_1 = (1), & g_2 = (3,4), \\
g_3 = (2,3), & g_4 = (2,3,4), \\
g_5 = (2,4,3), & g_6 = (2,4), \\
g_7 = (1,2), & g_8 = (1,2)(3,4), \\
g_9 = (1,2,3), & g_{10} = (1,2,3,4), \\
g_{11} = (1,2,4,3), & g_{12} = (1,2,4), \\
g_{13} = (1,3,2), & g_{14} = (1,3,4,2), \\
g_{15} = (1,3), & g_{16} = (1,3,4), \\
g_{17} = (1,3)(2,4), & g_{18} = (1,3,2,4), \\
g_{19} = (1,4,3,2), & g_{20} = (1,4,2), \\
g_{21} = (1,4,3), & g_{22} = (1,4), \\
g_{23} = (1,4,2,3), & g_{24} = (1,4)(2,3).
\end{array}$$

Since $\mathbb{F}_5\mathcal{S}_4$ is direct sum of five simple components generated by the primitive central idempotents:

$$\begin{aligned}
e_1 &= (44444444444444444444444), \\
e_2 &= (41144114411441144144114), \\
e_3 &= (10022001200200210022001), \\
e_4 &= (13300333022002303220032 3), \\
e_5 &= (12200223033003203330023 3).
\end{aligned}$$

with dimensions 1,1,4,9 and 9, respectively, let us suppose $\mathfrak{C} = \langle e_5 \rangle$. Then $e_0^+ = e_1 + e_2 + e_3 + e_4$, $n = 24$, $k = 9$, $d = 8$ and $t = 3$.

If $\underline{r} = (32001204041114214023003 2)$, then decoding $\mathfrak{r}$:

*Step 1:* The syndrome is $S = (10240321003400000020403 1)$. So, go to

*Step 2:* If we choose the $3-$set $\{g_1, g_2, g_3\}$, then we see that the matrix $\mathcal{M}(g_1, g_2, g_3)$ has rank 4 and so we discard it. Considering the $3-$set $\{g_4, g_{13}, g_{17}\}$, we have that the matrix

$$\mathcal{M}(g_4, g_{13}, g_{17}) = \left(\begin{array}{ccc|c}
0 & 0 & 2 & 1 \\
3 & 2 & 2 & 0 \\
3 & 3 & 2 & 2 \\
0 & 2 & 0 & 4 \\
0 & 0 & 0 & 0 \\
3 & 2 & 3 & 3 \\
2 & 3 & 2 & 2 \\
0 & 0 & 2 & 1 \\
0 & 0 & 0 & 0 \\
3 & 2 & 2 & 0 \\
2 & 2 & 3 & 3 \\
2 & 2 & 0 & 4 \\
2 & 0 & 0 & 0 \\
3 & 3 & 3 & 0 \\
2 & 3 & 3 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
2 & 3 & 3 & 0 \\
2 & 3 & 2 & 2 \\
0 & 0 & 0 & 0 \\
2 & 2 & 0 & 4 \\
2 & 2 & 2 & 0 \\
3 & 2 & 3 & 3 \\
0 & 0 & 2 & 1
\end{array}\right)$$

has rank 3. Then $X_1 = 0$, $X_2 = 2$ and $X_3 = 3$. Consequently,

$$\underline{e} = 2g_{13} + 3g_{17} = (000000000000200030000000)$$

and therefore the code-word sent is

$$\underline{c} = (32001204041144211023003 2).$$

As we said above, there is not only a unique $3-$set of $\mathcal{S}_4$ that contain positions $g_{13}$ and $g_{17}$ but all of them generate the same expression for $\underline{e}$. Indeed taking $\{g_{13}, g_{15}, g_{17}\}$, we have that the matrix $\mathcal{M}(g_{13}, g_{15}, g_{17})$ has rank 3 and therefore $X_1 = 2$, $X_2 = 0$ and $X_3 = 3$. Thus, $\underline{e} = 2g_{13} + 3g_{17}$ as calculated above.

## IV. Meggitt's Algorithm Generalization

Cyclic codes were introduced by E. Prange in [17] and are a family of linear codes that have a rich algebraic structure that allows obtaining many properties and a great variety of decoding methods (some in [18], [19]).

Every cyclic code can be seen as a $G-$code where $G$ is a cyclic group. Then group codes naturally generalize cyclic codes. In [20] and [21], J. E. Meggitt defined a decoding algorithm for cyclic codes. Here we show how to extend this

algorithm to use to decode with group codes.

For this, note that $g\underline{e}e_0^+ = gS$ for all $g \in G$. Furthermore, chosen a specific position $g_{i_0} \in G$, any word of $\mathbb{K}G$ can be obtained by multiplying a particular element of the group by an element of $\mathbb{K}G$ having to $g_{i_0}$ within its support. Therefore, we can consider the set of elements, called *class leaders*, of $\mathbb{K}G$ having weight less than or equal to $t$ and whose support contains a specific position $g_{i_0} \in G$. Then, we can make a list of the syndrome of class leader in a similar way to the one used in cyclic codes. This list is called *syndrome reduced list* and it is elaborated with the condition that, given two class leaders $\underline{a}$ and $\underline{b}$ (both elements of $\mathbb{K}G$) in this list, then there is no an element $g \in G$ such that $g\underline{a} = \underline{b}$. The order of complexity to make a syndrome reduced list is

$$\mathcal{O}\left(nt \times (q-1)^t \times \binom{n-1}{t-1}\right).$$

Once this list has been computed and word $\mathfrak{r}$ has been received, the following algorithm is applied:

*Algorithm 2*

*Step 1:* Compute the syndrome. If $S = 0$, then there are no errors and the algorithm ends. Otherwise, continue to

*Step 2:* Take $g \in G$ and compute $S_g = gS$.

a. If $S_g$ is syndrome of some class leader $\underline{e}'$ in the syndrome reduced list, then the error is $\underline{e} = g^{-1}\underline{e}'$ and the algorithm ends.
b. Otherwise, the element $g$ is discarded and another element of $G$ is considered and Step 2 is repeated with it.

The algorithm finishes when a element $g$ satisfying (P):

*$S_g$ is syndrome of some class leader in the syndrome reduced list*,

allows us to find the error or when all elements of $G$ have been checked and none satisfies property (P). In the last case, we conclude that the number of errors is greater than $t$ and the group code can not correct them for $\mathfrak{r}$.

**Observation 2.** This decoding algorithm generalizes the Meggitt's decoding algorithm and it has a complexity of the order $\mathcal{O}(n^3)$. In particular, if $\mathfrak{C}$ is a binary group code of a semisimple group algebra. The complexity to make the reduced list is

$$\mathcal{O}\left(nt \times \binom{n-1}{t-1}\right).$$

This makes it very appropriate to apply this algorithm in codes of this type.

**Example 2.** Let $\mathbb{K} = \mathbb{F}_2$ and $G = \mathcal{C}_7 \times \mathcal{C}_7$, where $G = \langle a, b\rangle$. Here $\mathbb{F}_2(\mathcal{C}_7 \times \mathcal{C}_7) = \langle e_1\rangle \oplus \cdots \oplus \langle e_{17}\rangle$. The dimension of the first simple component is 1 and the others simple component have dimension 3. If we fix the basis

$$\{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4, a^5,$$
$$a^4b, a^3b^2, a^2b^3, ab^4, b^5, a^6, a^5b, a^4b^2, a^3b^3, a^2b^4, ab^5, b^6,$$
$$a^6b, a^5b^2, a^4b^3, a^3b^4, a^2b^5, ab^6, a^6b^2, a^5b^3, a^4b^4, a^3b^5, a^2b^6,$$
$$a^6b^3, a^5b^4, a^4b^5, a^3b^6, a^6b^4, a^5b^5, a^4b^6, a^6b^5, a^5b^6, a^6b^6\},$$

and $\mathfrak{C} = \langle e_0\rangle$, where

$$e_0 = (011111000011111000000000$$
$$10000010111110101110101),$$

then $n = 49$, $k = 18$, $d = 12$ and $t = 5$. Let us consider the syndrome list reduced of the 194.580 class leaders whose weight is less than or equal to 5 and its support contains the position $1 \in \mathcal{C}_7 \times \mathcal{C}_7$. If

$$\underline{r} = (000100010111001101100111$$
$$10111000101010110011101 00),$$

then, decoding with this algorithm:

*Step 1:* We have that $S \neq 0$ and it continues to
*Step 2:* If we consider $g = a^3b$ then

$$S_g = (111011011101010111001111$$
$$000111110010111101101111 0),$$

whose class leader is $\underline{e}' = 1 + a^2b^2 + a^4b^5 + a^6b^4 + a^5b^6$. So, the error is

$$\begin{aligned}\underline{e} &= (a^3b)^{-1}\mathfrak{e}' \\ &= a^4b^6(1 + a^2b^2 + a^4b^5 + a^6b^4 + a^5b^6) \\ &= ab^4 + a^3b^3 + a^6b + a^2b^5 + a^4b^6\end{aligned}$$

or

$$\underline{e} = (000000000000000000010000$$
$$10001000100000000000001000).$$

In the table I we see the approximate number of operations in the precalculations and in the decoding for each one of the algorithms presented of this example.

## V. Conclusions

1) We have presented two decoding algorithms for semisimple group codes.
2) When the field is big enough, the most suitable algorithm is the syndrome decoding algorithm.
3) The generalized Meggitt decoding algorithm requires of many precalculations to obtain a syndrome reduced list,

TABLE I
APPROXIMATE NUMBER OF OPERATIONS IN ALGORITHMS PRESENTED

| Algorithm | Precalculations | Decoding |
|---|---|---|
| 1 | $2.4 \times 10^3$ | $8.9 \times 10^{10}$ |
| 2 | $1.0 \times 10^8$ | $2.4 \times 10^5$ |

but it is more efficient than syndrome decoding algorithm. Hence, the most appropriate decoding algorithm to apply for a binary semisimple group code is the Meggitt's generalization.

## ACKNOWLEDGMENTS

This work was supported by the project MINECO-18-MTM2017-83506-C2-2-P.

## REFERENCES

[1] J. McEliece: "A public-key cryptosystem based on algebraic coding theory", in *DSN Progress Report, Jet Propulsion Laboratory, California Institute of Technology*, pp. 114-116, 1978.

[2] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal and D. Schipani: "Enhanced Public Key Security for the McEliece Cryptosystem" in *Journal of Cryptology*, vol. 29, pp. 1-27, 2016.

[3] V. Sidelnikov: "A public-key cryptosystem based on binary Reed-Muller codes" in *Discrete Mathematics and Applications*, vol. 4, n. 3, pp. 191-207, 1994.

[4] H. Janwa and O. Moreno: "McEliece public cryptosystem using algebraic-geometric codes" in *Des. Codes Cryptogr.*, n. 8, pp. 293-307, 1996.

[5] L. Minder and A. Shokrollahi: "Cryptanalysis of the Sidelnikov Cryptosystem" in *Naor M. (eds) Advances in Cryptology - EUROCRYPT 2007. Lecture Notes in Computer Science*, vol. 4515, pp. 347–360, 2007.

[6] A. Couvreur, I. Márquez-Corbella and R. Pellikaan: "A polynomial time attack against algebraic geometry code based public key cryptosystem" in *IEEE International Symposium on Information Theory*, pp. 1446-1450, 2014.

[7] T. Berger, P. Cayrel, P. Gaborit and A. Otman: "Reducing Key Length of the McEliece Cryptosystem" in *Conference: Progress in Cryptology - AFRICACRYPT 2009, Second International Conference on Cryptology in Africa, Gammarth, Tunisia, June 21-25, 2009. LNCS*, vol. 5580, pp. 77-97, 2009.

[8] Z. Li, C. Xing and S. Ling Yeo: "Reducing the Key Size of McEliece Cryptosystem from Automorphism-induced Goppa Codes via Permutations" in *IACR International Workshop on Public Key Cryptography, PKC 2019: Public-Key Cryptography - PKC 2019*, pp. 599-617, 2019.

[9] J. Faugère, L. Perret and F. de Portzamparc: "Algebraic attack against variants of McEliece with Goppa polynomial of a special form" in *Advances in Cryptology-ASIACRYPT 2014*, vol. 8873, pp. 21-41, 2014.

[10] J. Bernal, Á. del Río and J. Simón: "An intrinsical description of group codes" in *Designs, Codes and Cryptography*, vol. 51, n. 3, pp. 289-300, 2009.

[11] C. García, S. González, V. Markov, C. Martínez and A. Nechaev: "Group codes over non-abelian groups" in *Journal of Algebra and its Applications*, vol. 12, n. 7, 2013.

[12] C. García, S. González, V. Markov, C. Martínez and A. Nechaev: "New examples of non-abelian group codes" in *Advances in Mathematics of Communications*, vol. 10, n. 1, pp. 1-10, 2016.

[13] C. García, S. González, V. Markov and C. Martínez: "Non-abelian group codes over an arbitrary finite field" in *Journal of Mathematical Sciences*, vol. 223, n. 5, pp. 504-507, 2017.

[14] J. Bernal, Á del Río and J. Simón: "Partial Permutation Decoding for Abelian Codes" in *IEEE Transactions on Information Theory*, pp 274-278, 2012.

[15] M. Elía and C. García: "Ideal Group codes and their Syndrome Decoding" in *21st International Symposium on Mathematical Theory of Networks and Systems. Groningen, The Netherlands July*, pp. 7-11, 2014.

[16] R. Curtis, I. Reiner: "Representation theory of finite groups and associative algebras", pp. 166-170, 1962.

[17] E. Prange: "Cyclic error-correcting codes in two symbols" in *Technical notes issued by Air Force Cambridge Research Labs.*, TN-57-103, 1957.

[18] E. Prange: "Some cyclic error-correcting codes with simple decoding algorithms" in *Technical notes issued by Air Force Cambridge Research Labs.*, TN-58-156, 1958.

[19] F. MacWilliams: "Permutation decoding of systematic codes" in *Bell System Tech. J.*, n. 43, pp. 485-505, 1963.

[20] J. E. Meggitt: "Error-correcting codes for correcting bursts of errors" in *IBM J. Res. Develop.* n. 4, pp. 329-334, 1960.

[21] J. E. Meggitt: "Error-correcting codes and their implementation for Data Transmission Systems" in *IRE Trans. Inform. Theory IT-6*, pp. 459-470, 1961.

# e-ticketing mediante NFTs

M. Magdalena Payeras-Capellà
Universitat de les Illes Balears
Crta. Valldemossa Km. 7,5, Palma
mpayeras@uib.cat

Macià Mut-Puigserver
Universitat de les Illes Balears
Crta. Valldemossa Km. 7,5, Palma
macia.mut@uib.cat

Jordi Castellà-Roca
Universitat Rovira i Virgili
Av. Països Catalans, Tarragona
jordi.castella@urv.cat

Jaume Ramis-Bibiloni
Universitat de les Illes Balears
Crta. Valldemossa Km. 7,5, Palma
jaume.ramis@uib.es

Llorenç Huguet-Rotger
Universitat de les Illes Balears
Crta. Valldemossa Km. 7,5, Palma
l.huguet@uib.es

Miquel À. Cabot-Nadal
Universitat de les Illes Balears
Crta. Valldemossa Km. 7,5, Palma
miquel.cabot@uib.es

***Resumen*—Los tickets electrónicos representan, evitando el soporte físico, el derecho de acceso a un determinado servicio. Estos tickets deberán satisfacer determinados requisitos de seguridad. Además, en determinados servicios puede permitirse que la identidad del usuario no forme parte del ticket ni sea revelada durante su uso, de tal modo que el usuario pueda permanecer anónimo. Los tickets electrónicos también mantienen algunos de los problemas inherentes a los tickets convencionales, como la reventa no permitida, facilitada por la inexistencia de un elemento físico a transferir. Recientemente algunos prestadores de servicios han planteado el uso de NFTs para la gestión del derecho de acceso. Este artículo pretende analizar el uso de NFTs en sistemas de e-ticketing, evaluando sus retos y oportunidades.**

***Index Terms*—e-ticketing, blockchain, NFT, seguridad**

## I. Introducción

Las tecnologías de la información (IT) están cada vez más presentes en nuestra sociedad. Nos permiten disponer de servicios en línea y disfrutar de ellos con independencia de nuestra ubicación e instante de tiempo. Entre los impactos de las IT resulta destacable la transformación que han sufrido los sistemas de emisión de tickets.

Cuando el proceso de compra y obtención del ticket es totalmente electrónico es necesario que el proceso de validación también lo sea. Los usuarios deben custodiar sus tickets y validarlos para acceder al servicio. Los tickets se pueden definir como la representación del derecho de su propietario a usar un determinado servicio. Eventualmente, este ticket puede ser transferido por su propietario a otro, cediendo sus derechos asociados.

La disponibilidad del servicio también permite clasificar los tickets según si hay una limitación en su emisión (p.e. entradas en un concierto) o no hay limitación alguna (tickets de metro). También, hay servicios que pueden ser anónimos y otros no. El usuario que tiene el ticket es quien puede utilizar el servicio. El posible anonimato de los tickets también debe ser contemplado en la incorporación de la transferibilidad. El sistema transferible debe mantener las características de anonimato del sistema no transferible.

Hay multitud de propuestas para la gestión de tickets electrónicos, tratando de conseguir los requisitos de seguridad, privacidad y funcionalidad necesarios. Los mismos autores tienen propuestas de e-tickets que garantiza la exculpabilidad del usuario [1], con reusabilidad de los mismos [2] y con que mecanismos de transferibilidad [3][4].

La irrupción de las tecnologías basadas en blockchain y concretamente la reciente aparición de los tokens no fungibles (NFT) permiten replantear el modelo de tickets electrónicos y afrontar los problemas derivados de su uso incorrecto [5]. Los NFT se basan en un contrato inteligente en una red blockchain que los permita, como Ethereum o Cardano. Los NFTs se basan en estándares que definen su formato, como el ERC-721 de Ethereum. Entre sus usos principales, los NFT representan la existencia y la propiedad de diferentes tipos de activos, como vídeos, imágenes, tickets, obras de arte, acciones, puntos en sistemas de fidelización o incentivos, artículos coleccionables, certificación de propiedad de activos digitales, fan tokens, etc [6].

## II. Tokens no fungibles

NFT significa token no fungible. Para comprender fácilmente qué son exactamente y cómo se pueden usar, debemos aclarar la principal diferencia entre fungibilidad y no fungibilidad. La fungibilidad se refiere a la capacidad de un activo para intercambiarse con un activo similar sin afectar a su valor. La fungibilidad también define las características de un activo, como la divisibilidad y el valor. Las características individuales dictan su singularidad, al igual que determinados activos del mundo real. Cada NFT contiene información distintiva que lo distingue de cualquier otro NFT y lo hace fácilmente verificable. El primer NFT se introdujo en la cadena de bloques Ethereum en 2017.

Los tokens fungibles como Bitcoin y las monedas fiduciarias son idénticos o uniformes y se pueden intercambiar con otros tokens fungibles del mismo tipo sin ningún problema.

### *II-A. Estándares ERC.*

Existen varios estándares en la blockchain Ethereum para diferentes usos. Los tokens, en Ethereum, se implementan en el lenguaje de programación Solidity. ERC-20 es un estándar general para tokens que se usa principalmente en la cadena de bloques Ethereum con funcionalidad básica para transferir tokens a través de un contrato inteligente. El estándar hace que los tokens sean iguales unos a otros (en términos de tipo y valor). ERC-721 es el estándar de token no fungible en el que cada token es un activo único distinguible. Con este estándar, cada NFT tiene un propietario [7]. En concreto, cada NFT tiene una variable llamada *tokenId* que es globalmente única. A través del estándar ERC-721 cualquier persona puede

verificar la dirección del contrato y consultar las características. ERC-1155 (multi-token standard) es un nuevo estándar que admite las funciones de tokens fungibles (ERC-20) y no fungibles (ERC-721). También admite la conversión y acuñación de nuevos tokens. RC-1155 amplía la funcionalidad de *tokenId*, donde cada uno de ellos puede representar de forma independiente diferentes tipos de tokens configurables. El campo puede contener su información personalizada, como los metadatos, la hora de bloqueo, la fecha, el suministro cualquier otro atributo.

```
FUNCTIONS
balanceOf(owner)
ownerOf(tokenId)
safeTransferFrom(from, to, tokenId)
transferFrom(from, to, tokenId)
approve(to, tokenId)
getApproved(tokenId)
setApprovalForAll(operator, _approved)
isApprovedForAll(owner, operator)
safeTransferFrom(from, to, tokenId, data)
```

Figura 1. Funciones ERC-721

También existen otros estándares para NFT en blockchains como NEO y EOS.

### *II-B. Usos de NFTs*

Las aplicaciones de los NFT son múltiples y diversas. Sus características de unicidad, transferibilidad y trazabilidad les hacen especialmente interesante en muchos escenarios como el entorno del arte, la música, las colecciones y los tickets. Dado que los NFTs representan la existencia y la propiedad de diferentes tipos de activos, se utilizan con artículos multimedia, acciones, puntos en sistemas de fidelización, incentivos, certificación de propiedad de activos, premios o títulos.

## III. E-TICKETING

El ticket electrónico es un contrato entre un usuario y un proveedor de servicios. Si el usuario demuestra ser dueño del ticket, obtiene el derecho a utilizar el servicio bajo los términos y condiciones del mismo (por ejemplo, el tiempo de validez del ticket). Comúnmente, se requiere la validación del ticket para poder utilizar el servicio. Dependiendo de las condiciones del ticket, se puede validar una vez, varias veces predefinidas o indefinidamente hasta una fecha límite.

El ticket debe incluir elementos para garantizar la seguridad del sistema y la privacidad de los usuarios. Los requisitos relacionados con la seguridad y la privacidad pueden variar entre las distintas aplicaciones de los tickets. En algunos casos, la seguridad sería crítica, como la falsificación de tickets para viajes aéreos. En otros, los requisitos de privacidad, como el anonimato de los usuarios, son obligatorios. Estos requisitos se definieron en [8] y en [9] se revisaron para su aplicación a sistemas de transporte.

### *III-A. Información incluida en el ticket*

Los tickets electrónicos deben incluir información básica para su uso práctico. En este apartado se describen brevemente los campos de información que pueden (en función del servicio que representen) incluir los tickets electrónicos:

- *Número de serie:* identificación única de cada ticket electrónico.
- *Emisor:* entidad responsable de la emisión del ticket electrónico. Este emisor puede ser también el proveedor del servicio o un intermediario.
- *Prestador de servicios:* entidad que ofrece el servicio al usuario.
- *Usuario:* información sobre el propietario del ticket electrónico. En caso de existir este campo en el ticket, no se podría lograr el anonimato del usuario.
- *Servicio:* descripción del contrato de servicio.
- *Términos y condiciones:* definición de los términos y condiciones del ticket electrónico o, alternativamente, un enlace externo para permitir la consulta.
- *Tipo de ticket electrónico:* información sobre las condiciones de uso. Contiene los campos:

1. Transferibilidad: indica si se permite la transferencia del ticket a otro usuario.
2. Reusabilidad : información sobre el número permitido de usos del ticket electrónico.

- *Atributos:* otros atributos del ticket electrónico que dependen del servicio (p. ej., localidad en espectáculos o destino en servicios de transporte)
- *Tiempo de validez:* incluye dos marcas de tiempo, la fecha de inicio y la de vencimiento.
- *Fecha de emisión:* fecha de emisión del e-ticket. El campo de tiempo de validez se puede establecer mediante la inclusión de este campo junto con los términos y condiciones.
- *Campo que pruebe la identidad del emisor.* Tradicionalmente los e-tickets contienen la firma digital del emisor.

### *III-B. Requisitos de los sistemas de e-ticketing*

Podemos clasificar los requisitos de los tickets electrónicos en tres categorías. Por un lado, tenemos requisitos de seguridad y de privacidad y, por otro lado, tenemos requisitos funcionales. Algunos de ellos pueden tener un impacto en diversas categorías.

#### *III-B1. Requisitos de Seguridad:*

- **Integridad.** Debe ser posible verificar si el contenido del e-ticket ha sido modificado con respecto al emitido por el correspondiente emisor autorizado. Todos los participantes deben poseer la capacidad de verificar si un e-ticket ha sido manipulado.
- **Autenticidad.** Un usuario tiene que poder verificar quien ha emitido un e-ticket. El cumplimiento de este requisito ayudará a los usuarios a verificar si el emisor es un emisor autorizado.
- **No repudio de origen.** El usuario que ha enviado o generado un mensaje no debe ser capaz de negar que ha lo ha enviado o generado. Este requisito puede ser de utilidad en diversas etapas, pero resulta particularmente importante en relación a un e-ticket emitido válido: el emisor no debe ser capaz de negar haberlo emitido, y con un contenido específico.

- **No repudio de recepción.** Un usuario que recibe un mensaje no debe ser capaz de negarlo tras haberlo recibido. De nuevo, este requisito puede ser útil en varias etapas. Por ejemplo, un usuario que haya solicitado y recibido un e-ticket no debe poder negar que lo haya recibido; o un proveedor que ha recibido un e-ticket por un servicio, no debería ser capaz de negar su recepción.
- **Infalsificabilidad.** Exclusivamente los usuarios autorizados pueden emitir e-tickets válidos. Es decir, no debe ser posible falsificar e-tickets para ser utilizados como si hubieran sido emitidos por un emisor autorizado.
- **Equidad.** Al final de un intercambio entre dos o más partes, o bien todas ellas obtienen los artículos esperados o ninguna de ellas lo obtiene. Este requisito puede ser de utilidad para múltiples procesos relacionados con la gestión de e-tickets, como la emisión, el uso y la compensación.
- **No sobreutilización.** Les e-tickets sólo deben ser utilizados según lo acordado entre el emisor y el usuario. Los e-tickets no reutilizables no pueden ser utilizados de nuevo después de haber sido gastados. Los e-tickets reutilizables pueden ser usados tantas veces como se haya acordado en el momento de su emisión.
- **Exculpabilidad.** El proveedor de servicio no puede acusar falsamente de sobregasto a un usuario honesto y el usuario tiene la capacidad de demostrar que ya ha validado el e-ticket antes de utilizarlo (incapacitando así al proveedor de acusarlo falsamente).
- **Reusabilidad.** El e-ticket reutilizable puede utilizarse más de una vez. El sobregasto debe prevenirse o bien ser detectado. Los e-tickets deben incorporar medidas de seguridad que permitan usarlos durante los períodos de tiempo de validez o bien el número de veces acordado (o bien una combinación de ambos, tiempo y número de usos).
- **Transferibilidad.** Un usuario puede transferir su e-ticket a otros usuarios. Algunos tickets pueden transferirse a otros (p.e.: entradas de espectáculos, tickets de bus, ...). Obviamente éste no es el caso de los tickets identificados (p.e.: billetes de avión). Un individuo al que se le transfiere un e-ticket (no directamente de un emisor autorizado) debe poder verificar que dicho e-ticket es válido (ello será fácil en el caso de satisfacerse las propiedades de no repudio, integridad y autenticidad) y que no ha sido gastado por la entidad que realiza la transferencia (o entidades transferentes previas).

*III-B2. Requisitos de Privacidad:*

- **e-tickets identificados.** La identidad del propietario del e-ticket debe ser verificable. No todos los tickets presentan los mismos requisitos en lo que a anonimato se refiere, de manera que se deben distinguir distintos escenarios posibles para los e-tickets. El primero corresponde a los e-tickets no anónimos, en los que el servicio requiere la identificación del usuario. Esto significa que la identidad del usuario debe hallarse incrustada en el e-ticket, de manera que el proveedor del servicio pueda verificar que el usuario está autorizado a gastar dicho e-ticket.
- **e-tickets anónimos.** El propietario de un e-ticket debe permanecer anónimo. Y debe serlo delante del emisor, el verificador y el proveedor de servicio. Este requisito está relacionado con la manera de emitir y usar el e-ticket.
- **Anonimato totalmente revocable.** El anonimato de los usuarios puede ser revocado. La identidad de los usuarios está incrustada, de alguna manera (seudónimos, identidad real), en los e-tickets. Típicamente, tan solo un reducido subconjunto de actores puede revelar esta identidad.
- **Anonimato revocable selectivamente.** La identidad de un usuario fraudulento de un e-ticket a priori anónimo puede ser revelada. Este requisito es bastante similar al previo, pero es más restrictivo: sólo los usuarios deshonestos pueden perder el anonimato.

*III-B3. Requisitos Funcionales:* Existen otros requisitos que no están directamente relacionados con la seguridad, pero no por ello menos importantes.

- **Tiempo de validez.** Fecha de caducidad. Determina la validez del e-ticket durante un intervalo de tiempo.
- **Verificación offline**. El e-ticket puede verificarse sin ninguna conexión externa. En determinados escenarios, no será posible contactar con bases de datos externas o terceras partes de confianza para verificar si un e-ticket es válido o no. Esta solución es preferible a una verificación on-line.
- **Portabilidad.** Los e-tickets se tienen que poder almacenar en dispositivos móviles para su portabilidad. En su defecto deben poder ser accesibles en todo momento y lugar.
- **Tamaño reducido.** Los e-tickets deben ser tan cortos como sea posible.
- **Flexibilidad.** Los e-tickets pueden utilizarse en múltiples entornos. Se puede diseñar un e-ticket específico para cada aplicación, o bien se puede adaptar un e-ticket general para cada aplicación. Obviamente, la segunda solución es preferible si el objetivo es economizar recursos, además de permitir un mejor análisis de seguridad.
- **Facilidad de uso**. Aprender como utilizar los e-tickets tiene que ser fácil.
- **Eficiencia.** El procesamiento de un e-ticket debe consumir la menor cantidad de recursos posibles.
- **Flexibilidad de pago.** Se tienen que poder utilizar los sistemas habituales de pago para pagar los e-tickets.
- **e-tickets universales.** Los clientes deben poder gastar sus e-tickets en cualquier proveedor de servicio adecuado.
- **Disponibilidad.** Los e-tickets deben poder usarse cuando sea necesario.

## IV. Análisis de la aplicabilidad de los NFTs en sistemas de e-ticketing

En la aplicación de NFTs a la gestión de e-tickets aparecen retos por resolver al tiempo de las características intrínsecas de los NFTs permiten resolver de forma directa el cumplimiento de algunos de los requisitos mencionadas anteriormente.

En esta sección se presentaran los requisitos de los sistemas de e-ticketing que los NFT cumplen por definición, al tiempo que se determinan los requisitos que deben satisfacer los protocolos de gestión detectando los principales retos de esta aplicación.

*IV-A. Información incluida en el ticket NFT*

En la sección III-A se listan los elementos que deben formar parte del ticket. En los NFT esta información se puede almacenar en los metadatos asociados al NFT o bien utilizar estos metadatos para incluir un puntero a la ubicación de los datos off-chain.

*IV-B. Cumplimiento de requisitos de seguridad, privacidad y funcionalidad*

Los sistemas de gestión de NFT son esencialmente aplicaciones descentralizadas y, por tanto, mantienen las propiedades de los registros inmutables subyacentes. Por este motivo algunas de las propiedades analizadas, divididas en propiedades de seguridad, privacidad y funcionalidad, como en III-B, se satisfacen de origen gracias a las características intrínsecas de los NFTs. Otras propiedades, en cambio, deberían tenerse en cuenta en el diseño de los protocolos de gestión mientras que otras representaran retos a resolver.

*IV-B1. Propiedades de seguridad::*

- **Integridad.** La información del token se puede registrar en una cadena de bloques, lo que los hace inmutables y confiables. Los NFTs son resistentes a la manipulación ya que los metadatos de NFT y sus registros comerciales se almacenan de forma persistente y no se pueden manipular una vez que las transacciones se consideran confirmadas.
- **Autenticidad.** La estandarización de NFT hace que cada token sea identificable y auténtico. El NFT, con sus metadatos de token, así como su propiedad se puede verificar públicamente. Las actividades de los NFT incluyen la acuñación, venta y compra que son de acceso público.
- **No repudio de origen, no repudio de recepción.** Cuando el NFT es transferido y cambia de wallet la transferencia se registra en la cadena de bloques por lo que no puede negarse tanto desde el punto de vista de la emisión como de la recepción.
- **Equidad, Atomicidad.** El comercio de NFT se puede completar en una transacción atómica, consistente, aislada y duradera.
- **Infalsificabilidad.** La transparencia de blockchain permite verificar y probar la autenticidad de cada NFT, pudiendo prevenir de manera efectiva los problemas de falsificación. Esto hace que la creación y circulación de activos digitales falsos no tenga sentido porque cada elemento se puede rastrear hasta el emisor original.
- **No sobreutilización.** Cuando un NFT es transferido este deja de ser propiedad del usuario emisor de la transferencia. De este modo se impide el uso del NFT en una nueva transacción. Tampoco es posible transferir un NFT del que no se es propietario.
- **Exculpabilidad.** Si un usuario posee un NFT que representa un ticket y este se utiliza para acceder al servicio, el usuario puede mantener la propiedad del NFT que ahora estará marcado como $usado$. De este modo el usuario puede demostrar tanto que es propietario del ticket NFT como que este ha sido validado.
- **Reusabilidad.** La reusabilidad de los tickets puede permitirse de forma ilimitada durante un período de tiempo o de forma limitada para un numero de usos prefijados. La sobreescritura de las funciones del token ERC-721 permitiría la incorporación de contadores que pudieran llevar el control de la reusabilidad del token.
- **Transferibilidad.** Propiedad significa que un NFT solo puede ser transferido por el propietario del activo debido a los contratos inteligentes y los derechos asociados. Incluso el emisor de un NFT no puede replicar o transferir el NFT sin el permiso de su propietario. El estándar ERC-721 define las interfaces mínimas, incluidos los detalles de propiedad, la seguridad y los metadatos, y la propiedad se asigna al propietario del activo. La estructura del NFT ofrece una forma sencilla de verificar su propiedad y transferir el token entre propietarios. En determinados casos, sin embargo, se desea que el token no sea transferido y seria deseable contar con tokens no fungibles y no transferibles.

*IV-B2. Propiedades de privacidad::* Las propiedades relacionadas con la privacidad merecen especial atención, ya que la gestión del anonimato y su revocabilidad no esta resuelta en los NFTs.

- **Identificación del usuario del servicio.** En caso de que se deseen tickets identificados la identidad del usuario debe incluirse entre los datos del NFT.
- **Anonimato.** La identidad del usuario puede no formar parte del NFT. Sin embargo este está vinculado a la dirección de su propietario. El anonimato conseguido de este modo no es un anonimato fuerte, dependiendo de la red en la cual se desplieguen los contratos inteligentes que gestionan en NFT.
- **Anonimato revocable general.** Para tener un sistema totalmente revocable se debería contar con un mecanismo para obtener las identidades de todos los propietarios participantes en el sistema.
- **Anonimato revocable selectivo.** Esta característica es más difícil de conseguir porque solo debe permitirse la revocación de aquellos usuarios que actúan de forma fraudulenta. Por tanto no debe ser una revocación basada únicamente en las direcciones de los usuarios.

*IV-B3. Propiedades funcionales:* Finalmente se analizaran las propiedades relacionadas con la funcionalidad y usabilidad del sistema.

- **Tiempo de Validez.** El tiempo de validez del ticket puede aparecer entre los datos del NFT.
- **Verificación On-line/off-line.** La validación de los NFTs debe realizar-se online. Esta verificación requiere un tiempo haciendo que el proceso de validación a tiempo real, por ejemplo en la entrada de un evento, pueda convertirse en un problema.
- **Portabilidad.** Los NFTs no requieren ser transportados en un dispositivo. Los NFT emitidos están siempre disponibles para ser validados o transferidos.
- **Tamaño Reducido.** El tamaño de los tokens y los contratos que los manejan se asocian al coste de emisión por lo que deben ser tenidos en cuenta.
- **Flexibilidad.** Todos los NFT y sus productos correspondientes pueden negociarse e intercambiarse arbitrariamente.
- **Facilidad de uso.** Dentro del ecosistema, cada NFT tiene la información de propiedad actualizada, que es fácil de usar y clara. Sin embargo las herramientas de gestión no son de uso extendido entre el gran público.

*IV-C. Retos inherentes al uso de NFTs*

El uso de NFTs plantea varios desafíos. Al tratarse de código estandarizado que se ejecuta en una cadena de bloques, dependen en gran medida de las propiedades del protocolo de la cadena de bloques subyacente.

- **Coste.** Las transacciones realizadas cobre las redes blockchain tienen un coste. En la red Ethereum existe una tarifa dinámica para realizar cualquier transacción en la red llamada ”tarifa de gas”, siendo un desafío importante para los NFTs porque la mayoría de los NFTs se acuñan en la cadena de bloques de Ethereum. Esto podría ser un gran problema, especialmente para los NFTs baratos, porque este alto precio de la tarifa del gas no es asequible para ellos. Para abordar este problema, las cadenas de bloques Cardano o Polkadot o la versión Ethereum 2.0 en el futuro serían una buena opción para acuñar los NFTs.
- **Ataque del 51 por ciento.** No es probable que las cadenas de bloques más potentes sufran un ataque del 51 %, porque presentan alta descentralización.
- **Propiedad Intelectual.** Es importante evaluar los derechos de propiedad de un individuo sobre un determinado NFT. Los metadatos del contrato inteligente subyacente deben contener los términos y condiciones de posesión de un NFT.
- **Ciberseguridad.** Los problemas generales de ciberseguridad también afectan a los NFTs. El desarrollo y la seguridad de contratos inteligentes son una de las preocupaciones críticas en el entorno NFT.
- **Impacto ambiental.** Si las tecnologías basadas en cadenas de bloques se adoptan tan ampliamente como otras tecnologías nuevas pueden influir en el calentamiento global.
- **Privacidad.** La privacidad es un reto en los NFTs, tanto las identidades como la preservación de información confidencial de los activos digitales.
- **Usabilidad.** Los NFTs suelen carecer de facilidad de acceso para los usuarios debido a la naturaleza de los componentes de back-end y la falta de interfaces fáciles de usar.
- **Mantenimiento.** Un problema con el que los NFTs van a encontrarse es la convivencia de tokens con diferentes versiones de smart contracts. La actualización de los contratos inteligentes existentes todavía presenta muchos problemas técnicos y operativos.
- **Extensión.** El problema de extensibilidad se basa en dos aspectos, la interoperabilidad y la actualización de los NTFs. Interoperabilidad entre cadenas porque los ecosistemas NFT existentes están aislados entre sí y los NFTs en un ecosistema solo pueden comerciar dentro del mismo ecosistema o red. Además, las posibles bifurcaciones de las cadenas pueden causar conflictos en los NFTs.

### *IV-D. Retos específicos de los NFTs usados en e-ticketing*

Una vez analizados los posibles problemas inherentes al uso de NFTs nos centraremos en los aspectos a tratar en el diseño de sistemas de e-ticketing basados en NFTs.

- **Privacidad.** Como se ha explicado anteriormente, en algunos casos se requiere que el propietario del ticket este totalmente identificado mientras que en otros casos pueda ser anónimo. En un NFT se define al propietario por lo que este podría ser identificado aunque deben estudiarse las técnicas que permitan que esta identificación sea correcta y segura. Por otra parte cuando se quieran emitir tickets anónimos se plantea el reto de que estos no permitan, de ningún modo, identificar al propietario. Más allá, los mecanismos de revocación del anonimato deben ser estudiados. El anonimato representa un desafío ya que la privacidad no se garantiza en la blockchain, siendo posible dar sentido a los datos pseudónimos en cadenas de bloques públicas, donde la transparencia y el acceso público son una característica clave [10].
La privacidad de los NFTs aún se está estudiando en este momento. La mayoría de las transacciones de NFTs dependen de la plataforma Ethereum, que solo proporciona pseudoanonimato en lugar de anonimato completo. Los usuarios pueden ocultar sus identidades hasta cierto punto si el público conoce los vínculos entre sus verdaderas identidades y las direcciones asociadas. De lo contrario, toda la actividad de los usuarios bajo la dirección expuesta es visible.
Algunas soluciones que han utilizado anteriormente en protocolos de e-ticketing para obtener anonimato, como las firmas de grupo [3], el cifrado homomórfico, las pruebas de conocimiento nulo, las firmas de anillo o la computación multiparte aún no se han aplicado a los sistemas de gestión de NFTs debido a que sus primitivas criptográficas complejas presentan inconvenientes en los sistemas basados en blockchain ya que los costes asociados a la ejecución de funciones en los contratos inteligentes es un elemento clave y entra en conflicto con la implementación de sistemas de privacidad. [11] plantea el uso de ZKP para conseguir privacidad en el entorno Ethereum.
- **Adopción de la tecnología.** Las tecnologías subyacentes en el uso de NTFs, como blockchain, se encuentran aun en sus fases iniciales de aplicación. En toda nueva tecnología deben considerarse los problemas de adopción por parte de los usuarios y tener en cuenta que la facilidad de uso debe ser una característica fundamental del servicio.
- **Limitación de la transferibilidad.** En caso de que el proveedor de servicio quiera impedir la transferencia de los tickets no puede evitar que el titular de un ticket venda un token que se encuentra dentro de una cadena de bloques. Sin embargo, gracias a la tecnología se sabrá que se ha realizado la transacción. El contrato inteligente puede restringir el precio pagado por la reventa, pero no puede evitar que el dinero cambie de manos al mismo tiempo, como un pago en efectivo. Este problema es bastante difícil de resolver porque no puede tener control sobre las personas que intercambian efectivo fuera de línea, pero permite que el organizador del evento pueda identificar cada transferencia de tickets dado que la tecnología blockchain brinda trazabilidad. [12].
En los sistemas de tickets electrónicos como [3], [1], [4] se han introducido mecanismos para permitir la transferibilidad en aquellos casos en que se considere oportuno, manteniendo bajo control el numero de transferencias realizadas. En el caso de la representación de tickets mediante tokens se ha planteado la uso de insignias. Una insignia es un token que, una vez asignado, no se puede transferir. Pueden ser cuantitativas (por ejemplo, reputación, experiencia,...) o cualitativas (premios, títulos,...). Estas insignias serian tokens no fungibles y no transferibles.
Las propuestas de ERC1238: Non-transferrable Non-Fungible Tokens (NTT) [13] y [14], donde los autores presentan unos tokens ”ligados al alma”(SBT, por sus siglas en inglés) podrian ir encaminados a proporcionar esta funcionalidad.

## V. Oportunidades

El uso de NFTs en e-ticketing permite plantear nuevos usos o funcionalidades a los tickets.

### V-A. Regalías

Habitualmente, después de la venta de un ticket se desconoce si este cambia de manos y si esta transferencia se hace a cambio de dinero. Dado que cada servicio susceptible de ser comercializado mediante tickets puede tener unas determinadas características de transferibilidad ahora es posible poner unos condicionantes a la transferibilidad, como el pago de un porcentaje al emisor del servicio.

Un NFT puede ser programado para pagar regalías a su creador cada vez que se vende a un nuevo propietario. El proceso se realiza automáticamente y el emisor del ticket puede beneficiarse y ganar regalías cada vez que se completa una transacción del ticket.

Por ejemplo, en determinados casos el precio de un ticket depende del momento de su compra (venta anticipada). Una transacción posterior a este periodo puede ser programada para el pago de la diferencia en el precio.

### V-B. Propiedad fraccionada

En los usos más habituales de e-ticketing, como son el transporte público o las entradas de eventos, el fraccionamiento del ticket no es habitual. En otros escenarios, como los tickets que representan boletos de lotería, la propiedad compartida o fraccionada no es inusual. La propiedad fraccionada del NFT permite a diferentes usuarios poseer una parte de ese NFT específico.

### V-C. Tickets coleccionables

Determinados usos de los tickets electrónicos, como representar entradas a eventos tales como conciertos o competiciones deportivas, además de ser usados para acceder al servicio, pueden pasar a formar parte de colecciones.

## VI. Conclusiones

La característica principal de los NFT es poder representar la singularidad mejor que cualquier instrumento anterior basado en blockchain. Los tickets podrían considerarse como un conjunto de derechos de acceso a un servicio y, por lo tanto, la tokenización de los derechos en general podría considerarse un caso de uso viable para los sistemas basados en blockchain y, específicamente, también para los NFTs. Llegados a este punto es necesario evaluar si esta solución satisface los requisitos necesarios para un sistema de e-ticketing. Este trabajo presenta el listado de requisitos agrupados en requisitos de seguridad, privacidad y funcionalidad. Estos requisitos se han evaluado en el uso de NFTs para representar a los tickets. Como conclusión puede afirmarse que los NFTs ayudan, por sus características intrínsecas, a satisfacer propiedades de seguridad que antes requerían complejos protocolos para resolverse. Este es el caso, por ejemplo, de la transferibilidad o la exculpabilidad.

En cuanto a privacidad, son necesarios mecanismos que permitan que los NFTs puedan usarse de forma segura tanto en escenarios que requieren la identificación de los usuarios como en escenarios que deben proteger su privacidad. En éste último caso, la revocación del anonimato se presenta como un reto especialmente complejo. Otras propiedades en la que es necesario trabajar es en el control de la transferabilidad y la reusabilidad de los tickets. Actualmente, la funcionalidad, usabilidad y escalabilidad son características que presentan problemas en la gestión de los NFTs. Los resultados de este análisis determinan que los NFTs tienen una gran aplicabilidad en los sistemas de gestión de tickets al tiempo que identifican los puntos en que debe seguirse trabajando.

## Agradecimientos

El proyecto *Fair Exchange, Loyalty and TIckets with blockCHAIN (FeltiCHAIN)* RTI2018-097763-B-I00. está financiado por: FEDER/Ministerio de Ciencia e Innovación – Agencia Estatal de Investigación

## Referencias

[1] Arnau Vives-Guasch, Magdalena Payeras-Capellà, Macià Mut Puigserver, Jordi Castellà-Roca: .E-Ticketing Scheme for Mobile Devices with Exculpability", en *Data Privacy Management and Autonomous Spontaneous Security*, pp. 79-92, 2010.

[2] Arnau Vives-Guasch, Magdalena Payeras-Capellà, Macià Mut Puigserver, Jordi Castellà-Roca, Josep Lluís Ferrer-Gomila: .A Secure E-Ticketing Scheme for Mobile Devices with Near Field Communication (NFC) That Includes Exculpability and Reusability", en *IEICE Transactions on Information Systystems*, pp.78-93, 2012.

[3] Magdalena Payeras-Capellà, Macià Mut Puigserver, Jordi Castellà-Roca, Julio Bondia-Barcelo: "Design and Performance Evaluation of Two Approaches to Obtain Anonymity in Transferable Electronic Ticketing Schemes", en *Mobile Networks and Applications*, n. 22, pp. 1137-1156, 2017.

[4] Arnau Vives-Guasch, Magdalena Payeras-Capellà, Macià Mut Puigserver, Jordi Castellà-Roca, Josep Lluís Ferrer-Gomila: .Anonymous and Transferable Electronic Ticketing Scheme", en *Data Privacy Management and Autonomous Spontaneous Security*, pp. 100-113, 2013.

[5] Mengxuan Liu: 'A Hybrid Blockchain-Based Event Ticketing System", Thesis Master of Science, Department of Computer Science, University of Saskatchewan Saskatoon, 2021.

[6] W. Rehman, H. e. Zainab, J. Imran and N. Z. Bawany: "NFTs: Applications and Challenges", en *22nd International Arab Conference on Information Technology (ACIT)*, pp. 1-7, 2021.

[7] Adhithya Dev, Kevin Gomez, Syam Mathew, "Non-Fungible Tokens (NFT): New Emerging Digital Asset", en *Int. Journal of Research in Engineering and Science (IJRES)*, vol. 10, n. 4, pp. 01-07, 2022.

[8] Magdalena Payeras-Capellà, Macià Mut Puigserver, Josep Lluís Ferrer-Gomila, Jordi Castellà-Roca, Arnau Vives-Guasch: .Electronic Ticketing: Requirements and Proposals Related to Transport", en *Advanced Research in Data Privacy*, pp. 285-301, 2015.

[9] Macià Mut Puigserver, Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila, Arnau Vives-Guasch, Jordi Castellà-Roca: .A survey of electronic ticketing applied to transport", en *Computers & Security*, pp. 925-939, 2012.

[10] Aadarsh Mani, Sahil Verma, Sarthak Marwaha: 'A Comprehensive Study of NFTs", en *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, vol. 9, n. IV, pp. 1656-1660, 2021.

[11] Khatri, Y.: "EY Reveals Zero-Knowledge Proof Privacy Solution for Ethereum", https://www.coindesk.com/ey-reveals-zero-knowledge-proof-privacy-solution-for-ethereum/.

[12] Alessandro Perfetti: "Football meets blockchain: an innovative ticketing solution", Bachelor's Degree Thesis, Institution Luiss Guido Carli, 2020/2021.

[13] ERC1238: Non-transferrable Non-Fungible Tokens (NTT) #1238. https://erc1238.notion.site

[14] Weyl, Eric Glen and Ohlhaver, Puja and Buterin, Vitalik, "Decentralized Society: Finding Web3's Soul", 2022.

[15] Regner, Ferdinand et al.: "NFTs in Practice - Non-Fungible Tokens as Core Component of a Blockchain-based Event Ticketing Application", en *40th International Conference on Information Systems, ICIS*, 2019.

[16] Arsalan Parham, Corinna Breitinger: "Non-fungible Tokens: Promise or Peril?", 2022.

[17] Gang Wang, Mark Nixon: "SoK: tokenization on blockchain", en *14th International Conference on Utility and Cloud Computing (UCC '21) Companion*, 2021.

[18] Qin Wang, Rujia Li, Qi Wang, Shiping Chen: "Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges", 2021.

[19] A.D. Popescu: "Non-Fungible Tokens (NFT) - Innovation beyond the craze", en *5th Int. Conference on Innovation in Business, Economics & Marketing research (IBEM-2021)*, vol. 66, pp. 26-30, 2021.

# CCBHash (Compound Code Block Hash) para Análisis de Malware

Pablo Pérez Jiménez
NICS Lab, Málaga
ppj@lcc.uma.es

José Antonio Onieva González
NICS Lab, Málaga
onieva@uma.es

Gerardo Fernández
VirusTotal, Málaga
gerardofn@virustotal.com

***Resumen*—En estos últimos años, el análisis de malware ha adquirido una importancia cada vez mayor debido al aumento de ataques informáticos, cada vez más sofisticados. Uno de los objetivos que tiene esta rama de la ciberseguridad es encontrar similitudes entre distintos ficheros, permitiendo así detectar y clasificar malware e incluso, en algunos casos, realizar atribuciones. En este trabajo desarrollaremos un fuzzy hash capaz de caracterizar el malware generando una firma fácilmente comparable y almacenable de sus funciones. Ya que nuestra meta es poder detectar estas similitudes en grandes cantidades de datos en un periodo de tiempo razonable, el tamaño del hash debe ser limitado a la vez que guarde la máxima información posible.**

***Index Terms*—análisis de malware, fuzzy hashing, similitud**

## I. Introducción

Encontrar similitudes entre distintos ficheros es un problema común estudiado desde hace más de una década. En los últimos años el problema ha cobrado importancia para la búsqueda de similitudes entre ficheros maliciosos. Si bien es cierto que el objetivo es el mismo, también lo es que la definición del concepto de similitud puede diferir en función del campo en el que esté siendo estudiado. Cuando nos encontramos en el análisis de malware, la palabra similitud puede tomar dos significados. Dos ficheros pueden ser similares cuando el resultado de su ejecución es el mismo, o bien, cuando éste es distinto pero se usan funciones iguales o parecidas en el proceso.

En este trabajo, no olvidaremos ninguno de los dos enfoques anteriormente comentados. El objetivo es poder identificar similitudes entre una gran cantidad de ficheros maliciosos (del orden de petabytes), no solo a nivel de muestra, sino a nivel de segmento de código ensamblador. Al encontrarnos con petabytes de muestras, estas deben estar almacenadas de forma que se permita realizar una rápida comparación entre ellas. Para ello, debemos ser capaces de caracterizar una muestra completa de malware dividiéndola en trozos más pequeños de código ensamblador y generando un hash *compuesto* para cada uno de ellos. Dichos hashes se unirán posteriormente en un solo hash que identificará a la muestra completa, tal y como hacen los fuzzy hashes que ya conocemos [1]. A este fuzzy hash lo llamaremos CCBHash *(Compound Code Block Hash)*. Se debe tener en cuenta que estos hashes son calculados de manera offline, es decir, una vez que se ha obtenido el CCBHash, se almacena para su posterior uso. Esto permite que, aún siendo importante, el tiempo del cálculo del CCBHash no sea un factor totalmente limitante en nuestra herramienta.

Los hash tradicionales tienen como objetivo identificar inequívocamente a una muestra, utilizando un tamaño mucho menor que el original y con las mínimas colisiones posibles [2]. De esta manera, dos ficheros que sean exactamente iguales excepto en un bit darán lugar a dos hashes totalmente diferentes. Las funciones de fuzzy hashing tratan de conseguir lo contrario. Es decir, resumen una muestra en una firma relativamente pequeña pero, si dos ficheros únicamente cambian en un bit, los hashes generados deben ser (casi) idénticos, permitiendo así detectar similitudes a partir del hash.

Para poder encontrar similitudes en bloques de código ensamblador, hay que definir el tamaño de estos bloques o, mejor dicho, el inicio y fin de cada bloque. Dado que nuestro objetivo es almacenar los hashes resultantes en una base de datos, calcular las firmas para todos los bloques posibles sería algo inviable (tanto por espacio como por tiempo de la comparación). Por ello, CCBHash usará como bloques las propias funciones en ensamblador, quedando así acotado el número de bloques posibles.

El problema de la búsqueda de similitudes en malware es tan importante y actual que empresas como VirusTotal ofrecen servicios para ello usando diferentes métricas [3]. Por un lado, existen herramientas que buscan diferencias en segmentos de código ensamblador entre dos ficheros, como BinDiff [4] y Diaphora [5]. Por otra parte, hay numerosos estudios y soluciones que se enfocan en las similitudes entre varias muestras completas de malware. Sin embargo, las alternativas para buscar similitudes a nivel de bloque de código ensamblador, o funciones en nuestro caso, entre un número muy grande de muestras de malware escasean. Algunas de estas opciones, en concreto los fuzzy hashes como SSDEEP [6], consiguen tiempos de ejecución aceptables pero deben encontrar coincidencias exactas. Es decir, dividen una muestra en bloques más pequeños que deben ser iguales para poder detectar similitudes. A diferencia de SSDEEP, en CCBHash el hash de cada bloque de código será a su vez un fuzzy hash compuesto por atributos de este en lugar de un simple hash del mismo, permitiendo encontrar similitudes entre diferentes bloques y no solo coincidencias exactas.

Resumidamente, nuestro objetivo es combinar dos de estas ideas: utilizar tanta información del código como sea posible, como hacen BinDiff y Diaphora, y combinarla en un fuzzy hash como hace SSDEEP, con la diferencia de que cada componente del fuzzy hash será a su vez otro fuzzy hash en lugar de un hash tradicional. De esta forma, conseguiríamos un análisis de similitud lo más preciso y rápido posible sobre petabytes de ficheros.

## II. Estado del arte

Antes de comenzar con la solución propuesta, llevaremos a cabo un estudio del estado del arte sobre la búsqueda de similitudes en malware. Son muchas las técnicas utilizadas; sin embargo, teniendo en cuenta nuestro objetivo, hay varias que parecen destacar y que comentaremos más detalladamente.

### *II-A. Estrategia basada en n-grams*

Hay una gran cantidad de artículos y soluciones en el mercado que hacen uso de los n-grams para comparar distintos ficheros. Los n-grams no son más que ventanas deslizantes de tamaño variable, $n$ en este caso, donde el objetivo es almacenar todas las ventanas posibles de $n$ elementos para su posterior comparación. Estos elementos pueden ser muy diversos dependiendo del contexto y el campo donde se quieran utilizar. En cuanto a malware se refiere, estos elementos suelen ser bytes o instrucciones en ensamblador, permitiendo una comparación de más alto nivel de abstracción.

Sin embargo, muchos de los estudios existentes han dado un paso más y no tratan de usar únicamente bytes o instrucciones exactas, si no que hacen un pequeño análisis de la entrada filtrando aquellas características más relevantes. Por ejemplo, hay artículos [7] donde los elementos usados para los n-grams son los *opcodes* de las instrucciones en ensamblador, obviando los registros y operandos utilizados. Este método permite realizar una comparación centrándose en la funcionalidad del malware, haciendo que las características más variables entre distintas muestras no se tengan en cuenta.

Si bien la solución anterior puede ser interesante en muchos contextos, también presenta algunas debilidades. Una de las principales desventajas es la exactitud que requiere para poder encontrar similitudes. Lo entenderemos mejor con dos ejemplos. Como sabemos, en muchas ocasiones el orden de ejecución de un grupo de instrucciones no importa. Por ejemplo, al realizar una suma y una resta consecutiva por una constante el resultado será el mismo independientemente del orden en el que se realicen. De la misma forma, hay varios *opcodes* en código ensamblador que se utilizan para realizar las mismas operaciones, pudiendo conseguir objetivos idénticos con distintas instrucciones.

Para solucionar algunos de los problemas existentes con los n-grams de *opcodes*, hay estudios [8] que introducen el concepto de n-perms. Los n-perms no son más que una versión de los n-grams usando permutaciones. De esta forma, si hay varios n-grams idénticos almacenados, los repetidos se eliminan dejando un solo n-perm de cada clase. Además, los n-perms son insensibles al orden de ejecución, por lo que dos n-grams que contengan los mismos elementos en distinto orden serán tratados como iguales.

Usar n-perms provoca una mayor detección de similitudes para ficheros que usan los mismos *opcodes*. Sin embargo, este mayor nivel de abstracción puede dar lugar también a un mayor número de falsos positivos (FP) donde dos ficheros con funcionalidades totalmente distintas usen instrucciones similares. Llegamos de esta forma a un punto donde debemos decidir entre dos modelos, uno de ellos donde hay una detección menor pero más exacta acerca de los positivos, y otro donde la detección es mayor pero puede que la tasa de falsos positivos aumente.

### *II-B. Estrategia basada en grafos*

Los Control Flow Graphs (CFG), son grafos de flujo de control. En dichos grafos se almacena el flujo de ejecución que sigue un determinado ejecutable, generalmente a nivel de código ensamblador. Para entender mejor el grafo resultante podemos hacer uso de la herramienta IDA Pro [9], donde se usa este concepto para la representación de las muestras. En la Figura 1 observamos un pequeño ejemplo de un CFG con código ensamblador.

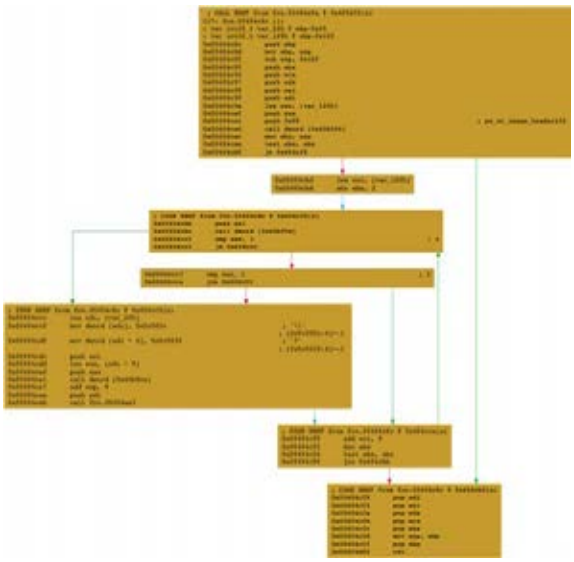


Figura 1. CFG con código ensamblador.

Como ocurre en todos los grafos, tenemos dos tipos de elementos: los nodos y las aristas, en este caso, dirigidas. Las aristas guardan la información acerca de los saltos existentes en la ejecución. De esta forma, cada salto, condicional o incondicional, se verá reflejado con una o varias aristas. En cuanto a los nodos, algunos estudios [10] prácticamente los dejan vacíos, dándole toda la importancia al flujo en sí. Sin embargo, en otros trabajos se usan dichos nodos para almacenar información mucho más diversa.

En [11] se propone el uso de atributos para caracterizar cada nodo, dando lugar al Attributed CFG (ACFG). Este enfoque trata de coger los nodos que genera una herramienta como IDA Pro y crear una firma basada en un vector de atributos para cada uno de ellos. Esta idea puede ser muy interesante si hacemos un pequeño estudio sobre qué atributos pueden caracterizar un bloque de código. Los atributos utilizados en este caso son los que se recogen en la Tabla I.

El uso de ACFG es muy interesante ya que permite almacenar información muy diversa en los nodos a la vez que permite combinar dicha estrategia con otras. Por ejemplo, en [12] se propone usar n-grams del CFG, donde los elementos sobre los que avanza la ventana son los nodos, que en este caso se caracterizan por el número de aristas entrantes y salientes, cubriendo todos los caminos posibles de $n$ nodos consecutivos siguiendo el flujo establecido. Incluso un CFG se puede usar simplemente para ordenar el código y luego tomar n-grams de *opcodes*.

Otros autores [13] tratan de combinar los CFG con las llamadas a la API del subsistema del sistema operativo, creando

Tabla I
ALGUNOS ATRIBUTOS PARA CARACTERIZAR NODOS EN LOS CFG.

| Tipo de atributo | Descripción |
|---|---|
| De la secuencia de código | # Constantes numéricas<br># Instrucciones de transferencia<br># Instrucciones de llamada<br># Instrucciones aritméticas<br># Instrucciones de comparación<br># Instrucciones de movimiento<br># Instrucciones de terminación<br># Instrucciones de declaración de datos<br># Instrucciones totales |
| De la estructura de los vértices | # Descendencia/hijos<br># Instrucciones en el vértice |

un grafo donde los nodos son caracterizados por las llamadas realizadas a la API. Este método, aunque con ventajas para ejecutables no maliciosos, presenta serios inconvenientes si tenemos en cuenta que hay malware creado específicamente para no usar llamadas a la API del subsistema.

Por último, hay trabajos como [14], [15] y [16] que también hacen uso de los Data Flow Graph (DFG) o Program Dependence Graph (PDG), donde se almacena un grafo sobre la dependencia de los datos, es decir, de los registros del procesador utilizados.

Estos enfoques, aunque presentan grandes resultados en cuanto a la búsqueda de similitud, también tienen algunas desventajas. La principal de ellas es el tiempo que requieren. Si bien la estrategia basada en n-grams presenta un coste computacional de $O(n)$, los enfoques basados en CFG aumentan hasta $O(n^3)$ [17], algo que para algunos escenarios puede ser inviable.

### *II-C. Estrategia basada en Fuzzy Hashing*

Existe una amplia cantidad de trabajos que estudia la utilidad y la precisión de las estrategias anteriormente mencionadas [18]. Sin embargo, hay otro tipo de enfoques con características muy interesantes, como son los fuzzy hashes.

Muchas herramientas conocidas emplean los fuzzy hashes para encontrar similitudes entre distintos ficheros. Sin embargo, ninguno de estos hashes es totalmente eficaz ya que depende mucho de la naturaleza de la muestra que esté siendo tratada. En VirusTotal podemos encontrar numerosas muestras donde, según el tipo de fuzzy hash utilizado, el porcentaje de similitud varía drásticamente.

Existen diferentes tipos de algoritmos de Fuzzy Hashing, entre los que destacan [19]:

- BBH (Block Based Hashing). Genera un hash para cada bloque de tamaño fijo de los datos de entrada y realiza la comparación entre los hashes de cada bloque. Cuanto más grandes sean los datos de entrada, mayor será el hash. El algoritmo dcfldd[1] by Harbour es un ejemplo de ello.
- CTPH (Context Triggered Piecewise Hashing). CTPH es muy similar a BBH pero usando trigger points en lugar de bloques de tamaño fijo. El problema del uso de trigger points es que se determinan en función del tamaño de los datos de entrada, lo que puede hacer que dos ficheros similares pero con tamaños muy diferentes tengan hashes totalmente distintos. El algoritmo SSDEEP pertenece a este grupo.
- SIF (Statistically Improbable Features). Trata de localizar un conjunto de características compuestas por secuencias de bits poco habituales, haciendo uso de la entropía, creando posteriormente un hash de dichas características. El algoritmo más conocido es sdhash [20].
- BBR (Block Based Rebuilding). Hace uso de datos externos a la propia entrada del algoritmo, que pueden ser aleatorios o constantes. Por ejemplo, a cada byte del fichero de entrada se le asigna 0xFF o 0x00 tras compararlo con sus bits vecinos. Si la mayoría de estos vecinos son 1, se convierte en 0x00, si no, en 0xFF. En general, la comparación se realiza mediante el cálculo de la distancia de Hamming. Un ejemplo de dicho grupo es el algoritmo mvHash-B [21].
- LSH (Locality Sensitive Hashing). Reduce el espacio de posibles entradas a una cantidad discreta de probabilidades, maximizando la probabilidad de que dos entradas similares generen una salida casi idéntica. Las técnicas utilizadas están relacionadas con la búsqueda de vecino más cercano.

Entre las herramientas analizadas en este campo hay dos que llaman especialmente nuestra atención, que son Machoc [22] y Machoke [23]. Ambas son muy similares en cuanto a su funcionalidad pero difieren ligeramente en la tecnología utilizada. Dichas herramientas obtienen el CFG de cada una de las funciones de un ejecutable, tal y como vemos en IDA Pro (de hecho, Machoke puede ser utilizada como plugin de esta), calculando para cada una un código hash basado en su CFG y concatenando los hashes para formar un fuzzy hash.

### *II-D. BinDiff y Diaphora*

Por último, es necesario hacer especial énfasis en dos herramientas ya mencionadas: BinDiff y Diaphora. Su objetivo es encontrar diferencias entre dos ficheros. Para ello, se centran en primer lugar en las funciones en código ensamblador. Luego, realizan una comparación entre las distintas funciones, tanto a nivel de atributos de función como de segmento de código dentro de la función. En este caso, los bloques consisten en cada uno de los nodos del CFG de la función. Por último, da un porcentaje de precisión y confianza a las similitudes encontradas, en función de los atributos que hayan coincidido en la comparación.

Estas herramientas usan atributos muy diversos para encontrar similitudes entre las funciones de dos ficheros, muchos de ellos a partir de las ideas extraídas de los apartados anteriores. Entre los atributos destacan: distintos grafos generados a partir de la propia función, el nombre de la función, las cadenas de caracteres existentes, las instrucciones de cada bloque de código, etc. El uso de tantas heurísticas hace que la comparación llevada a cabo sea bastante precisa y costosa. Sin embargo, esto puede hacerse gracias a que únicamente se están comparando dos muestras. Si el número de ficheros creciese, este enfoque sería inviable debido al tiempo que sería necesario.

Nuestra herramienta trata de conseguir un resultado similar al de BinDiff y Diaphora, pero realizando la comparación entre una cantidad enorme (con un total de almacenamiento del orden de petabytes) de funciones, que pasan a ser nuestro

[1] http://dcfldd.sourceforge.net/

componente principal de trabajo en la búsqueda de similitud. Por ello, reunirá varios de estos atributos, compactando cada atributo en un hash de tamaño acotado, que se unirán formando un sola firma para cada función. De esta forma, al saber el tamaño ocupado (o número de bytes) por cada atributo, será posible comparar cada uno de forma rápida e independiente.

## III. Solución propuesta

### *III-A. Marco teórico*

Lo primero que debemos tener en cuenta, y que hemos comentado anteriormente, es que nuestro objetivo es comparar funciones de código ensamblador entre una ingente cantidad de muestras. Es decir, para cada muestra necesitamos almacenar un código hash en el que se recoja información acerca de las diferentes funciones. La complicación se encuentra en cómo caracterizar las distintas funciones antes de utilizar algoritmos de hashing sobre ellas. Estas funciones las podemos obtener con herramientas como r2pipe de radare2 [24] o IDA Pro.

A continuación, es el momento de indicar cómo caracterizar a cada función. Lo haremos creando una firma para cada una de ellas, que estará compuesta por una serie de atributos a los que se les hará un hash (individualmente) y que serán concatenados en un único hash. Es decir, el hash de una muestra estará compuesto por hashes de funciones que a su vez estarán compuestos por hashes de atributos, que será la unidad mínima de comparación. Procedemos a seleccionar los atributos que formarán la firma de nuestra función. En primer lugar, seleccionaremos algunos atributos de más alto nivel, que traten a la función como una caja negra. Téngase en cuenta que la justificación para su selección es similar en todos ellos: valores similares o muy parecidos entre dos funciones distintas, indican comportamientos similares.

- Nombre de la función. Si bien los nombres asignados por un desensamblador no son relevantes, la existencia de información de debug en el ejecutable podría hacer que este atributo adquiera suficiente importancia.
- Tamaño de la función en bytes.
- Número de entradas y salidas. Proporciona un grado de utilización de la función dentro del malware así como de la necesidad de ésta por parte de otras.
- Número de bloques. Refleja un punto intermedio entre la cantidad de instrucciones y los saltos existentes.
- La complejidad ciclomática. Entendida como $A-N+2C$ siendo $A$ el número de aristas, $N$ el número de nodos y $C$ el número de nodos de salida.
- El propio CFG de la función. Para el cálculo del CFG se ha seguido un procedimiento similar al de [23].
- Número de variables locales.
- Número de argumentos de la función.

En segundo y último lugar, si bajamos el nivel de abstracción y nos fijamos en los bloques de código ensamblador (que observamos en IDA Pro) que componen la función, tenemos atributos como:

- Número de instrucciones.
- Tipo de instrucciones (y cantidad de cada tipo).

La elección de los atributos mencionados ha tenido en cuenta dos enfoques. El primero de ellos trata de encontrar código que actúe de forma similar, es decir, que realice operaciones similares. El segundo, y no menos importante, es detectar código que realice las operaciones de la misma forma.

Una vez que ya hemos reunido los atributos necesarios, es el momento de calcular el fuzzy hash de la función. Para ello, calcularemos un hash para cada uno de los atributos. El tamaño del hash calculado para cada atributo debe ser lo más pequeño posible a la vez que la probabilidad de colisiones sea lo suficientemente pequeña. En nuestro caso, tras diversas pruebas, el tamaño escogido ha sido de dos bytes, obteniendo finalmente un fuzzy hash de tamaño $2n$ bytes, siendo $n$ el número de atributos. En la Figura 2 podemos observar cómo sería el resultado si usáramos cinco atributos.


Figura 2. Hash de la función compuesto de hashes de los atributos.

CCBHash quedará limitado principalmente por esta elección de atributos llevada a cabo. Cuantos más atributos sean seleccionados mejor será la comparación, en detrimento de aumentar el tamaño del hash y el tiempo de comparación. Lo mismo ocurre si aumentamos el espacio asignado a cada uno de ellos. De esta forma, se ha debido llegar a un equilibrio entre calidad, espacio y rapidez.

Dado que el tamaño del hash de cada atributo es conocido, al comparar funciones de distintas muestras es posible hacerlo por atributo. De esta forma, dados los fuzzy hashes de dos funciones, podemos comparar paralelamente la parte correspondiente a cada atributo. Finalmente, sabríamos qué atributos han presentado coincidencias, pudiendo determinar un porcentaje de similitud en función del número de atributos iguales, así cómo la calidad del resultado en función de dichos atributos ya que hay heurísticas más fuertes que otras. En función del escenario donde se use este fuzzy hash, el criterio para considerar iguales dos funciones puede ser distinto. Un criterio aceptable sería que dos funciones son similares si al menos la mitad de los atributos coinciden. Aunque este valor es parametrizable y dependerá del contexto de uso de CCBHash. En la Figura 3 podemos ver un ejemplo con tres atributos.

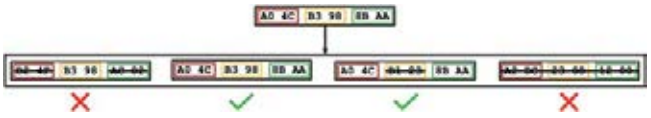


Figura 3. Buscando similitudes entre hashes de funciones.

Estableceremos, sin embargo, un sistema donde se ponderen los atributos en función de su importancia. De esta forma, heurísticas como el CFG o el tipo de instrucciones usadas tendrán más importancia que atributos como el nombre de la función o su tamaño (ya que estos son atributos fácilmente modificables y débiles en su grado de representatividad). En la Tabla II podemos ver como quedaría la puntuación según el atributo correspondiente. Una similitud perfecta daría como resultado 1 (100 %), mientras que dos funciones con ninguna

similitud tendría como resultado 0. Para nosotros, un resultado mayor que 0.7 (70 %) indicará que ambas funciones son similares. Este porcentaje se ha obtenido de manera empírica, y como explicamos en la sección IV, no es definitivo.

Tabla II
PUNTUACIONES SEGÚN ATRIBUTO.

| Atributo | Puntuación |
|---|---|
| Nombre de la función | 0.01 |
| Tamaño de la función | 0.05 |
| Número de entradas | 0.04 |
| Número de salidas | 0.04 |
| Número de variables locales | 0.04 |
| Número de argumentos de la función | 0.04 |
| Número de instrucciones | 0.10 |
| Complejidad ciclomática | 0.14 |
| Número de bloques | 0.04 |
| Control Flow Graph | 0.25 |
| Tipo de instrucciones | 0.25 |

Por último, solo queda calcular el CCBHash. Una vez que tengamos los hashes de cada función, los concatenamos y ya tenemos como resultado nuestro CCBHash. Dado que el tamaño del hash de cada función es conocido, al igual que lo era el de los atributos, se pueden comparar distintas funciones de manera simultánea. De esta manera, el CCBHash será de tamaño variable, ya que depende del número de funciones. En concreto, el tamaño de este es de $n \times f$ bytes, siendo $n$ el número de funciones y $f$ el tamaño del hash de la función, que será constante.

### *III-B. Marco práctico*

Para terminar, solo quedaría poner en práctica el diseño preliminar de nuestra herramienta. Para ello, vamos a utilizar muestras de distintas familias de malware, como DarkSide o WannaCry, para ver las similitudes que se encuentran con muestras de una misma familia y con muestras de distintas familias.

Para abordar el problema crearemos un script en Python, ayudándonos de la librería r2pipe que nos permite desensamblar una muestra y obtener información muy útil acerca de ella. Gracias a dicha herramienta, obtenemos las funciones del ejecutable, los bloques que forman cada función y las instrucciones que forman cada bloque, así como información asociada a cada uno de estos elementos. Para crear los hashes, usaremos la librería hashlib de Python [25], en concreto el algoritmo blake2b [26] ya que permite generar hashes de tamaño arbitrario rápidamente, superando en velocidad a algoritmos como MD5, SHA-1, SHA-2 o SHA-3.

En un MacBook Pro con procesador 2.9 GHz Intel Core i5, CCBHash tarda una media de 30 milisegundos por función para ejecutables de entre 10 KB y 4 MB. Sin embargo, de esos 30 ms, solo 0.5 ms equivalen a la aplicación del algoritmo blake2b y la concatenación de los hashes de cada atributo, y el resto a la obtención de las funciones y sus atributos con ayuda de la librería r2pipe.

En primer lugar, vemos los resultados con varias muestras de DarkSide. Obtenemos el fuzzy hash propuesto para las funciones de las muestras y buscamos similitudes. Atendiendo a dos muestras cualesquiera, por ejemplo las muestras con hashes 9CEE...7297 y AFB2...8178, y comparando el CCBHash obtenido, observamos que el 75 % de las funciones tiene más de un 50 % de similitud y hay un 56 % de funciones con más de un 90 % de similitud.

Probamos de nuevo con varias muestras de otra familia, en este caso WannaCry. Atendiendo nuevamente a dos muestras, con hashes 4186...D982 y 09A4...CAFA, observamos que el 52 % de las funciones tienen una similitud superior al 50 % y hay un 31 % de funciones con más de un 90 % de similitud. Repitiendo el proceso con distintas muestras de una misma familia obtenemos resultados similares, obteniendo siempre porcentajes superiores al 30 % de las funciones para similitudes del 70 %.

Por último, vemos qué ocurre con muestras de distintas familias, por ejemplo 9CEE...7297 de DarkSide y 4186...D982 de WannaCry. En este caso, obtenemos que el porcentaje de funciones con más de un 50 % de similitud es del 22 %, y si la similitud sube al 90 % el porcentaje de funciones ya baja al 4 %. En concreto, este 4 % equivale a tres funciones. Estas corresponden a funciones donde la única instrucción ejecutada sirve para llamar a la API de Windows. Podemos decir, según el criterio comentado anteriormente, que entre estas dos muestras ha habido un 4 % de falsos positivos. Nuevamente estudiando distintas muestras de familias diferentes obtenemos resultados similares, no superando el 10 % de falsos positivos (y en su mayor parte debido a funciones poco interesantes donde la única función es una llamada a la API).

En la Figura 4 vemos una gráfica del caso anterior del número de funciones que presenta un porcentaje de similitud concreto. Los altos porcentajes de similitud, superiores al 75 %, se deben a funciones casi idénticas donde únicamente cambia el nombre, número de variables locales, número de instrucciones o número de entradas y salidas. En la zona central, con similitudes en torno al 50 %, aparecen funciones donde alguno de los atributos más fuertes, como el CFG o los tipos de instrucciones, no coincide. Por último, tenemos aquellas funciones con similitudes inferiores al 25 %, donde los atributos con puntuaciones altas no coinciden y cuya similitud se debe a simples coincidencias.

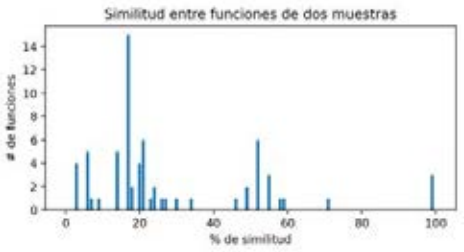



Figura 4. Similitud entre las funciones de dos muestras.

En cuanto a falsos negativos (FN), las pruebas realizadas hasta el momento han arrojado resultados prometedores: no se han encontrado falsos negativos. Es decir, no hemos encontrado muestras de la misma familia en la que no se encuentren similitudes. Como comentamos anteriormente, para nuestra herramienta es prioritario no dejar escapar ninguna similitud (por ello el uso de métricas débiles), aunque ello conlleve aceptar más falsos positivos en algunos escenarios. Es necesario, no obstante, fortalecer el diseño y realizar un número de pruebas estadísticamente relevante.

## IV. Conclusiones y trabajo futuro

Con este proyecto conseguimos una propuesta inicial para encontrar similitudes entre funciones de una cantidad ingente de muestras de malware. Esta herramienta, en su estado actual, podría utilizarse como un fuzzy hash más, complementando a los actuales y pudiendo encontrar similitudes que otros no encontrarían. Sin embargo, esta idea puede mejorarse. En especial en lo referente a la medición exhaustiva de resultados con un conjunto de muestras malware suficiente y una comparación detallada con respecto a los métodos presentados en la sección I en cuanto a los FN y FP.

En primer lugar, se está realizando un estudio sobre qué funciones son conocidamente no maliciosas o poco interesantes (eg. funciones con una o dos instrucciones como hemos observado en III-B). Con ello queremos incluir un filtro donde dichas funciones se obvien en el cálculo del CCBHash, ahorrando tiempo y espacio. Con objeto de maximizar la eficiencia, estamos estudiando distintas posibilidades con las que extraer los atributos de las funciones, ya que la librería r2pipe consumía la mayor parte del tiempo en el cálculo de nuestro hash.

También hemos comenzado un proceso de elección de atributos más potentes. Si bien los atributos usados actualmente detectan con éxito similitud entre funciones de malware, creemos necesario seguir investigando en este campo para encontrar aquellos atributos óptimos que encuentren cualquier tipo de similitud.

En relación a lo anterior, actualmente el hash creado tiene una longitud fija, donde cada característica de la función ocupa un tamaño fijo. En el futuro nos gustaría mejorar esta propuesta haciendo que cada atributo ocupe un tamaño que pueda diferir del resto de atributos. Incluso almacenando para cada atributo distintas formas de representación, como un valor numérico en función de un rango en lugar de un simple hash. Esto haría nuestro fuzzy hash más flexible, pero no debemos perder el contexto de ejecución de CCBHash (petabytes de funciones).

Creemos así que la herramienta desarrollada tiene gran utilidad en la detección de similitudes en malware, aportando características que no es posible encontrar con las opciones disponibles actualmente. Además, demuestra tener gran potencial y que, ahondando en su investigación, daría lugar a un nuevo fuzzy hash que podría no solo servir de complemento a otras herramientas, sino llegar a sustituir a algunas de las actuales.

## Agradecimientos

Este trabajo ha sido realizado gracias a la colaboración de la Universidad de Málaga (UMA) y VirusTotal bajo el amparo del proyecto SAVE cofinanciado por la Junta de Andalucía (PAIDI 2020) y el Fondo Europeo de Desarrollo Regional (FEDER).

## Referencias

[1] I. Abadía, "Evaluación de algoritmos de fuzzy hashing para similitud entre procesos," 2017, trabajo de Fin de Grado. [Online]. Available: http://webdiis.unizar.es/~ricardo/files/PFCs-TFGs/Fuzzy-Hashing-Procesos/Fuzzy-Hashing-Procesos.pdf

[2] M. Singh and D. Garg, "Choosing Best Hashing Strategies and Hash Functions," in *2009 IEEE International Advance Computing Conference*. IEEE, mar 2009, pp. 50–55. [Online]. Available: http://ieeexplore.ieee.org/document/4808979/

[3] V. Díaz, "Why is similarity so relevant when investigating attacks." [Online]. Available: https://blog.virustotal.com/2020/11/why-is-similarity-so-relevant-when.html

[4] "BinDiff de Zynamics." [Online]. Available: https://www.zynamics.com/bindiff.html

[5] "Repositorio de diaphora." [Online]. Available: https://github.com/joxeankoret/diaphora

[6] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digital Investigation*, vol. 3, pp. 91–97, 2006, the Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742287606000764

[7] L. Liu, B.-s. Wang, B. Yu, and Q.-x. Zhong, "Automatic malware classification and new malware detection using machine learning," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 9, pp. 1336–1347, sep 2017. [Online]. Available: http://link.springer.com/10.1631/FITEE.1601325

[8] A. Walenstein, M. Venable, M. Hayes, C. Thompson, and A. Lakhotia, "A.: Exploiting similarity between variants to defeat malware: "vilo" method for comparing and searching binary programs," in *In: Proceedings of BlackHat DC 2007. (2007) https: //blackhat.com/presentations/bh-dc-07/Walenstein/Paper/bh-dc-07-walenstein-WP.pdf.*

[9] Hex-Rays, "Ida pro." [Online]. Available: https://hex-rays.com/ida-pro

[10] G. Bonfante, M. Kaczmarek, and J. yves Marion, "Control flow graphs as malware signatures." [Online]. Available: https://hal.inria.fr/inria-00176235/document

[11] J. Yan, G. Yan, and D. Jin, "Classifying Malware Represented as Control Flow Graphs using Deep Graph Convolutional Neural Network," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, jun 2019, pp. 52–63. [Online]. Available: https://ieeexplore.ieee.org/document/8809504/

[12] Y. Li, J. Jang, and X. Ou, "Topology-Aware Hashing for Effective Control Flow Graph Similarity Analysis," no. December, pp. 1–6, apr 2020. [Online]. Available: http://arxiv.org/abs/2004.06563

[13] J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi, "Static detection of malicious code in executable programs," *Int. J. of Req. Eng*, 2001.

[14] Mayrand, Leblanc, and Merlo, "Experiment on the automatic detection of function clones in a software system using metrics," in *1996 Proceedings of International Conference on Software Maintenance*, 1996, pp. 244–253. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=\&arnumber=565012

[15] J. Kim, H. Choi, H. Yun, and B.-R. Moon, "Measuring Source Code Similarity by Finding Similar Subgraph with an Incremental Genetic Algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. New York, NY, USA: ACM, jul 2016, pp. 925–932. [Online]. Available: https://dl.acm.org/doi/10.1145/2908812.2908870

[16] J. Krinke, "Identifying similar code with program dependence graphs," in *Proceedings Eighth Working Conference on Reverse Engineering*. IEEE Comput. Soc, pp. 301–309. [Online]. Available: http://ieeexplore.ieee.org/document/957835/

[17] J. Liu, Y. Wang, and Y. Wang, "The Similarity Analysis of Malicious Software," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*. IEEE, jun 2016, pp. 161–168. [Online]. Available: http://ieeexplore.ieee.org/document/7866123/

[18] W. Song, "a framework for automated similarity analysis of malware," Master's thesis, Concordia University, September 2014, unpublished. [Online]. Available: https://spectrum.library.concordia.ca/id/eprint/978935/

[19] A. Lee and T. Atkison, "A Comparison of Fuzzy Hashes," in *Proceedings of the SouthEast Conference*. New York, NY, USA: ACM, apr 2017, pp. 18–25. [Online]. Available: https://dl.acm.org/doi/10.1145/3077286.3077289

[20] V. Roussev, "Data fingerprinting with similarity digests," in *Advances in Digital Forensics VI*, K.-P. Chow and S. Shenoi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 207–226.

[21] F. Breitinger, K. P. Astebøl, H. Baier, and C. Busch, "mvhash-b - a new approach for similarity preserving hashing," in *2013 Seventh International Conference on IT Security Incident Management and IT Forensics*, 2013, pp. 33–44.

[22] [Online]. Available: https://github.com/ANSSI-FR/polichombr/blob/dev/docs/MACHOC_HASH.md

[23] [Online]. Available: https://www.conix.fr/machoke-hashing/

[24] [Online]. Available: https://www.radare.org/n/r2pipe.html

[25] "Documentación de la librería hashlib de python." [Online]. Available: https://docs.python.org/3/library/hashlib.html

[26] [Online]. Available: https://www.blake2.net/

# Sistema de gestión de certificados digitales COVID-19 basado en blockchain

Rosa Pericàs-Gornals
Universitat Illes Balears
Crta. Valldemossa 7.5, Palma
rosa.pericas@uib.es

M. Magdalena Payeras-Capellà
Universitat Illes Balears
Crta. Valldemossa 7.5, Palma
mpayeras@uib.cat

Macià Mut-Puigserver
Universitat Illes Balears
Crta.Valldemossa 7.5, Palma
macia.mut@uib.cat

Llorenç Huguet-Rotger
Universitat Illes Balears
Crta.Valldemossa 7.5, Palma
l.huguet@uib.cat

***Resumen*—Como resultado a la declaración de pandemia por COVID-19, se han presentado diferentes propuestas de certificados Digitales COVID-19 basados en el uso de la blockchain. Teniendo en cuenta que los datos sanitarios tienen altos riesgos de privacidad, un sistema de gestión de datos sanitarios debe proporcionar altos requisitos de privacidad y seguridad. Por una parte se debe asegurar la confidencialidad de los datos médicos, siendo el usuario propietario de los datos el único actor que tenga el control total sobre estos. Por otra parte, las entidades involucradas en la generación y validación de certificados deben estar supervisadas por una autoridad reguladora. Además, el sistema debe proporcionar un proceso de verificación sencillo y que al mismo tiempo evite el riesgo de falsificación de certificados. En este artículo se presenta un protocolo para la gestión de certificados Digitales COVID-19 donde los usuarios deciden con quien desean compartir sus datos privados en un sistema jerárquico, además de proporcionar datos de su implementación y resultados obtenidos.**

***Index Terms*—Datos sanitarios, Blockchain, Privacidad, Reencriptación, Autosoberanía.**

## I. Introducción

La COVID-19 es la enfermedad causada por el coronavirus, conocido como SARS-CoV2, un tipo de virus que causa una variedad de dolencias que en los casos más severos pueden llegar a causar la muerte de las personas que lo padecen. La alta transmisibilidad del virus ha llevado a la necesidad de introducir fuertes restricciones globales, como confinamientos estrictos de la población para reducir los efectos y evitar el colapse hospitalario. Para poder reducir los efectos de la COVID-19 lo más rápido posible, al inicio de la pandemia hubo la necesidad de introducir una fuerte vigilancia de la población, registrando cuando una persona ha sido infectada por COVID-19 o si se ha vacunado. Entonces, se propuso el uso de certificados Digitales COVID-19, en los cuales se registran todos estos casos [1].

Cabe tener en cuenta que para conseguir los resultados esperados con dichos certificados el sistema de generación y mantenimiento debe imposibilitar la falsificación de certificados. Característica que justo después de la implantación de dichos certificados ya fue violada en diferentes países donde el uso se impuso como obligatorio [2], [3], [4]. Por esta razón, se deben introducir el mayor número de medidas de seguridad posibles para evitar la falsificación, como restringir el número de personas que pueden generar dichos certificados o añadir el uso de técnicas que garanticen la inmutabilidad de los datos.

En el protocolo diseñado se tiene muy presente la necesidad de evitar completamente la falsificación gracias al uso de la tecnología blockchain que introduce la descentralización del sistema y la inmutabilidad de los datos compartidos, además de habilitar la transparencia de los datos. También para evitar la falsificación se añade el uso de una autoridad reguladora, que introduce un control sobre las diferentes partes que forman el sistema.

Los últimos años la tasa actual de violación de datos ha aumentado llegando a la necesidad de crear reglamentos de protección de la privacidad como el Reglamento General de Protección de Datos de Europa (RGPD). Los certificados Digitales COVID-19 tratados en el sistema, al contener datos de salud, son considerados datos sensibles, por lo que requieren almacenamiento y compartición de forma segura y preservando la privacidad, para habilitar la imposición de penalidades severas por la violación de dichas leyes [5].

Para conseguir la confidencialidad necesaria de los datos sanitarios en el protocolo se ha introducido el uso de un servicio de Proxy de Reencriptación umbral (PRE), que proporciona las funcionalidades para el cifrado de los datos transmitidos entre el usuario propietario y las entidades solicitantes. Utilizando criptografía asimétrica de curva elíptica, el servicio PRE proporciona la transformación de un texto cifrado con la clave pública de Alice en un texto cifrado que podrá ser abierto únicamente usando la clave privada de Bob, sin la necesidad de intercambiar ninguna clave privada entre los usuarios, obteniendo una gestión autosoberana de los datos, con lo que los propietarios consiguen el control total de su información [6].

Hoy en día ya existen multitud de propuestas relacionadas con Certificados Digitales COVID-19, como en [7] donde se propone un Pasaporte Digital Sanitario, haciendo uso de una blockchain privada, juntamente con el uso de la prueba de autoridad. Así como señalan los autores, el sistema no proporciona un anonimato total de los usuarios finales, una alta seguridad contra la falsificación, ni permite a los miembros de la blockchain examinar los datos sin un permiso de acceso.

Existen algunas propuestas que siguen los estándares y las regulaciones actuales, como el modelo VacciFi [8] o los protocolos presentados en [9] y [10]. En el primero se sigue el Reglamento General de Protección de Datos y en el segundo y tercer caso el estándar "Verifiable Credentials" de la W3C.

Finalmente mencionar que también hay algunos protocolos que proponen el uso del sistema IPFS como [11] donde se presenta una solución para la gestión de resultados de pruebas de Covid-19 y vacunación utilizando blockchain juntamente con el sistema IPFS, dónde las entidades verificadoras sólo comprueban la firma digital del propietario sobre los datos de los resultados. En el protocolo propuesto por Hasan et al. en [12] se implementa un sistema de gestión de pasaportes médi-

cos digitales y certificados de inmunidad usando el sistema IPFS juntamente con un esquema de proxy de reencriptación, sin tener en cuenta la necesidad de añadir un control sobre las entidades de verificación de los certificados digitales.

Tal y como se ha presentado hoy en día hay diferentes propuestas para la gestión del COVID-19, que hacen uso de tecnologías seguras y innovadoras, pero a menudo no se tienen en consideración los altos requisitos de privacidad y la soberanía de los datos por parte del propietario, características que en el protocolo presentado se consideran esenciales.

Concretamente la contribución del protocolo presentado, respecto al estado del arte actual se basa en:

- Supervisión de los emisores y verificadores de los certificados COVID-19 digitales, evitando la falsificación.
- Completo cifrado de los datos contenidos en los certificados Digitales COVID-19.
- Soberanía de los datos por parte única del propietario.
- Fácil verificación de los certificados Digitales COVID-19 por entidades fiables y posibilidad de ejecutar la verificación por parte de entidades no fiables o usuarios.

El protocolo de gestión de certificados digitales COVID-19 presentado ha sido diseñado con el objetivo de tratar un tipo concreto de certificados de salud, pero este mismo protocolo fácilmente puede extenderse para la gestión de cualquier tipo de dato sanitario, ya que todo dato sanitario al ser considerado dato sensible tiene los mismos requisitos y necesidades.

## II. Protocolo

A continuación se presenta una breve introducción al protocolo diseñado, introduciendo las tecnologías utilizadas que proporcionan al protocolo sus características principales seguido de la presentación del conjunto de actores que pueden intervenir en las diferentes fases que forman el protocolo, una visión general de las diferentes funcionalidades del protocolo y finalmente una introducción a la implementación llevada a cabo, la cual se puede encontrar en nuestro repositorio de Github[1].

### *II-A. Tecnologías*

El protocolo para la gestión de certificados digitales COVID-19 se caracteriza por presentar cinco características principales: descentralización, inmutabilidad, confidencialidad, privacidad y auto-soberanía por parte del usuario propietario de los datos sanitarios. Dichas funcionalidades se obtienen gracias a la combinación de tres tecnologías innovadoras: blockchain, el sistema de almacenamiento distribuido IPFS y un servicio umbral de Proxy de Re-encriptación (PRE).

La blockchain, también conocida por cadena de bloques, es un registro digital, descentralizado y distribuido, que hace uso de transacciones firmadas criptográficamente agrupadas en bloques. La implementación del protocolo llevada a cabo utiliza concretamente la blockchain de Ethereum, la cual a partir de su implementación de la Máquina Virtual de Ethereum (EVM) permite la escritura y ejecución de código sobre la blockchain, a través del uso de Smart Contrats.

El sistema de almacenamiento InterPlanetary File System (IPFS) proporciona un sistema distribuido entre pares, donde los usuarios pueden almacenar cualquier tipo de contenido que será compartido con todos los usuarios de la red, obteniendo un sistema resistente a la censura. En la implementación este sistema es usado para realizar el almacenamiento de los certificados Digitales COVID-19, obteniendo un almacenaje distribuido y accesible permanentemente, gracias al servicio de *pinning* que evita la eliminación del contenido por parte del garbage collector.

Finalmente, para obtener la correcta confidencialidad y privacidad de los datos sensibles que forman los certificados Digitales COVID-19, se presenta el uso de un servicio de Proxy de Re-encriptación (PRE), obteniendo la funcionalidad de compartición de datos cifrados con terceras partes, sin la necesidad de descifrar los datos y volverlos a cifrar con las claves criptográficas de las terceras partes involucradas. Concretamente, se ha utilizado el servicio presentado por Nucypher [13], que proporciona las funcionalidades de cifrado utilizando criptografía asimétrica de curva elíptica. Un PRE utilizando la funcionalidad umbral proporciona una mayor seguridad al sistema, ya que un servicio PRE con un único proxy puede ser desafiado por ataques de colusión. Entonces, el PRE umbral proporciona una mayor seguridad al ser distribuido entre diferentes proxies, además de proporcionar una mayor descentralización del protocolo. Para facilitar la comprensión, en la implementación llevada a cabo se ha utilizado un único proxy, pero este puede ser fácilmente extendido a una distribución de proxies.

### *II-B. Actores involucrados*

El protocolo diseñado presenta la posibilidad de involucrar cinco actores diferentes en las diferentes fases que lo forman.

- **Autoridad Reguladora**: entidad encargada de la validación de los laboratorios que desean formar parte del sistema. Se encarga de la gestión de las entidades fiables que forman parte del sistema. En la implementación realizada es representada por la Organización Mundial de la Salud (OMS).
- **Laboratorio o centro de pruebas**: autoridad con el permiso adecuado para generar nuevos certificados digitales COVID-19 para sus pacientes introduciendo los resultados de las pruebas de COVID-19.
- **Usuario**: cliente de un laboratorio que solicita un certificado digital COVID-19. También tiene la funcionalidad de compartir sus certificados con entidades verificadoras o otros usuarios del sistema.
- **Entidad fiable**: entidad que realiza una solicitud a la autoridad reguladora para ser una entidad verificada, característica que proporciona la funcionalidad de verificación de un elevado número de certificados digitales, sin tener una fase intermedia en cada verificación de control por parte de la autoridad reguladora.
- **Entidad no fiable**: representa pequeñas organizaciones que desean proporcionar servicios seguros a sus clientes, pero no necesitan realizar una verificación rápida ni verificar un elevado número de certificados. Algunos ejemplos de este tipo de entidades no fiables son pequeñas escuelas de música, clubes de deporte, pequeños restaurantes,...

[1] https://github.com/secomuib/HighlyPrivateManagementSystemForDigitalCOVID-19Certificates

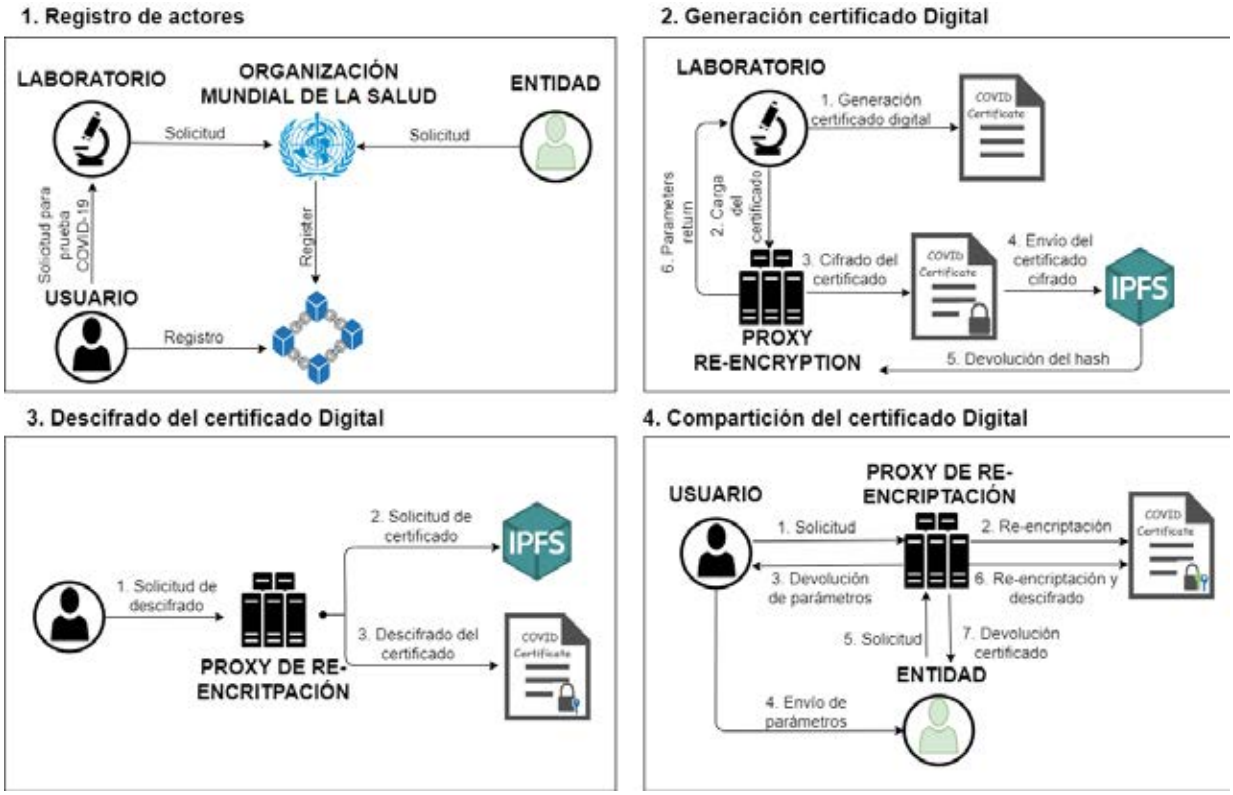



Figura 1. Diagrama de las operaciones principales del protocolo

### *II-C. Visión general del protocolo*

El protocolo presenta cuatro operaciones principales, representadas en la Figura 1 y detalladas a continuación. En la explicación se relacionan las diferentes acciones que forman cada operación con la representación de la Figura, indicando entre paréntesis el número de la acción.

1. **Registro de actores**. Los diferentes actores del sistema deben ejecutar su fase de registro para obtener su adición en el sistema. Por lo cual se pueden diferenciar subprotocolos para la gestión de laboratorios, de usuarios y de entidades fiables, los cuales permiten realizar el alta y la baja de estos tres tipos de actores en el sistema. El registro de laboratorios y entidades fiables debe ser realizado por la autoridad reguladora (la OMS), ya que debe verificar que el solicitante cumple todos los requisitos necesarios para el alta en el sistema, por lo cual externamente al sistema las organizaciones interesadas deben realizar las correspondientes solicitudes. En cambio, los usuarios realizan su registro directamente solicitando su entrada en el sistema a partir de una solicitud a través del propio sistema a la autoridad reguladora, con la cual se realizará el despliegue de su smart contract de usuario. Cabe destacar que las organizaciones que deseen acreditar para obtener el atributo de entidad fiable, previamente deben encontrarse registradas como usuarios del sistema.
2. **Generación de un certificado Digital**. Los actores usuario, externamente al sistema, solicitan una prueba COVID-19 a uno de los laboratorios asociados. Seguidamente, el laboratorio genera el certificado Digital COVID-19 con los resultados obtenidos (1) y envía dicha información al PRE (2). El certificado será cifrado por el servicio PRE (3) y seguidamente almacenado en el sistema de almacenamiento distribuido IPFS (4, 5). Finalmente, el laboratorio proporciona todos los parámetros del certificado cifrado al usuario propietario, para que este pueda solicitar el descifrado y la compartición para la verificación al servicio PRE (6).
3. **Descifrado de un certificado Digital**. El usuario propietario puede obtener su certificado digital COVID-19 en cualquier momento solicitando al servicio PRE la descarga de este desde el sistema IPFS (1, 2) y el correspondiente descifrado (3), proporcionando al PRE el CID del certificado (hash criptagráfico de los datos, el cual sirve para identificar los datos en IPFS) y su clave privada.
4. **Compartición de un certificado Digital**. En la compartición de un certificado Digital COVID-19 se pueden diferenciar dos fases: reencriptación del certificado y descifrado de un certificado Digital de un usuario externo para su verificación. El usuario propietario puede compartir su certificado Digital COVID-19 con diferentes entidades fiables u otros usuarios del sistema para su verificación, solicitando la reencriptación por parte del servicio PRE (1, 2, 3). Entonces, el usuario propietario debe enviar los parámetros de cifrado del certificado juntamente con la clave pública del usuario o entidad fiable receptora al PRE. El que responderá con los parámetros necesarios para el descifrado, que deben ser enviados a la tercera parte (4), esta únicamente debe realizar una solicitud de descifrado de los datos reencriptados al PRE (5). Primero el PRE realiza la descarga del certificado digital almacenado en el sistema

IPFS, seguidamente debe terminar la reencriptación de la clave de cifrado del certificado del usuario externo introduciendo la información privada de la tercera parte para poder proceder al correspondiente descifrado (6). Finalmente, el PRE realiza el descifrado del certificado y en caso exitoso lo proporciona al usuario o entidad fiable que había realizado la solicitud (7).

Así como ha sido indicado anteriormente, además de las entidades fiables, los usuarios del sistema (y entidades no fiables) pueden verificar certificados de otros usuarios del sistema. Principalmente se propone esta funcionalidad ya que se considera que posiblemente pequeñas organizaciones deseen poder ofrecer servicios seguros a sus clientes. De tal forma que sin la necesidad de realizar una solicitud compleja para formar parte del grupo de entidades fiables, estas pequeñas organizaciones pueden realizar solicitudes de verificación de los Certificados Digitales COVID-19 a sus clientes. Sin embargo, para añadir el control necesario, en estos casos siempre previamente a la compartición del certificado la autoridad reguladora realizará una verificación de la solicitud.

### *II-D. Implementación*

Por lo que hace la implementación del protocolo se debe tener en cuenta que todos los actores poseen direcciones de blockchain para poder comunicarse con los smart contracts definidos.

Concretamente, la autoridad reguladora contiene un smart contract que le permite realizar todas sus funcionalidades de gestión del sistema, seguidamente hay un smart contract para los laboratorios, proporcionando las pertinentes funcionalidades de generación de Certificados Digitales COVID-19. Y finalmente, un smart contract que comparten tanto las entidades fiables como los usuarios, el cual permite realizar las funcionalidades de usuario, además de la verificación de Certificados Digitales COVID-19 de usuarios externos utilizado por las entidades fiables y no fiables.

La figura 2 representa los tres smart contracts definidos con las posibles comunicaciones entre ellos y el servicio PRE definido en el protocolo. Las líneas continuas representan la relación entre el actor propietario del smart contract y su smart contract, las líneas punteadas de color rojo representan las diferentes comunicaciones entre los smart contracts, las cuales pueden ser bidireccionales o unidireccionales. Particularmente, el smart contract de la autoridad reguladora puede comunicarse con los diferentes smart contracts de laboratorio desplegados para realizar la baja de dicho laboratorio y también con los smart contracts de usuario, para realizar la gestión de entidades fiables. El smart contract de laboratorio únicamente puede enviar información a los usuarios para enviar los parámetros que permiten obtener el certificado cifrado propiedad del usuario. Finalmente, el smart contract de usuario puede comunicarse con la autoridad reguladora para realizar su registro en el sistema.

La línea punteada negra representa la comunicación entre el servicio PRE y el sistema de almacenamiento distribuido IPFS. Dicha comunicación cubre la carga y descarga de los certificados digitales COVID-19. Las líneas punteadas verdes representan la comunicación entre un laboratorio y el servicio PRE para realizar la solicitud de cifrado de un nuevo certificado y la comunicación entre un usuario propietario de un certificado y el servicio PRE, para realizar la reencriptación de dicho certificado y permitir el acceso a una entidad externa.

Finalmente, las líneas azules, representan la solicitud por parte de una entidad fiable para realizar la verificación de un certificado externo, así como la interacción con el servicio PRE para terminar la re-encriptación de dicho certificado que la entidad desea verificar.

## III. Propiedades

En esta sección se describirán brevemente un total de diez propiedades principales que definen el protocolo de gestión de Certificados Digitales COVID-19.

### *III-A. Disponibilidad*

Los certificados Digitales COVID-19 se encuentran almacenados cifrados en el sistema IPFS, donde son accesibles por cualquier usuario, pero únicamente podrán ser descifrados en caso que el usuario solicitante tenga permiso de acceso. Para obtener el almacenaje permanente se ha utilizado el servicio de pinning proporcionado por un nodo de Infura. Este almacenaje permanente se puede mejorar utilizando un nodo propio que forme parte de un grupo de nodos que comparten y almacenan sus datos, o utilizando servicios de nivel superior como es el caso de FileCoin.

### *III-B. Integridad*

Se garantiza la integridad de los certificados digitales, gracias a la inmutabilidad que proporciona la blockchain. Al almacenar el hash criptográfico del certificado que identifica los datos cifrados en el sistema IPFS, podemos detectar fácilmente si los datos han sido modificados, ya que el hash almacenado no coincidirá con el hash de los datos, ya que este habrá sido modificado. Además, si los parámetros de cifrado del certificado se modifican, el servicio PRE no será capaz de descifrar el certificado Digital COVID-19.

### *III-C. Confidencialidad*

El contenido de los certificados digitales COVID-19, aunque estos se almacenen en un sistema de almacenaje público como IPFS, al estar cifrados no son accesibles por el público general. Por otra parte, los parámetros para el descifrado se almacenan en el smart contract del usuario propietario, pero sin la clave privada del usuario no se puede acceder al descifrado de los datos.

### *III-D. Autenticación*

El protocolo proporciona autenticación de los datos gracias al hecho que la autoridad reguladora, en su smart contract, contiene toda la información de los usuarios registrados en el sistema. Además, en la solicitud para acreditar como entidad fiable, la autoridad reguladora debe solicitar la información necesaria para tener adecuadamente identificada la entidad, y debe verificar que en ningún caso se trata de una suplantación.

Gracias al uso de la tecnología blockchain, la dirección de Ethereum que solicita la transacción es almacenada, por lo que nos da una mayor facilidad para saber que entidad o usuario es el solicitante.

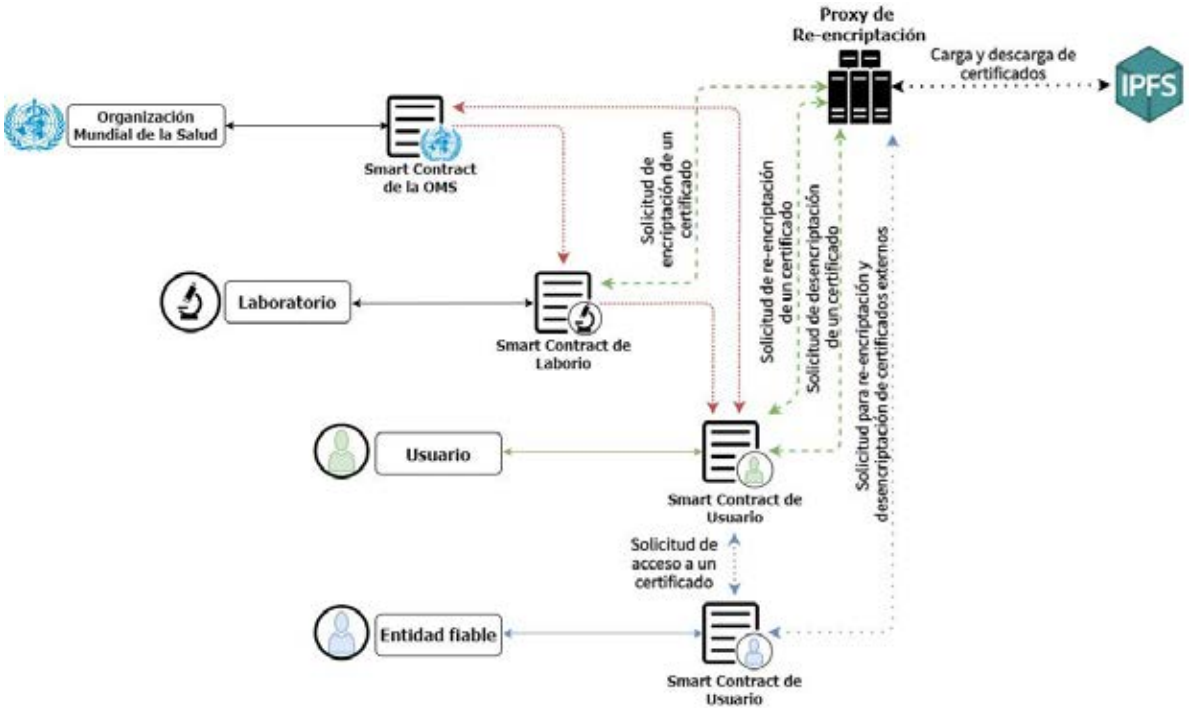



Figura 2. Diagrama de las posibles comunicaciones entre los Smart Contracts definidos en la implementación.

### *III-E. No Repudio*

El uso de un smart contract para cada usuario registrado en el sistema proporciona al protocolo la propiedad de no repudio de origen ni de destino, por el hecho que cada transacción se almacena públicamente en la blockchain juntamente con las direcciones del emisor y el receptor.

### *III-F. Auto-soberanía*

Los usuarios propietarios de los certificados Digitales COVID-19 tienen el control total sobre estos, los propietarios deciden con quien quieren compartir sus certificados.

Además en el código de los smart contract ha sido utilizado el modificador *onlyOwner* que restringe la ejecución de determinadas funciones al usuario propietario del smart contract. Por lo tanto, únicamente el usuario con la dirección de Ethereum apropiada puede firmar las transacciones para compartir sus certificados. También, en algunas funciones se restringe la ejecución por parte de un actor especifico del sistema, mediante el uso de los modificadores definidos *onlyUser*, *onlyLab* y *onlyWHO*.

### *III-G. Inmutabilidad*

Los bloques minados en la blockchain son identificados y enlazados con el hash de los datos que contienen. Por lo tanto si los datos contenidos en un bloque se manipulan o modifican estos cambiaran el hash del bloque y invalidarían la cadena. También en el sistema de almacenamiento IPFS, al utilizar el CID, si los datos de un certificado se modifican el CID consecuentemente cambiará, entonces este nuevo CID no se corresponderá con el almacenado en el registro de la blockchain.

### *III-H. Auditabilidad*

Gracias a las características de la blockchain pública: persistencia, inmutabilidad y el uso de marcas de tiempo, la existencia de un certificado específico puede ser verificado en cualquier momento por cualquier usuario, independientemente de si forma parte o no del sistema.

### *III-I. Autorización*

Los certificados Digitales COVID-19 pueden ser accedidos únicamente bajo la autorización del propietario. Los usuarios que quieren verificar un certificado Digital COVID-19 únicamente pueden acceder a él si la autoridad reguladora primero ha aceptado la solicitud de compartición del certificado, y seguidamente el usuario propietario lo ha aceptado, generando las claves de re-encriptación. Por otra parte, las entidades fiables sólo pueden acceder a un certificado digital COVID-19 si se encuentran registradas como tal y el usuario propietario del certificado acepta su solicitud.

### *III-J. Trazabilidad*

Todas las acciones definidas en el protocolo son realizadas a través de transacciones con smart contracts, que contienen las diferentes direcciones Ethereum que se encuentran involucradas, la acción realizada y los parámetros no confidenciales que se transmiten. Información muy útil en circunstancias donde puede ser necesario rastrear en cualquier momento las acciones realizadas juntamente con los actores involucrados.

## IV. Costes

Una vez desarrollada la implementación presentada, para poder realizar un análisis de la eficiencia y la efectividad del protocolo se realizó un análisis de los costes de las funciones implementadas. Concretamente utilizando la red de test Rinkeby de Ethereum. En la tabla I se presentan los costes obtenidos en las funciones principales del protocolo.

De la Tabla I podemos determinar que el despliegue del smart contract de la autoridad reguladora es la transacción con un coste mayor, seguido del registro de usuarios en el sistema,

| Función | Gas (weis) | USD (1Gwei) | USD (10Gwei) |
|---|---|---|---|
| **Autoridad Reguladora** | 4913478 | 9,04 | 90,42 |
| **Registro de usuario** | 2687489 | 4,95 | 49,45 |
| **Alta de un laboratorio** | 416236 | 0,77 | 7,66 |
| **Baja de un laboratorio** | 24834 | 0,05 | 0,47 |
| **Generación de un certificado** | 312792 | 0,58 | 5,76 |
| **Gestión de entidades** | 36099 | 0,07 | 0,66 |
| **Verificación de certificados** | 868014 | 1,6 | 15,97 |

Tabla I
TABLA DE LOS COSTES DE EJECUCIÓN DE LAS FUNCIONES PRINCIPALES DE LA IMPLEMENTACIÓN, CON UN PRECIO DE ETHER DE $1840,36.

función que además de realizar dicho registro también realiza el despliegue del smart contract de usuario. Coste producido por la necesidad de introducir todo el bytecode que forma los smart contracts en la EVM.

Por el hecho que los laboratorios tienen un número menor de funcionalidades, el despliegue de smart contracts de laboratorios resulta en unos costes menores respecto a los costes de los usuarios y la autoridad reguladora.

Las funciones de gestión de los certificados digitales tienen costes mucho menores que las que conllevan el despliegue de smart contracts. Las funciones de alta y baja de entidades fiables son las dos funciones con coste más bajo de toda la implementación, ya que únicamente requieren la publicación de pocos parámetros en la blockchain.

Finalmente, las funciones de generación de un certificado y la verificación de este, tienen un coste mayor, ya que se introduce un elevado número de parámetros en los smart contracts desplegados.

Como se puede comprobar las funciones con mayores costes son las que ejecutan el despliegue del sistema, pero una vez estas han sido ejecutadas, el resto de funciones tienen unos costes mucho menores, por lo que se puede considerar que el protocolo y la implementación realizada son relativamente efectivos en términos de coste.

## V. CONCLUSIONES

En este trabajo hemos presentado un protocolo basado en blockchain para la gestión de certificados Digitales COVID-19, utilizando un servicio de proxy de reencriptación, que proporciona alta privacidad, autenticidad y auto-soberanía de los datos por parte de los propietarios.

Las partes más importantes del protocolo diseñado son el uso del servicio de proxy de reencriptación para conseguir la confidencialidad de los datos y la auto-soberanía. El uso del sistema de almacenamiento distribuido IPFS para almacenar los certificados cifrados, y una autoridad reguladora encargada de la seguridad del sistema verificando los emisores y verificadores de los certificados digitales COVID-19. Finalmente, la tecnología Blockchain, así como se ha introducido en la sección de propiedades, permite que el protocolo obtenga altos requisitos de seguridad y privacidad, difíciles de conseguir con el uso de otras tecnologías. Pero al mismo tiempo es importante tener en cuenta que para conseguir las propiedades proporcionadas por la blockchain, los usuarios deben instalar una wallet en sus dispositivos, para poder realizar las transacciones.

El protocolo presentado está enfocado en la gestión de certificados Digitales COVID-19, pero con las mismas funcionalidades presentadas se puede direccionar para la gestión de cualquier dato médico o sensible.

Como trabajo futuro se pretende implementar una red de proxies de reencriptación distribuidos, además de dividir el código de dicho servicio para que los métodos que no necesitan utilizar datos privados del usuario se ejecuten externamente. Introducir en el sistema el uso de códigos QR para proporcionar una mayor facilidad de uso a los usuarios finales. Y finalmente, introducir el uso del servicio Smart Vault para obtener una mayor seguridad en el almacenamiento de los certificados en IPFS, el cual añade a través de smart contracts una ACL para limitar el acceso a los datos.

## AGRADECIMIENTOS

Este proyecto (RTI2018-097763-B-I00.) está financiado por: FEDER/Ministerio de Ciencia e Innovación–Agencia Estatal de Investigación

## REFERENCIAS

[1] EU Digital COVID Certificate, [Online]. Available: https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate

[2] Nik Martin, “Germany sees increase in fake vaccination certificates”, Deutsche Welle, November 27, 2021. [Online]. Available: https://www.dw.com/en/germany-sees-increase-in-fake-vaccination-certificates/a-59954734

[3] AP, “French authorities open 400 investigations into fake COVID-19 health passes”, Euronews., December 13, 2021. [Online]. Available: https://www.euronews.com/2021/12/13/french-authorities-open-400-investigations-into-fake-covid-19-health-passes#

[4] Krassen Nikolov and Zelijko Trkanjec, “Fake COVID passports flourish in southeastern Europe”, December 1, 2021. [Online]. Available: https://www.euractiv.com/section/health-consumers/news/fake-covid-passports-flourish-in-southeastern-europe/

[5] H. Jin, Y. Luo, P. Li, and J. Mathew, “A review of secure and privacy-preserving medical data sharing”, IEEE Access, vol. 7, pp. 61 656–61 669, 2019.

[6] D. Nuñez, I. Agudo, and J. Lopez, “Proxy re-encryption: Analysis of constructions and its application to secure access delegation,” Journal of Network and Computer Applications, vol. 87, pp. 193–209, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517301078

[7] C. M. Angelopoulos and V. Katos, “DHP framework: Digital health passports using blockchain. Use case on international tourism during the COVID-19 pandemic,” 2020, arXiv:2005.08922. [Online]. Available: https://arxiv.org/abs/2005.08922

[8] A. B. Haque, B. Naqvi, A. K. M. N. Islam, and S. Hyrynsalmi, “Towards a GDPR-Compliant Blockchain-Based COVID Vaccination Passport,” Applied Sciences, vol. 11, no. 13, p. 6132, Jul. 2021, doi: 10.3390/app11136132.

[9] M. Eisenstadt, M. Ramachandran, N. Chowdhury, A. Third, and J. Domingue, “Covid-19 antibody test/vaccination certification: There’s an app for that”, IEEE Open Journal of Engineering in Medicine and Biology, vol. 1, pp. 148–155, 2020.

[10] Abid A, Cheikhrouhou S, Kallel S, Jmaiel M. Novid- Chain: Blockchain-based privacy-preserving platform for COVID-19 test/vaccine certificates. Softw Pract Exper. 2021;1–27. https://doi.org/10.1002/spe.2983

[11] J. Odoom, R. S. Soglo, S. A. Danso and H. Xiaofang, “A Privacy-preserving Covid-19 Updatable Test Result and Vaccination Provenance based on Blockchain and Smart contract,” 2019 International Conference on Mechatronics, Remote Sensing, Information Systems and Industrial Information Technologies (ICMRSISIIT), 2019, pp. 1-6, doi: 10.1109/ICMRSISIIT46373.2020.9405872.

[12] H. R. Hasan, K. Salah, R. Jayaraman, J. Arshad, I. Yaqoob,M. Omar, and S. Ellahham, “Blockchain-based solution for covid-19 digital medical passports and immunity certificates,” IEEE Access, vol. 8, pp. 222 093–222 108, 2020.

[13] David Nuñez. UMBRAL: A THRESHOLD PROXY REENCRYPTION SCHEME. [Online]. Available: https://github.com/nucypher/umbral-doc/blob/master/umbral-doc.pdf

# Authenticated Encryption for Janus-Based Acoustic Underwater Communication

Branislav Petrović
Norwegian University of Science and Technology (NTNU)
P.O.Box 8900, Torgarden, 7491 Trondheim, Norway
branislp@stud.ntnu.no

Bálint Zoltán Téglásy
Norwegian University of Science and Technology (NTNU)
P.O.Box 8900, Torgarden, 7491 Trondheim, Norway
balint.teglasy@ntnu.no

Sokratis Katsikas
Norwegian University of Science and Technology (NTNU)
P.O.Box 191, 2802 Gjøvik, Norway
sokratis.katsikas@ntnu.no

***Resumen*—Wireless underwater communications commonly utilize acoustic waves due to their superior robustness and range compared to radio waves. But usage of acoustic waves introduces some constraints, such as a low data rate and high packet loss. In addition, most information sent using acoustic waves under water today is unencrypted and unauthenticated. With the development of underwater environments, there is an increased need for data protection among marine operators, since the underwater threat landscape is rapidly broadening with new kinds of attacks, such as eavesdropping, routing attacks, and data tampering. To overcome these problems, in this paper, we propose two security schemes integrated with the first standard for acoustic underwater communication, Janus. The aim is to counteract the threats, bearing in mind the limitations of the acoustic communication channels. The proposed schemes are based on symmetric cryptography. The ultimate goal is to provide a way for underwater nodes to exchange authenticated and encrypted information.**

***Index Terms*—underwater acoustic communication, authenticated encryption, Janus**

## I. Introduction

Wireless underwater communication networks are rapidly increasing in quantity and there exist several types of underwater networks that serve different purposes. Currently, few standardized solutions for underwater communication exist and not all are available for public use. The currently most established digital underwater communication standard is Janus [1], a physical-layer acoustic standard that is robust and makes interoperability among maritime operators possible. However, Janus by itself provides no security mechanism, neither in the form of encryption or authentication. As the Janus standard already serves as a basis for wireless communication, in this paper we focus on adding security mechanisms to it. Authentication of messages and entities is an important requirement, in addition to encryption, since not only confidentiality, but also integrity of data is to be ensured. Authenticated encryption is a mechanism that provides both integrity and confidentiality.

The threats to underwater acoustic communication are similar to those faced by radio communication above the sea surface. Similar techniques are used to disrupt the propagation of the waves and, thereby, also the communication. In addition, routing protocols that forward packets based on channel quality and the possibility to reach adjacent nodes, are also in use in underwater networks. Protocols of this kind can be manipulated by malicious users to cause disruptions in networks.

Ghannadrezaii et al. [2] classify threats to underwater communication into three main categories: Eavesdropping, Data tampering, and Routing attacks, while Domingo [3] also considers Jamming, Wormhole, and Sybil attacks.

The main constraints for providing security features under water are low data rate and high packet loss in this environment. These constraints make the use of many standardized encryption and authentication schemes infeasible due to their complexity and scale. Consequently, the authenticated encryption mechanisms must be computationally inexpensive and reduce the overhead to a minimum.

Bearing in mind the threat landscape and the constraints of the communication under water, it is realistic to assume that the capabilities of eventual adversaries enable performing the attacks mentioned above. Therefore, incorporating security mechanisms in underwater communication standards is necessary.

In this paper, we propose two solutions for the authenticated encryption in Janus-based underwater communications. The first proposal is based on the well-known scheme CCM - Counter and CBC MAC (CBC - Cipher Block Chaining, MAC - Message Authentication Code) [4], and the second proposal is based on a more recent scheme AEGIS [5]. We define Janus-compatible protocols capable of realizing these two authenticated encryption schemes and give arguments in favor of their usage.

The structure of the paper is the following: Section II discusses the Janus standard and the previous proposals that introduce some security mechanisms in it. In Section III, we propose two authenticated encryption schemes that could be used in Janus and give the arguments in favor of using them. Section IV concludes the paper.

## II. Background and related work

The Janus standard allows for a bandwidth of 80 bps and a range of 10 km. The baseline packet is 64 bits long. 34 of these bits are reserved for user data, and are called the Application Data Block (ADB). The rest of the packet consists of communication overhead that specifies different communication properties. The very limited amount of user

data that can be transmitted in a packet makes Janus best suited for small data exchanges, such as Command-and-Control or status messages. If larger amounts of data are to be sent, it must be either distributed into several packets or be sent as a *cargo* immediately following a packet. If several packets are used, the additional overhead must be encoded and modulated for each packet, resulting in additional use of computing resources and increased latency. If a cargo is specified, the channel is reserved during the transmission of the cargo and no other communication is to happen on the network while it is being transmitted.

Most current research of wireless underwater security considers encryption and authentication schemes based on symmetric cryptography and pre-shared information, as this is the only model that is realizable with today's physical underwater infrastructure. Nevertheless, methods based on public key cryptography that would be applicable in a hypothetical underwater PKI (Public Key Infrastructure) have also been proposed (see, for example, [2], [6]).

In order to provide a certain level of security, the recently published subclass of Janus, Venilia [7], specifies an encryption scheme for Janus packets using the custom-made Tiny Underwater Block (TUB) cipher [8]. With Venilia, 27 of the 34 bits in the Janus ADB are encrypted (27 bits is the block size of the TUB cipher). These 27 bits consist of an 8-bit message (so-called pre-canned message), a source address and destination address of 7 bits each, and an additional 5-bit CRC (Cyclic Redundancy Check). The remaining 7 bits in the ADB house a 5-bit IV (Initialization Vector) and a 2-bit epoch identifier, both of which are used as input to the TUB cipher. The 8-bit message field allows for 256 unique messages to be processed. These messages must be stored in a pre-shared code book that is put on all devices in a network.

A drawback of Venilia is that its operation is restricted to 8-bit Command-and-Control and status messages, which are stored in a predefined code book at each device in a communication network. The fact that Venilia does not define the use of cargo packets further restricts its communication capabilities. Consequently, if 30-bit Maritime Mobile Service Identity (MMSI) is going to be used for entity authentication, then Venilia cannot be used for encryption.

With Venilia, it is assumed that authentication has already taken place before the use of Venilia. Consequently, many kinds of attacks can be launched against communication that does not employ authentication, of which the most prominent ones are MITM (Man-In-The-Middle)-based attacks.

The fact that the IV is only used for key derivation and not in the encryption process, implies that the TUB cipher is used in the ECB (Electronic Codebook) mode, which is vulnerable to statistical attacks, although it is worth noting that the key management of Venilia provides certain protection against such attacks.

An approach using public key cryptography is described in [2]. Here, key establishment between nodes in an underwater network is performed using the Elliptic Curve Diffie-Hellman key exchange protocol. After two nodes have established an identical key each, they then encrypt subsequent communications with a symmetric encryption algorithm, such as AES (Advanced Encryption Standard). This scheme permits secure underwater machine-to-machine communication between the acoustic nodes in underwater networks. However, it assumes the existence of an underwater PKI, which is not specified.

The first complete proposal of an authentication procedure based on Janus is given by Téglásy et al. [9]. Here, two devices, initially unknown to each other, first identify each other as friend or foe by determining whether they possess the same pre-shared key. They achieve this by exchanging a timestamp, a clock accuracy descriptor, and two SYN and ACK flags in the ADB of the Janus packet. These values are encrypted with the 32-bit block version of the RC5 cipher using a pre-shared long-term key $K_n$ of at least 128 bits in length. The authentication is based on the validity of the timestamps exchanged by the devices and the fact that the sending device would be unable to encrypt the message without $K_n$. Assuming that the devices' clocks were synchronized during the exchange of $K_n$ at the start of the mission, a device checks if the timestamp it received is within the expected bounds compared to the mission duration. The expected bounds are adjusted according to possible deviations in clock synchronisation between the devices and the expected maximum distance that a message can travel. If the timestamp is valid, the receiver sends its own timestamp and clock accuracy descriptor back to the originating device, encrypted with the same key $K_n$. The protocol is shown in Fig. 1.

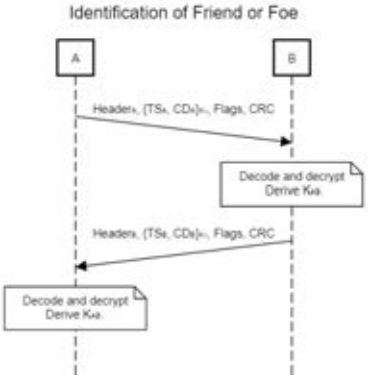



Figura 1. Protocol for identification of friend or foe [9].

To protect the communication in the case of compromise of $K_n$, the exchanged values are later used to compute a session key $K_{AB}$ with a custom Key Derivation Function (KDF). The KDF also utilizes RC5, albeit with a block size of 128 bits, to produce a 256-bit ciphertext. $K_{AB}$ allows communication to remain secure if $K_n$ is compromised after the establishment of $K_{AB}$ since subsequent messages will be encrypted with $K_{AB}$ instead of $K_n$. It is also infeasible for attackers to obtain previous session keys if they obtain $K_n$ and they have not eavesdropped on previous runs of the protocol. Since $K_{AB}$ is derived from a timestamp, a new timestamp sent by an attacker to a legitimate node as an authentication attempt will result in a different key due to the passing of time and, thus, a different timestamp.

As one device may derive session keys with many other devices, it is necessary to create a mapping of the derived session keys to their corresponding device identities. This is achieved with a shared lookup table at each device, containing all other devices' identities with their corresponding long-term

keys. When a device receives a message that is encrypted with a certain long-term key, it will attempt to decrypt the message with the long-term keys in its lookup table, sequentially. If one of the keys yields a successful decryption, the key that yielded the decryption is used with the KDF to derive a session key, and the session key is stored along with the identity and the long-term key in the table. The identity has the form of MMSI, which is a standard format used for tracking ships above the sea surface. The length of the MMSI is 30 bits.

Although this protocol provides authentication, encryption, and key establishment, it has certain limitations. They are outlined below.

*Timestamp forgery*: Authentication of messages that relies solely of the validity of timestamps is vulnerable since timestamps can be forged by an attacker relatively simply. As long as the attackers used a valid $K_n$ to encrypt their message, they will be able to successfully establish a session key with a valid device. Moreover, if attackers obtain $K_n$ before the protocol is run, they can decrypt any message that is encrypted with $K_n$, as well as derive $K_{AB}$.

*Timestamp replay*: Another concern is replay of timestamps. An attacker can relatively simply forge the correct timestamp by observing when the messages in the legitimate protocol are transmitted and recording the time at that moment. Since the acceptance of timestamps at legitimate devices is adjusted to allow errors caused by currents, clock drift, encryption, and decryption delays, there is a possibility that the forged timestamp will be accepted as legitimate.

*Limited forward secrecy*: After the establishment of $K_{AB}$, messages encrypted under it remain secure even if attackers possess $K_n$. However, this is only true if the attackers have not eavesdropped on or intercepted any previous messages. If previous protocol runs with messages encrypted with $K_n$ are eavesdropped and stored, compromise of $K_n$ at any point in time will give attackers the possibility to decrypt the messages and calculate previous session keys that depend on $K_n$.

*Limited encryption strength*: RC5 was chosen because of its security provided by a long key and its relatively simple software implementation. However, Biryukov et al. [10] show that, with partial differential cryptanalysis, it is possible to derive all 25 subkeys for the 12 rounds of the 32-bit block version of RC5, with $2^{44}$ chosen plaintexts. Even though it is not likely that this amount of messages will be encrypted under the same key, employing a stronger encryption algorithm would also eliminate this theoretical possibility.

*Poor scalability*: The lookup table stored on each device requires substantial storage space and memory to store, read, and write to. Additionally, attempting to decrypt a received message with all long-term keys consumes large amounts of time and computational resources, resulting in poor performance, especially in resource-constrained devices, such as sensors. If large networks with many devices are to be supported by an authentication protocol of this kind, the scalability and performance must be improved.

Some limitations mentioned above, such as limited encryption strength and poor scalability, have been addressed in our proposals. Further study is needed in order to conclude whether our proposed solutions also address the timestamp replay/forgery and forward secrecy challenges in a satisfactory way.

## III. The new authenticated encryption schemes for Janus

### *III-A. Authenticated Encryption with CCM*

Given the security and bandwidth requirements, a potentially suitable algorithm for providing assurance of the confidentiality and the integrity of Janus data is the CCM mode of operation for block ciphers, proposed by Whiting et al. [4] and standardized as Special Publication 38-C by NIST [11]. The counter mode it applies for encryption allows the avoidance of transmitting complete blocks of the underlying block cipher, even though the data must be padded for partitioning into complete blocks. The only additional overhead in the data that is sent are the nonce and the MAC tag. It is therefore a good option for our needs as it is a standard that provides a high level of security, and the Janus baseline packet and cargo can be formatted as input to the CBC-MAC and encryption algorithms.

According to [11], CCM is only defined for block ciphers with a block size of 128 bits. Currently, the only approved cipher for the algorithm is AES. There are three inputs to the algorithm: the plaintext $P$ (also called the payload) to be authenticated and encrypted, the associated data $AD$ that will only be authenticated and not encrypted, and a nonce $N$, which is a unique value associated with the other two inputs. Typically, the payload consists of user data that needs to be confidential, while the associated data are packet headers that remain unencrypted for the proper functioning of routing mechanisms in networks. In our case, the payload consists of the ADB in the Janus baseline packet and optional cargo, while the associated data is the 22-bit preamble before the ADB, and the 8-bit checksum. The nonce can be the timestamp of an autonomous underwater vessel or a random number. CCM defines two major operations: generation-encryption, in which the MAC tag is generated and the payload is encrypted, and decryption-verification, where the payload is decrypted and the MAC tag is verified.

We consider the example where only the 64-bit baseline Janus packet is to be processed by CCM. The length of the secret key $K_n$ is 256 bits. The MAC tag $T$ should, by the specification, have a length of at least 4 bytes to prevent forgery attacks. The octet length of $N$, $n$, should, by the specification, be at least 7. The total length of $AD$ is 30 bits, while the length of $P$ is 34 bits.

*III-A1. Argumentation for the use of CCM:* CCM is used first and foremost due to its familiarity as a mode of operation for AEAD (Authenticated Encryption with Associated Data) and its long-lasting and widespread use in WLAN (Wireless Local Area Networks). Despite the age of the algorithm, very few practical attacks against both confidentiality and authenticity have been developed, meaning that it still provides a high level of security. Another argument for its usage in underwater networks is its ability to partially encrypt a message, while authenticating both the encrypted and unencrypted parts. Since the Janus packets in our authentication protocol have messages of this form (i.e., it is possible to format the Janus header as associated data, while only encrypting the ADB), formatting our messages for the structure of CCM is relatively simple.

Jonsson [12] provides a formal analysis of CCM with the

conclusion that a high level of confidentiality and authenticity is provided in line with other standardized modes of operation for authenticated encryption, such as GCM (Galois/Counter Mode) or OCB (Offset Codebook Mode). The attractive properties of CCM are listed below:

1. A specific mechanism that handles parts of messages that are only to be authenticated and not encrypted is provided by default. This is done without additional ciphertext overhead, something that requires enhancements in many other authenticated encryption modes.
2. AES is used only in the forward direction for encryption and its inverse is never used. This contributes to the reduction of the code size of implementation.
3. The `ctr` and `cbcmac` algorithms are widely deployed and have been in use for a long time, meaning that CCM is based on well known and proved technology. Existing implementations are also highly optimized.
4. All intellectual property rights have been released to the public domain, making CCM freely available for any purpose.

Regarding authenticity, Jonsson claims that it is hard to extract any non-trivial information about the input blocks and the output blocks of the CBCMAC algorithm, even if all the plaintexts of a message exchange are known. With respect to confidentiality, the goal of an attacker would again be to distinguish a ciphertext from a bit string chosen uniformly at random from the set of all possible bit strings of the given length. The two ways that this can be done are either that the attacker executes a birthday attack against the `ctr` output blocks or that an anomaly occurs within the `cbcmac` computation, such as an internal collision or a MAC tag that is identical to some `ctr` output block.

*III-A2. Application in Janus-based communication:* Regarding the application in Janus, CCM can be directly applied, using the smallest recommended values of $t$, $n$, $a$, and $p$, such that the standardized levels of confidentiality and authenticity are provided. However, the output of the processing of a Janus baseline packet cannot itself fit into a single baseline packet due to the need to transmit the nonce $N$ and the MAC tag $T$ in addition to the associated data $AD$ and the encrypted payload $C$. Instead, either a cargo must be specified or several baseline packets must be used. Which option is better depends on several factors, such as the scale of the network and the amount of data that the devices send on average. Since the aim of Janus is to be an open standard, we believe that users with different requirements may wish to use either option. Following are descriptions of the incorporation of CCM into the protocol for authentication of underwater assets, using both options - without cargo and with cargo.

*III-A3. Without usage of cargo:* To perform full authentication i.e., both authentication of messages and entity authentication, it is necessary to transmit a 29-bit timestamp $TS$, a 3-bit clock accuracy descriptor $CD$, and two 1-bit SYN and ACK flags $F$, as well as the MMSI of both device $A$ and $B$, in each direction. The transmission of both MMSIs allows the sending device to authenticate itself and to indicate which device the message is designated for. Thus, there is no longer a need to store session keys at each device. This improves scalability, as the storage of the lookup tables and the lookup procedures would require much storage space and processing as the network grows. For CCM, the required input at the receiving device are the nonce $N$, the associated data $AD$, and the ciphertext $C$, which consists of the plaintext $P$ and MAC tag $T$, both encrypted. Hence, our goal is to partition the values used in the authentication protocol into the format of CCM.

For CCM to function, $N$ must have a length of at least $56$ bits and cannot fit into the 34-bit ADB. However, a way to resolve this is to set an initial length of $N$ to 32 bits and then duplicate that value locally at each device to produce a $64$-bit string, which is a valid length for the generation-encryption and decryption-verification operations. The entropy of $N$ will in this case still be the same as for a 32-bit string, but this is satisfactory for most applications.

Regarding the associated data, $F$ should be authenticated, as it provides relevant status information in the protocol. Taking this into account, $AD$ consists of the 22-bit Janus header and the two 1-bit flags, so it is 24 bits long. In our case, the Janus header will remain the same for all CCM-related packets sent in one direction that are part of the authentication protocol. This allows for the usage of the Janus header of any packet to directly be included in $AD$, without the need to partition $AD$ in the ADB. Thus, the first baseline packet can be used to transport both $N$ and the 22 bits of $AD$ that are made up of the Janus header. The two remaining bits of $AD$ can either be transmitted in the same packet or in the next one.

The timestamp and clock accuracy descriptor consist of 32 bits in total and can therefore fit in the ADB together with the flags F. $TS$ and $CD$ can also directly be encrypted in the counter mode of CCM such that a 32-bit ciphertext is produced. Hence, the second packet in the protocol is used to transport $TS$ and $CD$ in encrypted form and $F$ in plaintext.

The MMSIs have a length of 30 bits each and must therefore be located in their respective baseline packets. Therefore, the third and fourth packets are used to transport $\text{MMSI}_A$ and $\text{MMSI}_B$.

The authentication tag $T$ is the final element that needs to be transmitted. It is possible to set the size of $T$ to $4$ bytes and conveniently place it in the ADB.

The complete protocol can be seen in Fig. 2. As each participant transmits five $64$-bit baseline packets, the theoretical delay will be $64 \cdot 5/80 = 5s$, in addition to the propagation delay of the acoustic signals, which depends on the distance between the participants.

*III-A4. With usage of cargo:* The usage of a cargo would allow for transmitting all the required data in a single packet, thus avoiding the need to transmit $AD$ and a CRC for every baseline packet, which leads to reduced bandwidth consumption. However, the channel would be reserved for the transmission of the cargo, preventing any other devices from transmitting while the cargo is being transmitted.

In the setting with cargo, $AD$ again consists of the 22-bit Janus header and $F$, but, additionally, the first byte of the ADB is considered as associated data, as it provides meta-information about the packet. Since the first byte of the ADB is used for the cargo specification, 26 bits are left in it for user data. Thus, the 26 MSBs of $N$ can be put here, while the 6 LSBs are transmitted as cargo. Following $N$, the encrypted

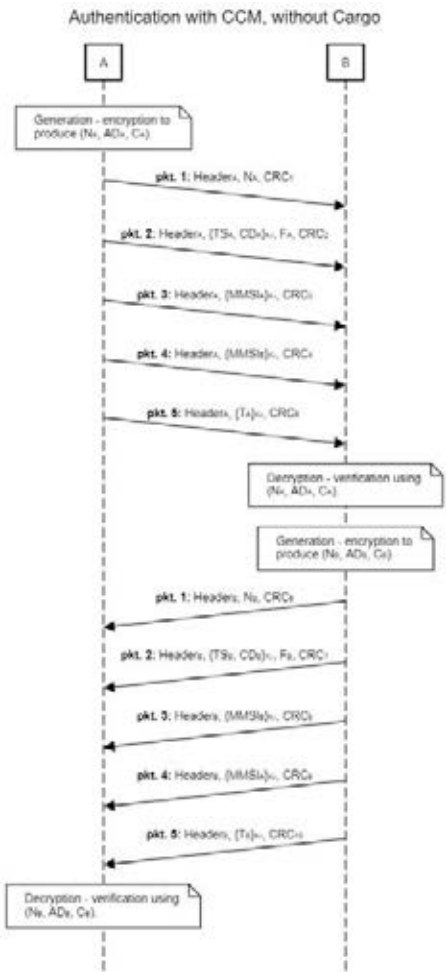



Figura 2. CCM used in authentication protocol, without usage of cargo.

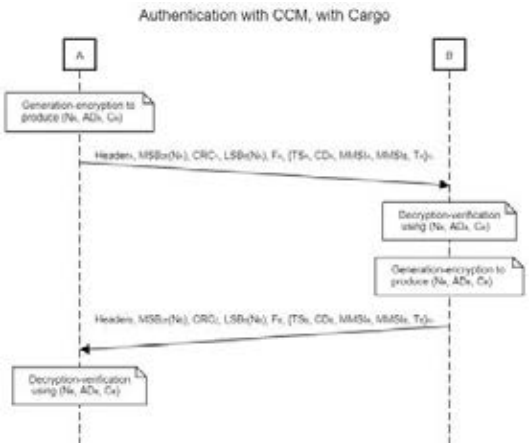



Figura 3. CCM used in authentication protocol, with usage of cargo.

payload is transmitted, consisting of the ciphertexts of $TS$, $CD$, the two MMSIs, and $T$. The protocol can be seen in Fig. 3.

For the case of one packet of this format, the total amount of data is $30 + 26 + 8 + 6 + 2 + 29 + 3 + 30 + 30 + 32 = 196$ bits, with the cargo consisting of the last 132 bits. With the data rate of 80 bps provided by Janus, the theoretical encoding time for the entire packet is $196/80 = 2{,}45$ s. For the cargo alone, the encoding time is $132/80 = 1{,}65$ s. This means that at least 1,65 s of channel reservation time must be specified in the first byte of the ADB. According to the lookup table for channel reservation used by Janus, the minimum time that can be reserved with the reservation bits in the ADB to accommodate this cargo length is $\approx 1{,}79$ s. Compared to the encoding time for one baseline packet alone, $64/80 = 0{,}8$ s, this packet format does not impose a substantial overhead with respect to the added benefits of both confidentiality and integrity of all data needed to perform message authentication, entity authentication, and session key establishment.

### *III-B. Authenticated encryption with AEGIS*

A potentially even more suitable algorithm than CCM is AEGIS, proposed by Wu and Preneel [5]. Unlike CCM, which is a special mode of operation for a block cipher, AEGIS is a dedicated authenticated encryption algorithm, which also employs an underlying block cipher, but uses a message to update the state of the cipher. As with CCM, AES is most commonly used as the underlying block cipher and we shall also describe this version for our purposes.

For consistency with other relevant solutions, such as Venilia, we focus on the 256-bit key version, namely AEGIS-256, which utilizes a 256-bit key and 256-bit IV, operates on 16-byte message blocks and uses 6 AES round functions.

AEGIS-256 defines a state update function, which updates an internal state using 6 iterations of AES. The main operations are the initialization, processing of the associated data, encryption, and finalization. Each of the operations updates the state and uses it to perform their respective tasks. The tag $T$ can be of any length up to 128, which is the AES block size.

*III-B1. Argumentation for the use of AEGIS:* AEGIS was one of the winning submissions to the CAESAR contest for authenticated encryption algorithms. Hence, we believe that an underwater solution based on AEGIS will keep underwater security schemes up to date with their counterparts above water, even if CCM gets discontinued in WiFi networks. In addition, AEGIS has already been deployed in autonomous vessels that apply the Robot Operating System (ROS) [13]. Since the underwater environment has similar devices and characteristics as the surface vessels using ROS, we believe the application of AEGIS under water is a natural next step.

Wu and Preneel provide a security analysis of AEGIS in addition to its specification. They name three requirements for the secure operation of AEGIS:

1. Each key $K$ used for initialization must be generated uniformly at random.
2. An IV must not be used more than once during the lifetime of a key and each key and IV pair must only be used with one size of the authentication tag $T$.
3. If the verification of the tag $T$ fails, the decrypted plaintext and wrong $T$ must not be disclosed to the public.

*III-B2. Application in Janus-Based Communication:* The protocol for authentication with AEGIS is very similar to the CCM version, described above, both with and without Janus cargo packets. The same values are transmitted in both schemes, resulting in an equal amount of bits. The nonce $N$ from CCM is denoted IV here, but these values serve a very

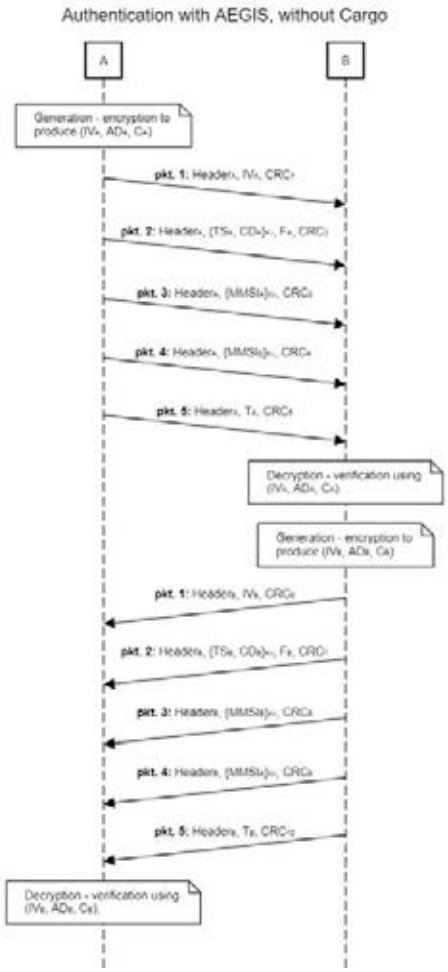



Figura 4. AEGIS used in authentication protocol, without usage of cargo.

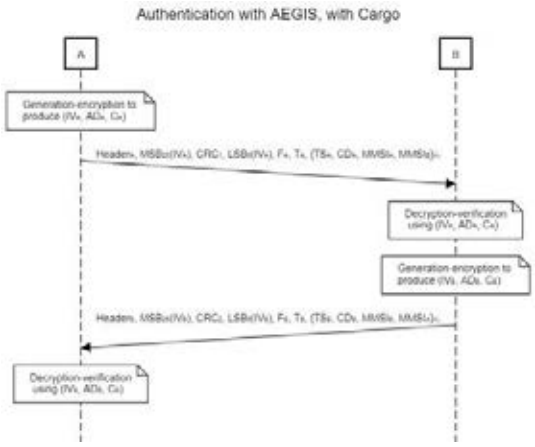



Figura 5. AEGIS used in authentication protocol, with usage of cargo.

similar purpose. The protocol without cargo can be seen in Fig. 4, while the version with cargo is shown in Fig. 5.

## IV. Conclusion

In this paper, we have proposed two mechanisms of authenticated encryption integrated in the Janus-based underwater acoustic communication system. We gave the arguments in favor of these proposals and we believe that they can provide a satisfactory level of security in Janus. We have shown that the two proposed solutions have potential to be accepted for use with Janus when longer messages are to be securely transmitted. We provided the calculations that show that the computational overhead related to these mechanisms is acceptable.

## Referencias

[1] Potter J., Alves J., Green D., Zappa G., Nissen I., McCoy K., "The JANUS underwater communications standard", *2014 Underwater Communications and Networking (UComms)*, pp. 1–4, 2014.

[2] Ghannadrezaii H., Bousquet J., "Securing a Janus-based flooding routing protocol for underwater acoustic networks", *Proceedings of OCEANS 2018*, pp. 1–7, 2018.

[3] Domingo M., "Securing underwater wireless communication networks", *IEEE Wireless Communications*, Vol. 18, No. 1, pp. 1–21, 2011.

[4] Whiting D., Housley R., Ferguson N., "AES encryption & authentication using CTR mode & CBC-MAC", IEEE 802.11-02, 2002.

[5] Wu H., Preneel B., "AEGIS: A fast authenticated encryption algorithm", *Proceedings of SAC 2013*, pp. 1–21, 2013.

[6] Petrioli C., Saturni G., Spaccini D., "Feasibility study for authenticated key exchange protocols on underwater acoustic sensor networks", *Proceedings of WUWNET '19*, pp. 1–5, 2019.

[7] Hobbs A., Holdcroft S., "JANUS class 17 Venilia: Secure pre-canned messaging", *Dstl Cyber and Information Systems*, 2021.

[8] Hobbs A., Holdcroft S., "Tiny Underwater Block cipher (TUBcipher): 27-bit encryption scheme for JANUS class 17", *Dstl Cyber and Information Systems*, 2021.

[9] Téglásy B., Wengle E., Potter J., Katsikas S., "Authentication of underwater assets", paper in preparation, 2022.

[10] Biryukov A., Kushilevitz E., "Improved cryptanalysis of RC5", *Proceedings of EUROCRYPT'98*, pp. 85–99, 1998.

[11] Dworkin M., "Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality", NIST Special Publication 800-38C, 2004.

[12] Jonsson J., "On the security of CTR + CBC-MAC", *Proceedings of SAC 2003*, pp. 76–93, 2003.

[13] Volden Ø., Solnør P., Petrović S., Fossen T., "Secure and efficient transmission of vision-based feedback control signals", *Journal of Intelligent & Robotic Systems*, Vol. 103, No. 2, 2021.

# Ransomware: An Interdisciplinary Analysis

M. Robles-Carrillo
Network Engineering & Security Group
https://nesg.ugr.es
University of Granada (Spain)
mrobles@ugr.es

P. García-Teodoro
Network Engineering & Security Group
https://nesg.ugr.es
University of Granada (Spain)
pgteodor@ugr.es

***Abstract*—Ransomware has evolved from being a simple form of cyber-attack to become a national and international security risk. Statistical data and doctrinal analyses agree that there is an increasing trend of this type of threat, in both quantitative and qualitative terms, since it is expanding, diversifying and perfecting its attack approaches. Prevention, detection, recovery and response techniques have evolved considerably, although they are still insufficient to deter this phenomenon. This paper performs an analysis of ransomware from an interdisciplinary, technical and legal perspective. For this purpose, we explain the problem and research on this subject, the legal challenge that it poses, the need to address the technical and legal dimensions, and the response guidelines from both perspectives.**

***Index Terms*—Ransomware, Technical, Legal, Interdisciplinary**

## I. Introduction

Ransomware basically consists of 'kidnapping' personal data and/or devices until a ransom is paid by the victims. From this basic criminal condition, ransomware has evolved in its techniques -more varieties, more devices to use it on and more functional complexity–, targets and purposes even to the point of being assimilated to terrorist activity or becoming an instrument of warfare.

Ransomware has indeed acquired a higher, unfortunate and alarming protagonism. As various data and sources show, this is no exaggeration. In May 2022, Lincoln College closed its doors after 175 years of activity as a result of an attack that, by hijacking and encrypting data, prevented the management of its student recruitment and fund-raising campaigns necessary for the operation of this university (https://www.securityweek.com/ransomware-attack-nail-coffin-lincoln-college-closes-after-157-years). In May 2022, one year after the Colonial Pipeline attack, the US Department of Transportation Administration imposed a fine of close to 1 million dollars because of the failure of the company's ransomware contingency plan (https://www.scmagazine.com/analysis/critical-infrastructure/us-proposes-1-million-fine-for-colonial-pipeline-ransomware-attack). In May 2022, Cybersecurity and Infrastructure Security Agence (CISA) announced the formation of a joint ransomware task force (https://securityboulevard.com/2022/05/cisa-announces-joint-ransomware-task-force/). In May 2022, too, the President of Costa Rica declared a national state of emergency due to the Conti ransomware attacks against government entities (https://www.bleepingcomputer.com/news/security/costa-rica-declares-national-emergency-after-conti-ransomware-attacks/). In fact, also in May, the US State Department offered a 10,000,000 dollars reward for information on the leaders of the Conti ransomware (https://www.state.gov/reward-offers-for-information-to-bring-conti-ransomware-variant-co-conspirators-to-justice/). In the wake of the Colonial Pipeline case, the US Department of Justice gave ransomware investigations a priority similar to that of terrorism (https://www.reuters.com/technology/exclusive-us-give-ransomware-hacks-similar-priority-terrorism-official-says-2021-06-03/). NATO warned about the impact of ransomware attacks (https://www.govtech.com/blogs/lohrmann-on-cybersecurity/nato-adds-cyber-commitments-potential-ransomware-response. In the context of the conflict in Ukraine, the importance of ransomware attacks has also become evident. For some time now, ransomware has become a criminal, terrorist or even warlike activity that threatens security in general and information security in particular. As a consequence, it is considered a global security issue [1].

The aim of this paper is to provide an interdisciplinary, technical-legal analysis addressing the problem of ransomware, the scope of the research, the legal challenge, the technical-legal dimensions and the guidelines for combating this phenomenon.

## II. The Ransomware Problem

Ransomware has naturally become a common and shared concern for States, institutions, agencies and international organizations.If it used to be just another type of cyber-attack, for some time now it has acquired entirely new dimensions.

In October 2020, the G7 States have included a Ransomware Annex to their final Statement Meeting. They recognized its particularity as a global threat and committed themselves to coordinate action to address and mitigate it (https://home.treasury.gov/system/files/136/G7-Ransomware-Annex-10132020_Final.pdf). In June 2021, the G7 Leaders have identified the fight against ransomware among their priorities (https://www.g7uk.org/wp-content/uploads/2021/06/Carbis-Bay-G7-Summit-Communique-PDF-430KB-25-pages-3.pdf). In June as well, the European Union and the United States have adopted a Joint Statement in which they stated their common concern on ransomware (https://www.consilium.europa.eu/en/press/press-releases/2021/12/17/joint-eu-u-s-statement-following-the-eu-u-s-justice-and-home-affairs-ministerial-meeting-washington-d-c-16-december-2021/). In October 2021, more than thirty countries, as well as the European Union, led by the United States, have adopted the Joint Statement of the Ministers and Representatives from the Counter Ransomware Initiative. According to it, *"ransomware is an escalating global security threat with serious economic and security consequences"* (https://www.whitehouse.gov/briefing-room/statements-releases/2021/10/14/joint-statement-o

f-the-ministers-and-representatives-from-the-counter-ransomware-initiative-meeting-october-2021/).

There are also important international bilateral initiatives such as the commitment between the United States and Israel concerning the creation of a U.S.-Israeli Task Force to combat ransomware (https://home.treasury.gov/news/press-releases/jy0479). As an example of public-private cooperation, 'No More Ransom' (NMR) is a project launched in 2016 by the Dutch National Police, Europol, Intel Security and Kaspersky Lab, that introduces a different level of cooperation between law enforcement and the private sector to fight ransomware (https://www.eulisa.europa.eu/Newsroom/PressRelease/Documents/PR-NMR.pdf).

There is, indeed, a paradigm shift that alerts about the scope and dangerousness of this type of attack. The RTF (Ransomware Task Force) considers it an *"urgent national security risk around the world"* [2]. Although important, many of these initiatives do not have adequate technical support and legal basis to address a threat like ransomware that targets an ever growing number of individuals, companies and institutions. There are a considerable misalignment and imbalance between policy initiatives, technical research and legal issues.

## III. Research on Ransomware

Technical studies on ransomware are numerous and exhaustive. Many of them deal with the concept, evolution, families [3], [4], anatomy [5] and characteristics of ransomware behaviour [6]. Zimba and Chishimba also explain its evolution, but with the specific aim of making a categorization framework based on the virulence of a given attack [7]. The taxonomy of ransomware, mitigation techniques and ransom payment guidelines are also analyzed [8], as well as prevention, monitoring and damage control [9], [10]. Some authors analyze a specific typology such as crypto ransomware [11] or focus on specific devices or operating systems [12]. Rehman *et al.* have plead about the need of a better technological vision and stronger defenses [13]. Trautmand and Ormerod have arrived to the same conclusion developing a study about Wannacry as well as related ransomware cases considered an emerging threat to corporations [14]. With a different approach, Bander *et al.* have prepared a survey of the existing research into ransomware as a novel ransomware taxonomy [15]. These and many other authors have produced a large and comprehensive technical bibliography in contrast to the small number of legal or interdisciplinary studies. This extensive range of technical solution proposals contrasts with the paucity of legal rules and sanctions to combat ransomware.

However, the deepest problem lies on the fact that many of the victims do not use existing legal remedies to report the attack and prosecute the offence. In addition, besides the widespread tendency to manage the problem outside the law, ransomware has even become a business for insurance companies. Contracting an insurance policy to face a possible ransomware is equivalent to providing a legal guarantee of payment for the commission of an unlawful illegal activity. It is, moreover, an additional incentive for the crime because, if the victim cannot pay, the insurance company will. This is, quite simply, the perversion of the law: to turn it into a guarantee for the benefit of the criminal.

Ransomware poses a major legal challenge, indeed. The main problem is not just the high degree of impunity enjoyed by the perpetrators of this type of attack, but also, and especially, the loss of confidence of the victims in both the legal order and the justice system themselves.

## IV. The Legal Challenge

Ransomware is possibly at this time the greatest exponent in practical terms of the loss of trust in law and justice. For whatever reasons, whether it is the fear of losing control over data or of the cyber-attack itself or the loss of prestige, the acceptance of or the acquiescence to extortion raises not only a technical and/or criminal issue, but a deeper social, political and legal problem. Juridically, there are two issues: how law may combat ransomware and how ransomware may erode the trust in law.

The exponential growth of ransomware is explained by its economic profitability [16], and also by the high degree of impunity that characterizes this criminal practice [17], [2]. Neither the development of accurate technical remedies nor the implementation of social or business behavioural practices have proven to be enough to neutralise or reduce the cases. Most of the research on ransomware to date has focused primarily on its technical aspects, with comparatively little attention being given to understanding other aspects as the socio-technical or legal sides.

Legally, ransomware involves a variety of complex criminal actions including the hijacking of data and/or devices, the alteration and/or destruction of data and/or devices, extortion, the illegal demand for the payment of ransom, the laundering of the proceeds of crime and the possible use of ransom to commit other illegal activities. To understand the complexity of ransomware from a legal point of view and to regulate it accordingly are the necessary first steps to prosecute and sanction it. If there are no legal sanctions or if they are weak or insufficient, there will be impunity - an impunity that will fuel the use of ransomware,

Technical solutions to problems are necessary but cannot replace or displace the law because technology and law have different functions and they must complement each other in order to be effective. On the other side, law has also necessarily to be adapted to the technological changes.

The relationship between technology and law is not easy. Regulatory change is slower than technological change. They can also be more complex. Norms are the result of legally pre-established procedures involving institutions with the necessary power and legitimacy to create, modify and implement norms. After their adoption, norms become binding. Actions and behaviours are to be adapted to the norm and not the reverse. But for rules to be effective, they must provide the right response to needs and problems. To do this, the reality to be regulated must be properly understood, even if it is technically complex. Tatar *et al.* argue that *"Inconsistency between the way in which the law is structured, and the way in which technologies actually operate is always an interesting and useful topic to explore. When a law conflicts with a business model, the solution will often be changing the business model. However, when the law comes into conflict with the architecture of hardware and software, it is less clear how the problem will be managed"* [18].

In fact, the problem arises in both directions. If the regulation is in conflict with the technical component, there is a problem. Likewise, if the technical component conflicts with the norm, there is also a problem. In the latter case, the technical component is illegal. In the first case, the law is ineffective. Whatever the case, there exists a problem. Any solution, in order to be both legal and effective, necessarily involves a careful simultaneous understanding of technical and legal aspects.

## V. Technical and Juridical Dimensions

There exist two main types of ransomware [4], [19], [8], [5]: *Device lockers*, aimed at locking the device screen and displaying a full-screen image, and *Crypto-ransomware*, which ciphers user's personal files and document. The most relevant international legal framework for the prosecution and punishment of malicious activities in cyberspace is the Convention on Cybercrime adopted in the Council of Europe in 2001. In this so-called *Budapest Convention*, the provisions specifically related to ransomware are: Article 4, concerning crypto-ransomware; and Article 5, relating to locker-ransomware.

In similar terms to the Budapest Convention, Directive 2013/40/EU regulates *"Illegal access to information systems"* (Article 4), *"Illegal system interference"* (Article 5) and *"Illegal data interference"* (Article 6). However, article 8 establishes, in addition, the legal regime regarding the cases of incitement, aiding and abetting and attempt. According to Article 8.1, Member States *"shall ensure that the incitement, or aiding and abetting, to commit an offence referred to in Articles 3 to 7 is punishable as a criminal offence"*. Along with it, States shall ensure that the attempt to commit an offence referred to in Articles 4 and 5 is punishable as a criminal offence.

The criminalisation of incitement, complicity and the attempt to commit an offence is a fundamental difference with respect to the Budapest Convention system. It is an effective approach to the punishment of this offence as well as to limit or reduce the impunity associated with ransomware. It is not only the criminal result that is criminalised, but also the conduct aimed at achieving that result, regardless of whether it is achieved or not. The fact that there are an illegal access or an illegal interference and requirement that the action is committed "intentionally" and "without right" must be sufficient to justify the punishment of such conduct even when the expected result of such an attack is not achieved.

Although in the case of the European Union the regulation seems to be more effective, there is an in-depth problem with the legal approach to ransomware in general. Legal response to ransomware is basically the same as to malware in general [20]. However, ransomware differs from other types of cybercrime for several reasons [21]. It is an exception to the traditional data security breach concept [21], [22]. Not only data is affected, but also privacy, which is another legitimate legal right protected by law. Actually, when the target is a critical infrastructure, the attack implies a contravention of the regulations established to protect it. Regulations on the security of networks and information systems or on the security of electronic communications are also threatened and/or violated by ransomware attacks. Ransomware is a complicated modality of extortion [23], [24].

In order to better understand the basics and scope of ransomware and the reasons why those legal responses are not enough, an understanding of its objectives and operation is necessary. Figure 1 shows some of the most relevant ransomware samples appeared over time are [25], [26], [27].

As regard its operation, aimed at fighting properly against ransomware, some studies exist where specific samples are collected and analyzed in order to characterize them and extract and learn common behaviors [28], [29]. First of all, it is important to note that ransomware usually goes through several common stages:

1) Infection/propagation. As any other types of malware, usual infection vectors include spam emails, SMSs, malicious webs (drive-by-download), and the use of infected devices [19]. In this first spreading stage, exploitation of system vulnerabilities is also a principal infection vector.
2) Privilege escalation and permission gain. Once the malware is downloaded onto the device, special privileges may be required to access some functionality (*e.g.*, PIN modification to lock screen).
3) Ransomware execution. As explained, ransomware is intended to kidnap the user's device, either by ciphering the information or by locking the access. In the first case, some of the most common types of personal files affected are database-related files, web pages, and data and photos. In the case of locker-ransomware, access to system files can be performed to get locking the device (*e.g.*, by changing the entry PIN).
4) Ransom message. Once the malicious action is carried out, all current families display threatening messages (maybe by email) to monetary extort the user.
5) External communications. As other typology of malware (*e.g.*, botnets), it is usual the communication of the infected device with an external server. The reason for that can be varied: extraction of personal information (leakage), exchange of commands (command and control, or C&C), malware update, etc.

According to its operation, from a legal point of view, ransomware is more than a simple illegal access to or interference with systems, communications or data. It cannot be treated merely as such. Ransomware can be a sum of several of these different infractions, or perhaps it should be a specific type of illicit act. Criminalising ransomware as a specific type of offence could be a valid and effective option for a legal response to this particular type of cyber attack. The uniqueness of ransomware within the overall typology of cyber attacks, due to its nature, characteristics and performance, may justify its autonomous typification as an independent crime.

## VI. Response Guidelines

Following the Joint Statement of the Ministers and Representatives from the Counter Ransomware Initiative, the fight against ransomware *"will include improving network resilience to prevent incidents when possible and respond effectively when incidents do occur; addressing the abuse*

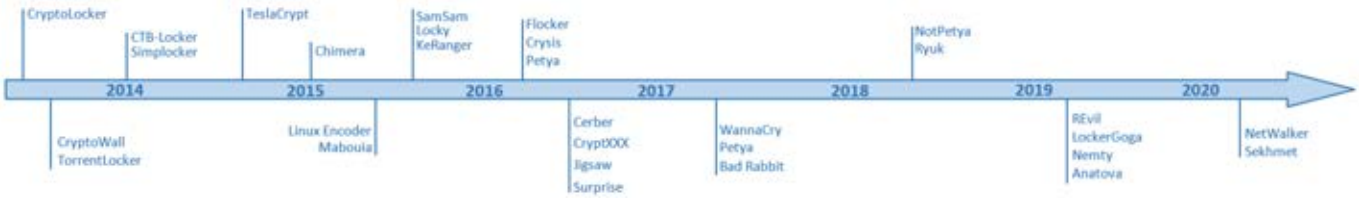



Fig. 1. Timeline of some relevant ransomware families and variants.

*of financial mechanisms to launder ransom payments or conduct other activities that make ransomware profitable; and disrupting the ransomware ecosystem via law enforcement collaboration to investigate and prosecute ransomware actors, addressing safe havens for ransomware criminals, and continued diplomatic engagement"* (https://www.whitehouse.gov/briefing-room/statements-releases/2021/10/14/joint-statement-of-the-ministers-and-representatives-from-the-counter-ransomware-initiative-meeting-october-2021/).

Similarly to generic malware, three are the typical technical defense lines to defeat ransomware: *prevention*, aimed at trying to avoid its occurrence; *detection*, to be aware as soon as possible of its potential appearance; and *recovery*, to mitigate its effects in case of operation and thus to increase the resilience of the target system. Attribution, prosecution and sanction must also be guaranteed.

### A. Preventing Ransomware

The first defense line must necessarily be that of prevention [5]. Prevention requires the use of technical measures as well as social, educational and legal instruments and policies. Awareness, education and training on cybersecurity are essential. Regulation for the prevention, prosecution and punishment of ransomware is also fundamental both to fulfil a deterrent function and to avoid the impunity that encourages the commission of these crimes.

Technically, the best way to avoid a given pernicious threat is to put into action some mechanisms aimed at preventing the occurrence of the threat. For that, best practices are recommended to minimize risks (Table I).

From a juridical point of view, at this point, the main challenge is to provide a clear and comprehensive regulation that includes both the prosecution of these crimes and the obligation to establish and regularly adapt all the necessary security and protection measures to combat ransomware. Typification of ransomware as an autonomous offence would provide a clear understanding of the prohibited behaviour and its legal consequences.

Moreover, the procurement of insurance policies to respond in case of attack should be prohibited. This practice has been developed faster because it is wrongly recommended as a solution for the possible victims and because it is a business for insurance companies. The insurance contracts actually have an unlawful cause and the *de facto* beneficiary of these policies is the ransomware attacker. For the latter, it is a guarantee of the success of the attack because in any case the payment is covered by the insurance contract. It is an additional incentive for the commission of the crime. The victims, even if they do not have to pay the ransom, are already paying the insurance. In addition to having an economic cost, this can lead to a relaxation of their own security measures, which, in turn, can result in an increase in the number of attacks.

Alongside with that, the possibility of prohibiting and penalizing the payment of ransom is being widely debated. In this case, unlike the previous one, the issue is much more complex. On the one hand, it could be a way to combat this crime to follow the route of the payment. Although, on the other, it would mean accepting a double victimization of the victims as they are first attacked and then punished for responding to the extortion. Legal arguments such as force major or state of necessity could be invoked to justify such an action.

### B. Ransomware Detection

Despite several prevention mechanisms are adopted, infections are still possible. That is mainly due to human factors (*e.g.*, social engineering) [21], but also because of the usual existence of vulnerabilities and misconfigurations in software and systems. As a consequence, also detection mechanisms need to be deployed around our environment, aimed at early detecting the potential operation of ransomware to thwart its effects. Actually, *"early detection is not so effective once the victim is infected"* [30].

There are different detection approaches in the literature [31]. Most current detection solutions refer to signature-based tools, like *SurtRight's HitmanPro.Kickstart* or *Avast's Ransomware Removal*. Such approaches rely on the detection of well-known activity patterns. For instance, McAfee reincludes more than 8 million ransomware signatures including *CTB-Locker*, *CryptoWall*, and their variants [32]. Instead, several other current detection solutions rely on analyzing behaviours [33]. A holistic taxonomy of countermeasures for ransomware is introduced in [34], where both technical, education based, as well as policy and law related issues are considered.

Although they are numerous and worthy, none of the available solutions at present is effective enough against ransomware. Early detection is a main challenge. Although potentially accurate in detection, any valid solution should additionally be as quick as possible. Otherwise, the action regarding the encryption of the system can be completed and, thus, the detection itself will become useless.

In addition to these technical issues, the detection phase is particularly relevant from the juridical point of view. The applicable law and the competent jurisdiction to prosecute this crime, including the complex issue of obtaining and preserving evidence, have to be determined on the basis of the place and time of detection. As it is well known, investigation and criminal prosecution of these offences may fall under different jurisdictions or, because of the negative

TABLE I
DEFENSE MECHANISMS AGAINST RANSOMWARE.

| Prevention | Detection | Recovery |
|---|---|---|
| • Security policies<br>• Training<br>• Legitimate software<br>• Updates<br>• Privilege management<br>• Service deployment<br>• Internet protection<br>• Data backup<br>• Removable devices | • File system activities<br>• API calls<br>• Registry access<br>• C&C communications<br>• Encryption procedures | • Payment<br>• Cleaning/replacement<br>• Backup restore<br>• Law enforcement<br>• Agencies notification |

conflict between them, under no jurisdiction, thus amounting to absolute impunity. Activation of legal measures at the detection stage may be essential to avoid such a situation.

Two main legal issues arise here. First, by nature, ransomware is a transnational crime. Second, there are several and different legislation in the various countries. The place in which the intrusion is detected may be relevant for the purpose of determining the applicable law. In the end, the best mechanisms to tackle the problem are international cooperation and mutual assistance also in the recovery phase.

### C. Recovery from Ransomware

The disposal of a recovery plan for business continuity is a key aspect in cybersecurity defense (as in any other ICT field). That includes the measures to solve the problem reported (the existence of a ransomware in this case) and to restore the system to the previous (non-infected) status. Never pay may be a main rule.

Some mechanisms discussed to recover the system from a ransomware event are as follows:

- Once clear that the ransom should not be paid, monetary or otherwise, the next step is to isolate the infected machine to clean it. However, how to be sure about that? Since it is possible malware persists even after system formatting (take into account that malicious software can be embedded into personal files like pdf), the best option is device replacement instead of simple cleaning.
- In this line, it is recommended to restore the data affected by ransomware from data backups (see *Prevention* above).

Whichever the recovery plan adopted, it is important to urgently report incidents to law enforcement agencies to make it possible to prosecute and punish the illegal action. Otherwise, intrusion will go unpunished and, probably, reattempted since it has not been sanctioned. A technical response without legal complaint action is a quick short-term solution. But it is not the best option in the medium or long term because the intrusion is more likely to be repeated if there are not legal consequences.

### D. Attribution, Prosecution and Sanction

The problem of technical traceability and anonymity that characterize cyber actions make it difficult to attribute the actions to a perpetrator or perpetrators. In ransomware, there is an additional problem: the general propensity of the victims to pay the ransom and not to use the existing legal channels to denounce the facts. As a result, ransomware has really become like an iceberg paradigm. Under-reported crime is a reality. Moreover, even when the facts are reported, the prosecution of the crime is complicated by the difficulty of accessing the evidences needed to incriminate the perpetrators. Ransoms are often paid through cryptocurrency, so they are difficult to trace [2].

Attribution, prosecution and sanctions are challenging tasks. First, ransomware is a complex attack both from the technical and juridical point of view. Although it constitutes an access or unlawful interference, it is more complicated than that because it generally damages a larger number of legal assets and rights. Second, while technical responses are being adapted to respond to new variants of attack, there have not been real changes in the current regulations in spite of the fact that they have proven not to be effective in deterring or punishing these illegal conducts. Third, while statistics on cases, victims, ransoms and payments are increasing, there are no sufficiently indicative and reliable, comprehensive or aggregate statistics on reported, investigated, prosecuted or sentenced ransomware cases. Finally, while ransomware has been marginalising the recourse to law and justice, law has been unable to adapt itself to face the challenges posed by technological progress as well as, particularly, the danger of ransomware.

In the end, only a percentage of ransomware attacks are reported, only a percentage within these can be traced and attributed, and only a percentage within these can be prosecuted and ultimately criminally sanctioned. Without sanction, there is no deterrence. Without punishment, there is impunity. With impunity, crime remains profitable and will not stop. In these circumstances, hardly surprisingly, ransomware has moved beyond its status as a crime to be used as a terrorist or even a warfare tool becoming a global security threat.

## VII. CONCLUSIONS

Ransomware has become a serious security issue [8]. The overwhelming majority of the proposed solutions against ransomware are of a technical nature. There are few solutions of any other nature. There are no interdisciplinary proposals despite the fact that the two main reasons for ransomware success as a criminal activity are two: *effectiveness* and *impunity*. Effectiveness is mainly a technical issue but impunity is definitely a legal problem. Impunity is the main incentive for this unlawful act.

Legal solution requires some basic changes. The first one is the typification of ransomware as an autonomous and specific crime taking into account its technical uniqueness. Secondly,

the penalisation of incitement, complicity and the attempt to commit this offence would be an effective approach to limit or reduce the impunity. Ransomware is an intrusion that would have to be prosecuted and sanctioned by its simple execution intentionally and without right because it affects by itself to the integrity of the devices or data. Thirdly, the prohibition and penalization of insurance contracts for the payment of ransom is a necessary measure to prevent legal instruments from becoming an additional incentive for criminals in a regrettable perversion of the system. Finally, international cooperation is the main instrument for dealing with the impunity arising from the fact that it is a generally transnational crime.

At the end, without a proper legal sanction, without a proper punishment, the crime will be repeated and extended because of the lack of harmful consequences for the aggressor. Technical solutions intended to mitigate the problem are essential, but without an adequate legal support it is difficult to fight effectively against this pandemic. It is imperative to coordinate and merge the technical and legal approaches to provide a feasible response to the problem of ransomware. Security and safety are at stake.

## ACKNOWLEDGEMENT

This work is partly supported by the Spanish Ministry of Economy and Competitiveness and ERDF (European Regional Development Fund) funds through project PID2020-114495RB-I00 and by the Network Engineering and Security Group (NESG).

## REFERENCES

[1] Sophos, "The State of Ransomware 2021," 2021, online resource: https://secure2.sophos.com/en-us/medialibrary/pdfs/whitepaper/sophos-state-of-ransomware-2021-wp.pdf.

[2] RansomwareTaskForce, "Combatting Ransomware. A Comprehensive Framework for Action: Key Recommendations from the Ransomware Task Force," 2021, available at https://securityandtechnology.org/ransomwaretaskforce/report/.

[3] K. Gandhi and P. Viral, "Survey on Ransomware: A New Era of Cyber Attack," *International Journal of Computer Applications*, vol. 168, no. 3, pp. 38–41, 2017, dOI: https://doi.org/10.5120/ijca2017914446.

[4] S. Aurangzeb, M. Aleem, M. Iqbal, and M. Islam, "Ransomware: A Survey and Trends," *Journal of Information Assurance and Security*, vol. 12, no. 2, pp. 48–58, 2017.

[5] P. Kumar and H. Bin Hj Ramlie, "Anatomy of Ransomware: Attack Stages, Patterns and Handling Techniques," in *Advances in Intelligent Systems and Computing*, 2021, pp. 205–214.

[6] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "A Survey on Detection Techniques for Cryptographic Ransomware," *IEEE Access*, vol. 7, pp. 144 925–144 944, 2019, dOI: https://doi.org/10.1109/ACCESS.2019.2945839.

[7] A. Zimba and M. Chishimba, "Understanding the Evolution of Ransomware: Paradigm Shifts in Attack Structures," *Int. Journal of Computer Network and Information Security*, vol. 11, no. 1, pp. 26–39, 2019.

[8] M. Humayun, N. Jhanjhi, A. Alsayat, and V. Ponnusamy, "Internet of things and ransomware: Evolution, mitigation and prevention," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 105–117, 2021.

[9] J. Tailor and A. Patel, "A Comprehensive Survey: Ransomware Attacks Prevention, Monitoring and Damage Control," *International Journal of Research and Scientific Innovation (IJRSI)*, vol. IV, no. VIS, pp. 116–121, 2017.

[10] F. Malecki, "Best practices for preventing and recovering from a ransomware attack," *Computer Fraud& Security*, vol. 3, pp. 8–10, 2019.

[11] F. Tang, B. Ma, J. Li, F. Zhang, J. Su, and J. Ma, "RansomSpector: An introspection-based approach to detect crypto ransomware," *Computers& Security*, vol. 97, p. 101997, 2020.

[12] M. Scalas, D. Maiorca, M. F., C. A. Visaggio, F. Martinelli, and G. Giacinto, "On the effectiveness of system API-related information for Android ransomware detection," *Computers& Security*, vol. 86, pp. 166–182, 2019.

[13] H. Rehman, E. Yafi, M. Nazir, and K. Mustafa, *Security Assurance Against Cybercrime Ransomware*. Springer, 2019, vol. 866, dOI: https://doi.org/10.1007/978-3-030-00979-3_3.

[14] L. Trautman and P. Ormerod, "Wannacry, Ransomware, and the Emerging Threat to Corporations," *Tennessee Law Review*, vol. 86.503, pp. 505–556, 2019.

[15] A. Bander, M. Maarof, and S. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, vol. 74, pp. 144–166, 2018.

[16] M. Conti, A. Gangwal, and S. Ruj, "On the Economic Significance of Ransomware Campaigns: A Bitcoin Transactions Perspective," *Computers & Security*, vol. 79, pp. 162–169, 2018, dOI: https://doi.org/10.1016/j.cose.2018.08.008.

[17] EUROPOL, "Internet Organised Crime Threat Assessment. European Union Agency for Law Enforcement Cooperation," 2020, available at https://www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2020.

[18] U. Tatar, Y. Gokce, and N. B., "Law versus technology: Blockchain, GDPR, and tough tradeoffs," *Computer Law & Security Review*, vol. 38, no. 1, p. 05454, 2020.

[19] A. Mohanta, M. Hahad, and K. Velmurugan, *Preventing Ransomware: Understand, prevent, and remediate ransomware attacks*. Packt, 2018.

[20] E. Nováčková, *Current Cyberthreats and Relevant Legal Instruments in EU and Canada*. Prage Law Working Papers Series-Charles University Law Faculty, 2018.

[21] J. Sherer, "Ransomware -Practical and Legal Considerations for Confronting the New Economic Engine of the Dark Web," *Richmond Journal of Law &Technology*, vol. XXIII, pp. 1–49, 2017.

[22] M. Brewczyńska, S. Dunn, and A. Elijahu, "Data Privacy Laws Response to Ransomware Attacks: A Multi-Jurisdictional Analysis," *In: Reins L. (eds) Regulating New Technologies in Uncertain Times. Information Technology and Law Series'*, vol. 32, 2019, dOI: https://doi.org/10.1007/978-94-6265-279-8_15.

[23] H. Salvi and R. Kerkar, "Ransomware: A Cyber Extortion," *Asian Journal of Convergence in Technology*, vol. 2, no. 2, pp. 1–6, 2015.

[24] ComissionAdHoc, "Le droit penal à l'épreuve des cyberattaques," 2021, rapport du Club de Juristes, Paris.

[25] TrendLabs, "2016 1H Security Roundup: The Reign of Ransomware," 2016, online resource: http://www.trendmicro.co.uk/vinfo/uk/security/research-and-analysis/threat-reports/roundup.

[26] TrendMicro, "Ransomware: Past, Present, and Future," 2016, online resource: https://documents.trendmicro.com/assets/wp/wp-ransomware-past-present-and-future.pdf.

[27] KeepnetLabs, "Top 11 Ransomware Attacks in 2020-2021," 2021, available at https://www.keepnetlabs.com/top-11-ransomware-attacks-in-2020-2021/.

[28] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks," in *12th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2015, pp. 1–20.

[29] Cybereason, "Cybereason's Newest Honeypot Shows How Multistage Ransomware Attacks Should Have Critical Infrastructure Providers on High Alert," 2020, https://www.cybereason.com/blog/cybereason-honeypot-multistage-ransomware.

[30] J. Gómez-Hernández, L. Álvarez González, and P. García-Teodoro, "Locker: Thwarting ransomware action through a honeyfile-based approach," *Computers & Security*, vol. 73, pp. 389–398, 2018.

[31] C. Bijitha, R. Sukumaran, and H. Nath, "A survey on ransomware detection techniques," *In: Sahay S., Goel N., Patil V., Jadliwala M. (eds) 'Secure Knowledge Management In Artificial Intelligence Era. Communications in Computer and Information Science'*, vol. 1186, pp. 55–68, 2020, dOI: https://doi.org/10.1007/978-981-15-3817-9_4.

[32] McAfee, "How to Protect Against Ransomware," 2020, https://www.mcafee.com/enterprise/en-us/assets/solution-briefs/sb-how-to-protect-against-ransomware.pdf.

[33] A. Arabo, R. Dijoux, T. Poulain, and G. Chevalier, "Detecting Ransomware Using Process Behavior Analysis," *Procedia Computer Science*, vol. 168, pp. 289–296, 2020, dOI: https://doi.org/10.1016/j.procs.2020.02.249.

[34] L. Connolly and D. Wall, "The rise of crypto-ransomware in a changing cybercrime landscape: Taxonomising countermeasures," *Computers & Security*, vol. 87, pp. 1–18, 2019.

# AndroCIES: Automatización de la certificación de seguridad para aplicaciones Android

Manuel Ruiz
NICS Lab,
Universidad de Málaga
Campus de Teatinos s/n
29071, Málaga
mrr@lcc.uma.es

Rubén Ríos
NICS Lab,
Universidad de Málaga
Campus de Teatinos s/n
29071, Málaga
ruben@lcc.uma.es

Rodrigo Román
NICS Lab,
Universidad de Málaga
Campus de Teatinos s/n
29071, Málaga
roman@lcc.uma.es

Antonio Muñoz
NICS Lab,
Universidad de Málaga
Campus de Teatinos s/n
29071, Málaga
amunoz@lcc.uma.es

Juan Manuel Martínez
DEKRA Testing and Certification, S.A.U.
Málaga TechPark, Severo Ochoa, 2&6, 29590, Málaga
juanmanuel.martinez@dekra.com

Jorge Wallace
DEKRA Testing and Certification, S.A.U.
Málaga TechPark, Severo Ochoa, 2&6, 29590, Málaga
jorge.wallace@dekra.com

***Resumen*—El auge de las plataformas móviles está impulsando el desarrollo de un gran número de aplicaciones, muchas de las cuales salen al mercado sin las convenientes comprobaciones de seguridad. Recientemente, Google está apostando por hacer este problema más visible y concienciar a los usuarios de la necesidad de instalar aplicaciones verificadas por laboratorios independientes. Sin embargo, la certificación de aplicaciones suele ser una tarea ardua y no exenta de errores. Por ello, en este trabajo, presentamos la herramienta AndroCIES, que es capaz de automatizar en gran medida las evaluaciones necesarias para la certificación de aplicaciones móviles, reduciendo en torno a un 20 % el tiempo empleado en este proceso.**

***Index Terms*—certificación, seguridad, aplicaciones, análisis estático**

## I. Introducción

Desde su primer lanzamiento público en septiembre de 2008, el sistema operativo para dispositivos móviles, Android, ha recibido una gran acogida por parte de los fabricantes, de los usuarios y de la comunidad de desarrolladores de software. En 2020, había más de 1000 millones de usuarios de Android [1] y su proveedor oficial de aplicaciones, *Google Play*, actualmente cuenta con más de 2,5 millones de apps en su tienda online [2]. Este auge se debe, en gran parte, a su desarrollo accesible basado en un lenguaje de programación común (Java) y la enorme disponibilidad de librerías externas.

Debido a la enorme popularidad de Android, no es difícil encontrar un sinfín de aplicaciones de baja calidad y mal diseñadas desde el punto de vista de la ciberseguridad. De hecho, según [2], casi el 40 % de las aplicaciones existentes en Google Play son de baja calidad. La existencia de este tipo de aplicaciones no sólo afecta negativamente a la seguridad de los usuarios, sino también a la reputación de sus desarrolladores [3]. A fin de evitarlo, Google ha puesto en marcha en su tienda oficial el programa de Evaluación de Seguridad de Aplicaciones Móviles (MASA por sus siglas en inglés), con el objetivo de proporcionar a los usuarios un mecanismo fiable que les permita reconocer fácilmente aquellas aplicaciones que han sido validadas por laboratorios de seguridad independientes [4].

El programa MASA se basa fundamentalmente en los criterios de seguridad establecidos por las normas OWASP (Open Web Application Security Project) para dispositivos móviles. OWASP ha definido una serie de requisitos de seguridad para aplicaciones móviles (MASVS [5]) además de unos casos de prueba (MSTG [6]), que pueden realizarse a través de diversas herramientas para el análisis del código fuente de las aplicaciones (análisis estático) y del comportamiento durante su ejecución (análisis dinámico).

Sin embargo, cabe destacar que, a pesar de existir guías de evaluación y herramientas que permiten realizar gran parte de las pruebas indicadas en tales guías, el proceso sigue siendo una tarea muy manual, que consume gran cantidad de tiempo y es propenso a errores. Así pues, se hace necesaria la creación de herramientas que permitan automatizar lo más posible este proceso de evaluación de aplicaciones móviles, facilitando la tarea de los laboratorios de certificación al tiempo que se reduce la posibilidad de pasar por alto problemas de seguridad, más aún cuando Google Play reconocerá en su tienda a aquellas aplicaciones que hayan sido validadas respecto a un conjunto de requisitos de MASVS.

### *I-A. Motivación y objetivos*

Entre las líneas de negocio de la empresa DEKRA Testing and Certification (en adelante DEKRA) se encuentra la certificación de seguridad de aplicaciones móviles [7]. Su dilatada experiencia en este ámbito y la creciente demanda de este servicio por parte de sus clientes, les ha hecho detectar la necesidad de mejorar su proceso de certificación de aplicaciones. A pesar de contar con diversas herramientas y guías, como la OWASP MSTG, su proceso de certificación requiere de una gran intervención humana. Las herramientas disponibles en el mercado, aunque ofrecen información muy valiosa, no permiten dar un claro veredicto para todos los casos de prueba que requiere el estándar. En otras ocasiones, esta información se encuentra repartida en diferentes partes de la salida de una o varias herramientas. Es aquí donde se hace necesario contar con expertos capaces de interpretar los resultados aportados por las herramientas para así poder

ofrecer un veredicto final sobre el caso de prueba. Esto es un tarea tediosa y propensa a errores.

Esta necesidad ha sido materializada en un proyecto de investigación conjunto entre DEKRA y la Universidad de Málaga, denominado CIES. Entre los objetivos de este proyecto se encontraba el desarrollo de una herramienta de evaluación que permitiese a DEKRA reducir el tiempo y esfuerzo empleado en la evaluación de aplicaciones móviles por el equipo de certificación. Más concretamente, la herramienta está diseñada para facilitar la automatización de pruebas de certificación frente a estándares actuales y futuros, como el OWASP MASVS, que es el estándar de facto de la industria y el elegido por Google Play.

A tal fin, la herramienta debería combinar, clasificar y categorizar la información procedente de varias herramientas de análisis, en función de los casos de prueba establecidos por los diferentes estándares. Idealmente la herramienta debería permitir presentar un veredicto automatizado o, en su defecto, agruparía toda la información necesaria para que el experto pudiera realizar un veredicto con facilidad. De esta forma, no sería necesario que el experto ejecutara varias herramientas y recopilara la información resultante de su ejecución, sino que se limitaría a seguir paso a paso la información expuesta para realizar la verificación del estándar, ahorrando tiempo y esfuerzo en el proceso.

### *I-B. Estructura del documento*

El resto del artículo se organiza según la estructura descrita a continuación. En primer lugar, la sección II, se analiza brevemente el estado del arte y de la técnica, entrando en detalle en los principales estándares actuales y en desarrollo. A continuación, la sección III introduce los requisitos y necesidades de la herramienta desarrollada. En la sección IV se describe la arquitectura de la herramienta en base a las secciones anteriores, mientras que en la sección V se aborda la implementación realizada y su funcionamiento. En la sección VI, se muestran las ventajas que otorga la herramienta. Para finalizar, en la sección VII, se muestran las conclusiones del trabajo y posibles líneas de investigación y desarrollo futuro.

## II. Trabajos relacionados

### *II-A. Estándares de seguridad en móviles*

Existen diversos estándares que definen los requisitos mínimos de seguridad y privacidad que una aplicación móvil debe cumplir. Entre ellos, destacan el OWASP Mobile Application Security Verification Standard (MASVS) [5] y el ioXt Mobile Application Profile [8]. El OWASP MASVS es un estándar que establece requisitos de seguridad para aplicaciones móviles y se utiliza en conjunto con la OWASP Mobile Security Testing Guide (MSTG) [6], un manual para el análisis de la seguridad de aplicaciones e ingeniería inversa. En la MSTG se describen procesos técnicos para verificar los casos de prueba necesarios para cumplir con el OWASP MASVS. El ioXt Mobile Application Profile también es un estándar dedicado a la certificación de seguridad de aplicaciones móviles. Dentro de este estándar se definen también una serie de especificaciones de seguridad que otorgan al fabricante los derechos para usar la marca de ioXt Compilance.

Ambos estándares constan de requisitos similares, pero presentan diferencias entre ellos. Aunque ambos son muy completos, el ioXt Mobile Application Profile define requisitos de un nivel más abstracto que el OWASP MASVS, centrándose más en el diseño y la arquitectura que la implementación. Además, el OWASP MASVS presenta una guía de verificación, la ya mencionada OWASP MSTG, lo que le permite centrarse en problemas más concretos. Dado el carácter específico de OWASP MASVS y al interés de Google Play por este dentro de su MASA [4], en este trabajo se ha priorizado el OWASP MASVS y la OWASP MSTG aunque, como se detalla en la sección V, AndroCIES también implementa parte del estándar ioXt.

El OWASP MASVS está compuesto por dos niveles de seguridad. El nivel L1 cubre las buenas prácticas respecto a la seguridad en el desarrollo de aplicaciones. Abarca requisitos básicos relacionados con la calidad del código, el manejo de datos sensibles, y la interacción con el entorno Android. Se ajusta a todas las aplicaciones. Por otra parte, el nivel L2 va dirigido a aplicaciones críticas que necesitan unos requisitos de seguridad más estrictos. En este nivel, la seguridad debe ser parte de la arquitectura de la aplicación, y debe existir un modelo de amenaza. Está orientado a aplicaciones que manejan información altamente sensible, como aplicaciones de banca electrónica.

El nivel L1 de MASVS consta de un total de siete categorías dedicadas a diferentes aspectos de la aplicación: arquitectura, almacenamiento, criptografía, autenticación, comunicación, plataforma y código. Cada una de estas categorías consta de varios requisitos, algunos de estos son de alto nivel mientras que otros son más específicos. En total hay 47 requisitos definidos en el nivel L1 de MASVS. Google MASA considera únicamente un subconjunto de estos requisitos.

### *II-B. Estado del arte*

El análisis de seguridad de aplicaciones se ha tratado en la literatura desde muchos enfoques diferentes pero los predominantes son el análisis estático y el análisis dinámico. El análisis estático consiste en examinar el código fuente de la aplicación, aunque en ocasiones suele partirse de los propios binarios, sin llevar a cabo ningún tipo de ejecución. Durante este análisis, se buscan malas prácticas de seguridad, que pueden ser intencionadas (p.ej., la aplicación incluye un código malicioso) o no intencionadas (p.ej., el propio código incluye credenciales). En cambio, el análisis dinámico se centra en analizar el comportamiento de la aplicación durante su ejecución sobre un dispositivo real o emulado. En este tipo de análisis es importante lanzar un número suficiente de ejecuciones que abarque el mayor número de flujos de ejecución posibles dentro de la aplicación.

El análisis de seguridad de aplicaciones se ha tratado en numerosas ocasiones en la literatura, por ello, nos centraremos en hacer referencia a los trabajos de revisión más recientes y relevantes hasta la fecha. En cuanto al análisis estático, Li et al. [9] realizan una revisión de la misma desde el punto de vista de la detección de fallos no intencionados, mientras que Pan et al. [10] y Jusoh et al. [11] la estudian desde el punto de vista del análisis de malware. Por otra parte, el análisis dinámico también ha recibido gran atención por parte de la comunidad científica. Cabe destacar el trabajo de Kong

et al. [12], donde igualmente se recoge una revisión de la literatura.

De los dos tipos de análisis mencionados, el análisis estático es el más prestado a la automatización ya que presenta una menor complejidad al basarse únicamente en archivos fuente y no necesitar la presencia de una plataforma de ejecución.

### *II-C. Estado de la técnica*

Debido al auge y popularidad del sistema operativo Android, se han desarrollado numerosas herramientas destinadas al análisis de sus aplicaciones.

Algunas de estas herramientas de análisis (p.ej., Ostorlab [17], Kryptowire [18] y NowSecure [19]) son servicios privados en la nube ofrecidos por empresas, mientras que otras (p.ej., AndroShield [13], AndrotomistLite [14], MARA Framework [15] y MobSF [16]), son proyectos de código abierto que puede utilizar cualquiera. Las herramientas ofrecidas como servicios cloud privados, aunque bastante completas, no han sido consideradas para la construcción de nuestra herramienta de automatización al ser difícil su integración y evitar así la dependencia de servicios de terceros.

Las aplicaciones de código abierto son fácilmente integrables y adaptables, permitiendo ser ejecutadas de forma local. Sin embargo, no todas ofrecen una funcionalidad suficiente para abarcar los casos de prueba establecidos por los estándares revisados en la sección II-A. AndroShield es la que, desde nuestro punto de vista, ofrece una funcionalidad más limitada. Esta herramienta permite analizar tanto el código fuente como el Android Manifest[1], pero con un nivel de detalle insuficiente. AndrotomistLite es una herramienta ligeramente más avanzada que la anterior. La parte de análisis estático está compuesta por APKProfiler [20], una herramienta capaz de descompilar las aplicaciones y extraer información del Manifest, el código y el certificado de la misma. A continuación tenemos MARA Framework, que consta de un conjunto de 14 herramientas capaces de cubrir muchos aspectos de interés, como el análisis de permisos, del Manifest, de código, de certificados, y varios más. La principal desventaja que presenta MARA Framework es la forma en que presenta los resultados, ya que simplemente aporta las salidas de las herramientas por separado, cada una en su propio formato. Finalmente, la herramienta MobSF, aunque también hace uso de varias herramientas entre las cuales se encuentra algunas de las presentes en MARA Framework, con los datos obtenidos añade un análisis propio.

En resumen, la herramienta MobSF cubre prácticamente todos los aspectos de análisis estático abordables por los estándares de seguridad en aplicaciones móviles. Además, la información de salida generada se recoge en una única base de datos y, como ventaja adicional, realiza un juicio de severidad para alguna de la información presentada. Sin embargo, de las herramientas de código abierto que se mencionan, casi ninguna realiza un análisis de las librerías externas que utilizan las aplicaciones. MobSF es la única que realiza este análisis pero de manera muy superficial. Más adelante, en la sección IV, mostraremos cómo es posible solucionar esta limitación de MobSF. En la Tabla I se ofrece un resumen de la comparación realizada.

[1]En el Manifest se definen datos de interés para analizar la seguridad de una aplicación Android como los permisos o las características hardware necesarias para su ejecución.

## III. Análisis de Requisitos

El objetivo principal de este trabajo es la construcción de una herramienta que permita automatizar el proceso de certificación de seguridad de aplicaciones móviles basadas en Android. Tanto este objetivo como los requisitos que se detallan a continuación han sido obtenidos a lo largo de varias reuniones entre los miembros del equipo de investigación y desarrollo de la Universidad de Málaga y el equipo de certificación de DEKRA.

- **Completitud**: la herramienta debe ser capaz de evaluar todos los casos de prueba del MASVS L1 que DEKRA considera más relevantes, los cuales están alineados con lo establecido por el MASA de Google Play.
- **Extensibilidad**: la herramienta debe ser capaz de permitir la evaluación de estándares actuales y futuros con relativa facilidad en base a los casos de prueba establecidos en tales estándares sin necesidad de rediseñar la herramienta.
- **Portabilidad**: la herramienta debe ser capaz de ser desplegada en diferentes sistemas operativos y entornos, físicos o virtualizados.
- **Persistencia**: la herramienta debe ser capaz de garantizar la persistencia de los resultados obtenidos tras analizar las aplicaciones.
- **Soporte multi-usuario**: la herramienta debe ser capaz de ofrecer el servicio de análisis a múltiples expertos a la vez.
- **Usabilidad**: la herramienta debe ofrecer una interfaz de usuario sencilla, que permita al experto analizar los resultados forma clara y concisa.

Con todo esto, objetivo y requisitos, se procede al diseño de la herramienta, que detallaremos en la siguiente sección.

## IV. Diseño

Durante el diseño de la herramienta se ha adoptado una serie de decisiones que permiten acometer los diversos requisitos planteados en la sección anterior. Respecto a la completitud del análisis, se han añadido dos herramientas dedicadas exclusivamente al análisis de librerías externas denominadas LibScout [21] y OWASP Dependency Check [22]. Ambas siguen un funcionamiento similar: realizan un perfil de las librerías que se están utilizando, recogen los identificadores y los comparan con sus bases de datos para ver si se detecta alguna vulnerabilidad. Además, también se ha realizado un fork propio [23] de la herramienta MobSF donde se añaden algunas reglas de detección para el código y una extensión del análisis del Android Manifest.

Para permitir la modificación y creación de nuevos estándares, se ha adoptado una arquitectura modular. Esta arquitectura puede ver reflejada en la Figura 1, dentro del contenedor amarillo de mayor tamaño. Partiendo desde arriba, el módulo de Front-End se encargará de recoger los informes proporcionados por los módulos dedicados a cada uno de los estándares. Estos, a su vez, procesan la información transmitida por los módulos de sus secciones. Las secciones son los módulos encargados de analizar la información y, si la complejidad lo permite, proporcionar un veredicto. Si alguno de los estándares tuviera secciones que definen requisitos similares, los módulos pueden ser reutilizados.

Tabla I
COMPARACIÓN DE HERRAMIENTAS DE ANÁLISIS ESTÁTICO PARA APLICACIONES ANDROID

| | Análisis | | | | | | | Clasificación de | Formato de |
|---|---|---|---|---|---|---|---|---|---|
| | Permisos | Manifest | Código | Certificado | URLs | Librerías externas | SaaS | severidad | salida |
| AndroShield [13] | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | Página web |
| AndrotomistLite [14] | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | Fichero txt |
| MARA Framework [15] | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | Ficheros txt y json |
| MobSF [16] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | Base de datos |
| Ostorlab [17] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Página Web |
| Kryptowire [18] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Página Web |
| NowSecure [19] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Página Web |

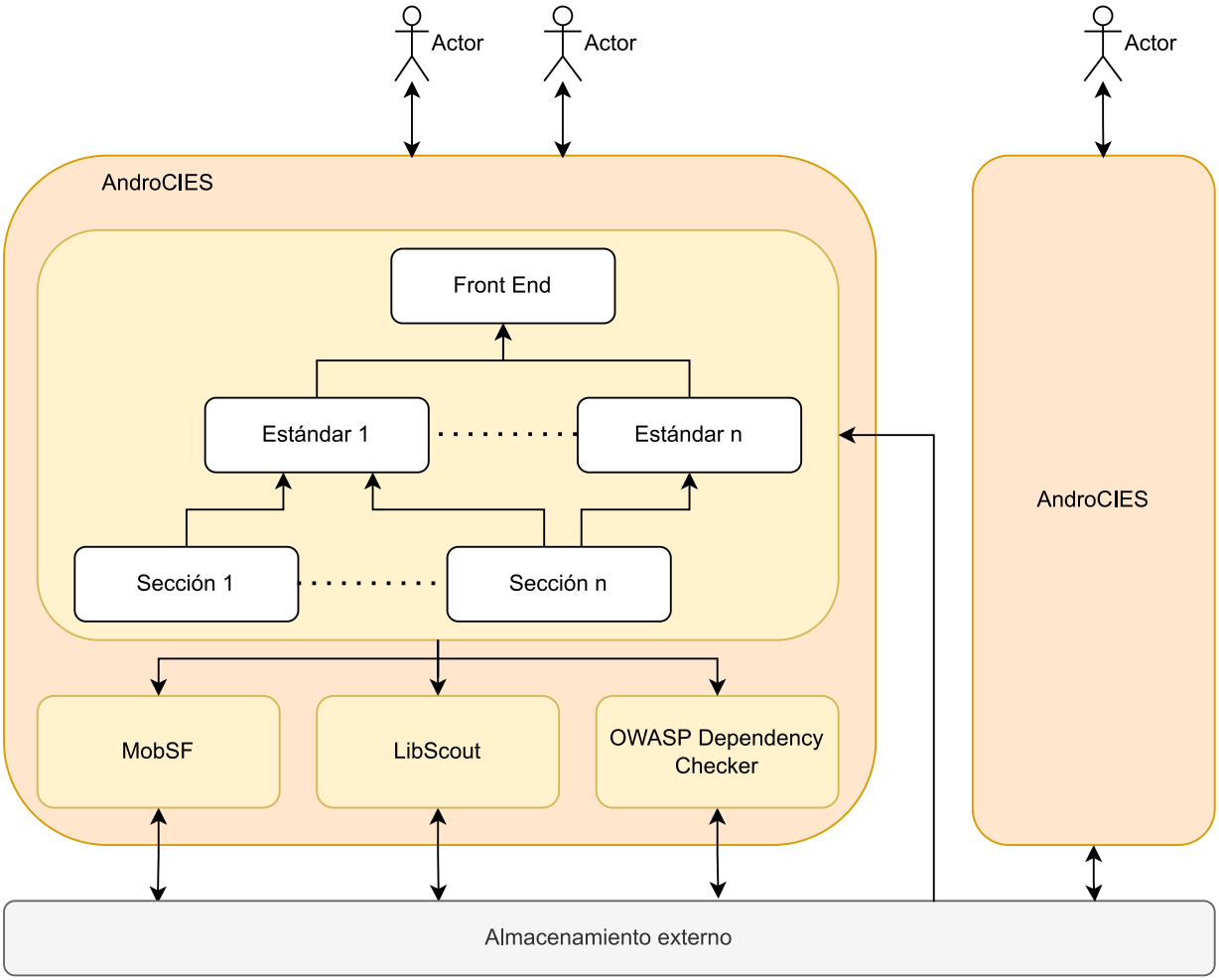



Figura 1. Estructura de la herramienta

Para conseguir la portabilidad del sistema, se ha diseñado un despliegue basado en contenedores. A la hora de dividir las herramientas en contenedores, debido a las dependencias, al tamaño y a la disponibilidad de las mismas, se ha decidido crear dos contenedores. El primero contendría la herramienta desarrollada, así como las herramientas de LibScout y OWASP Dependency Check, mientras que el segundo estaría formado por la herramienta MobSF. Sin embargo, debido a la volatilidad de los mismos, ha sido necesario crear una zona de almacenamiento externo para garantizar la persistencia de los datos. Ambos contenedores tendrán acceso a dicho almacenamiento.

Para garantizar la facilidad de uso y una experiencia multiusuario, se ha decidido una interfaz accesible mediante navegador web. Esta interfaz será abordada utilizando una arquitectura del tipo Modelo-Vista-Controlador. De esta forma, se eliminan las dependencias entre la información generada por la herramienta y la muestra de los datos al usuario, permitiendo la reutilización de la información en otros sistemas. Además, cabe mencionar que debido al desacoplamiento el almacenamiento externo, es posible lanzar dos instancias de la herramienta accediendo a las mismas bases de datos. Esto permite dedicar dos máquinas distintas al análisis y ver en cada una de ellas los resultados obtenidos de ambas, lo que facilita la experiencia de varios usuarios de forma paralela.

La arquitectura general del sistema se puede ver en la Figura 1. En ella se representan dos instancias de AndroCIES conectadas al mismo almacenamiento externo. Se observan los

componentes internos de AndroCIES, como MobSF, LibSCcout, OWASP Dependency Check y la parte desarrollada en JavaScript encargada de la recogida de la información y el análisis de datos.

## V. Implementación y funcionamiento

Tras las consideraciones e ideas presentadas en las secciones III y IV, se ha decidido realizar la implementación con NodeJS. NodeJS es un framework de ejecución de código JavaScript que facilita la implementación de la interfaz web y la modularidad. Además, para la interfaz web se han utilizado plantillas dinámicas de HTML mediante Handlebars. Para la contenerización se ha utilizado Docker y como orquestador se utiliza Docker Swarm, facilitando así el despliegue de la aplicación.

El funcionamiento es el siguiente:

1. El usuario se conecta a la interfaz web y selecciona las aplicaciones sobre las que quiere realizar el análisis.
2. AndroCIES se encarga de enviar las aplicaciones seleccionadas al resto de herramientas de análisis y comenzar el proceso.
3. Las herramientas proporcionan los resultados del análisis en su propio formato. MobSF almacena el resultado en una base de datos, mientras que LibScout y OWASP Dependency Check lo almacenan en ficheros JSON.
4. AndroCIES recoge la información a través de los módulos de entrada, donde se procesa la información.
5. Los módulos de entrada pasan la información a sus módulos superiores, los de estándar, quienes organizan la información en el formato de salida.
6. Los módulos de estándar pasan la información generada al módulo de Front-End que será el encargado de mostrarlo por pantalla.
7. El usuario obtiene el informe generado por pantalla.

La mayoría de módulos de sección obtienen la información de la base de datos de MobSF. Principalmente se extraen datos de la sección del análisis de código, pero también participan otras como la parte de análisis de permisos, de certificado, del Android Manifest, y de los elementos exportados, entre otros. Las herramientas LibScout y OWASP Dependency Check se especializan en el análisis de librerías externas, y son utilizadas por una sola sección. Además, también existen módulos propios de la herramienta dedicados a analizar el archivo de Android Manifest para ampliar la información proporcionada por MobSF.

En la Figura 2 se muestra la interfaz gráfica de la herramienta desarrollada. En la parte superior existe un menú que permite la navegación entre las diferentes vistas de la herramienta. Una de ellas permite seleccionar las aplicaciones a analizar, mientras que la otra, la mostrada en la figura, presenta los informes generados. En la parte izquierda se pueden ver las diferentes secciones de los estándares implementados. En concreto, se ve el estándar OWASP MASVS y sus diferentes categorías, seguido del estándar ioXt y algunas de sus categorías. Este último se encuentra aún en desarrollo. En el informe generado, lo primero que se observa es el nombre de la aplicación analizada (UMA) junto a un identificador único y dos enlaces con información adicional: el informe generado por MobSF sin procesar y el archivo *AndroidManifest.xml*. A continuación, se muestran los resultados para cada una de las secciones del OWASP MASVS, en este caso, los dos primeros apartados de la primera categoría. De esta forma, la información relevante que aparecería dispersa en otras herramientas, es agrupada y mostrada de manera amigable en la sección correspondiente. Por ejemplo, en la sección *Storage-2*, se detecta la existencia de permisos de escritura en el almacenamiento externo y no se detecta de creación de archivos temporales. Esta información aparecería separada en las secciones de análisis de permisos y análisis de código en la herramienta MobSF, lo cual dificulta la realización de un veredicto sobre este requisito.

## VI. Evaluación y Resultados

Tras la implementación de la herramienta, se realizó un primer análisis de seguridad a varias aplicaciones reconocidas de la tienda oficial Google Play. El objetivo de este análisis era comprobar la efectividad de la herramienta a la hora de evaluar los casos de prueba del OWASP MASVS. Estas aplicaciones fueron analizadas de manera independiente por el grupo de investigación de la Universidad de Málaga y por parte de DEKRA, utilizando sus procesos tradicionales. Dichas aplicaciones pertenecen principalmente a las categorías de almacenamiento de fotos, traducción de texto, gestión de contactos y gestión de conexiones VPN.

Los resultados fueron muy satisfactorios ya que los resultados obtenidos por ambos equipos fueron prácticamente idénticos.

Tras esta primera validación, el equipo de certificación de DEKRA ha estado utilizando AndroCIES para el análisis de seguridad de otras aplicaciones móviles. En promedio, se ha determinado una reducción de un 20 % del tiempo empleado para la evaluación respecto al estándar OWASP MASVS utilizando la metodología definida en la OWASP MSTG. Esta es una reducción considerable, teniendo en cuenta el margen de mejora existente si se continúa el trabajo en la misma línea.

## VII. Conclusión

El auge en el desarrollo de aplicaciones móviles y el interés de Google en una Play Store más segura y transparente está empujando a los laboratorios especializados a mejorar sus procesos de certificación de seguridad de aplicaciones. En este trabajo se ha realizado un análisis de los principales aspectos que son necesarios para cumplir con los principales estándares de seguridad propuestos por la industria y se han estudiado herramientas que permiten evaluar, aunque no de manera completa, el cumplimiento de esos estándares. A partir de este estudio se ha desarrollado AndroCIES, un herramienta capaz de realizar de manera automática estas pruebas y presentar los resultados de forma intuitiva y organizada a los expertos, reduciendo así el tiempo dedicado al proceso de verificación del estándar OWASP MASVS en torno a un 20 %.

A pesar de que los resultados ofrecidos por la herramienta son prometedores, existen varios aspectos susceptibles de mejora, que abordaremos en el futuro. Aunque una posible línea de trabajo es la adición de nuevos estándares para la certificación de aplicaciones móviles, se trata de una tarea relativamente sencilla de acometer debido al diseño modular de nuestra herramienta. Así pues, consideramos que las dos líneas principales de trabajo futuro serían, por un lado, la

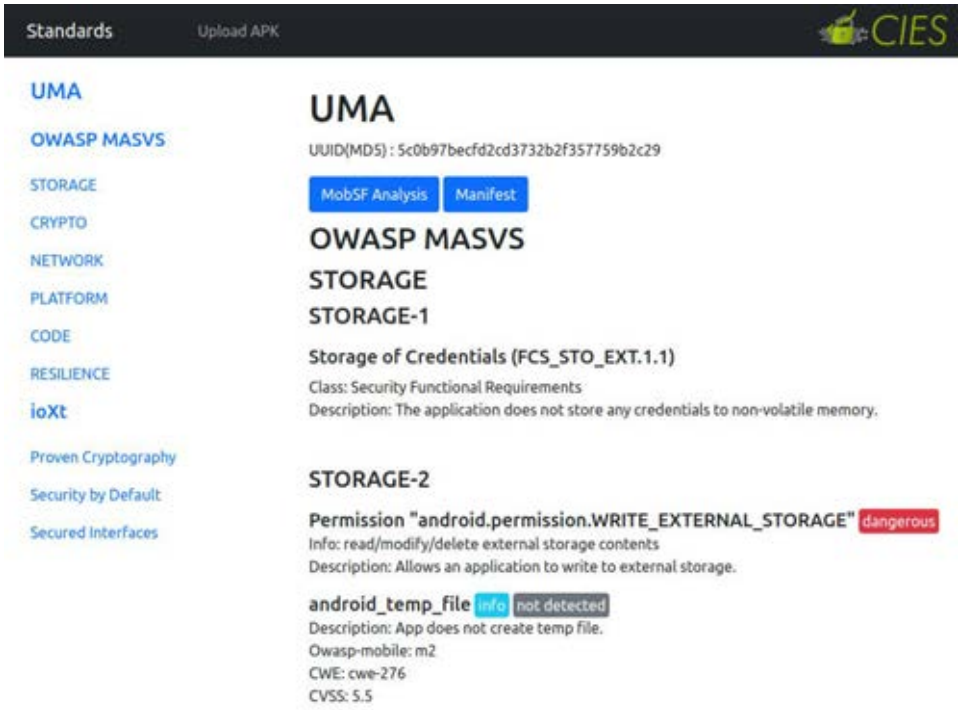



Figura 2. Interfaz de la herramienta

incorporación de técnicas de análisis dinámico de código y, por otro lado, el otorgar a la herramienta de cierta inteligencia. En este sentido, la inteligencia iría orientada hacia la detección de posibles falsos positivos, así como a aprender de los veredictos realizados por el analista experto que hace uso de la propia herramienta.

## Agradecimientos

Este trabajo ha sido financiado en parte por la Corporación Tecnológica de Andalucía (CTA) bajo la subvención 21/1046 y la Consejería de Empleo, Empresa y Comercio de la Junta de Andalucía a través de la Agencia de Innovación y Desarrollo de Andalucía (IDEA) (pendiente aprobación).

## Referencias

[1] M. Mena Roa. (2021) Android e ios dominan el mercado de los smartphones. [Online]. Available: https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/

[2] AppBrain. (2022) Number of android apps on google play. [Online]. Available: https://www.appbrain.com/stats/number-of-android-apps

[3] H. Wang, H. Li, L. Li, Y. Guo, and G. Xu, "Why are android apps removed from google play? a large-scale empirical study," in *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2018, pp. 231–242. [Online]. Available: https://doi.ieeecomputersociety.org/

[4] Google. (2022) Mobile application security assessment. [Online]. Available: https://appdefensealliance.dev/masa

[5] OWASP. (2022) Owasp mobile application security verification standard (masvs). [Online]. Available: https://github.com/OWASP/owasp-masvs

[6] ——. (2022) Owasp mobile security testing guide (mstg). [Online]. Available: https://github.com/OWASP/owasp-mstg

[7] DEKRA. (2021) Dekra autorizado por ioxt alliance para realizar evaluaciones de ciberseguridad en aplicaciones móviles. [Online]. Available: https://www.dekra.es/es/dekra-autorizado-por-ioxt-alliance-para-realizar-pruebas-ciberseguridad-aplicaciones-moviles-vpn/

[8] ioXt Alliance. (2020) ioxt mobile application profile. [Online]. Available: https://static1.squarespace.com/static/5c6dbac1f8135a29c7fbb621/t/604aa3fa668a8e3b50630433/1615504379349/Mobile_Application_Profile.pdf

[9] L. Li, T. Bissyandé, M. Papadakis, S. Rasthofer, A. Bartel, D. Octeau, J. Klein, and L. Traon, "Static analysis of android apps: A systematic literature review," *Information and Software Technology*, vol. 88, pp. 67–95, 8 2017.

[10] Y. Pan, X. Ge, C. Fang, and Y. Fan, "A systematic literature review of android malware detection using static analysis," *IEEE Access*, vol. 8, pp. 116 363–116 379, 2020.

[11] R. Jusoh, A. Firdaus, S. Anwar, M. Z. Osman, M. F. Darmawan, and M. F. Ab Razak, "Malware detection using static analysis in android: a review of feco (features, classification, and obfuscation)," *PeerJ Computer Science*, vol. 7, 2021.

[12] P. Kong, L. Li, J. Gao, K. Liu, T. Bissyande, and J. Klein, "Automated testing of android apps: A systematic literature review," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 45–66, 3 2019.

[13] A. Amin, A. Eldessouki, M. T. Magdy, N. Abdeen, H. Hindy, and I. Hegazy, "Androshield: Automated android applications vulnerability detection, a hybrid static and dynamic analysis approach," *Information*, vol. 10, no. 10, p. 326, 2019.

[14] V. Kouliaridis. (2022) Androtomistlite. [Online]. Available: https://github.com/billkoul/AndrotomistLite

[15] C. Kisutsa. (2019) Mobile application reverse engineering and analysis (mara) framework. [Online]. Available: https://github.com/xtiankisutsa/MARA_Framework

[16] A. Abraham. (2022) Mobile security framework (mobsf). [Online]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF

[17] Ostorlab. (2022) Mobile security testing. [Online]. Available: https://www.ostorlab.co/

[18] Kryptowire. (2022) Kryptowire mobile app security testing. [Online]. Available: https://www.kryptowire.com/

[19] NowSecure. (2022) Nowsecure: Deliver secure mobile apps. [Online]. Available: https://www.nowsecure.com/

[20] I. Grevenitis. (2022) An android application decompilation and feature extraction library. [Online]. Available: https://github.com/Giannisgre/APKProfiler

[21] E. Derr. (2019) Libscout. [Online]. Available: https://github.com/reddr/LibScout

[22] OWASP. (2022) Owasp dependency-check. [Online]. Available: https://owasp.org/www-project-dependency-check/

[23] M. Ruiz Ruiz. (2022) Fork mobsf. [Online]. Available: https://github.com/roxax19/Mobile-Security-Framework-MobSF

# Aprendizaje Federado con Agrupación de Modelos para la Detección de Anomalías en Dispositivos IoT Heterogéneos

Xabier Sáez-de-Cámara
Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA)
Mondragon Unibertsitatea
xsaezdecamara@ikerlan.es

Jose Luis Flores
Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA)
jlflores@ikerlan.es

Cristóbal Arellano
Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA)
carellano@ikerlan.es

Aitor Urbieta
Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA)
AUrbieta@ikerlan.es

Urko Zurutuza
Mondragon Unibertsitatea
uzurutuza@mondragon.edu

***Resumen*—Hace ya unos años que varios ataques de denegación de servicio distribuido realizados por la *botnet* de Mirai, formada en gran medida por dispositivos IoT, dejaron inaccesibles distintas plataformas de Internet durante un tiempo. Dese entonces, se ha observado una tendencia creciente de ciberataques contra los dispositivos IoT, y la complejidad de estas amenazas también está aumentando. Los métodos de aprendizaje automático están mostrando resultados prometedores para detectar estas amenazas; sin embargo, las arquitecturas basadas en la computación en la nube o el perímetro para el entrenamiento de estos modelos presentan múltiples inconvenientes en entornos IoT, como la sobrecarga de la red o el aislamiento de los datos. En este trabajo presentamos una arquitectura de *Federated Learning* (FL) para el entrenamiento de modelos no supervisados de detección de anomalías en redes IoT. La arquitectura incluye un algoritmo de agrupación de dispositivos integrado en el proceso de FL para abordar los problemas causados por la alta heterogeneidad en estos entornos. Evaluamos la propuesta sobre un banco de pruebas con 360 dispositivos IoT simulados, mostrando la detección de varios ataques de denegación de servicio y comunicación mando y control.**

***Index Terms*—Botnet, Detección de anomalías, Detección de intrusiones, Internet of Things, Machine learning**

## I. Introducción

La creciente adopción del Internet de las Cosas (IoT) está permitiendo una mayor digitalización y optimización de los recursos [1]. Sin embargo, este nivel de conectividad, unido a las malas prácticas de seguridad en estos dispositivos [2], conlleva una mayor superficie de ataque tanto para los dispositivos IoT domésticos como industriales [3], [4]. En consecuencia, esta situación ha dado lugar a la proliferación de *malware* diseñado para explotar estos dispositivos [5], [6].

Las distintas estrategias de mitigación, como por ejemplo el uso de sistemas operativos especializados, la eliminación de servicios no esenciales o los mecanismos de actualización [7], no garantizan un entorno seguro, ya que los errores de configuración, vulnerabilidades y el descubrimiento de *zero-days* hacen que los dispositivos IoT sigan siendo propensos a ser atacados [8]. Asimismo, el uso de herramientas de seguridad avanzadas como los sistemas de detección de intrusiones (IDS, *Intrusion Detection System*) presentan dificultades para mantenerse al día en entornos IoT debido al uso de técnicas de ofuscación [5] y de largos retrasos entre el análisis de *malware* y la publicación de las reglas correspondientes [9].

El uso de métodos basados en el aprendizaje automático (ML, *Machine Learning*) han mostrado resultados prometedores en esta área [10]. Sin embargo, a pesar de las ventajas del ML, la infraestructura necesaria para entrenar modelos en entornos IoT de gran escala presenta varias dificultades. Por una parte, las arquitecturas centralizadas basadas en la nube exhiben problemas como el alto consumo de ancho de banda, congestión de recursos de red, alta latencia o picos de tráfico [11]. Por otra parte, las técnicas de computación en el perímetro, que surgen como alternativa a las arquitecturas centralizadas, presentan problemas como el aislamiento de los datos, que dificulta la aplicación de ML ya que se reduce el volumen total de datos disponibles para el entrenamiento al realizarse aisladamente en cada dispositivo, perdiendo la posibilidad de agregar en el mismo conjunto de datos información de múltiples dispositivos y modelar comportamientos normales observados en otros segmentos de red [12].

Una alternativa prometedora que podría abordar los problemas mencionados es el aprendizaje federado (FL, *Federated Learning*) [13]. FL permite entrenar un modelo global a partir de datos distribuidos en múltiples dispositivos remotos sin la necesidad de centralizar previamente los datos. Sin embargo, todavía hay algunas dificultades que deben considerarse para un uso práctico de FL. Aunque FL asume que la generación de datos puede no ser independiente e idénticamente distribuida (IID) en todos los clientes, en la práctica, las configuraciones altamente no IID pueden obstaculizar la convergencia del modelo global [14]. Esto sucede en entornos altamente heterogéneos, como en grandes redes de dispositivos IoT. Las contribuciones de este trabajo se resumen en:

- Una arquitectura basada en FL para el entrenamiento de modelos no supervisados de detección de anomalías aplicada a redes de dispositivos IoT heterogéneos.
- Un método no supervisado para la agrupación de dispositivos IoT para abordar los problemas de convergencia del modelo global en entornos FL heterogéneos. El método está totalmente integrado en el proceso de FL y no necesita intervención humana.
- Resultados experimentales de la arquitectura propuesta sobre un banco de pruebas de 360 dispositivos IoT simulados que se comunican mediante MQTT y CoAP.

## II. Trabajos relacionados

Recientemente, han surgido varias propuestas sobre el uso de FL para la detección de intrusiones en IoT. Nguyen et al. [15] presentan DÏot, un sistema no supervisado para la detección de anomalías de red sobre dispositivos IoT de consumo. Agrupan dispositivos con un comportamiento similar usando una herramienta externa de identificación de dispositivos. Posteriormente, usan FL para entrenar un modelo global por cada grupo. Aplicado en un entorno similar, Rey et al. [16] desarrollan un framework basado en FL para detectar ciberataques contra dispositivos IoT y también ataques adversariales contra FL. En Popoola et al. [17] y Rahman et al [18] se presentan propuestas de sistemas de detección de intrusiones basados en FL, y se comparan con arquitecturas centralizadas y en el perímetro. Con un enfoque a entornos industriales, Li et al. [19] presentan un sistema que combina FL con un cifrado Paillier para aumentar la confidencialidad de las actualizaciones del modelo durante el entrenamiento. Otros trabajos muestran el uso de FL para entrenar IDSs basados en ML en dispositivos que se comunican con protocolos industriales como Modbus por Mothukuri et al. [20] o DNP3 por Kelli et al. [21]. Este último, combina FL con *active learning* para la personalización de los modelos.

La mayoría de las propuestas emplean métodos supervisados para el entrenamiento de los modelos. Sin embargo, en despliegues reales, la obtención de datos de red etiquetados no es viable a nivel práctico. La extensión de FL a los métodos no supervisados sigue siendo un reto [14]. Además, los experimentos realizados en los artículos citados se limitan a escalas pequeñas del orden de 10 clientes, que dista de los entornos de gran escala de IoT. Por otra parte, sólo unos pocos trabajos consideran la heterogeneidad de los dispositivos IoT. En los casos en los que se considera, se requiere una segmentación manual de los dispositivos [22] o la ayuda de herramientas externas [15] que no están integradas en el proceso de FL.

Fuera del ámbito de la ciberseguridad, hay varias propuestas que abordan el problema de convergencia de los modelos en casos con datos no IID. En Sattler et al. [23] proponen *Clustered FL*, que agrupa los clientes basándose en la similitud del coseno de las actualizaciones de gradiente de cada cliente después de que el modelo de FL haya convergido. Briggs et al. [24] introducen un paso de agrupación jerárquica de los clientes de acuerdo a la similitud de sus actualizaciones locales de modelos. Después, inician un proceso de FL para cada grupo. A diferencia de estos artículos, este trabajo aborda el tema de métodos de agrupación de dispositivos IoT integrado en FL aplicado al ámbito de la ciberseguridad.

**Algoritmo 1:** FL con agrupación de modelos para clientes heterogéneos.

**Function**
`AgrupaciónModelos(`$lista\_parametros\_modelo$`)`**:**
  $\mathcal{W} \leftarrow$ lista vacía
  **for w in** $lista\_parametros\_modelo$ **do**
    colapsar $\mathbf{w}$ a una dimensión y añadir a $\mathcal{W}$
  **end**
  $\mathcal{W} \leftarrow$ aplicar reducción de dimensionalidad PCA a $\mathcal{W}$
  $\mathcal{S} \leftarrow$ lista vacía
  $\mathcal{L} \leftarrow$ lista vacía
  **for** $n \leftarrow 2$ **to** *número máximo de grupos* **do**
    K-means clústering de $\mathcal{W}$ en n grupos
    añadir etiquetas de número de grupo a $\mathcal{L}$
    añadir métrica de validación de clúster a $\mathcal{S}$
  **end**
  $K \leftarrow$ número de grupos con mejor resultado en $\mathcal{S}$
  **return** (etiquetas de $\mathcal{L}$ que correspondan a $n = K$, $K$)

**Input:** Un conjunto de clientes $\mathcal{C}$
**Result:** Un conjunto de modelos globales entrenados
inicial parámetros del modelo $\mathbf{W_0}$ en el servidor
$\epsilon \leftarrow$ número de épocas locales para la agrupación
**foreach** *cliente* $c \in \mathcal{C}$ **in parallel do**
  recibir $\mathbf{W_0}$ del servidor
  $\mathbf{W_c}, n_c \leftarrow$ `LocalTrain(`$\mathbf{W_0}$, $\epsilon$`)`
  enviar $\mathbf{W_c}$ al servidor
**end**
$\mathcal{W} \leftarrow$ lista de todos los $\mathbf{W_{c \in C}}$ recibidos
$\mathcal{L}$, $K \leftarrow$ `AgrupaciónModelos(`$\mathcal{W}$`)`
**foreach** *etiqueta* $k \in \{1, ..., K\}$ **in parallel do**
  $\mathcal{C}_k \leftarrow$ subconjunto de clientes $\in \mathcal{C}$ con etiqueta $\mathcal{L} = k$
  $\mathbf{W_G^{C=k}} \leftarrow$ promedio de $\mathcal{W}$ con etiqueta $\mathcal{L} = k$
  $\mathbf{W_G^{C=k}} \leftarrow$`AprendizajeFederado(`$\mathcal{C}_k, \mathbf{W_G^{C=k}}$`)`
**end**

## III. FL con agrupación de modelos

El proceso propuesto de FL con agrupación de modelos consta de dos agentes principales: los clientes o dispositivos IoT que realizan el entrenamiento local de los modelos; y el servidor de agregación, que coordina todo el proceso de FL y la agrupación de los clientes. El pseudocódigo se describe en el Algoritmo 1.

En el primer paso, el servidor de agregación inicia los pesos $\mathbf{W_0}$ del modelo y los hiperparámetros. Estos valores se envían a todos los clientes. A continuación, cada cliente entrena parcialmente $\mathbf{W_0}$ durante $\epsilon$ épocas usando los datos de entrenamiento locales, y después envía el modelo parcialmente entrenado al servidor de agregación. El servidor agrupa los clientes en $K$ grupos basándose en los pesos parcialmente entrenados, siguiendo la hipótesis de que los clientes que tengan distribuciones de datos similares convergerán a modelos con parámetros similares siempre que partan del mismo modelo inicial. Para ello, primero se aplica el análisis de componentes principales (PCA, *Principal Component Analysis*) para reducir la dimensionalidad de los parámetros con el objetivo de evitar los problemas de agrupamiento con datos de alta dimensionalidad y, a su vez, acelerar el proceso de agrupación. Posteriormente se usa K-means para agrupar los clientes con parámetros de $\mathbf{W_0}$ similares en $K$ grupos. El valor óptimo de $K$ se realiza en base a métricas de validación interna como Silhouette, Calinski-Harabasz, Davies-Bouldin y S-Dbw [25].

Para cada grupo, se inicia un proceso de FL. Los parámetros parcialmente entrenados de cada grupo se agregan mediante FedAvg [13] (media ponderada), que se convierte en el modelo inicial del que parte FL. En total se entrenan $K$ modelos globales. El proceso de FL usado es el FL generalizado propuesto por Reddi et al. [26] en el que el entrenamiento local en cada cliente se realiza con CLIENTOPT [26] y el agregado de los modelos por parte del servidor se realiza mediante SERVEROPT [26]. Ambos son abstracciones sobre los optimizadores habituales como SGD, Adam o RMSprop. El proceso de FL se repite durante $R$ rondas, y por cada ronda los clientes entrenan localmente el modelo durante $E$ épocas.

### *III-A. Modelo de detección de anomalías*

En este trabajo se han empleado Autoencoders para generar los modelos de detección de anomalías. Los Autoencoders son redes neuronales que se entrenan de un modo no supervisado para replicar los datos de entrada $\mathbf{x} \in \mathbb{R}^n$ en su capa de salida $\mathbf{x}' \in \mathbb{R}^n$ bajo algunas restricciones para evitar aprender la función de identidad. Los parámetros del modelo se entrenan para minimizar el error cuadrático medio (ECM) entre $\mathbf{x}$ y $\mathbf{x}'$. La función de pérdida de la *Ec.* (1) se construye sumándole al ECM el término de regularización $L_2$, donde $\mathbf{w}$ representa los parámetros del modelo y $\lambda$ es la constante que controla la contribución de la regularización.

$$\mathcal{L} = \text{ECM} + L_2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i')^2 + \lambda \sum_i w_i^2 \qquad (1)$$

## IV. BANCO DE PRUEBAS IOT

El objetivo del banco de pruebas es poder simular un gran número de dispositivos IoT con el que generar datos de entrenamiento y evaluar la propuesta de FL. Para ello se ha usado GNS3 [27] para simular una topología de red con distintos dispositivos IoT, servicios y atacantes.

Para simular un entorno heterogéneo, se han desarrollado tres tipos de dispositivos IoT que se comunican usando los protocolos MQTT y CoAP, dos protocolos ampliamente usados en esta clase de dispositivos. Cada tipo de dispositivo se ha implementado mediante contenedores Docker y las bibliotecas Eclipse Paho y libcoap para implementar los patrones de comunicación de red de MQTT y CoAP respectivamente. Los dispositivos se conectan mediante switches de red implementados con el software de Open vSwitch. Entre los servicios se incluye un broker de MQTT y un cliente CoAP que envía peticiones periódicamente a los dispositivos. Los tres tipos de dispositivo IoT simulados son los siguientes:

- **mqtt tipo 1**: Se comunica con el broker mediante el protocolo MQTT sobre TCP. Abre una única conexión con el broker al inicio de la transmisión y la mantiene activa enviando periódicamente datos de telemetría simulados y mensajes de *keep-alive*. La carga útil de la telemetría es pequeña, con aproximadamente 50 bytes por mensaje.
- **mqtt tipo 2**: Se comunica con el broker mediante el protocolo MQTT sobre TCP. Cada vez que se envían datos de telemetría, se abre una nueva conexión al broker y se cierra al terminar. La carga útil es mayor, con aproximadamente 1700 bytes por mensaje.
- **coap tipo 1**: Se comunica con los clientes empleando el protocolo CoAP sobre UDP. Periódicamente envía datos de aproximadamente 20 bytes por cada petición.

La topología creada con GNS3 incluye en total 360 instancias de dispositivos IoT simulados, 120 de cada tipo. Para incluir mayor variedad, cada instancia presenta pequeñas desviaciones en la periodicidad de las comunicaciones y el volumen de datos transmitidos.

El banco de pruebas incluye un atacante implementado con el servidor de mando y control (C&C, *Command and Control*) Merlin [28] y el programa hping3 [29] para realizar ataques de denegación de servicio (DoS, *Denial of Service*). Parte de los dispositivos IoT descritos anteriormente incluyen el agente de Merlin. Estos dispositivos comprometidos forman una *botnet*, es decir, el conjunto de dispositivos bajo las órdenes del servidor de C&C. En total se han comprometido tres dispositivos, uno de cada tipo.

### *IV-A. Generación de datos*

*IV-A1. Tráfico normal:* El tráfico normal se compone de trazas de paquetes de red capturados en cada uno de los dispositivos mientras se ejecuta la simulación, sin incluir ningún ataque. El tráfico normal se divide en dos conjuntos: uno para entrenamiento (80 %) y otro para validación (20 %).

*IV-A2. Tráfico bajo ataque:* El tráfico de ataque consiste en las trazas de paquetes de red capturados en los dispositivos comprometidos con el agente de Merlin. Los datos incluyen una mezcla de tráfico legítimo (comunicación normal del dispositivo IoT) y el tráfico de red entre el dispositivo comprometido y el servidor de C&C o la víctima de los ataques de DoS. A continuación, se muestra una lista de los comportamientos maliciosos y ataques realizados:

**Comunicación C&C:** El atacante inicia el servidor C&C de Merlin a la espera de conexiones entrantes de todos los dispositivos comprometidos que ejecuten el agente de Merlin. Los canales de C&C se mantienen abiertos mediante el envío de mensajes periódicos.

**Transferencia de datos:** El atacante transfiere el binario de hping3 a cada uno de los dispositivos comprometidos mediante el canal de C&C. Posteriormente, los *bots* pueden usar hping3 para realizar ataques contra las víctimas.

**Ejecución remota de código:** El atacante ejecuta remotamente comandos en los dispositivos comprometidos para preparar el entorno antes de realizar los ataques.

**Denegación de servicio:** Debido a que muchas de las muestras de *malware* real de IoT se basan en el código fuente de Mirai [5], los ataques incluidos se realizan de acuerdo a su comportamiento, tal como se describe en su código fuente [30], pero implementados usando hping3. Se incluyen: (i) ICMP *flood*, (ii) UDP *flood* a diferentes puertos aleatorios de la víctima, ataques de (iii) TCP SYN y (iv) TCP ACK.

## V. IMPLEMENTACIÓN Y RESULTADOS

En este apartado se describe la metodología de entrenamiento seguida y los resultados experimentales obtenidos. La metodología consta de distintas fases, que van desde el procesamiento de los datos capturados hasta la detección de anomalías: (V-A) procesamiento de datos (incluyendo el filtrado y la extracción de características), (V-B) selección

del modelo e hiperparámetros, (V-C) FL con agrupación de modelos y (V-D) resultados de la detección de anomalías.

### V-A. Procesamiento de datos de red

En primer lugar, las trazas de red obtenidas mediante el banco de prueba se filtran para eliminar ciertos paquetes. Después se seleccionan las características de red relevantes y, por último, esas características se preprocesan para adecuarlas a la entrada de los modelos de ML.

*V-A1. Filtrado de datos y características:* Como primer paso, se filtran los archivos *pcap* para descartar todos los paquetes IPv6 y ARP. Después, por cada uno de los paquetes de red filtrados, se extraen las 11 características descritas en la Tabla I. El uso de estas características se debe a que los ataques de DoS realizados con Mirai [30] y sus variantes, incluyen opciones que afectan o modifican esos valores. `len`, `ip_tos`, `ip_ttl`, `tcp_win` y `h` se normalizan dividiéndolos por el valor máximo alcanzable de cada característica. `iat` se transforma aplicando $log(1 + \texttt{iat})$. A las variables categóricas `ip_flags`, `ip_proto` y `tcp_flags` se les aplica la codificación *one-hot*.

Tabla I
CARACTERÍSTICAS DE RED SELECCIONADAS JUNTO A SU DESCRIPCIÓN.

| Característica | Descripción |
|---|---|
| `len` | Longitud del paquete en bytes. |
| `iat` | tiempo respecto al paquete anterior. |
| `h` | Entropía base 2 del paquete. |
| `ip_tos` | IP tipo de servicio. |
| `ip_flags` | IP campos (MF, DF, R bits). |
| `ip_ttl` | IP tiempo de vida. |
| `ip_proto` | IP protocolo (TCP, UDP, ICMP). |
| `src_port` | Puerto origen. |
| `dst_port` | Puerto destino. |
| `tcp_flags` | TCP campos (F, S, R, P, A, U, E, C, N). |
| `tcp_win` | TCP tamaño de ventana. |

*V-A2. Procesamiento de características federado:* Dada la naturaleza distribuida de los conjuntos de datos en los entornos de FL, la transformación de ciertas características requiere una consideración especial. En particular, para `src_port` y `dst_port` se propone una discretización federada que tiene en cuenta la distribución de los números de puerto origen y destino en todo el conjunto de datos. Primero, cada cliente calcula individualmente un histograma con los puertos observados y lo envía al servidor de agregación. Cuando el servidor tenga los resultados de todos los clientes, este discretiza los datos en $b$ grupos de tal modo que se consiga aproximadamente una frecuencia similar de elementos en cada uno de los grupos (basado en los cuantiles de la distribución). Finalmente, el servidor envía la estrategia de discretización a todos los clientes y se discretiza `src_port` y `dst_port` aplicando la codificación *one-hot*. En nuestro caso se ha establecido $b = 10$ después de probar con distintos valores.

La estrategia de discretización federada se aprende usando exclusivamente el conjunto de datos de entrenamiento, después se aplica directamente a los conjuntos de validación y ataque. Las 11 características de la Tabla I se transforman en un conjunto de 34 características al finalizar el proceso.

### V-B. Selección del modelo e hiperparámetros

En FL hay un número mayor de hiperparámetros a ajustar comparado con entornos ML centralizados. En particular, hay que tener en cuenta el modelo de ML (capas, nodos por capa, funciones de activación, etc.), el optimizador a nivel local CLIENTOPT y a nivel del agregador SERVEROPT junto a sus respectivos *learning rates* $\eta$ y $\eta_s$, número de épocas locales $E$, rondas $R$ de FL y número de clientes muestreados $M$ por cada ronda. Debido a la dificultad de explorar simultáneamente todas las combinaciones, se procederá a optimizar ciertos parámetros paso a paso seleccionando aquellos que minimicen el ECM de validación en el menor número de rondas o épocas.

*V-B1. Selección del modelo de Autoencoder:* Para definir la arquitectura del Autoencoder, se han explorado distintas combinaciones usando un subconjunto reducido de los datos de entrenamiento correspondientes a un cliente. Tanto la capa de entrada como de salida del Autoencoder se ha fijado a 34 nodos (igual al número de características). Para el encoder se han evaluado distintas combinaciones de hasta 3 capas ocultas en el que cada capa tiene la mitad de nodos que la capa precedente. Después de 2 capas ocultas no se ha encontrado una mejora significativa en el ECM de validación. El modelo final es un encoder de dos capas con 17 y 8 nodos y un decoder simétrico de 8 y 17 nodos. Se usa la función de activación $ReLU$ tras cada capa y un tamaño de *batch* de 32.

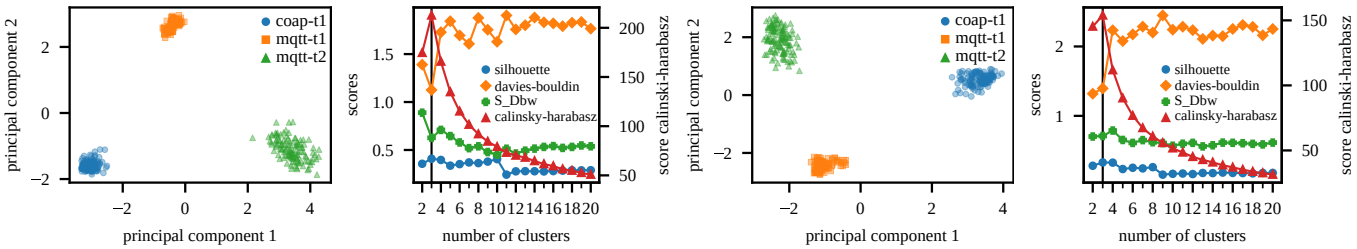
El ajuste del resto de hiperparámetros se realizará de un modo federado usando el conjunto de datos de entrenamiento completo en el que participan todos los clientes.

*V-B2. Ajuste de hiperparámetros de FL:* Para seleccionar los optimizadores que implementen CLIENTOPT y SERVEROPT, se han comparado múltiples combinaciones de SGD y Adam. Para SGD se ha usado sin momento y con momento fijado en 0,9, tal como se sugiere en [31]. Para Adam se han usado dos combinaciones distintas: la primera fijando los dos decrecimientos exponenciales $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y la constante para estabilidad numérica $\epsilon = 1 \times 10^{-8}$ (los valores por defecto del optimizador Adam en la biblioteca PyTorch); y la segunda combinación usando $\beta_1 = 0{,}9$, $\beta_2 = 0{,}99$, $\epsilon = 0{,}001$, tal como se sugiere en [31]. El *learning rate* del cliente se fija a $\eta = 1 \times 10^{-3}$ y el multiplicador de la regularización $L_2$ mencionado en la *Ec.* (1) fijado a $\lambda = 1 \times 10^{-5}$. El *learning rate* del servidor se fija a $\eta_s = 1$ para SGD y $\eta_s = 1 \times 10^{-2}$ para Adam.

Después de $R = 60$ rondas de FL con $E = 1$, la combinación de optimizadores que minimiza el ECM de validación más rápidamente es el siguiente: Adam con $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$, $\epsilon = 1 \times 10^{-8}$ para CLIENTOPT; y SGD sin momento para SERVEROPT. Para algunas otras combinaciones, el modelo no converge y presenta un error cada vez mayor. A continuación, se refina el valor de los *learning rates* $\eta$ y $\eta_s$ simultáneamente mediante *grid search* en el que $\eta$ toma valores $10^{-1}; 10^{-2}; 10^{-3}; 10^{-4}$ y $\eta_s$ $0{,}1; 0{,}5; 1{,}0; 1{,}5$. El menor ECM se obtiene con $\eta = 10^{-3}$ y $\eta_s = 1{,}0$.

### V-C. FL con agrupación de modelos

Usando el modelo de Autoencoder y los hiperparámetros mencionados anteriormente, se pone en marcha el proceso de FL con agrupación de modelos descrito en el Algoritmo 1.




(a) Agrupación para $\epsilon = 2$. (b) Métricas para $\epsilon = 2$. (c) Agrupación para $\epsilon = 16$. (d) Métricas para $\epsilon = 16$.

Figura 1. Resultados de la agrupación de dispositivos para $K = 3$. Proyección 2D de los modelos y valores de las métricas de validación.

*V-C1. Agrupación de dispositivos:* El servidor de agregación inicia aleatoriamente los parámetros del Autoencoder y lo distribuye a todos los 360 clientes IoT. Cada cliente entrena el modelo localmente durante $\epsilon$ épocas, y esos modelos parcialmente entrenados se envían al servidor de agregación para iniciar el proceso de agrupación. Después de aplicar PCA, se selecciona el número de componentes necesario para retener por lo menos el 90 % de la varianza. Se usa el algoritmo K-means para agrupar los modelos en ese espacio de dimensionalidad reducida. Se prueban distintos números de grupos $K$ de 2 a 20, y para cada uno de los casos, se miden las siguientes cuatro métricas de validación interna: Silhouette, Calinski-Harabasz, Davies-Bouldin y S-Dbw [25]. El número óptimo de grupos $K$ se selecciona automáticamente mediante una votación por mayoría de las métricas mencionadas.

Se han repetido los experimentos para distintos valores de $\epsilon = 1, 2, 4, 8, 16$ y 32. El resultado para $\epsilon = 2$ y 16 se puede observar en la Figura 1. Tal como muestran las imágenes, con $K = 3$ grupos, a todos los dispositivos del mismo tipo se les asigna correctamente al mismo grupo. Con $K = 2$, los dos tipos de dispositivos basados en MQTT se agrupan en el mismo grupo. En adelante, se fija $K = 3$ para el resto de la experimentación.

*V-C2. Resultados de FL por cada grupo:* Por cada uno de los tres grupos identificados, se agregan los parámetros parcialmente entrenados del Autoencoder mediante FedAvg y el resultado pasa a ser el modelo inicial del que parte el proceso de FL. Se realizan un total de $R = 30$ rondas de FL para cada grupo. Los experimentos se repiten para distintos valores del número de épocas de entrenamiento local $E = 2, 4$ y 8. En la última ronda, el caso con $E = 2$ muestra valores más altos en el ECM de validación en comparación con los casos con más épocas de entrenamiento locales. Para $E = 4$ y 8, ambos muestran valores del ECM comparables; sin embargo, $E = 8$ necesita menos rondas de FL para conseguirlo a expensas de un mayor coste computacional local en cada dispositivo. Los resultados para la progresión del entrenamiento para $E = 4$ en el caso de los tres grupos se muestra el la Figura 2.

La Figura 2 también compara la progresión del entrenamiento FL con el método aislado sin FL, en el que cada dispositivo inicia su propio modelo de Autoencoder y lo entrena sobre los datos locales hasta la convergencia sin ningún tipo de cooperación (usando los mismos hiperparámetros). FL muestra una convergencia más rápida y una menor variabilidad en la función de pérdida que en el caso aislado.

### V-D. Detección de anomalías

Después de completar el entrenamiento durante 30 rondas de FL, cada dispositivo dispone localmente del modelo global correspondiente a su grupo. En este apartado se evaluará la tasa de detección de anomalías de los 3 modelos globales resultantes usando el conjunto de datos de ataques.

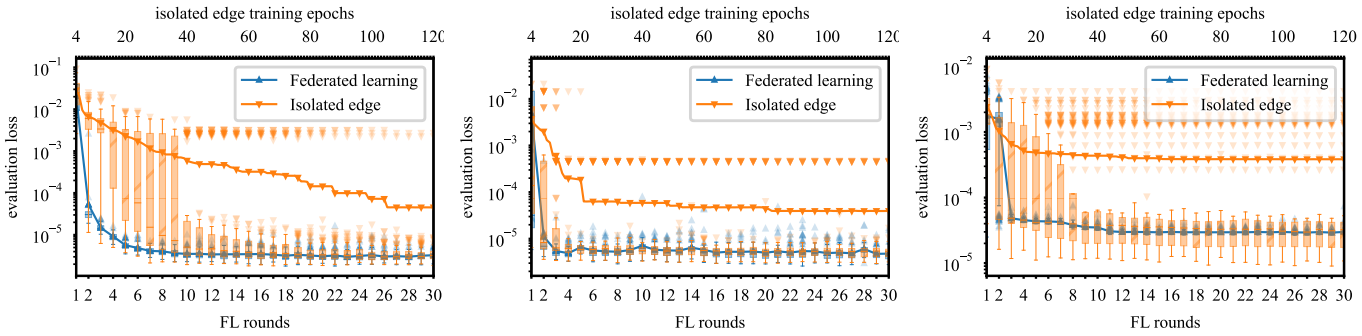
Debido a que es un modelo no supervisado, primero se evaluará el modelo en el conjunto de validación de cada dispositivo para estimar el valor del umbral de detección de anomalías. Cuando el ECM entre la salida del modelo y la entrada sea mayor que el umbral, el paquete de red se considerará anómalo. Este valor se calcula localmente, por lo que cada uno de los dispositivos podrá tener un umbral distinto. Una opción simple consiste en asignar el valor máximo del ECM del conjunto de validación como el umbral.

Para obtener una estimación del desempeño del modelo, se ha anotado el conjunto de datos de ataques teniendo en cuenta la dirección IP del servidor de C&C y la víctima. Se considera cualquier paquete que tenga como origen o destino estas dos direcciones como parte de un ataque.

Las métricas usadas para evaluar los modelos son la precisión, el valor F1 y el coeficiente de correlación de Mathews (MCC, *Matthews Correlation Coefficient*). El modelo global correspondiente al grupo 1 presenta una precisión de $= 0{,}9957$, F1 $= 0{,}9970$ y MCC $= 0{,}9890$. El modelo global del grupo 2 obtiene una precisión de $= 0{,}8935$, F1 $= 0{,}9199$ y MCC $= 0{,}7863$. Finalmente, El modelo global correspondiente al grupo 3 tiene una precisión de $= 0{,}9696$, F1 $= 0{,}9471$ y MCC $= 0{,}9276$.

## VI. Conclusiones

La arquitectura FL presentada no necesita ningún tipo de etiquetado de datos, lo que la hace apropiada para despliegues reales en los que no es factible obtener un etiquetado preciso del tráfico de red antes de entrenar los modelos de ML. Para abordar los problemas que surgen en entornos heterogéneos, la arquitectura incluye un algoritmo de agrupación que funciona inspeccionando los parámetros de los modelos parcialmente entrenados. El método está integrado en el proceso de FL y no depende de herramientas externas o de métodos manuales, facilitando así la implementación de arquitecturas basadas en FL. Sin embargo, no es sencillo realizar una implementación práctica de FL, debido al alto número de hiperparámetros a ajustar. Por ejemplo, cada situación puede requerir un equilibrio distinto entre el número de rondas de FL y número de épocas de entrenamiento local dependiendo del coste de




(a) Grupo 1 (120 dispositivos coap tipo 1). (b) Grupo 2 (120 dispositivos mqtt tipo 1). (c) Grupo 3 (120 dispositivos mqtt tipo 2).

Figura 2. Progresión de entrenamiento de FL para los 3 grupos identificados comparado con entrenamiento aislado (sin FL).

la transmisión de datos y de si se quiere mover la carga computacional al servidor de agregación o a los clientes IoT.

## Agradecimientos

Este trabajo ha sido financiado por la Comisión Europea mediante el programa Horizon Europe bajo el proyecto IDUNN (núm. 101021911). También está parcialmente financiado por las Ayudas Cervera para Centros Tecnológicos del Centro para el Desarrollo Tecnológico Industrial (CDTI) bajo el proyecto EGIDA (CER-20191012), y por el Gobierno Vasco bajo el programa ELKARTEK, proyecto REMEDY - REal tiME control and embeddeD securitY (KK-2021/00091). Urko Zurutuza es miembro del grupo de investigación de Sistemas inteligentes para Sistemas Industriales de Mondragon Unibertsitatea (IT1676-22), apoyado por el departamento de Educación, Universidades e Investigación del Gobierno Vasco.

## Referencias

[1] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, Oct. 2018.

[2] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[4] M. H. U. Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera, "The role of big data analytics in industrial Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 247–259, Oct. 2019.

[5] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis *et al.*, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110.

[6] P.-A. Vervier and Y. Shen, "Before Toasters Rise Up: A View into the Emerging IoT Threat Landscape," in *Research in Attacks, Intrusions, and Defenses, RAID 2018*, ser. Lecture Notes in Computer Science, Bailey, M and Holz, T and Stamatogiannakis, M and Ioannidis, S, Ed., vol. 11050, 2018, pp. 556–576, 21st International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Heraklion, GREECE, SEP 10-12, 2018.

[7] G. Kambourakis, C. Kolias, and A. Stavrou, "The Mirai Botnet and the IoT Zombie Armies," in *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, ser. IEEE Military Communications Conference, 2017, pp. 267–272, IEEE Military Communications Conference (MILCOM), Baltimore, MD, OCT 23-25, 2017.

[8] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[9] A. Costin and J. Zaddach, "Iot malware: Comprehensive survey, analysis framework and case studies," *BlackHat USA*, 2018.

[10] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Feb. 2020.

[11] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[12] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M. S. Hossain, "Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.

[13] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[14] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[15] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DIoT: A Federated Self-learning Anomaly Detection System for IoT," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, ser. IEEE International Conference on Distributed Computing Systems, 2019, pp. 756–767, 39th IEEE International Conference on Distributed Computing Systems (ICDCS), Richardson, TX, JUL 07-09, 2019.

[16] V. Rey, P. M. S. Sánchez, A. H. Celdrán, G. Bovet, and M. Jaggi, "Federated learning for malware detection in iot devices," *CoRR*, vol. abs/2104.09994, 2021.

[17] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in iot edge devices," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[18] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things intrusion Detection: Centralized, On-Device, or Federated Learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, Sep. 2020.

[19] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber-Physical Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[20] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, 2021.

[21] V. Kelli, V. Argyriou, T. Lagkas, G. Fragulis, E. Grigoriou, and P. Sarigiannidis, "Ids for industrial applications: A federated learning approach with active personalization," *Sensors*, vol. 21, no. 20, 2021.

[22] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyber-physical systems," in *2019 28th International Conference on Computer Communication and Networks, ICCCN*, 2019, pp. 1–8.

[23] Sattler, Felix and Müller, Klaus-Robert and Samek, Wojciech, "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2020.

[24] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *2020*

*International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020.

[25] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE International Conference on Data Mining*. IEEE, Dec. 2010, pp. 911–916.

[26] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[27] J. Grossmann *et al.* Graphical network simulator 3. [Online]. Available: https://www.gns3.com/

[28] R. V. Tuyl. Merlin is a cross-platform post-exploitation http/2 command & control server and agent written in golang. [Online]. Available: https://github.com/Ne0nd0g/merlin

[29] S. Sanfilippo. hping network tool. [Online]. Available: https://github.com/antirez/hping

[30] J. Gamblin. Leaked mirai source code for research/ioc development purposes. [Online]. Available: https://github.com/jgamblin/Mirai-Source-Code

[31] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, B. A. y Arcas, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly *et al.*, "A field guide to federated optimization," *CoRR*, vol. abs/2107.06917, 2021.

# Arquitectura para la Detección de Noticias Falsas Basada en *Watermarking* y *Machine Learning*

Julián Salas
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
jsalaspi@uoc.edu

Victor Garcia-Font
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
vgarciafo@uoc.edu

Jordi Serra-Ruiz
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
jserrai@uoc.edu

Tanya Koohpayeharaghi
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
tkoohpayeharaghi@uoc.edu

David Megías
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
dmegias@uoc.edu

Helena Rifà-Pous
Internet Interdisciplinary Insitute
Center for Cybersecurity Research of Catalonia
Universitat Oberta de Catalunya, Barcelona
hrifa@uoc.edu

***Resumen*—La facilidad de creación y distribución de noticias falsas se está convirtiendo en una amenaza creciente tanto para particulares como para empresas e instituciones. Los principales facilitadores de la distribución de noticias falsas son las redes sociales, que permiten generar e intercambiar contenidos todos los días, a todas horas. Es por ello que desarrollar contramedidas efectivas es de primordial importancia. Considerando lo anterior, en este artículo, proponemos y describimos una arquitectura de un sistema de detección de noticias falsas que se está desarrollando dentro del proyecto "Detection of fake newS on SocIal MedIa pLAtfoRms" (DISSIMILAR). Dicho proyecto está diseñado para la detección de noticias falsas en medios digitales, es decir, imágenes, vídeo y audio y, para cumplir con sus objetivos, combina técnicas de marca de agua, procesamiento de señales y aprendizaje automático.**

***Index Terms*—noticias falsas, marcas de agua digitales, aprendizaje automático, procesamiento de señales.**

## I. Introducción

Las noticias falsas y bulos, han estado presentes en la historia de la humanidad incluso antes del advenimiento del Internet. En general, las noticias falsas suelen considerarse como un tipo de periodismo amarillo que contiene piezas de noticias legítimas que pueden ser engaños. En el mundo digitalizado actual, los principales canales para difundir la desinformación son plataformas de redes sociales y otros tipos de medios en línea. Los datos falsos incluyen no solo información de texto también incluyen imágenes digitales, vídeos o archivos de audio manipulados.

La difusión de noticias falsas en las plataformas de redes sociales ya ha tenido impacto en la vida real, no digital. Por ejemplo, en 2016, durante las elecciones presidenciales de EE. UU., varios tipos de noticias falsas sobre los candidatos fueron ampliamente difundidas en las redes sociales. Como se señala en [1], se estimó que más del 41,8 % del tráfico de datos de noticias falsas en las elecciones se transmitió por medio de redes sociales en línea. Con un efecto más significativo y generalizado que los canales tradicionales (es decir, televisión, radio o medios impresos). Otro ejemplo reciente está relacionado con la actual pandemia de coronavirus SARS-CoV-2. Las campañas de desinformación relacionadas con el virus en sí, su gravedad, origen, posibles formas de infección, tratamiento y, finalmente, la vacunación han comenzado propagándose más rápido que el propio virus.

Distinguimos entre dos tipos distintos de noticias falsas. Algunas noticias falsas son creadas a partir de contenidos de origen legítimo que han sido manipulados de manera maliciosa, por ejemplo, reemplazar el audio de un vídeoclip o incluso crear un DeepFake a partir de un vídeo auténtico. Por otra parte, podemos encontrar noticias falsas que se crean desde cero sin manipular un contenido original legítimo.

El proyecto DISSIMILAR proporcionará un primer conjunto de tecnologías, combinando marcas de agua digitales y aprendizaje automático, que se integrarán con las plataformas de redes sociales para la detección de noticias falsas. El prototipo permitirá agregar y reemplazar diferentes componentes, con esto esperamos proporcionar un mecanismo para integrar otras tecnologías de otros colaboradores en la plataforma.

El resto del documento está estructurado de la siguiente manera. En la Sección II presentamos el trabajo relacionado. Luego, en la Sección III, explicamos los fundamentos necesarios para entender el marco propuesto. A continuación, la Sección IV hace énfasis en la importancia y la novedad del uso de *Watermarking* digital para la detección de noticias falsas. En la Sección V se describe la arquitectura general del proyecto, mientras que en la Sección VI se presenta la plataforma de evaluación planificada. Finalmente, la Sección VII presenta las conclusiones y líneas de investigación futuras.

## II. Trabajo relacionado

En las redes horizontales de comunicación, por ejemplo, sitios de redes sociales (SNS) [2] los usuarios se vuelven tanto remitentes como receptores de mensajes. Con los medios digitales, se refuerza la idea de prosumidores [3], ya que los usuarios se convierten en productores y consumidores capaces de circular y recrear contenidos. Así, la difusión de rumores [4], o la difusión de noticias falsas [5], se han visto reforzadas con la creciente relevancia de los SNS y la popularización de la cultura participativa [6].

La difusión de noticias falsas ha exigido a los medios de comunicación un esfuerzo más significativo para demostrar que defienden la integridad. Han surgido diferentes proyectos dedicados a desenmascarar bulos en varios países. Por ejemplo, FactCheck.org (EE.UU.), maldita.es (España), o Demagog.org.pl (Polonia) o FactCheck Initiative (Japón) han sido ampliamente adoptados por los consumidores. Dichos sitios web suelen mostrar los metadatos de los contenidos multimedia para argumentar la originalidad de los medios. Sin embargo, los metadatos se pueden cambiar fácilmente y no muestran lo que se ha cambiado en el contenido, por lo que se requieren controles adicionales.

Por otro lado, Trustproject [7] proporciona un protocolo que incluye ocho indicadores de confianza para "ampliar el compromiso del periodismo con la transparencia, la precisión, la inclusión y la equidad para que el público pueda tomar decisiones informadas sobre noticias". Los protocolos han sido adoptados por sitios de noticias en todo el mundo, incluidas empresas prominentes como BBC, South China Morning Post y Bay Area News Group. Si bien estos proyectos han sido bien recibidos, carecen de herramientas tecnológicas para automatizarlos. Incluso los proyectos que utilizan aprendizaje automático para detectar noticias falsas, como puede verse en [8], principalmente se basan en herramientas de procesamiento de lenguaje natural, y, por lo tanto, se basan en el texto y no utilizan contenido multimedia para la detección.

Con estas consideraciones, la marca de agua digital es una técnica prometedora para abordar los problemas de protección de derechos de autor, autenticación de contenido, detección de manipulación y otros [9]. En algunas aplicaciones, se incrusta una huella digital única que identifica al receptor en cada copia individual del contenido distribuido. Esta aplicación actúa como elemento disuasorio de redistribución al permitir que el propietario del contenido rastree la fuente de la copia redistribuida [10], [11]. Por lo tanto, la aplicación de técnicas de marca de agua digital podría ser beneficiosa en la identificación y rastreo de noticias falsas. Este concepto no ha sido investigado hasta ahora en los estudios existentes. Aunque ha habido algunos intentos para contrarrestar *deep fakes* [12] o noticias falsas en imágenes, no han sido utilizadas para otros tipos de contenido digital, y nunca se han aplicado en un sistema más completo integrado con plataformas de redes sociales.

Con respecto a las técnicas de detección, los trabajos típicamente existentes para detectar vídeos falsos se centran en encontrar características imperceptibles que aparecen en ellos. Por ejemplo, el método propuesto en [13] se basa en la detección del parpadeo, que es una señal fisiológica que no se presenta bien en los vídeos falsos sintetizados. Observando las capas y filtros de la CNN, el método en [14] descubrió que los ojos y la boca juegan un papel primordial en la detección de rostros falsificados con herramientas de *DeepFake* [15].

Para acelerar los avances en la detección de medios falsos, el reto de detección de *DeepFake* (DFDC) [16] fue publicado por Amazon Web Services (AWS), Facebook, Microsoft, el Comité Directivo de Integridad de los Medios de *Partnership on AI* y académicos. El objetivo del reto es estimular a los investigadores de todo el mundo a desarrollar tecnologías innovadoras que puedan ayudar a detectar *DeepFakes* y medios manipulados.

En contraste con el trabajo presentado anteriormente, nuestro objetivo (en el proyecto DISSIMILAR) es desarrollar modelos para detectar contenido de medios digitales falsos, centrándose en las señales creadas por las operaciones de procesamiento y dispositivos de registro. Las señales se insertan intencionalmente como marcas de agua. La combinación de las marcas de agua digitales (*Watermarking* digital) y herramientas de detección basadas en aprendizaje automático (*Machine Learning*) permitirá a los usuarios discriminar fácilmente entre contenido original y falso sin necesidad de evaluación y control desde un servicio centralizado.

## III. Antecendentes

Esta sección introduce una breve visión de las tecnologías que serán usadas durante el desarrollo del proyecto, incluyendo *Watermarking* y *Machine Learning* (ML) para la detección de las noticias falsas (*fake news*).

### *III-A. Watermarking digital*

Las marcas de agua digitales [17], [9] son un conjunto de técnicas que consisten en incrustar datos (marca de agua) en un objeto o soporte digital, por lo general manteniendo la calidad perceptual del objeto. Los portadores tradicionales de las marcas de agua son los contenidos multimedia, como imágenes, audio o vídeo, pero también pueden ser texto e incluso los protocolos de red. La marca incrustada está relacionada con el objeto portador, y el objeto portador en sí mismo es valioso (típicamente más valioso que la marca de agua). La marca de agua se puede utilizar, por ejemplo, para proporcionar evidencia sobre el titular de los derechos de autor o los usuarios autorizados del contenido. Las aplicaciones tradicionales de las marcas de agua digitales incluyen la protección de los derechos de autor, la autenticación de contenido, la monitorización de transmisiones de radio y televisión, el seguimiento de transacciones y el control de copias, entre otras.

En función de la aplicación concreta que se pretenda se requerirán ciertas propiedades. Las propiedades más frecuentes para los esquemas de *Watermarking* incluyen las siguientes:

- Robustez/fragilidad: según sea necesario que un esquema sea resistente a procesamiento de señales (robusto), no lo sea (frágil), o sea robusto frente a un conjunto concreto de ataques admitidos, como la compresión con pérdida (semifrágil).
- Extracción ciega o informada de la marca: si se requiere el objeto original sin marcar para la extracción de la marca (extracción informada) o se puede extraer sin necesidad de usar el original (extracción ciega).

- Imperceptibilidad o transparencia: nivel de distorsión admitido para el ruido introducido en el proceso de incrustación de la marca de agua.
- Capacidad de incrustación: cantidad de información que se puede incrustar y luego recuperar del objeto marcado. Suele darse en términos relativos, como bits por píxel (bpp) en el caso de imágenes y bits por segundo (bps) para audio y vídeo.
- Seguridad de la marca: hace referencia a la protección del sistema frente a ataques intencionados que pretenden eliminar la marca, incrustar una marca diferente u obtener las claves secretas usadas para la incrustación o la extracción de la marca. En caso de eliminación de la marca, no se consideran dentro de la propiedad de seguridad los ataques de procesamiento de señales que no intentan explotar el conocimiento sobre el mecanismo de incrustación utilizado.

Las aplicaciones de *Watermarking* que pueden ser útiles para la detección de noticias falsas se detallan en el apartado IV.

### *III-B. Análisis Forense de Contenido Multimedia*

Como hemos visto, en el caso de la esteganografía, el mensaje es escondido dentro de un contenido portador para que pueda ser enviado por un canal abierto. Una parte maliciosa podría usar estos métodos para comunicarse sin levantar sospechas a nadie. Pero, por el contrario, existe también el análisis de esos contenidos portadores que son sospechosos de tener mensajes insertados, este análisis se conoce como estegoanálisis, y actualmente es un campo muy prolífero en nuevas técnicas, incluso aplicando *Machine Learning* para detectar o clasificar ese contenido portador con mensajes insertados.

Esto ha generado estudios en las irregularidades del contenido multimedia que son generadas por esas modificaciones al insertar el mensaje oculto. Así, desde el punto de vista del análisis forense de contenido multimedia en vídeo, audio o imágenes, se buscan anomalías en ellos. Para ello se realiza lo siguiente:

- Identificación del dispositivo de adquisición del contenido multimedia digital.
- Validación de la integridad.
- Extracción de información relevante del contenido que pueda identificarlo.

Así podemos utilizar las trazas intrínsecas que deja en la adquisición los dispositivos, como pueden ser el ruido del sensor, la distorsión de las lentes, el ruido ambiente, entre otros. Además, tenemos también el ruido o eliminación de este, mediante las operaciones posteriores de edición del contenido, o por la modificación intencionada de una parte. La compresión con perdida, el filtrado, el remplazo de zonas, etc. harán que tenga trazas identificables por ser diferentes al resto del contenido. Aquí entran también las técnicas de análisis forense con la asistencia de *Deep Learning*, que detectan ya algunas de esas modificaciones. Estas técnicas son capaces de detectar contenido falso con una gran exactitud.

### *III-C. Detección basada en ML*

Los algoritmos de *Machine Learning* crean modelos de aprendizaje a partir de datos de entrenamiento. Estos algoritmos son capaces de modelar las características de los datos para clasificarlos y distinguir si pertenecen a diferentes grupos. De esta manera, crean límites de decisión que separan los grupos eficientemente para poder asignar un grupo adecuado a un dato que no ha sido previamente clasificado.

Para la clasificación del contenido falso, inicialmente tendríamos que extraer las características del contenido para entrenar algoritmos de ML que generen un modelo de decisión para asignarlo a una clase: positiva o negativa. Generalmente, es muy costoso determinar cuáles son las mejores características de cada grupo de imágenes o contenido multimedia para entrenar un clasificador, ya que requiere un profundo conocimiento del dominio y del tipo de imágenes que van a ser utilizadas. En contrapartida, podemos utilizar técnicas de *Deep Learning* (DL) que calculan todo este proceso automáticamente a partir del conjunto de entrenamiento.

Algunas de las técnicas de *Deep Learning* más utilizadas son las *Recurrent Neural Networks* (RNNs) y las *Convolutional Neural Networks* (CNNs). Las redes RNNs capturan la información secuencial presente en los datos de entrada, por ejemplo, la dependencia entre las palabras dentro de un texto. Mientras que las CNNs capturan las características espaciales de una imagen, como la disposición de los píxeles y la relación entre ellos dentro de la imagen. Así se permite identificar con precisión a los objetos, la localización de estos dentro de la imagen o la relación entre ellos.

Estas técnicas de DL permiten por ejemplo modificar o crear contenido falso cambiando la cara de una persona con la de otra y sincronizar el movimiento de la cara adecuadamente a un discurso hablado. Las manipulaciones faciales pueden ser: *síntesis de cara entera*, *cambio de identidad*, *manipulación de atributos*, y *cambios de expresión* [18].

Para detectar estas técnicas, ya existen algunos estudios de clasificación de caras reales o generadas artificialmente. Los primeros trabajos se centraron en las anomalías creadas en las primeras versiones de los vídeos falsos. La inconsistencia entre los labios y el lenguaje es analizada en [15].

En [19], se mejora la capacidad de detección implementando el uso de la memoria a largo plazo (LSTM) que es en lo que se basan las RNNs. Algunos detalles como el color de los ojos, la falta de reflejos o la pérdida de detalles en los ojos y dientes son utilizados en [20] para hacer una clasificación del contenido falso usando modelos de regresión logística y percetrón multicapa (MLP) [21]. Agarwal [22] presenta un sistema de detección basado en expresiones faciales y movimientos de la cabeza utilizando técnicas de *Support Vector Machine* (SVM) [23]. Los cambios en el patrón de parpadeo de los ojos ha sido estudiando en [24].

A partir del contorno de la cara y los píxeles adyacentes, un sistema de detección [13], [25] basado en CNN, como VGG16, ResNet50, ResNet101, y ResNet152, detecta la presencia de anomalías alrededor de esas zonas modificadas. En [14] se proponen enfoques basados en funciones mesoscópicas y de estegoanálisis, en el que se desarrolla un sistema basado en CNN. El sistema de detección basado en la arquitectura XceptionNet ha proporcionado los mejores resultados.

## IV. Uso de watermarking digital para la detección de noticias falsas

En el proyecto DISSIMILAR se plantea el uso de *Watermarking* de tres maneras diferentes. Cabe destacar que la

literatura que plantea el uso de técnicas de *Watermarking* para la detección noticias falsas es tremendamente reducida, lo que indica el potencial innovador del proyecto que estamos llevando a cabo.

Las aplicaciones de *Watermarking* que pueden ayudar a la detección de noticias falsas son las siguientes:

- **Verificación de la fuente de noticias legítimas:** si los medios y agencias de comunicación verifican sus contenidos digitales usando marcas de agua robustas, será relativamente sencillo detectar cuando un contenido provenga de una fuente fidedigna, comprobando si posee la marca de agua usada en origen para certificar su procedencia. Para esta aplicación deberán usarse marcas de agua robustas.
- **Detección de manipulaciones fraudulentas:** aunque un contenido se pueda verificar con marcas de agua robustas, esto no excluye que un usuario malicioso pueda modificarlo para alterarlo y propagar noticias falsas que procedan aparentemente de una fuente fiable. En este caso, las marcas frágiles o semifrágiles ayudarán a detectar modificaciones fraudulentas con mayor fiabilidad que las técnicas de análisis forenses.
- **Trazabilidad de las noticias falsas:** uno de los mayores retos en la detección de noticias falsas es el de identificar fuentes de este tipo de bulos para ayudar a los usuarios a identificarlos. Si las plataformas sociales incluyen marcado robusto en los contenidos que son aportados por los usuarios, puede ser relativamente sencillo trazar el origen de contenidos fraudulentos y añadir dicha fuente a una base de datos de fuentes no confiables.

Así pues, más allá de las aplicaciones tradicionales del *Watermarking* digital, el proyecto DISSIMILAR abre la puerta a tres áreas nuevas de aplicación de estas técnicas. El interés de la comunidad científica en estas técnicas había decaído notablemente en los últimos años, dado que el problema de la vulneración de los derechos de autor (principal aplicación del *Watermarking* digital) ha disminuido a causa de la aparición de las plataformas de *streaming* que ofrecen contenidos con *copyright* a precios asequibles para una gran parte de la población. No obstante, la problemática de las noticias falsas abre la puerta a nuevas aplicaciones de estas técnicas que, en combinación con el aprendizaje automático, pueden ayudar a los usuarios a distinguir entre contenidos legítimos y fraudulentos. Una primera muestra de este tipo de soluciones se puede encontrar en [26].

## V. Propuesta inicial de arquitectura y análisis comparativo

DISSIMILAR combinará diferentes tecnologías para ayudar a los usuarios en la detección de noticias falsas. En la Fig.1 se muestra a alto nivel la arquitectura propuesta para este proyecto y la interacción con los dos tipos básicos de usuario: productores y consumidores de contenido. En primer lugar, los productores de contenido pueden generar noticias genuinas o falsas y publicarlas en una red social. En segundo lugar, los consumidores de contenido son los usuarios típicos de las redes sociales, objetivo de los productores de contenido malicioso que publican información manipulada. El proceso de detección propuesto se divide en dos pasos y cada paso consta de dos módulos:

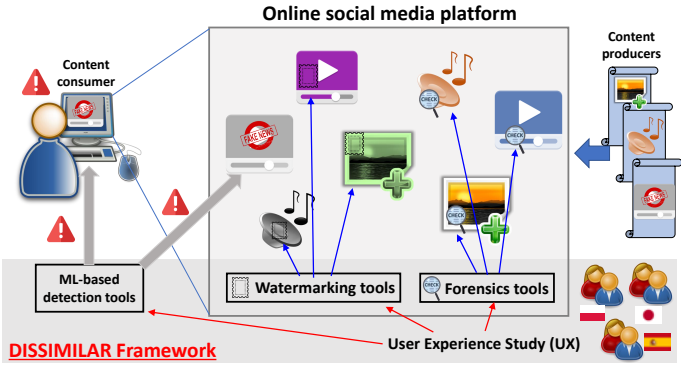



Figura 1. Representación a alto nivel de la arquitectura de DISSIMILAR.

1. **Paso 1: Verificación de la fuente y autenticación de contenido.** Esta fase tiene como objetivo verificar si la fuente de un contenido en línea (audio/voz, vídeo o imagen) es confiable, e intentar determinar si el contenido ha sido modificado. Esta fase consta de los siguientes dos módulos:
   *a*) **Módulo de verificación de la fuente.** Intenta verificar la fuente del contenido utilizando un detector de marca de agua sólido (la fuente debe haber incrustado una marca de agua sólida antes de distribuir el contenido) o (si no se detecta ninguna marca de agua) realizando una búsqueda en Internet para ubicar el contenido. Este módulo utiliza una base de datos de fuentes confiables.
   *b*) **Módulo de autenticación.** Si la fuente ha sido identificada como confiable, se puede usar un módulo de autenticación para tratar de determinar si hay rastros de falsificación en el contenido. Se puede llevar a cabo usando una marca de agua frágil o semifrágil en la fuente o, se pueden usar herramientas de análisis forense, en su mayoría basadas en métodos de ML, para determinar si ha habido alguna modificación maliciosa.

   Por ejemplo, para el módulo de autenticación podemos aplicar la técnica forense de [27] basada en (CNNs) que permite detectar si una imagen ha sido manipulada, junto con las marcas de agua aplicadas en [26] que son resistentes a las modificaciones de *DeepFakes* y permiten detectar la suplantación de la voz.
   Después de verificar la fuente confiable de un contenido y determinar que no ha sido falsificado, el contenido se etiqueta como legítimo ("no falso") y se envía un mensaje al usuario detallando el análisis realizado. De lo contrario, se etiqueta como "sospechoso" y se requiere un mayor escrutinio en el Paso 2.
2. **Paso 2: Detección y rastreabilidad de noticias falsas.** Una vez que un contenido se etiqueta como sospechoso en el Paso 1, este paso intenta determinar si el contenido es realmente una noticia falsa, legítima o indeterminada. Si un contenido es etiquetado como "noticias falsas", un módulo intenta identificar su fuente para agregarlo a una lista negra (base de datos) de fuentes poco confiables. Esta fase consta de los siguientes dos módulos:
   *a*) **Módulo de detección de falsificaciones.** Un con-

Tabla I
COMPARATIVA DE LA ARQUITECTURA.

| Aproximación | Métodos |
|---|---|
| Activos | Watermark [9]<br>Watermark & Blockchain [30], [26] |
| Pasivos | Estegoanálisis [31], [32]<br>Mesoscopic Network [14]<br>LSTM [33]<br>Frequency Domain [34] |
| Propuesta | Combinación de Activos & Pasivos<br>Trazabilidad de la fuente (Digital Fingerprint) |

tenido sospechoso se clasifica utilizando ML con tres posibles resultados: contenido legítimo ("no falso"), "noticias falsas" o "indeterminado". En este último caso, se le enviará una alarma al usuario, quien deberá decidir si la información es confiable. Si el usuario lo desea, se recogerá su opinión sobre la fiabilidad del contenido, y podrá utilizarse para puntuar el contenido en caso de que otro usuario lo envíe para su evaluación.

*b*) **Módulo de rastreo.** Si un contenido se clasifica como noticias falsas, este módulo intenta rastrear la fuente para tratar de identificarla y agregarla a una base de datos de fuentes no confiables. Este paso se puede lograr mediante marcas de agua trazables (también conocidas como huellas dactilares digitales) o mediante la recopilación de algún otro tipo de información (datos de procedencia) que pueda ser útil para la trazabilidad.

Como ejemplos de las técnicas aplicadas en este paso, en [28] tenemos una clasificación de técnicas para detectar *DeepFakes* y en [29] una discusión de los retos y ventajas de nuestra aproximación que combina técnicas de *Watermarking* con ML, para la detección y rastreo de las noticias falsas.

Esta arquitectura es muy flexible, ya que algunas de las tecnologías que incorpora (como las marcas de agua digitales) no son obligatorias, pero pueden ayudar significativamente en la detección de noticias falsas si se utilizan. Además, esta arquitectura permite distinguir los dos tipos distintos de noticias falsas: las que se crean a partir de un contenido legítimo que luego se manipula, y las que se crean desde cero, sin manipular la fuente original.

Al final del proyecto DISSIMILAR, esperamos desarrollar un prototipo abierto para que se puedan integrar otras soluciones, allanando el camino hacia una plataforma de uso gratuito para ayudar a los usuarios en la detección de noticias falsas. A diferencia de otras soluciones, DISSIMILAR permitirá a los consumidores de contenido decidir cuándo usar el sistema para ayudarlos a identificar noticias falsas, prevenir la censura o el control centralizado, preservando así derechos fundamentales como la libertad de expresión en Internet.

### *V-A. Análisis comparativo*

En los últimos años, muchos investigadores se han centrado en el problema del contenido falso y sus técnicas de defensa. Los enfoques convencionales se clasifican básicamente en dos tipos: activos y pasivos. La Tabla I muestra la comparación de los métodos existentes y la arquitectura propuesta. En las técnicas activas, cierta información se codifica en el momento de la generación multimedia, por ejemplo, se agrega una marca de agua al contenido [9]. La marca de agua se utiliza para identificar si el contenido multimedia ha sido manipulado. Además, las partes manipuladas en el objetivo se pueden detectar utilizando la marca de agua extraída. Sin embargo, las marcas de agua digitales no son infalibles [35], problema que puede contrarrestarse utilizando una blockchain [36], [26] para mantener un registro de marcas de agua y de las características del contenido a prueba de manipulaciones. Alattar et al. [30] ha propuesto una prueba de concepto de un sistema de prevención y detección de noticias falsas en vídeo utilizando tecnologías de marca de agua y blockchain.

En las técnicas pasivas, se analizan algunos rastros de manipulación para identificar contenidos falsos. El uso de *DeepFakes* produce artefactos difíciles de identificar para los humanos, pero que pueden ser reconocidos por máquinas y análisis forenses. Las inconsistencias, las irregularidades en el fondo y las huellas dactilares de las *Generative Adversarial Networks* (GAN) son ejemplos de artefactos espaciales. La detección de fluctuaciones en el comportamiento de una persona, las señales fisiológicas, la coherencia y la sincronización de cuadros de vídeo son ejemplos de artefactos temporales. Tratar con píxeles y explotar las correlaciones es uno de los enfoques directos para aclarar las variaciones entre lo real y lo falso. Para aumentar la eficacia de detección y mejorar la capacidad de generalización, en la literatura se han investigado técnicas basadas en *Deep Neural Networks* (DNN).

La arquitectura propuesta es la combinación de enfoques activos y pasivos. Durante la verificación de la fuente y la autenticación del contenido, tanto la existencia de la marca de agua como la verificación de los rastros de manipulación se ejecutan para clasificar el contenido de destino, ya sea falso o no. Además de la clasificación, la arquitectura propuesta involucra el módulo de rastreo, que permite identificar la fuente con la ayuda de la huella digital.

## VI. PLATAFORMA EXPERIMENTAL DEL PROYECTO

Los métodos de detección de noticias falsas introducidos necesitan un entorno de evaluación robusto y confiable. Durante dicho proceso de evaluación, se requerirán datos reales recopilados directamente de Internet para demostrar que los enfoques desarrollados son eficientes y efectivos en entornos del mundo real. Teniendo en cuenta que la mayoría de los datos utilizados para desarrollar técnicas de detección de noticias falsas se basan en texto (como ya observamos en la Sección II). Consideramos que es necesario diseñar, desarrollar e implementar una plataforma experimental.

El sistema consta de tres partes esenciales: (1) recolector de páginas web, (2) base de datos dedicada, y (3) interfaz de usuario basada en web.

El **recolector de páginas web** es responsable de visitar páginas web seleccionadas y descargar material multimedia (imágenes digitales y archivos de audio y vídeo). El material descargado se almacena en el disco duro y se analiza instantáneamente utilizando los complementos provistos (que contienen los esquemas de detección propuestos).

Todos los resultados de los complementos analíticos relacionados con un archivo multimedia descargado se colocan como metadatos en la **base de datos dedicada**. El sistema desarrollado permite agregar fácilmente nuevos complementos

analíticos que brindan varias funcionalidades al sistema, por ejemplo, detección de marcas de agua incrustadas, identificación de la posible modificación de medios para la creación de noticias falsas o adición de información esteganográfica para el envío secreto de comandos a máquinas infectadas.

El último elemento de la plataforma desarrollada es la **interfaz de usuario basada en web**, que gestiona el proceso de recopilación de datos multimedia de Internet. Además, permite buscar metadatos en la base de datos del sistema.

Un aspecto crucial se refiere a informar a los propietarios de servidores web que la recolección que realizamos no es una actividad hostil. Por supuesto, debería existir la posibilidad de que los propietarios de servidores web eliminen sus direcciones de forma permanente de la lista, que utilizamos con fines experimentales. Actualmente, investigamos varios métodos para informar a los propietarios de servidores web: el primero utiliza la misma dirección IP utilizada para la recolección como servidor web. La página web alojada contiene información sobre nuestra investigación y un formulario que los propietarios del sitio web pueden utilizar para eliminar permanentemente sus direcciones de nuestros análisis. El segundo, utiliza algunos encabezados HTTP personalizados y proporciona información sobre el proyecto DISSIMILAR.

## VII. Conclusiones y trabajo futuro

La distribución de noticias falsas a través de las plataformas sociales en línea se ha convertido en un problema creciente para la sociedad. En este artículo presentamos un marco de trabajo diseñado para detectar noticias falsas en el contenido multimedia desarrollado dentro del proyecto DISSIMILAR. En este trabajo hemos caracterizado la arquitectura del sistema y la plataforma del banco de pruebas que se utilizará para evaluar varios enfoques de detección de noticias falsas.

El objetivo final del proyecto DISSIMILAR es crear un prototipo que demuestre que una combinación de técnicas de ocultación de datos, aprendizaje automático y técnicas forenses de datos multimedia, crearía una plataforma de detección de noticias falsas, eficiente y eficaz que tendría más éxito que las soluciones parciales por sí solas. El prototipo creado se enriquecerá en primer lugar con un conjunto de algoritmos de marcas de agua y aprendizaje automático utilizando un diseño centrado en el usuario. Como propuesta a futuro, tenemos la intención de compartir la plataforma con la comunidad de seguridad para que otros investigadores y desarrolladores puedan contribuir con soluciones mejoradas y posiblemente más eficaces modificando algunos de sus componentes.

## Agradecimientos

Los autores agradecen la financiación obtenida por el proyecto Detection of fake newS on SocIal MedIa pLAtfoRms “DISSIMILAR” EIG CONCERT-Japan con la subvención PCI2020-120689-2 (Gobierno de España), y a los proyectos RTI2018-095094-B-C22 “CONSENT” y PID2021-125962OB-C31 “SECURING” del Ministerio de Ciencia e Innovación.

## Referencias

[1] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, May 2017.

[2] M. Castells, *Communication Power*. United Kingdom: Oxford University Press, 2009.

[3] A. Toffler, *The Third Wave*. William Morrow & Company, 1980.

[4] C. R. Sunstein, “On rumors, How falsehoods spread, why we believe them, and what can be done,” Princeton and Oxford, 2014.

[5] H. Allcott and M. Gentzkow, “Social Media and Fake News in the 2016 Election,” *Journal of Economic Perspectives*, 2017. [Online]. Available: https://doi.org/10.1257/jep.31.2.211

[6] H. Jenkins, M. Ito, and D. Boyd, *Participatory Culture in a Networked Era*. Polity, 2016.

[7] TrustProject. The trust project – news with integrity. [Online]. Available: https://thetrustproject.org/

[8] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, “Behind the cues: A benchmarking study for fake news detection,” *Expert Systems with Applications*, vol. 128, pp. 201–213, 2019.

[9] D. Megías, “Data hiding: New opportunities for security and privacy?” in *Proceedings of the European Interdisciplinary Cybersecurity Conference (EICC 2020),* Art. No. 15, Rennes, France, 2020, paper 15, pp. 1–6.

[10] G. R. Blakley, C. Meadows, and G. B. Purdy, “Fingerprinting long forgiving messages,” in *Advances in Cryptology — CRYPTO ’85 Proceedings*, H. C. Williams, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 180–189.

[11] D. Boneh and J. Shaw, “Collusion-secure fingerprinting for digital data,” *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[12] P. Korus and N. Memon, “Content authentication for neural imaging pipelines: End-to-end optimization of photo provenance in complex distribution channels,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8613–8621.

[13] Y. Li, M.-C. Chang, and S. Lyu, “In ictu oculi: Exposing ai created fake videos by detecting eye blinking,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018.

[14] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “Mesonet: a compact facial video forgery detection network,” in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018.

[15] P. Korshunov and S. Marcel, “Deepfakes: a new threat to face recognition? assessment and detection,” *CoRR*, vol. abs/1812.08685, 2018. [Online]. Available: http://arxiv.org/abs/1812.08685

[16] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C.-Ferrer, “The deepfake detection challenge (DFDC) preview dataset,” *CoRR*, vol. abs/1910.08854, 2019. [Online]. Available: http://arxiv.org/abs/1910.08854

[17] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[18] R. Tolosana, R. V.-Rodriguez, J. Fierrez, A. Morales, and J. O.-Garcia, “Deepfakes and beyond: A survey of face manipulation and fake detection,” *Information Fusion*, vol. 64, pp. 131–148, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303110

[19] P. Korshunov and S. Marcel, “Speaker inconsistency detection in tampered video,” in *Proc. European Signal Processing Conference (EUSIPCO 2018)*, 2018.

[20] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” in *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 83–92.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[22] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li, “Protecting world leaders against deep fakes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[23] P. L. Shrestha, M. Hempel, F. Rezaei, and H. Sharif, “A support vector machine-based framework for detection of covert timing channels,” *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 274–283, 2016.

[24] T. Jung, S. Kim, and K. Kim, “Deepvision: Deepfakes detection using human eye blinking pattern,” *IEEE Access*, vol. 8, pp. 83 144–83 154, 2020.

[25] Y. Li and S. Lyu, “Exposing deepfake videos by detecting face warping artifacts,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[26] A. Qureshi, D. Megías, and M. Kuribayashi, “Detecting deepfake videos using digital watermarking,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC*, 2021, pp. 1786–1793.

[27] D. Takeshita, M. Kuribayashi, and N. Funabiki, "Feature extraction suitable for double jpeg compression analysis based on statistical bias observation of dct coefficients," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2021, pp. 1808–1814.
[28] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, "Deepfake detection for human face images and videos: A survey," *IEEE Access*, vol. 10, pp. 18 757–18 775, 2022.
[29] M. T. Ahvanooey, M. X. Zhu, W. Mazurczyk, K.-K. R. Choo, M. Conti, and J. Zhang, "Misinformation detection on social media: Challenges and the road ahead," *IT Professional*, vol. 24, no. 1, pp. 34–40, 2022.
[30] A. Alattar, R. Sharma, and J. Scriven, "A system for mitigating the problem of deepfake news videos using watermarking," in *Electronic Imaging - Media Watermarking, Security, and Forensics 2020*. Ingenta, January 2020, pp. 1–9.
[31] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.
[32] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec'17, 2017, pp. 159–164.
[33] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, "Two-branch recurrent network for isolating deepfakes in videos," in *ECCV 2020*. Springer-Verlag, 2020, pp. 667–684.
[34] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *ECCV2020*, ser. Lecture Notes in Computer Science, vol. 12357. Springer, 2020, pp. 86–103.
[35] A. Qureshi and D. Megías, "Blockchain-based P2P multimedia content distribution using collusion-resistant fingerprinting," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1606–1615.
[36] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

# Implementación de cifrado broadcast para mensajes cortos en WiFi

José Luis Salazar
Universidad de Zaragoza
C/María de Luna, 3 50018, Zaragoza
jsalazar@unizar.es

Julían Fernandez-Navajas
Universidad de Zaragoza
C/María de Luna, 3 50018, Zaragoza
navajas@unizar.es

Jose Ruiz-Mas
Universidad de Zaragoza
C/María de Luna, 3 50018, Zaragoza
jruiz@unizar.es

Guillermo Azuara
Universidad de Zaragoza
C/Atarazana, nº2, 44003 Teruel
gazuara@unizar.es

***Resumen*—El uso actual de mensajes cortos en redes inalámbricas está creciendo considerablemente. Las aplicaciones de mensajería en terminales móviles con cobertura inalámbrica son muy comunes en centros de afluencia masiva de personas. Esto requiere mejorar su eficiencia sin perder la seguridad en un entorno tan hostil. En este artículo, proponemos una mejora en el uso del medio a través de un nuevo paradigma de cifrado de transmisión multicanal. Mostraremos la seguridad de nuestro modelo centrado en dos puntos: mensajes cortos y mantenimiento de la privacidad en un marco compartido. Para implementarlos, mejoramos la eficiencia de las comunicaciones, reduciendo las cabeceras de seguridad a una sola que será compartida por todos los receptores, mientras que el payload se multiplexa mediante el Teorema Chino de los Restos. De esta manera reducimos la longitud del paquete (menos encabezados) y mantenemos la ratio encriptado/texto plano cercano a uno.**

***Index Terms*—cifrado de transmisión multicanal, seguridad comprobable, eficiencia de utilización del canal**

## I. Introducción

El uso extendido de los servicios móviles en las últimas décadas ha llevado a un uso creciente de paquetes pequeños [1], especialmente para ciertos servicios con fuertes restricciones de interactividad, por ejemplo, VoIP (Voice over IP) o juegos en línea. Para lograr esto, se deben enviar pequeños fragmentos de información (muestras de voz, actualizaciones de juegos) con una frecuencia alta. Dado que cada uno de estos paquetes incluye las cabeceras impuestas por las diferentes capas de la arquitectura del protocolo, se acentúa la ineficiencia de estos servicios en tiempo real.

En este contexto, encontrar un buen equilibrio entre la eficiencia de las comunicaciones y su seguridad se convierte en un desafío, a menudo influenciado por los diferentes requisitos del servicio. El problema se vuelve más relevante cuando las restricciones físicas afectan la Calidad de Servicio (QoS), como sucede en las comunicaciones inalámbricas.

Por un lado, un claro ejemplo de esta ineficiencia (datos útiles respecto al total de bytes transmitidos) lo encontramos en el envío de paquetes pequeños sobre 802.11: según la definición de la trama MAC del estándar [2], la cabecera más la FCS tiene una longitud de 40 octetos. Por tanto, para un paquete VoIP de 40 bytes, la eficiencia es 1/2. Sin embargo, para paquetes grandes, el problema es insignificante porque la carga útil puede representar la gran mayoría del tamaño de la trama. Además, el uso de un medio compartido requiere algún tiempo para los mecanismos de control de acceso a los medios, lo cual es otra fuente de ineficiencia. Sin embargo, existen soluciones exitosas destinadas a abordar este problema, por ejemplo, la agregación de tramas en 802.11 [3]. En la literatura se han propuesto otras soluciones para escenarios cableados, [4][5]. Además, el uso de paquetes broadcast se pude considerar como una forma de aumentar la eficiencia de la red, especialmente en escenarios inalámbricos [6].

Por otro lado, la seguridad también tiene una influencia negativa en la eficiencia: la adición de encabezados de seguridad puede aumentar los gastos generales de manera significativa. Por lo tanto, los enfoques que abordan conjuntamente la agregación y la seguridad se vuelven muy importantes: si un encabezado seguro puede cubrir una cantidad de paquetes pequeños, su sobrecarga se amortiza entre ellos.

En [7] se hace una propuesta para mejorar el equilibrio de seguridad y eficiencia. Para solucionar el problema, agregan conjuntamente varios paquetes cifrados pequeños y se transmiten, asegurando que cada uno de los subpaquetes sólo pueda ser descifrado por una única entidad: su legítimo receptor.

Dicha propuesta adopta el sistema clásico de transmisión segura de televisión por cable (*Tabla 1, primera fila*), donde una serie de encabezados $u_i$ individualizados (necesarios para construir la clave privada de cada usuario para el descifrado), generalmente se envían juntos en una sola trama multiplexada. Además, se requiere un campo común *Hdr*, y finalmente se agrega el contenido cifado de cada usuario ($c_i$) .

Sin embargo, en [7] (*Tabla 1,segunda fila*) se fusionan todos los encabezados individuales para obtener un solo encabezado *Hdr'*. Esto reduce la cantidad total de información a transmitir, proporcionando ahorros reales en términos de ancho de banda. Al mismo tiempo, sólo el destinatario legítimo *i* de cada paquete podrá descifrar la información $u_i$, utilizando *Hdr'* y su propia clave privada.

La propuesta combina el esquema subyacente de cifrado broadcast, pero usando más eficientemente el espectro: se agregan los paquetes cifrados con el Teorema Chino de los Restos e implementan el cifrado aleatorio donde la semilla aleatoria es compartida por todos los usuarios.

Nuestra propuesta es consecuencia de una puesta a punto de

este algoritmo para su explotación en WiFi. Los términos en los que se describe el algoritmo y se demuestra su seguridad, son lo suficientemente genéricos para encontrar otras funciones intermedias alternativas a la propuesta, que mejoran la eficiencia en el ecosistema WiFi sin perder su fortaleza en la seguridad. Las mejoras se enfocan hacia dos problemas de implementación: el tamaño de los números primos usados, que determinan el de la clave privada, y la eficiencia en la generación de la semilla aleatoria.

## II. Cifrado broadcast para mensajes cortos

En esta sección haremos una breve descripción del modelo de cifrado broadcast para luego apreciar cómo se ha adaptado para mensajes cortos.

### *II-A. Modelo de cifrado broadcast*

Para describir este tipo de cifrado usaremos el modelo para un sistema de cifrado broadcast propuesto en [10]. Formalmente, dicho sistema consta de cuatro algoritmos probabilísticos:

- **Configuración**($\lambda$): Toma como entrada el parámetro de seguridad $\lambda$, genera los parámetros globales del sistema y devuelve la clave secreta de cifrado *EK*.
- **Extraccción**(*i*, *EK*): Toma como entrada el índice *i* del usuario, junto con la clave de cifrado *EK* y genera las claves privadas del usuario ($p_i$, $x_i$).
- **Cifrado**($u_1$, $u_2$, ..., $u_n$, *EK*, $m_1$, $m_2$, ..., $m_n$): Toma como entrada los identificadores de *n* usuarios ($u_i$), *n* mensajes ($m_i$) y la clave de cifrado *EK*. Produce (*Hdr*, *ET*) donde *Hdr* es un número aleatorio para el cifrado/descifrado, y *ET* es el texto cifrado, calculado con todos los textos planos.
- **Descifrado**(*Hdr*, *ET*, *i*, $p_i$, $x_i$): Toma como entrada el encabezado aleatorio *Hdr*, el texto cifrado *ET*, el índice del usuario *i* y sus claves privadas $p_i$ y, $x_i$, y genera el texto descifrado $m_i$, para *i*.

Para asegurar la corrección del sistema se requiere que $\forall i \in \{1, \ldots, n\}$, si $EK \leftarrow$ **Configuración**($\lambda$), $(p_i, x_i) \leftarrow$ **Extracción**$(i, EK)$, (*Hdr*, *ET*) $\leftarrow$ **Cifrado**($u_1$, $u_2$, ..., $u_n$, *EK*, $m_1$, $m_2$, ..., $m_n$), entonces $m_i \leftarrow$ **Descifrado**(*Hdr*, *ET*, *i*, $p_i$, $x_i$)

### *II-B. Modelo de cifrado broadcast para mensajes cortos*

La adaptación a mensajes cortos de [7] se hace por medio de criptografía simétrica, pero utilizando aritmética modular (Teorema Chino de los Restos), que resultando ser más lenta, queda amortiguada por la corta longitud de los mensajes.

- **Configuración**($\lambda$): El algoritmo toma como entrada el parámetro de seguridad $\lambda$. Los parámetros globales del sistema se generan de la siguiente manera: primero se eligen *n* primos $p_i$ (uno por usuario) y *n* enteros aleatorios $x_i \in {Z^*}_{p_i}$, tales que mcd($x_i$, $p-1$) = 1. Se define $N = \prod_{i=1}^{n} p_i$ y *EK* = (($p_1$, $x_1$), ($p_2$, $x_2$), ..., ($p_n$, $x_n$)) donde cada par corresponde a la clave secreta de descifrado de cada usuario, que le será enviada a traves del algoritmo **Extraccción**
- **Cifrado**($u_1$, $u_2$, ..., $u_n$, *EK*, $m_1$, $m_2$, ..., $m_n$): Elige un entero aleatorio *Hdr* $\xleftarrow{\$}$ ${Z^*}_{min_j\{p_j\}}$ y define $Hdr_i = min\{g \geq Hdr \backslash g$ es un generador de ${Z^*}_{p_i}\}$ para calcular

$$ET = (\sum_{i=1}^{n}(m_i + Hdr_i^{x_i}(modp_i))\cdot \cdot \frac{N}{p_i}[(\frac{N}{p_i})^{-1}(modp_i])(modN) \quad (1)$$

para dar como salida (*Hdr*, *ET*).
- **Descifrado**(*Hdr*, *ET*, *i*, $p_i$, $x_i$): Este algoritmo calcula

$$m_i = (ET - Hdr_i^{x_i})(modp_i) \quad (2)$$

que es consecuencia del Teorema Chino de los Restos.

Tabla I
Comparación de información cifrada en [7]

| | |
|---|---|
| Modelo clásico | ($u_1$, $u_2$, ... , $u_n$, *Hdr*, $c_1$, $c_2$, ... , $c_n$) |
| Propuesta | (*Hdr'*, $c_1$, $c_2$, ... , $c_n$) |

### *II-C. Entorno de aplicación del algoritmo: WiFi*

Una vez presentada la solución propuesta en [7], estudiemos el trade-off de eficiencia en un escenario inalámbrico: el estándar IEEE 802.11 (WiFi).

Dadas las necesidades de comunicación de paquetes pequeños en términos de seguridad y eficiencia, las restricciones que implica el uso de este algoritmo limitan principalmente el tamaño de las claves utilizadas y su generación. Nuestro principal objetivo es encontrar un equilibrio entre la eficiencia y la seguridad que casi por definición suelen ser temas antagónicos. Según la propuesta, sus algoritmos serán más eficientes, siempre que el tamaño de los paquetes del usuario tenga un límite superior. De hecho las ganacias serán óptimas si el tamaño de la clave es solo un poco más alto que el tamaño del paquete. Así, en servicios que envían paquetes del mismo tamaño (por ejemplo, VoIP), se puede seleccionar fácilmente un tamaño de clave óptimo.

Cuando analizamos un enlace inalámbrico, no sólo debemos tener en cuenta el ancho de banda (en términos de bytes enviados), sino que también es necesario considerar los mecanismos de acceso al medio. Si tenemos que esperar a que el medio esté libre, cuando tengamos que enviar muchos paquetes a través de él, acumularemos mucho tiempo de espera. Por lo tanto, el envío de una cantidad menor de tramas (una sola trama multicast para varios usuarios en lugar de una numerosa cantidad de tramas unicast) puede mejorar la utilización del espectro. La versión 802.11n del estándar incluye dos mecanismos de agregación: A-MPDU (Aggregated MAC Protocol Data Unit), que envía varias MPDU juntas, y A-MSDU (Aggregated MAC Service Data Unit), que hace lo mismo a nivel de MSDU [3], el uso de estos mecanismos de agregación para el envío de tramas multicast ha sido propuesto en la literatura [8][9].

Con estos sistemas de agregación, parece adecuado plantearse la adopción del sistema propuesto en [7]. En concreto resulta ser más económico a nivel de transmisión, emplear A-MSDU pues nos evitamos la transmisión de las cabeceras MAC de los MPDUs. Por otra parte, se gana en seguridad (tanto real, como percibida), ya que cada MSDU agregado en la transmisión broadcast permanecerá inaccesible a aquellos que no son sus legítimos receptores. Recordemos que en una

transmisión broadcast 'standard', el mensaje es accesible a todos los destinatarios incluidos en el grupo de dicha dirección broadcast de destino.

Por tanto, resulta bastante intuitivo hacer el reparto de papeles en este escenario: los usuarios y la cabecera seguirían ejerciendo el mismo rol y los mensajes a cifrar serían los diferentes MSDUs.

### *II-D. Problemas de implementación*

Cuando nos ponemos a implementar el algortimo en este entorno nos encontramos con un nuevo conflicto de intereses entre la seguridad y la eficiencia.

Por una parte, cuanto mayor sea el tamaño de la clave secreta, mayor será la seguridad implementada, pero peor será la eficiencia en la transmisión. Esta ineficiencia viene marcada por dos hechos: el incremento notorio del tamaño del mensaje a transmitir comparado con el que al final se transmite; y el hecho de que elevar el tamaño de los elementos en el anillo $Z_p$, eleva de manera exponencial el tiempo de cálculo.

Por otra parte, el cálculo de $Hdr'$ se plantea como una herramienta para la implementación de una función pseudo-aleatoria que marca una cota inferior de seguridad para todo el proceso. De esta manera los autores consiguen demostrar probabilísticamente su seguridad. Sin embargo, el cálculo de generadores de los anillos de manera tan espontánea exige también una potencia de cálculo que se vuelve a enfrentar a la eficiencia de las comunicaciones.

## III. Soluciones propuestas

Tras lo expuesto en la última sección expondremos propuesta para solucionar los dos problemas planteados: tamaño de la clave y generación de la fuente aleatoria.

### *III-A. Ajuste del tamaño de la clave*

Para resolver el primer impedimento, la solución que aportamos es la de fijar el tamaño de los primos $p_i$ a una longitud fija, es decir, establecer un nivel mínimo de seguridad $\lambda$. Por ejemplo, un valor en nuestro días mínimamente aceptable serían números primos de 1024 bits. Si con los avances tecnológicos este nivel debiera incrementarse por cuestiones de seguridad, sólo tendríamos que subir ese valor. De esta manera podríamos visualizar la transmsión de cada MSDU como un canal de seguridad de 1024 bits.

Con este límite, debemos replantearnos el algoritmo y empezar a limitar la política de claves para responder a las siguientes preguntas: ¿Qué hacemos si nos llegan paquetes de mayor tamaño? En media, de esta manera ¿cuánto paquete 'vacío' estoy cifrando?.

A la primera pregunta, una vez visualizada la transmisión segura a través de canales, resulta muy intuitivo y aprentemente sencillo contestar. Primero, cuantifiquemos cuantos canales vamos a necesitar en función del tamaño máximo de mensaje:

$$j = \lceil \frac{m_i}{\lambda} \rceil \tag{3}$$

y definimos

$$N_i = \prod_{k=1}^{j} p_{ij} \tag{4}$$

donde $p_{ij}$ será el primo que se asigne al usario $i$ para su canal de seguridad $j$.

Esto implica un pequeño cambio en la elección de los parámetros y función de cifrado. Las claves del usuario $i$ serán: $((p_{i1}, ..., p_{ij}), x_{ij} = \prod_{k=1}^{j} x_{ik})$, donde $x_{ij} \in {Z^*}_{N_i}$ es el producto de las claves secretas que se deberían asignar a esos canales de seguridad. Con estos parámetros definimos:

$$m_{ij} = m_i + Hdr_i^{x_{ij}}(modN_i) \tag{5}$$

Por lo tanto, si en esos $n$ canales tenemos $l$ usuarios, deberíamos reescribir la fórmula de cifrado como

$$\begin{aligned} ET = (\sum_{i=1}^{l}(\sum_{k=1}^{j}(m_{ij}(modp_{ik})\cdot \\ \cdot\frac{N}{p_{ik}}[(\frac{N}{p_{ik}})^{-1}(modp_{ik})])(modN) \end{aligned} \tag{6}$$

En la función de descifrado, donde podemos ahorrarnos la reconstrucción de todo el MSDU a partir de la fórmula:

$$m_i = (ET - Hdr_i^{x_{ij}})(modN_i) \tag{7}$$

Respecto a la segunda cuestión, hay que tener en cuenta la distribución probabilística de la variable aleatoria que defina la longitud de los mensajes. Si suponemos que la media de esta variable es $M$ y la capacidad de los canales que hemos creado es $C$, la ratio del espacio desaprovechado en cada MSDU, será:

$$\frac{M(modC)}{\lceil \frac{M}{C} \rceil} \tag{8}$$

por lo que resulta de vital importancia conocer la distribución probabilística de la longitud de los MSDU para intentar minimizar el numerador y así mejorar la eficiencia.

### *III-B. Generador aleatorio*

Los autores parten del cálculo de una semilla aleatoria común $Hdr \xleftarrow{\$} {Z^*}_{min_j\{p_j\}}$ de la que luego cada usuario calcula $Hdr_i = min\{g \geq Hdr \backslash g$ es un generador de ${Z^*}_{p_i}\}$ y lo usan para enmascarar su mensaje $(m_i + {Hdr_i}^{x_i}(modp_i))$.

Sin duda lo hacen de esta manera tan meticulosa, para poder usar la pseudoeleatoriedad de la exponenciación modular en la demostración de seguridad del algoritmo. Sin embargo, habría sido suficiente con haber modelado esa función con oráculo aleatorio para que la demostración hubiera sido igual de válida; y dejar al futuro usuario del algoritmo la implementación de una función pseudoaleatoria adecuada.

Además, el cálculo de generadores en un anillo puede resultar costoso, sobre todo cuando estamos hablando de comunicaciones en tiempo real. Por lo tanto el objetivo se establece en la búsqueda de un algoritmo que genere diferentes salidas aleatorias para cada usuario, a partir de una semilla fija y consumiendo pocos recursos.

Con estas premisas, parece que una solución básica es el empleo de una función HMAC que genere el valor aleatorio a partir de la semilla y un valor compartido en ambos extremos de la transmisión, como podría ser la clave secreta y/o un valor de otro HMAC previo. Para adecuar el tamaño de la salida de esta función a la longitud que se precise se puede encadenar varias salidas de dicha función HMAC. En este caso no definiremos una función concreta, dejando elegir al desarrollador aquélla que crea más adecuada para su entorno.

De manera análoga, la renovación de las claves $(p_i, x_i)$ podría requerir hacerse con más frecuencia que cuando se usa criptografía de clave pública. En este caso, se podrían hacer offline y tener un almacén de claves igual en ambos extremos de la comunicación, manteniendo su uso sincronizado, pero siguiendo el mismo método de encadenamiento de HMACs.

## IV. Conclusiones

En este artículo hemos presentado una puesta en práctica de un algoritmo para el envío de mensajes cortos cifrados en modo broadcast o multidifusión. El escenario de la puesta en marcha ha sido un entorno WiFi. Para la adecuación a este entorno se han mejorado dos aspectos del algoritmo original: el ajuste de la longitud de las claves y la generación de números pseudoaleatorios

Para la adecuación del tamaño de las claves se ha procedido a diseñar unos canales de seguridad de un tamaño fijo (marcado por el tamaño de las claves). A cada canal se le asigna unas claves secretas, que serán compartidas en origen y destino. Y a cada usuario se le asigna un conjunto de canales de los cuales conocerá dichas claves.

Para la generación de números pseudoaleatorios, hemos hecho una propuesta de mejora genérica, basada en el uso de HMACs como generadores pseudoaleatorios. De hecho este generador es el que va a llevar el peso de la seguridad y eficiencia del sistema ya que su semilla será compartida por todos los usuarios de la multidifusión. El fundamento de la mejora de la eficiciencia reside en que las operaciones básicas son calculadas con funciones hash (que son de coste computacional bajo), sustituyendo a operaciones de aritmética finita (entre ellas exponenciaciones) con un coste computacional sensiblemente mayor.

Por último, añadir que una implementación beta ya ha sido realizada con éxito, transmitiendo señales desde un punto de acceso WiFi a un teminal de manera cifrada, tal y como se describe en este artículo.

## V. Agradecimientos

Esta publicación es parte de la ayuda T31_20R al grupo CeNIT de la Universidad de Zaragoza, financiada por la Diputación General de Aragón.

## Referencias

[1] Huawei, Smartphone Solutions White Paper, Issue 2, 2012.07.17. Available at: https://www.huawei.com/mediafiles/CBG/PDF/Files/hw_193034.pdf, accessed 6 June 2022.

[2] IEEE Std. 802-11 (1997). IEEE standard for wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification, http://www.ieee802.org/11/, accessed 6 June 2022.

[3] Ginzburg, B., Kesselman, A.: Performance analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11n. In *IEEE 2007 Sarnoff Symposium*, pp. 1-5, IEEE. Princeton (USA), 2007.

[4] Saldana, J., Fernández-Navajas, J., Ruiz-Mas, J. et al.: Improving Network Efficiency with Simplemux.In: *IEEE CIT 2015, International Conference on Computer and Information Technology*, pp. 446-453, IEEE, Liverpool (UK), (2015).

[5] Saldana, J., Fernández-Navajas, J., Ruiz-Mas, J. et al.: Emerging Real-Time Services: Optimising Traffic by Smart Cooperation in the Network. *IEEE Communications Magazine*, vol. 11, 127-136, (2013).

[6] Coronado, E., Riggio, R., Villalón, J., et al.: Efficient real-time content distribution for multiple multicast groups in SDN-based WLANs. *IEEE Transactions on Network and Service Management*. vol. 15, n.1, 430-43 (2017)

[7] Salazar, J.L., Saldana, J., Fernández-Navajas, J., et al: Short Message Multichannel Broadcast Encryption. In: *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*. CISIS 2019. Advances in Intelligent Systems and Computing, vol 1267. Springer, Cham.

[8] Park, Y.D., Jeon, S., Kim, K., et al.: Ramcast: Reliable and adaptive multicast over IEEE 802.11n wlans. *IEEE Communications Letters* vol. 20, n. 7, 1441-1444, (2016)

[9] Park, Y.D., Jeon, S., Jeong, J.P., et al.: FlexVi: PHY Aided Flexible Multicast for Video Streaming over IEEE 802.11 WLANs. *IEEE Transactions on Mobile Computing*. vol. 19, n. 10, 2299-2315, (2020)

[10] Baudron, O., Pointcheval, D., Stern., J.: Extended notions of security for multicast public key cryptosystems. In *ICALP2000. LNCS*, vol. 1853, 499-511,Springer, Heidelberg (2000)

# Anomaly Detection Using Improved $k$-Means Clustering on Apache Flink

Aleksander Styrmoe
Norwegian University of Science and Technology (NTNU)
P.O.Box 8900, Torgarden, 7491 Trondheim, Norway
alekssty@stud.ntnu.no

Slobodan Petrović
Norwegian University of Science and Technology (NTNU)
P.O.Box 191, 2802 Gjøvik, Norway
slobodan.petrovic@ntnu.no

***Resumen*—The k-means algorithm is a popular clustering algorithm widely used in unsupervised machine learning. Anomaly-based Intrusion Detection Systems (IDS) can use it to detect attacks on hosts and networks in situations where traditional signature-based IDS are not effective. However, the original $k$-means algorithm may be too slow for IDS applications due to high data rates in today's networks. Certain improvements to the original $k$-means algorithm have been proposed, yet few of these improvements have been applied in intrusion detection. This paper proposes an effective anomaly-based intrusion detection system that implements some improvements of the $k$-means algorithm on the distributed computing platform Apache Flink. These improvements exploit the triangle inequality property of the distance measure used in clustering. By applying the $k$-means speedups and by implementing them on Apache Flink, the efficiency of an anomaly-based IDS using $k$-means clustering is improved, in spite of some computational overhead that such an implementation introduces.**

***Index Terms*—intrusion detection systems, clustering, distributed computing**

## I. Introduction

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) play a significant role in protecting personal, cooperative and national assets by detecting, identifying and, if possible, stopping attacks. IDS can either be signature-based, also called misuse-based, or anomaly-based. The signature-based IDS scan for signatures, or fingerprints, of known attacks when looking for intrusions, while the anomaly-based IDS look for deviations from previously defined normal behavior in hosts and networks. Consequently, anomaly-based IDS are efficient against so-called *zero-day attacks*, which constitute a significant threat in today's computing systems.

To detect anomalies, unsupervised machine learning and clustering are often used. Clustering algorithms in IDS should be accurate and fast enough to handle large traffic loads. The most often used clustering algorithm in anomaly detection is the $k$-means algorithm. It is easy to implement and its time complexity is linear in the number of feature vectors to cluster. However, the original $k$-means algorithm is sometimes too slow for application in IDS. Improvements to the $k$-means algorithm have been proposed as the original algorithm performs some unnecessary distance computations. Most of these improvements utilize the triangle inequality [1], [2], [3], [4].

This paper proposes a fast implementation of some of the $k$-means improvements utilizing the triangle inequality for application in anomaly detection. Its efficiency is based on parallelisation offered by a distributed computing platform Apache Flink [5]. Such an implementation of $k$-means does not reduce the accuracy of anomaly detection. Apache Flink supports $k$-means clustering, but only the basic (so-called naïve) version, which includes many unnecessary distance computations. This may slow down the clustering process. We implement multiple improvements and variations of the $k$-means algorithm applied in IDS and compare them with each other and to the original $k$-means algorithm.

## II. Background

### *II-A. The original k-means algorithm*

The $k$-means algorithm, [6], [7] is a widely used clustering algorithm for its efficiency (linear time complexity in the number of vectors to cluster) and rapid convergence. $k$ is the number of clusters, which must be known in advance. This may be a disadvantage in general, since this number is not always known. However, in anomaly detection, this is not a problem, since we can always cluster in at least two clusters: "normal" and "attack". The $k$-means algorithm minimizes the sum-squared-error (SSE) function (1), where $C_j$ is the cluster $j$, $c_j$ is the mean vector of the cluster $j$ and $d(x, y)$ is the distance measure (usually the Euclidean distance) between the vectors $x$ and $y$.

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, c_j)^2 \tag{1}$$

The mean vector of a cluster is called *centroid*. The $k$-means algorithm works as follows:

1. Initialize $k$ centroids (very often, at random).
2. While algorithm has not converged:
   *a*) Assign each vector to its currently closest centroid.
   *b*) Move each centroid to the mean of its currently-assigned vectors

For step 1, different initialization methods are used. Often, $k$ vectors are chosen at random. Other methods of initialization have been proved to be more effective in making the algorithm converge faster. One such method is the `k-means++` [8], where $k$ initial centroids are chosen using a special distribution.

The time complexity of the $k$ means algorithm is $O(knr)$, where $r$ is the dimensionality of each vector to cluster and $n$ is the number of iterations before convergence. The number of iterations $n$ is often estimated to be relatively small (see, for example, [9]).

The $k$-means algorithm originally works in offline, or batch mode. It starts when all the vectors are presented in advance. When applied in IDS, the batch mode accumulates $N$ vectors representing the monitored traffic at a time and then runs the $k$-means algorithm on the whole batch of vectors. By contrast, the on-line version of the $k$-means algorithm does not accumulate batches but processes one vector at a time by executing step 2 of the algorithm above until the convergence. This version is slower than the batch mode since, for each new vector, the whole cluster structure and the centroids have to be updated (see, for example, [10]).

### *II-B. Improvements to the k-means algorithm*

The original $k$-means algorithm is called the naïve $k$-means algorithm since it performs multiple unnecessary distance computations. Improvements to the algorithm utilizing the triangle inequality and other techniques have been proposed. The triangle inequality holds if the chosen distance measure is a *metric*. Then for any three points, $a$, $b$ and $c$, $d(a,b) \leq d(a,c) + d(b,c)$. This section reviews some of the improvements that we propose to use in anomaly detection. They utilize the triangle inequality to skip some distance computations. Let $x$ be a vector from the vector set (batch) to cluster, $c$ is its currently assigned centroid and $c'$ is another candidate centroid. The following descriptions of improvements to the $k$-means algorithm have been proposed.

Compare-means (Philips [4]) By the triangle inequality, we can skip the calculation of $d(x,c')$, if $2d(x,c) \leq d(c,c')$. This means $x$ cannot be assigned to $c'$. The reason for this condition is the following. By the triangle inequality, $d(c,c') - d(x,c) \leq d(x,c')$. If $2d(x,c) \leq d(c,c')$ we can write $2d(x,c) - d(x,c) \leq d(x,c') \implies d(x,c) \leq d(x,c')$. Now we know that $c'$ is far enough from $c$. In this improvement, distances between centroids need to be calculated each time centroids move (after each iteration).

Sort-means (Philips [4]) This variant uses the same condition as compare-means, but is faster than compare-means since it searches for centers in a different order. This variant has a larger overhead by maintaining a $k \times k$ matrix storing centroid-to-centroid distances each time a new centroid gets calculated. By sorting each row of the matrix, we can see if any other centroid, $c'$, is far enough from $c$, meaning $d(c,c') \geq 2d(x,c)$, by searching the row of $c$ in increasing distance to $c$. If one centroid is found, the search can stop, $x$ gets a new centroid and more importantly, we have skipped looking at some far-away unnecessary centroids [11].

Upper and lower bounds (Elkan [2]) Elkan's idea is to use upper and lower bounds instead of exact values to skip some distance computations. Let $c$ be a centroid before one iteration of the $k$-means algorithm. Let $c*$ be the new centroid of the same cluster after recomputing the cluster mean after the iteration (we say that the centroid $c$ has *moved* to $c*$). Multiple conditions are checked to skip computations. As Philips, Elkan also uses the following fact: $\frac{1}{2}d(c,c') \geq d(x,c) \implies d(x,c') \geq d(x,c)$, and $d(x,c')$ is not needed to be computed. An upper bound, $u(x)$, for $d(x,c)$ is maintained. At the beginning, the upper bound starts as $u(x) = d(x,c)$. After each iteration, the upper bound is maintained by adding $d(c,c*)$ to $u(x)$. If $u(x) \leq \frac{1}{2}d(c,c')$, by checking the minimum $d(c,c')$ over all $c' \neq c$, we do not need to compute $d(x,c')$ and $x$ does not change the cluster. A lower bound for $d(x,c)$, $l(x,c)$, is also used in this modification of the $k$-means algorithm. If $c$ did not move too much during the iteration, $l(x,c) - d(c,c*)$ is a good approximation for $d(x,c*)$. If $u(x) \leq l(x,c')$ or $u(x) \leq \frac{1}{2}d(c,c')$, the calculation of $d(x,c')$ is unnecessary and $x$ does not change cluster to $c'$.

One lower bound (Hamerly [3]) This is a modification of Elkan's algorithm, which uses only one lower bound, $l(x)$, instead of $k \times n$ lower bounds. $l(x)$ represents the minimum distance any centroid, except for the currently-assigned one, can be at that point. It maintains the maximum distance any centroid has moved and uses this to update $l(x)$. If $l(x) < u(x)$ happens to be true, more computation is needed and $x$ might switch cluster. This algorithm uses less memory than Elkan's algorithm, but calculates distances more often. This algorithm works better in low dimension [11].

$b$ lower bounds (Drake and Hamerly [1]) This variant reduces the overhead in Hamerly's algorithm and the memory usage in Elkan's algorithm. It maintains $1 < b < k$ lower bounds for each vector. The first $b-1$ lower bounds represent the distance to the $b-1$ closest centroids, except for its currently-assigned one. The last lower bound represents a lower bound for the $k-b$ furthest-away centroids and is updated by either subtracting by the distance the furthest-away centroid has moved, or by subtracting the longest distance any centroid has moved [11].

### *II-C. Parallelization and distributed computing*

By using distributed computing, we split the operations on large datasets among multiple computer nodes. The same operations are performed on each node. Several practically useful open-source distributed computing systems exist today, the most prominent ones being Apache Hadoop [12], Apache Spark [13], and Apache Flink [5]. These systems use the technique called MapReduce – it is the process of mapping the data to different operators using a key first (while filtering or sorting the data) and then performing a Reduce function on all operators (like a summary function) before the result is generated. While Apache Hadoop is known for its fault tolerance and scalability, that framework does not support iterative algorithms (see, for example, [14]). Apache Spark is a framework that adds more functionality to the MapReduce paradigm, enabling more algorithms to be implemented that are better suited for less fault-tolerant applications. The framework supports both batch processing (where the operators process batches of data), and streaming (where we divide the stream of data into small micro-batches that execute operators).

In this paper, we concentrate on Apache Flink, which is a relatively new distributed computing framework that supports stateful operations and provides different APIs to operate on different layers to handle both bounded and unbounded

data sources[1]. The Table API and the SQL API operate on the highest layer, while the DataStream API operates on the lower layer enabling more custom data operations. A programmer can change between the different APIs freely. Flink supports both batch operations and streaming natively. The Flink architecture consists of one centralised JobManager connected to a group of worker nodes called TaskManagers. The user connects to the JobManager either through the Flink Client or the Command Line Interface (CLI) to execute jobs that are executable Java or Scala programs. The JobManager creates a group of operators, sends it to the TaskManagers and distributes the workload to the TaskManagers, where the operators are located.

Apache FlinkML is a new machine learning framework inside Apache Flink that implements some machine learning algorithms. The FlinkML library operates at different Stages. The first Stage is an Estimator responsible for training the machine learning model and it implements a *fit* method. This method takes in training data and outputs a Model. Next, the Model is another estimator that implements a method called *transform*, which takes in data and uses the model data to produce a result or *prediction*. The library also introduces a framework for iterative algorithms that support multiple streams to be fed back into the iteration, emitted and put into the iteration.

*II-C1. $k$-means in FlinkML:* Ready-made offline and online implementations of the $k$-means algorithm exist and they have been added to the FlinkML library[2]. Since the machine learning paradigm in Flink operates with a fit operation to create a model, the Offline $k$-means implementation in the ML library of Apache Flink is also split into two operations that derivate from the original $k$-means algorithm. The two operations are *fit* and *transform*. The fit operation creates a "KMeansModel" that essentially contains the centroids. The second operation is the transform operation that takes input vectors and assigns them to their closest centroids from the "KMeansModel" in one iteration without adjusting the centroids. The Offline $k$-means implementation in FlinkML terminates after a fixed number of iterations and does not look at when the algorithm has converged.

## III. Related work

### *III-A. Naïve $k$-means and Elkan's improvement on Hadoop*

Despite Apache Hadoop not supporting iterative algorithms, AlGhamdi et al. [14] implemented the $k$-means algorithm on Apache Hadoop by introducing a driver that took control over the iterations and fed the MapReduce framework for each iteration. Along with implementing the naïve algorithm on MapReduce (KMMR-N), the authors also implemented Elkan's improvement of the $k$-means algorithm by using bounds to utilize the triangle inequality. The bounds were stored in two different methods. The first method, KMMR-EV, extended an input vector into an extended vector (EV), also containing the bounds. The second method, KMMR-BF, included storing the bound information in a separate file. The results showed that KMMR-EV reduced the running time compared to KMMR-N with speedups of up to $4{,}5\times$. KMMR-BF was even faster with speedups of up to $6{,}8\times$ compared to KMMR-N. However, this varied with the number of dimentions, records and clusters. The overhead of KMMR-EV could outweigh the gain in time from skipping distance calculations in high dimentions, high $k$ and large number of records so much that it becomes slower than KMMR-N.

### *III-B. Naïve $k$-means and Elkan's improvement on legacy Apache Flink*

Ringdalen [15] used an earlier version (1.9) of Apache Flink and implemented the naïve and Elkan's versions of the $k$-means algorithm on the Apache Flink framework. No machine learning support was used, so it was not possible to use *stages* and iterations with multiple data streams. Ringdalen compared the naïve and Elkan's versions of the $k$-means algorithm on Flink and also compared them to a naïve implementation using classical computing. Results showed that Elkan's version performed worse than the naïve version when increasing the number of clusters. Only increasing the parallelism and having a small $k$ made Elkan's version perform better than the naïve version. However, the overhead of introducing tagged tuples to store bounds did outtake the gained time by skipping distance calculations. With parallelism of $4$, Elkan's version had an increase in performance of just $1{,}8\,\%$ with $k = 2$.

## IV. The new $k$-means-based IDS on Flink

The anomaly detection system that we propose has the improved $k$-means algorithm implemented on Apache Flink as its principal component. To be implemented on Flink/FlinkML, this algorithm must sustain several adjustments, which we present below.

### *IV-A. Introducing domains in $k$-means*

We need to solve the problem regarding non-numerical features before implementing a $k$-means clustering-based IDS. The solution can be to take the same approach as Yu et al. [16] and introduce the concept of *domains* that we introduce here. A domain is a group of features that has the same value of all non-numerical attributes (i.e., those attributes that cannot be processed directly by the $k$-means algorithm). A domain can be identified as a string being the conjunction of all non-numeric fields. The $k$-means algorithm is performed on each domain separately.

By introducing the concept of domains, we have not altered the $k$-means algorithm but rather formalized the pre-processing and post-processing. In some domains, using a clustering algorithm makes sense. For example, if one domain consists of HTTP traffic, and another one consists of SMTP traffic, both domains probably have normal traffic and anomalies. However, some domains may consist of only one class by nature. For example, some combination of flags and services may always represent anomalies. Even so, we must pay attention to such combinations since they may constitute the root to a zero-day attack in the future. This would lead to multiple false positives (FP) or false negatives (FN) that should not happen in that domain. This challenge is not solved by $k$-means. In such cases, an anomaly detection system should be assisted by signature-based detection that

[1]Bounded means the data is in a batch form, while unbounded means a continuous stream of data.

[2]https://github.com/apache/flink-ml/tree/master/flink-ml-lib/

would investigate those specific domains. We suggest to not automatically approve the domains that look like *only-normal* domains as they may correspond to zero-day attacks in the future.

### *IV-B. Adjusting $k$-means in FlinkML to support domains*

We run one instance of $k$-means while having domains as an integrated part. In the FlinkML $k$-means implementation, one data stream is used for only one element. This is an array holding the centroids. We have expanded the KMeans implementation to support multiple arrays of centroids, one for each domain. These elements are stored in MapStates at operators, and when a vector arrives, the correct array of centroids is retrieved. All centroid and point objects also have a domain attached as a field. Fig. 1 shows an example of the centroid datastream after initial centroids have been chosen.


Figura 1. The centroid datastream after random centroids have been selected.

### *IV-C. Adjusting $k$-means in FlinkML to stop at convergence*

The original offline $k$-means implementation in FlinkML has a fixed number of iterations that the user sets when initialising the KMeans object. The correct implementation should involve looking at when the algorithm converges i.e., when the vectors have stopped changing clusters and all centroids' movement is zero. We have changed the data flow in the KMeans program in FlinkML to support convergence by introducing a filter that emits centroids and points when the algorithm has converged. The data flow is given in Fig. 2.

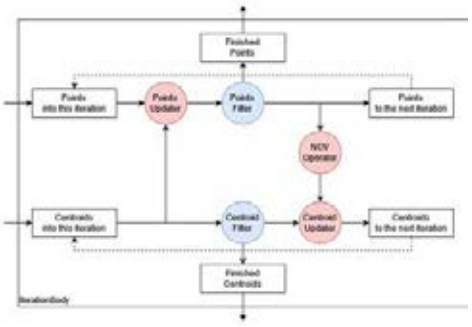



Figura 2. FlinkML $k$-means implementation that stops at convergence.

After the centroids have been updated in the operator called CentoridUpdater, the centroids may have a movement of 0. If so, the centroids are fed back into a new iteration so the PointUpdater can mark the vectors as finished (finally classified). After this, the vectors with the finished flag set and the centroids with $movement = 0$ are filtered out of the stream.

The domains can converge at different times. If all centroids in a domain have stopped moving, the domain can be marked as finished, and the output of that particular domain can be emitted.

### *IV-D. Supporting multiple improvements of $k$-means*

To make implementation of each improvement of $k$-means as similar as possible, the unique logic of each improvement was placed in objects. All vectors and centroid objects contain information relevant to the improvement stored in them and methods that update internal fields, such as bounds and other fields based on all centroids. The rest of the data flow is the same for each improvement and can be seen in Fig. 2. Algorithms 1 and 2 show how PointUpdater and CentroidUpdater operate.

**Algorithm 1** Pseudo-code for PointUpdater

```
operator PointUpdater:
  processElement(point):
    centroids = getState()
    point.update(centorids)
  processBroadcastElement(centroids):
    storeInState(centorids)
```

**Algorithm 2** Pseudo-code for CentroidUpdater

```
operator CentroidUpdater:
  processElement(newCentroidValueObject):
    centroids = getState()
    for centroid in centroids:
    centroid.move(newCentroidValueObject)
    for centroid in centroids:
    centroid.update(centroids)
  processBroadcastElement(centroids):
    storeInState(centorids)
```

All vectors contain fields *assigned* and *finished*. All centroids contain fields *ID* and *movement*.

By using the framework prestented above, we have reduced the improvement-dependent side of the implementation to just Centorid and Point objects.

## V. Experimental work

### *V-A. Input data*

We have tested our proposed anomaly detection system on the well-known and currently widely accepted data set NSL KDD [17]. It includes 41 features and is a subset of the outdated KDDCUP '99 dataset. Most features of the vectors are numbers. They can be processed directly by the proposed $k$-means-based IDS. Four features are boolean values. We include these features in measuring distance by assigning 1 to "true" and 0 to "false". In addition, the feature "su_attempted" is discrete with 3 values, 0, 1, and 2.

The features, "protocol_type", "service", and "flag", are non-numerical. Assigning numbers to each value and computing the distance does not make sense since such data values do not have natural ordering. As explained above, we split the data set into the different domains that correspond to the values of these features and operate our $k$-means implementation on each domain separately. The challenge with this approach is that we may have one pair of centroids (corresponding to attack traffic and normal traffic) for each domain. This is also taken care of in our framework.

NSL KDD is labeled i.e., there is an additional feature - the label in the dataset. This feature is included in the Point objects to later perform the calculation of accuracy, but it is not included in the very $k$-means algorithm input.

### *V-B. Experimental setup and $k$-means implementation*

Two objects, *Point* and *Centroid*, are used in our $k$-means implementation, both inheriting from the Vector object. Both objects have a method called *distance* that is used to calculate the Euclidean distance to another vector. The *Point* object has a field to identify which cluster the vector is assigned to. The initial value of this field is $-1$. The *Centroid* object has one field for unique identification in the domain and a field for storing the movement of the centroid since the last iteration. The initial value of the movement field is the maximum value of this field. The Point object has a boolean value for indicating that the domain has converged and that the object can be emitted. Depending on the version of $k$-means, both *Point* and *Centroid* have an *update* method for updating some values based on all centroid values. The *Centroid* object also has a *move* method that updates the vector coordinates and the movement field.

When implementing the different improvements of the $k$-means algorithm, we make objects inheriting from *Point* and *Centroid* objects to make sure we change as little as possible, only adding the improvements. This means the Flink Job still performs the same method calls for each version, but the implementations of *move* and *update* methods inside the objects are overwritten.

As a starting point, we chose to use the original OfflineKMeans implementation in FlinkML. We changed the KMeans class to support our objects, removed the Estimator interface and changed the return value of the fit method to be both the resulted centroids datastream and the resulted vector datastream (including their assigned cluster ID).

The flow inside the fit method of the KMeans object is shown in Fig. 3.

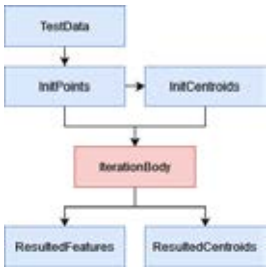



Figura 3. Flow in our $k$-means implementation.

The IterationBody is executed for each iteration. It assigns vectors to new clusters and then calculates new centroid values. When a domain $k$-means algorithm converges, the domain's vectors and centroids are output from the iteration body.

The following are DataStreams in IterationBody

- Centroids The input datastream of centroids fed into the iteration.
- Points The input datastream of vectors fed into the iteration.
- NewPoints Contains vectors with newly assigned cluster IDs. If no centroids have moved since the last iteration, these vectors are flagged as finished.
- CentroidsStillWithUs Centroids that will continue in the iteration. Centroids that are not flagged as finished.
- PointsStillWithUs Vectors that will continue in the iteration. Vectors that are not flagged as finished.
- NewCentoridValues The tuple containing domain, new vector values and IDs of centroids.
- ResultedCentorids Centroids that are marked as finished. The output of the iteration.
- ResultetPoints Vectors that are marked as finished. The output of the iteration.

The following are Operators in IterationBody as seen in Figure 2.

- PointUpdater Takes in vectors and a broadcasted centroid stream stored in the operator's state. If centroids in the domain have moved, this operator calls update on the vectors, and the Point object calculates new clusters. If not, the vector is marked as finished.
- CentroidUpdater Takes in tuples containing new vector values of centroids, along with the centroids. Calls update on each centroid.
- CentroidFilter A filter that splits the centroids into two streams, one for finished centroids and one for centroids in domains that have not converged.
- PointFilter Same as CentroidFilter, but for Points.
- NCVOperator Calculates new centroid values (NCV) for each domain based on updated points.

## VI. Results and discussion

If correctly implemented, all the improvements of the $k$-means algorithm must give the same results when it comes to *accuracy* of the classification. We have verified this with our proposed IDS - all the implementations gave the same *Positive Predictive Value - PPV* (2).

$$PPV = \frac{BR * TPR}{BR * TPR + (1 - BR) * FPR}, \tag{2}$$

where $BR$ is the *base rate* - the probability of an attack (this is calculated on the test data set NSL KDD by counting the total number of attacks in it), $TPR$ is the *True Positive Rate*, which is the probability of correct detection by the IDS, and $FPR$ is the *False Positive Rate*, which is the probability of false alarm produced by the IDS. For our proposed system, we have achieved the value of the $PPV$ of $\approx 0{,}7$, which is satisfactory for a system without training.

The property of our proposed IDS that distinguishes it from the ordinary implementations of $k$-means-based anomaly detection systems is the *efficiency*. We chose the reduction in the number of distance computations during the execution of the $k$-means algorithm compared to the number of distance computations in the naïve $k$-means as a measure of efficiency. The results obtained on the NSL KDD data set are presented in Table I

From Table I, we can see that the improvements of $k$-means discussed in this paper significantly reduce the number of necessary distance computations during the execution of the algorithm. With no parallelization, when the Apache Flink is

Tabla I
REDUCTION IN NUMBER OF DISTANCE COMPUTATIONS.

| Version of k-means | % of distance computations skipped |
|---|---|
| Naïve k-means | $0{,}0\,\%$ |
| Compare-Means | $39{,}0\,\%$ |
| Elkan's improvement | $81{,}1\,\%$ |
| Hamerly's improvement | $59{,}8\,\%$ |

run on an ordinary office computer, there is only a small gain in speed using Compare-Means. This is due to the overhead in the number of operations introduced by the various $k$-means improvement methods. With parallelization, when Apache Flink runs on a computing cluster, the total execution time is signifficantly shorter than the time needed for execution of the $k$-means algorithm on an ordinary office computer.

## VII. CONCLUSION

In this paper, we propose an anomaly detection system based on the improved versions of the original $k$-means algorithm, where these improvements are implemented on the distributed computing platform Apache Flink. As expected, the implementations do not affect the accuracy of the classification, which remains on the satisfactory level. But the improvement in efficiency is significant since the reduction in the number of distance computations needed to complete the execution of the $k$-means algorithm achieves $81\,\%$. This leads to huge speed-ups if Apache Flink is run on distributed computing hardware. Also, Section IV-A introduce domains that has significant impacts on k-means-based IDS.

## REFERENCIAS


[1] Drake J., Hamerly G., "Accelerated $k$-means with adaptive distance bounds", *Proceedings of 5th NIPS Workshop on Optimization for Machine Learning*, Lake Tahoe, USA, 2012.

[2] Elkan C, "Using the triangle inequality to accelerate $k$-means", *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, Washington DC, USA, pp. 147–153, 2003.

[3] Hamerly G., "Making $k$-means even faster", *Proceedings of the SIAM International Conference on Data Mining, SDM 2010*, Columbus, Ohio, USA, pp. 130–140, 2010.

[4] Philips S., "Acceleration of $k$-means and related clustering algorithms", *Proceedings of ALENEX 2002*, LNCS 2409, pp. 166–177, 2002.

[5] `https://flink.apache.org/`.

[6] Forgy E., "Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications", *Biometrics*, Vol. 21, No. 3, pp. 768, 1965.

[7] Lloyd S., "Least squares quantization in PCM", *IEEE Trans. Info. Theory*, Vol. IT-28, No. 2, March 1982, pp. 129-137.

[8] Arthur D., Vassilvitskii S., "`kmeans++`: The advantages of careful seeding", *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, USA, pp. 1027-–1035, 2007.

[9] Anderberg M., Cluster Analysis for Applications, Elsevier, 1973.

[10] MacQueen J. B., "Some methods for classification and analysis of multivariate observations", *Proc. 5th Symp. Math. Statist. and Probability*, Berkeley, USA, pp. 281–297, 1967.

[11] Celebi M. (Ed.), Partitional Clustering Algorithms, Springer Verlag, 2015.

[12] `https://hadoop.apache.org/`.

[13] `https://spark.apache.org/`.

[14] AlGhamdi S, Di Fatta G., "Efficient parallel $k$-means on MapReduce using triangle inequality", *Proceedings of 15th IEEE Intl. Conf. on Dependable, Autonomic and Secure Computing*, Orlando, Florida, USA, pp. 985–992, 2017.

[15] Ringdalen O., "Applying $k$-means with triangle inequality on Apache Flink, with applications in intrusion detection", Master thesis, NTNU Gjøvik, Norway, 2020.

[16] Yu Z., Tsai J., Intrusion Detection - A Machine Learning Approach, Imperial College Press, 2011.

[17] `https://www.unb.ca/cic/datasets/nsl.html`.

# Análisis de ataques a bases de datos de publicación continua en privacidad sintáctica

Adrián Tobar Nicolau
Universidad Politécnica de Cataluña
Barcelona, España
adrian.tobar@upc.edu

Javier Parra-Arnau
Universidad Politécnica de Cataluña
Barcelona, España
javier.parra@upc.edu

Jordi Forné
Universidad Politécnica de Cataluña
Barcelona, España
jordi.forne@upc.edu

***Resumen*—La publicación de información privada puede ser muy útil en el minado de datos, pero una anonimización previa es necesaria para garantizar la privacidad de los individuos de quienes se obtiene dicha información. El tipo de amenazas a las que el proceso de anonimización se tiene que enfrentar varía en función del marco de publicación. En el caso de las publicaciones dinámicas es necesario tener en cuenta, tanto los ataques diseñados para el caso estático, como los propios de la publicación múltiple. En este artículo nos fijamos en este tipo de ataques propios del entorno de publicación dinámica. Presentamos los principales ataques propios de las bases de datos dinámicas, es decir, que son editables vía inserciones, eliminaciones, actualizaciones y reinserciones de tuplas.**

***Index Terms*—privacidad sintáctica, bases de datos dinámicas, anonimización**

## I. Introducción

La publicación de bases de datos ha permitido el desarrollo del minado de datos y la mejora de los clasificadores en diferentes campos tales como el de sistema de soporte a las decisiones o el análisis estadístico. Ahora bien, la publicación de bases de datos privadas conlleva un riesgo para la privacidad de los individuos que la conforman. Para proteger a los participantes de estas bases de datos se han establecido diferentes regulaciones y restricciones a la publicación y diseminación de la información [1]. La comunidad de Control de Revelación Estadística [2] (*Statistical Disclosure Control*) ha desarrollado diversos mecanismos para garantizar la privacidad de los individuos a la vez que se preserva gran parte de la utilidad de los datos. En un principio, el único caso considerado fue el de publicación simple o estático que consideraba una única publicación de los datos. Según aumentaba la necesidad de abarcar más estructuras de datos, nuevos modelos fueron estudiados, en particular, los marcos de publicación dinámica. Existen cuatro categorías principales de publicación dinámica:

- Publicación múltiple. Se hacen varias publicaciones de la misma base de datos cada una conteniendo un subconjunto de los atributos. Este caso es de interés cuando la base de datos es de gran tamaño y diferentes instituciones están interesadas en unos pocos atributos. Al no necesitarse la publicación completa, se puede aumentar la utilidad de cada publicación parcial.
- Publicación secuencial. Versiones, cada vez más completas, de la base de datos se van publicando. Este caso se considera cuando la base de datos está siendo formada y publicada simultáneamente. En general, se asume que el conjunto de tuplas es fijado y el de atributos va aumentando.
- Flujo de datos. Se recibe constantemente información nueva y esta tiene que ser publicada periódicamente. Solo se publica la información nueva. Se prioriza la rápida publicación y el orden en que ha llegado la información.
- Publicación continua. Se hacen publicaciones de una base de datos cambiante. Entre publicaciones, la base de datos puede variar a través de inserciones de tuplas nuevas, eliminaciones, reinserciones de tuplas previamente eliminadas o actualizaciones de microdatos. El conjunto de atributos no cambia. La base de datos se publica completamente, lo que causa repeticiones de información entre publicaciones.

En este artículo se realiza una clasificación de los ataques a la privacidad propios de la publicación continua, en particular, ataques considerados en privacidad sintáctica. Con este fin, a continuación, damos detalles sobre las posibles variaciones en las bases de datos y atacantes del marco de publicación continua.

## II. Preliminares

Esta sección introduce la información básica necesaria para una correcta presentación del contenido de este documento. Empieza con una subsección de notación; continúa con una exposición de los posibles tipos de bases de datos y termina con una clasificaciones de los posibles atacantes.

### *II-A. Notación*

*Definición 2.1:* Usamos las siguiente notación:

- $t, t[QI], t[SD]$: tupla, cuasi identificadores (QI) y dato sensible (SD) de la tupla $t$.
- $T_i$: base de datos en el instante $i$.
- $\mathbf{T} = \{T_1, ..., T_n\}$: valores históricos de la base de datos.
- $T^*$: base de datos anonimizada.
- $\mathbf{T}^* = \{T_1^*, ..., T_n^*\}$: valores históricos de la base de datos anonimizada; también denominados publicaciones.
- $\mathbf{TT} = \cup_{i=1}^{n} T_i$: unión de las bases de datos; lista de tuplas sin repeticiones (resp. $\mathbf{TT}^*$).
- $\mathbf{TT}_R = \sqcup_{i=1}^{n} T_i$: unión disjunta de las bases de datos; lista de tuplas con repeticiones (resp. $\mathbf{TT}_R^*$).
- $Q(t, T^*)$: clase de $t$ en $T^*$, es decir, tuplas en $T^*$ que tienen los mismos QIs que $t$ (incluyendo a $t$).
- $SD(A)$: conjunto de valores sensibles de las tuplas en el conjunto A. Lo denota como firma de $A$.
- $C(p, T_i^*)$: conjunto de tuplas en $T_i^*$ que concuerdan con el individuo $p$, es decir, las tuplas que tienen los mismos cuasi identificadores que $p$ o generalizaciones de estos.

Tabla I: Tipos de bases de datos.

| Bases de datos | INS | DEL | UPD | REINS |
|---|---|---|---|---|
| Estática | No | No | No | No |
| Incremental | Si | No | No | No |
| Dinámica | Si | Si | No | No |
| Completamente Dinámica | Si | Si | Si | Si |

Dos tuplas concuerdan si pueden corresponder al mismo individuo.

- Intervalo de participación $[x, y]$ de $t$: intervalo en el que la tupla ha participado en la base de datos, es decir, para todo $i \in [x, y]$ se cumple $t \in T_i$.

### *II-B. Tipos de bases de datos dinámicas*

La Tabla I muestra los tipos de bases de datos que se consideran en publicación continua (originalmente presentado en [3]). La base de datos incremental solo contempla la inserción (INS) de nuevas tuplas. La base de datos dinámica permite la inserción de nuevas tuplas además de su eliminación (DEL). Finalmente, la base de datos completamente dinámica permite inserciones, eliminaciones, la actualización (UPD) de microdatos de cada tupla y la reinserción (REINS) de tuplas previamente eliminadas.

### *II-C. Tipos de atacante*

La Tabla II muestra los diferentes tipos de atacante de la literatura en relación a ataques de bases de datos de publicación continua. Los parámetros son:

- Conocimiento de la participación: información de qué individuos aparecen en cada publicación. Esta información puede ser de todos los individuos, indicado con un Si, o de uno solo, indicado con un "singular".
- Conocimiento de los cuasi identificadores (QI): saber la información no sensible de los individuos de la base de datos. En combinación con información externa, los QI pueden ser usados para identificar parcialmente a los individuos de la base de datos. Puede ser de un solo individuo (singular), de un subconjunto (acotado) o de todos los participantes (Si).
- Conocimiento temporal (*temporal knowledge*, TK): poder determinar qué inserciones, eliminaciones y reinserciones se han hecho en la base de datos. Puede ser total o limitado a un subconjunto de tuplas.
- Conocimiento de información sensible (*sensitive data knowledge*, SDK): conocimiento de información sensible de la base de datos, es decir, conocer el SD de un subconjunto de los individuos. Siempre es acotado a un subconjunto. Puede ser probabilístico (P.).
- Conocimiento probabilístico QI/SD (*sensitive background knowledge*, SBK): información probabilística de la correlación entre los QIs y los SDs. Por ejemplo, como de probable es que un hombre de avanzada edad tenga reuma en comparación a un hombre joven.
- Conocimiento de correlación (*correlation background knowledge*, CBK): conocimiento probabilístico de como se actualizan los SDs. Por ejemplo, un individuo es más probable que tenga cáncer en fase 2 si ha tenido previamente cáncer en fase 1. Nótese que este conocimiento no puede ser usado sin tener alguna información sobre el SD previo de la tupla.

## III. Clasificación de ataques

En función del tipo de base de datos y el conocimiento de un atacante, diferentes ataques son posibles a la hora de extraer información de una base de datos dinámica. Es relevante recordar que una base de datos dinámica es susceptible a ser atacada con ataques diseñados para una base de datos estática en cada una de sus publicaciones. A continuación, presentamos los principales ataques en la literatura.

### *III-A. Ataque de intersección*

El ataque de intersección [4] es posible cuando un atacante es capaz de identificar parcialmente a su objetivo, es decir, reducir a un pequeño subconjunto de tuplas candidatas a ser la tupla de su objetivo. El ataque consiste en intersecar los valores sensibles de cada subconjunto para obtener así el valor sensible o una corta lista de valores candidatos.

Sea A un atacante y $\mathbf{T} = \{T_1, T_2, ..., T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, ..., T_n^*\}$ los valores históricos de una base de datos y sus anonimizaciones. Un ataque de intersección actúa de la siguiente manera. Sea $p$ un individuo con QI conocidos.

- Para cada $T_i^* \in \mathbf{T}^*$, usa los QIs de $p$ para calcular $C_i = C(p, T_i^*)$, i.e., el conjunto de tuplas que pueden ser $p$.
- Para cada intervalo máximo de participación $[x, y]$ donde $p$ no haya cambiado su SD, calcula la intersección

$$\bigcap_{i=x \wedge C_i \neq \emptyset}^{y} SD(C_i),$$

donde $SD(C)$ denota el conjunto de SDs de $C$. El valor sensible de $p$ está en dicha intersección.

La Tabla IV muestra en qué casos el ataque de intersección (INT) se puede utilizar. Obsérvense las Figuras 1a y 1b, correspondientes a dos publicaciones de una base de datos incremental. Un gestor de los datos podría considerar que $l$-diversidad es suficiente para garantizar la privacidad pero, bajo un atacante con conocimiento de QIs y la participación en la base de datos del individuo 1, el atacante puede razonar a partir de la Figura 1a que el individuo tiene VIH o fiebre y de la Figura 1b que tiene VIH o acné deduciendo así que su SD es VIH.

### *III-B. Ataque de correspondencia*

Los ataques de correspondencia fueron originalmente presentados en [4] y formalizados en [5]. El conocimiento temporal permite comparar publicaciones y extraer información de la (no) participación de los individuos. Se consideran tres posible tipos: *forward-attack, cross-attack* y *backward-attack*. Véase la Tabla III.

Sea A un atacante y $\mathbf{T} = \{T_1, T_2, ..., T_n\}$, $\mathbf{T}^* = \{T_1^*, T_2^*, ..., T_n^*\}$ los valores históricos de una base de datos y sus anonimizaciones. Sea $p$ un individuo con QI conocidos. Para cada intervalo máximo de participación $[x, y]$ donde $p$ no haya cambiado su SD, para cada par $T_i^*, T_{i+1}^*$ con $i \in [x, y-1]$, un *forward-attack* actúa de la siguiente manera:

- Calcula el conjunto $C_i = C(p, T_i^*)$
- Calcula el conjunto de tuplas en $T_{i+1}^*$ que concuerdan con las de $C_i$ con QIs generalizaciones de los de $p$. La tupla de $p$ está en ese conjunto.

Un *cross-attack* actúa de la siguiente manera:

Tabla II: Tipos de atacantes.

| ATACANTE | PARTICIPACIÓN | QI | T.K. | S.D.K. | S.B.K. | C.B.K. |
|---|---|---|---|---|---|---|
| TRIVIAL | No | No | No | No | No | No |
| MÍNIMO | SINGULAR | SINGULAR | No | No | No | No |
| INCOMPLETO | Si | ACOTADO | No | No | No | No |
| OBJETIVO | Si | SINGULAR | Si | No | No | No |
| LIMITADO | Si | Si | ACOTADO | No | No | No |
| COMPLETO | Si | Si | Si | No | No | No |
| INTERIOR | Si | Si | Si | ACOTADO | No | No |
| PROBABILÍSTICO | Si | Si | Si | P. ACOTADO | Si | Si |

(a) Primera publicación

| Id | SEXO | EDAD | S.D. |
|---|---|---|---|
| 1 | - | $[20-22]$ | VIH |
| 2 | - | $[20-22]$ | FIEBRE |

(b) Segunda publicación

| Id | SEXO | EDAD | S.D. |
|---|---|---|---|
| 1 | HOMBRE | [19-20] | VIH |
| 3 | HOMBRE | [19-20] | ACNÉ |
| 2 | MUJER | 22 | FIEBRE |
| 4 | MUJER | 22 | TOS |

(c) Tercera publicación

| Id | SEXO | EDAD | S.D. |
|---|---|---|---|
| 1 | HOMBRE | [19-20] | VIH |
| 3 | HOMBRE | [19-20] | ACNÉ |
| 2 | MUJER | [22-23] | FIEBRE |
| 5 | MUJER | [22-23] | VIH |

Figura 1: Publicaciones 2-diversas.

Tabla III: Tipos de ataque de correspondencia.

| | Participación | Publicación atacada | Publicación informativa |
|---|---|---|---|
| F-attack | $T_1, T_2$ | $T_1^*$ | $T_2^*$ |
| C-attack | $T_1, T_2$ | $T_2^*$ | $T_1^*$ |
| B-attack | $T_2$ | $T_2^*$ | $T_1^*$ |

- Calcula el conjunto $C_{i+1} = C(p, T_{i+1}^*)$
- Calcula el conjunto de tuplas en $T_i^*$ que concuerdan con las de $C_i$ con QIs generalizaciones de los de $p$. La tupla de $p$ está en ese conjunto.

Un *backward-attack* actúa de la siguiente manera:

- Calcula el conjunto $C_x = C(p, T_x^*)$
- Calcula el conjunto $X$ de tuplas en $T_x^*$ que también participan en $T_{x-1}^*$ . La tupla de $p$ está en $C_x \setminus X$.

La Tabla IV muestra en que casos el ataque de correspondencia (COR) se puede utilizar.

Obsérvense las Figuras 2a y 2b correspondientes a una base de datos incremental. Un atacante puede realizar un *forward-attack* sobre estas con el conocimiento de los QI de un usuario y su participación. En la Figura 2a, de las tres primeras tuplas no todas pueden ser de la forma *Francia, abogado, fiebre* ya que en la Figura 2b solo aparecen dos tuplas que puedan tener esta forma. De manera similar, un *cross-attack* razonaría que como solo hay dos tuplas con SD VIH en la Figura 2a, al menos una de las tuplas de la Figura 2b con ID 4,5 o 6 no participó en la Figura 2a. En el caso de un *backward-attack*, al menos una de las tuplas con ID 1,2,3 de la Figura 2b tiene que haber participado en la Figura 2a, ya que de lo contrario una tupla de la Figura 2a no estaría presente en la figura 2b.

(a) Primera publicación

| Id | PROCEDENCIA | PROFESIÓN | S.D. |
|---|---|---|---|
| 1 | EUROPA | ABOGADO | FIEBRE |
| 2 | EUROPA | ABOGADO | FIEBRE |
| 3 | EUROPA | ABOGADO | FIEBRE |
| 4 | EUROPA | ABOGADO | VIH |
| 5 | EUROPA | ABOGADO | VIH |

(b) Segunda publicación

| Id | PROCEDENCIA | PROFESIÓN | S.D. |
|---|---|---|---|
| 1 | REINO UNIDO | - | FIEBRE |
| 2 | REINO UNIDO | - | FIEBRE |
| 3 | REINO UNIDO | - | FIEBRE |
| 9 | REINO UNIDO | - | VIH |
| 10 | REINO UNIDO | - | VIH |
| 4 | FRANCIA | - | VIH |
| 5 | FRANCIA | - | VIH |
| 6 | FRANCIA | - | VIH |
| 7 | FRANCIA | - | FIEBRE |
| 8 | FRANCIA | - | FIEBRE |

Figura 2: Publicaciones 2-diversas y 5-anónimas.

### *III-C. Ausencia y presencia crítica*

La ausencia crítica [6] es un fenómeno que se manifiesta cuando una tupla con SD poco frecuente es eliminada de la base de datos. Si el método usado para la anonimización no tiene en cuenta la desaparición de un SD, un atacante con conocimiento temporal podría detectar su ausencia y usarlo a su favor. De de misma forma la presencia crítica es el fenómeno contrario dónde un SD nuevo es añadido a la base de datos; este fenómeno es un caso particular de *backward-attack*.

Los ataques de ausencia y presencia crítica actúan bajo los siguientes razonamientos:

- (Ausencia) Cada dato sensible que haya reducido su frecuencia debe corresponder a alguna tupla eliminada de la base de datos.
- (Presencia) Cada dato sensible que haya aumentado su frecuencia debe corresponder a una tupla nueva o reinsertada.

La Tabla IV muestra en qué casos la ausencia crítica (AUS) y presencia crítica (PRE) se pueden utilizar. Obsérvense las

(a) Primera publicación

| ID | EDAD | SEXO | SD |
|---|---|---|---|
| 1 | [18-50] | HOMBRE | FIEBRE |
| 3 | [18-50] | HOMBRE | ACNÉ |
| 2 | [21-22] | MUJER | FIEBRE |
| 4 | [21-22] | MUJER | VIH |

(b) Segunda publicación

| ID | EDAD | SEXO | SD |
|---|---|---|---|
| 1 | [20-22] | - | FIEBRE |
| 4 | [20-22] | - | VIH |
| 3 | [18-50] | HOMBRE | ACNÉ |
| 5 | [18-50] | HOMBRE | FIEBRE |

Figura 3: Publicaciones 2-diversas.

Figuras 1a, 1b y 1c. Un ataque de presencia critica revela que entre la publicación de las Figuras 1a y 1b se han añadido dos tuplas con SD *acné* y *tos* es más, si se tiene conocimiento de los QIs de los nuevos participantes, se deduce que el hombre tiene acné y la mujer tos. Por otro lado, un ataque de ausencia sobre las Figuras 1b y 1c muestra que el individuo que ha dejado de participar tenía SD tos ya que esta ya no aparece en la Figura 1c.

### *III-D. Ataque de equivalencia*

El ataque de equivalencia [7] se diferencia del resto en que, en vez de buscar una revelación del dato sensible, busca relacionar conjuntos disjuntos de tuplas con el mismo dato sensible. Se considera un ataque de equivalencia el encontrar dos conjuntos $P_1$ y $P_2$ tal que:

- $P_1 \cap P_2 = \emptyset$ y $P_1, P_2 \subset \mathbf{TT}^*_R$.
- $\forall t_1, t_2 \in P_1 \cup P_2$, $t_1$ y $t_2$ no corresponden al mismo individuo.
- $|P_1| = |P_2| = e$
- $SD_R(P_1) = SD_R(P_2)$.

La Tabla IV muestra en qué casos el ataque de equivalencia (EQUI) se puede utilizar. Obsérvense las Figuras 3a y 3b. Un atacante con conocimiento del QI del individuo 3 puede deducir que está en la clase $\{1, 3\}$ de la Figura 3a y en la clase $\{3, 5\}$ de la Figura 3b. Como ambas clases tienen el mismo conjunto de SDs, los individuos 1 y 5 tienen el mismo SD.

### *III-E. Ataque interior*

El ataque interior [8] es aquel en el que el atacante tiene información sensible de una parte de las tuplas de la base de datos y con esta intenta extraer más. Este ataque puede usarse en combinación con el ataque de equivalencia o con las correlaciones históricas (definido a continuación). De manera similar, se puede realizar el mismo ataque con información probabilística de los datos sensibles. Sean $T^*_i, T^*_j \in \mathbf{T}^*$ con $i \neq j$ dos publicaciones, $Q_1 \in T^*_i, Q_2 \in T^*_j$ dos clases con $SD_R(Q_1) = SD_R(Q_2)$ y $U_1, U_2 \subset U(\mathbf{TT}_R)$ dos conjuntos de individuos. Decimos que $U_1$ y $U_2$ están en correlación histórica [8] si:

- $U_1 \subset U(Q_1)$.
- $U_2 \subset U(Q_2)$.
- $U(Q_1) \setminus U_1 = U(Q_2) \setminus U_2$.
- Los atributos de los individuos en $U(Q_1 \cup Q_2)$ no han cambiado en el intervalo $[i, j]$.

En [8] se demuestra cómo dos conjuntos $U_1, U_2$ en correlación histórica satisfacen $SD_R(U_1) = SD_R(U_2)$. En general, un ataque probabilístico será el uso de ataques de equivalencia y correlaciones históricas para relacionar tuplas comprometidas con el resto de la base de datos causando así un efecto cascada de revelación de SDs. La Tabla IV muestra en qué casos el ataque interior (RIOR) se puede utilizar. Obsérvense las Figuras 3a y 3b. Siguiendo el ejemplo de ataque de equivalencia se deduce que los individuos 1 y 5 tienen el mismo SD. Un atacante con el conocimiento de que el SD del individuo 3 es *acné*, también deduce que tanto 1 como 5 tienen SD fiebre. Sabiendo que el individuo 1 también participa en la Figura 3b puede también concluir que el individuo 4 tiene VIH. Finalmente, volviendo a la Figura 3a, descubre que el individuo 2 tiene fiebre revelando así el SD de todos los participantes.

### *III-F. Ataque probabilístico*

Los ataques probabilísticos [9] con SBK consisten en usar conocimiento probabilístico de la base de datos para poder afirmar con cierta probabilidad $p$ el valor del SD de una tupla. Si un atacante es capaz de realizar dicha asociación se dice que ha hecho un ataque de asociación con valor $p$. Estos ataques se basan en la existencia de correlaciones entres QIs y SDs, con el conocimiento de estas correlaciones es posible asignar probabilidades a las diferentes asignaciones individuo, valor sensible, eligiendo así la más probable. En el caso de bases de datos completamente dinámicas, se usa también información probabilística de las posibles actualizaciones del SD, es decir, CBK.

La Tabla IV muestra en qué casos el ataque probabilístico (PROB) se puede realizar. Obsérvese la Figura 3b. Consideremos un atacante probabilístico con el conocimiento de que los QI de los individuos 3 y 5 son *18, hombre* y *50, hombre*, respectivamente. Con su SBK sabe que el *acné* es más probable en gente joven asociando así que el individuo 3 tiene *acné* como SD y el individuo 5 fiebre. Con esta información, usando ataques de equivalencia, el adversario podría hacer un ataque probabilísticamente análogo al del ataque interior.

## IV. Métodos de protección

En esta sección presentamos el estado del arte en nociones sintácticas para bases de datos de publicación continua. La Figura 4 muestra las nociones en relación a las combinaciones de atacantes y bases de datos dinámicas.

- BCF-anonimato. Es una noción limitada a bases de datos incrementales que limita la efectividad de los ataques de correspondencia.
- m-invarianza. Una de las principales nociones en publicación continua. Impone que las clases de cada tupla no varíen sus conjuntos de SDs, de esta manera no solo limita los ataques de intersección sino que acota superiormente la probabilidad de revelación del SD.
- Pvr-safety. Esta noción contempla atacantes interiores y acota la probabilidad de producir un efecto cascada de filtración de SD asegurando un tamaño mínimo de clase para cada tupla.

Tabla IV: Combinaciones de base de datos/atacante dónde los diferentes ataques pueden realizarse.

| | INCREMENTAL | DINÁMICA | COMPLETAMENTE DINÁMICA |
|---|---|---|---|
| MÍNIMO | INT | | |
| INCOMPLETO | INT | | |
| OBJETIVO | INT, COR, PRE | INT, COR, AUS, PRE | INT, COR, AUS, PRE |
| LIMITADO | INT, COR, PRE, EQUI | INT, COR, AUS, PRE, EQUI | INT, COR, AUS, PRE, EQUI |
| COMPLETO | INT, COR, PRE, EQUI | INT, COR, AUS, PRE, EQUI | INT, COR, AUS, PRE, EQUI |
| INTERIOR | INT, COR, PRE, EQUI, RIOR | INT, COR, AUS, PRE, EQUI, RIOR | INT, COR, AUS, PRE, EQUI, RIOR |
| PROBABILÍSTICO | INT, COR, PRE, EQUI, RIOR, PROB | INT, COR, AUS, PRE, EQUI, RIOR, PROB | INT, COR, AUS, PRE, EQUI, RIOR, PROB |

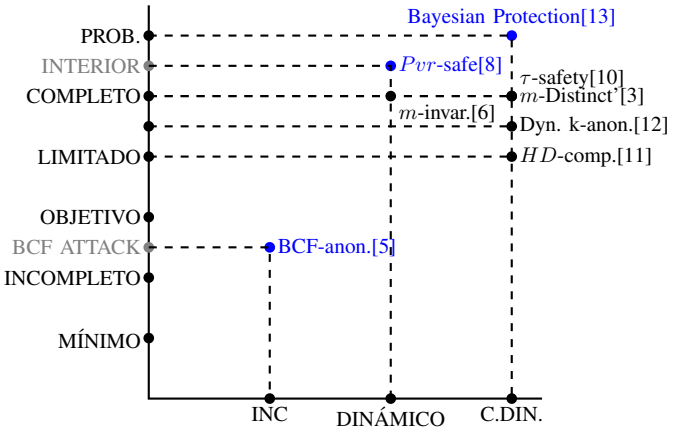



Figura 4: Nociones sintácticas.

- HD-composition'. Método que fuerza la estructura de las clases a través de un sistema de portador (holder) y señuelo (decoy) para esconder los SDs entre varios candidatos. Acota la probabilidad de revelación del SD.
- k-anonimato dinámico (Dyn. k-anon.). Extiende k-anonimato a la publicación continua limitando la capacidad de los ataques de intersección.
- m-Distinct'. Método similar a m-invarianza para bases de datos completamente dinámicas. Acota la probabilidad de revelación del SD pero requiere propiedades extra sobre la base de datos.
- $\tau$-safety. Extensión de m-invarianza sobre bases de datos completamente dinámica. Acota la probabilidad de revelación del SD pero asume actualizaciones arbitrarias, es decir, que las actualizaciones no dependen de los valores previos.
- Protección Bayesiana. Método usado contra clasificadores bayesianos. Altera la base de datos para reducir la efectividad de un clasificador entrenado con SBK y CBK.

## V. Conclusiones

En este artículo hemos analizado las principales vulnerabilidades que se presentan propiamente en una base de datos dinámica que es publicada de forma continua. Con este fin, hemos llevado a cabo una clasificación sistemática de los principales ataques específicos de las bases de datos dinámicas, y hemos analizado los mecanismos de protección más relevantes de la literatura. Como línea de trabajo futuro, nos planteamos comparar la eficacia de los ataques probabilísticos en bases de datos dinámicas con respecto a los ataques estáticos.

## Agradecimientos

El proyecto que dio lugar a estos resultados recibió el apoyo de una beca de Fundación "la Caixa" (ID 100010434) y del programa "European Union's Horizon 2020 research and innovation" bajo la subvención Marie Skłodowska-Curie No 847648. El código de la beca es LCF/BQ/PR20/11770009. Este trabajo también ha sido subvencionado por el Gobierno de España bajo el proyecto de investigación "Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)" (PID2020-113795RB-C31/AEI/10.13039/501100011033).

## Referencias

[1] "General data protection regulation." [Online]. Available: https://gdpr-info.eu/

[2] L. Willenborg and T. DeWaal, *Elements of Statistical Disclosure Control*. New York: Springer-Verlag, 2001.

[3] F. Li and S. Zhou, "Challenging More Updates: Towards Anonymous Re-publication of Fully Dynamic Datasets," *arXiv e-prints*, p. arXiv:0806.4703, Jun. 2008.

[4] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *Secure Data Manage.*, 2006.

[5] B. C. M. Fung, K. Wang, A. W.-C. Fu, and J. Pei, "Anonymity for continuous data publishing," in *Proc. 11st Int. Conf. Extending Database Technol.: Adv. Database Technol.*, ser. EDBT '08. New York, NY, USA: Assoc. for Comput. Machinery, 2008, p. 264–275.

[6] X. Xiao and Y. Tao, "M-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. 2007 ACM SIGMOD Int. Conf. Manage. Data*, ser. SIGMOD '07. New York, NY, USA: Assoc. for Comput. Machinery, 2007, p. 689–700.

[7] Y. He, S. Barman, and J. F. Naughton, "Preventing equivalence attacks in updated, anonymized data," in *Proc. 2011 IEEE 27st Int. Conf. Data Eng.*, ser. ICDE '11. USA: IEEE Comput. Soc., 2011, p. 529–540.

[8] D. Riboni and C. Bettini, "Cor-split: Defending privacy in data re-publication from historical correlations and compromised tuples," in *Scient., Stat. Database Manage.*, M. Winslett, Ed. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 562–579.

[9] O. Temuujin, J. Ahn, and D.-H. Im, "Efficient l-diversity algorithm for preserving privacy of dynamically published datasets," *IEEE Access*, vol. 7, pp. 122 878–122 888, 2019.

[10] A. Anjum and G. Raschia, "Anonymizing sequential releases under arbitrary updates," in *Proc. Joint EDBT/ICDT 2013 Workshop*, ser. EDBT '13. New York, NY, USA: ACM, 2013, p. 145–154.

[11] Y. Bu, A. W. C. Fu, R. C. W. Wong, L. Chen, and J. Li, "Privacy preserving serial data publishing by role composition," *Proc. VLDB Endow.*, vol. 1, no. 1, p. 845–856, Aug. 2008.

[12] J. Salas and V. Torra, "A general algorithm for k-anonymity on dynamic databases," in *Data Priv. Manage., Cryptocurrencies and Blockchain Technol.*, J. Garcia-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, Eds. Cham: Springer-Verlag, 2018, pp. 407–414.

[13] F. Amiri, N. YAzdani, A. Shakery, and S.-S. Ho, "Bayesian-based anonymization framework against background knowledge attack in continuous data publishing," *Trans. Data Priv.*, vol. 12, 2019.

# A Comparison of Layer 2 Techniques for Scaling Blockchains

Adrià TORRALBA-AGELL
Universitat Oberta de Catalunya and
CYBERCAT,
Center for Cybersecurity Research of Catalonia
atorralbaag@uoc.edu

Cristina PÉREZ-SOLÀ
Universitat Oberta de Catalunya and
CYBERCAT,
Center for Cybersecurity Research of Catalonia
cperezsola@uoc.edu

***Abstract*—Since the creation of Bitcoin, back in 2009, many other cryptocurrencies have appear, and its usage has been growing year after year. With this huge popularity, doubts about the ability of blockchains to become worldwide payment systems (or even universal mediums for general decentralised systems) begin to arise and, with them, solutions started to be explored. In this paper we first explain the blockchain scalability problem and then present a brief review and a comparison among some of the state-of-the-art techniques that are used to scale blockchains on Layer 2 (or *off-chain*), analysing properties related to their Usability, Security and Cost.**

***Index Terms*—zero-knowledge techniques, blockchain, scaling blockchain, payment channels, zkRollups, optimistic rollups.**

## I. Introduction

With the apparition of the Bitcoin white paper in 2009 [1] we have seen a huge rise in popularity over the blockchain technology. The massive adoption and application of this technology has brought several changes to the way we interact with the world. From creating digital, secure and decentralised currencies [1] to implementing fair, secure voting system [2], going through enabling secure and reliable digital identification over the Internet [3].

With this rise in popularity, thousands of different applications; such as dAPPs [4], DeFi [5], NFTs [6] and blockchain games [7]; appeared that had made use of this technology. With it, many blockchains had suffered from heavy congestion resulting in poor performance and/or high transaction fees[1].

With this problem, many proposals have been presented in an aim to make blockchain networks more performant. The adoption of some of these solutions has lead to agitated debates within the community, and some of them even originated hard forks that ended up with the creation of new cryptocurrencies.

In general, there are two primary ways to scale networks: scaling the main network (or Layer 1), or creating networks on top of it (or Layer 2). In this paper, we focus on the later solutions, and provide a comparison of their properties in terms of Usability, Security and Cost. We devote special attention to *zkRollups*, one of the Layer 2 solutions that is currently receiving much thought from both the research and the developing communities. *zkRollups* are based on Zero-Knowledge Proofs, and provide a way to compress a batch of transactions onto a succinct proof, that is easier (and faster) to verify compared to checking and verifying every single transaction in the batch. Yadav et al. [8] have analysed multiple solutions at all different Layers (in particular, they have considered solutions for Layer 0, Layer 1 and Layer 2).

The rest of this paper is organised as follows: in Section II, we present the concrete problem of blockchains with scalability, in Section III we present the actual existing solutions to the aforementioned problem. In Section IV, we present a comparison among the different solutions that implement some kind of scalability technology for blockchains. Finally, in Section V, we present and draw the conclusions for this article, as well as the future work.

## II. Blockchain scalability problem

As we stated in the previous Section, blockchain networks usually suffer from scalability problems. Before exploring the solutions that are currently being studied and deployed to solve this problem, we first review the concept of scalability and explain the impact trivial solutions may have on the security and decentralisation of the networks.

The most common metric for measuring blockchain scalability is **transaction throughput**. All blockchains have a (in some cases variable) block size; which determines the amount of transactions that can be fitted in a block; and a block time; which determines how many units of computation can be processed per block and how fast a new block may be added. These two characteristics determine the *throughput* of a blockchain (as the amount of transactions per second the blockchain is able to confirm). This metric has the benefits of being easy to compute and somehow useful to compare different payment systems (even with traditional non-blockchain based ones). However, it falls short in capturing the diversity of operations a single transaction may convey.

Other popular metrics are **latency** (the time it takes for a transaction to be considered final); **bootstrap time** (the time it takes for a new node to synchronize with the network); **cost per confirmed transaction** (in terms of computation, network and storage resources); or **cost to maintain a full node** (also in terms of computation, networking and storage resources).

### A. The Blockchain Trilemma

The challenge of scaling blockchains comes from the need to scale without compromising the security and decentralization properties of the network. This problem is usually

[1]For instance, the Ethereum network reached values over 100 USD on transaction fees during peaks of both Ether price and gas transaction fees on December 2017 or August 2020.

referred to as the Blockchain Trilemma[2]. Let us explain each of the three properties and how are they interlinked in the context of blockchains:

**Security** reefers to the fact that every transaction published to the network is immutable and valid. In order to improve the speed of the network, it can be useful to reduce the number of nodes leading to more performant but more centralised and less secure networks. Blockchains like Nano (XNO) and IOTA are known to be networks that are quick and decentralised in exchange of less security.

**Decentralisation** reefers to how the control of the network is split across its participants. Having high decentralisation usually trades-off with the speed of the network, since the more decentralised a network is, the more verifiers you need to have in order to process transactions. Cryptocurrencies like XRP or EOS are known to prioritise speed and security with the cost of decentralisation.

**Scalability** reefers to the capacity of a network to support high transactional throughput and the ability to sustain growth in the future. Scalability usually trades-off with decentralisation and security since, the more decentralised a network is, the longer it takes to process transactions leading to slower performance. Bitcoin (BTC) and Ethereum (ETH) are known to prioritise decentralisation and security in exchange for scalability.

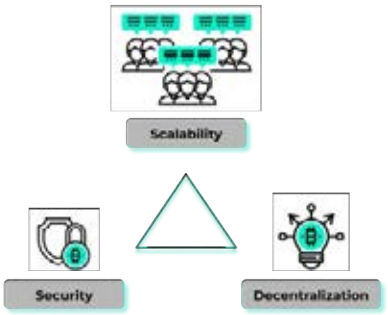



Figure 1. Blockchain Trilemma. Icons from [9]

Taking into account the interactions between scalability and both security and decentralization, the following two constraints are critical to successfully scale blockchains.

**Hardware Requirements** The speed of a blockchain network is determined by the ability of the weakest node in the network to verify transactions and hold its state. Hence, it is desirable to keep the costs to run a node (i.e. the hardware, bandwidth and the storage requirements) as low as possible in order to enable as many participants as possible to the network. Table I summarises the hardware requirements for Bitcoin, Ethereum and Solana blockchains.

**State Growth** State growth refers to how quickly the blockchain grows in the sense that, the more throughput a blockchain allows to happen per unit of time, the quicker the blockchain grows. Since the full nodes of the network store its history –an ever growing ledger– and those nodes should verify all transactions, new nodes can struggle with huge syncing times when joining the network.

[2]Concept coined by Vitalik Buterin, co-founder of the Ethereum Network.

## III. Blockchain scalability solutions

This section summarizes the two main approaches to scale blockchains, and reviews the different techniques that are being discussed and deployed for each one of the approaches (with emphasis on Layer 2 solutions).

### A. Layer 1 scaling (L1)

On the one hand, Layer 1 scalability solutions are focused on the consensus algorithm, the network and the data structure of the blockchain itself. Since the solutions in this layer are performed directly over the chain, these solutions are also commonly named as *on-chain* solutions. One of the main challenges here is to handle block size limit, since its increase directly affects transaction throughput but has consequences on decentralization. Other approaches to Layer 1 go through the implementation of techniques that enable splitting the work of building and verifying blocks across many nodes in the network (sharding).

### B. Layer 2 scaling (L2)

On the other hand, Layer 2 scaling solutions offer to withdraw computation from the main network (Layer 1) and perform this work *off-chain*. This is, instead of performing all the computing-consuming part of the activity onto the blockchain directly, you can perform the bulk and heavy part of the job over the network in the Layer 2.

There are three main approaches that implement Layer 2 scaling.

*1) Payment Channel Networks (PCN):* This system enables the construction of a peer-to-peer network on top of the main blockchain network that allows its participants to perform as many transactions as desired without the main restrictions inherited by the anchored blockchain. However, those payment channels have to overcome several other issues regarding security and reliability. The most well known implementations of PCN are the Lightning Network [10] for the Bitcoin blockchain, and the Raiden Network [11] for the Ethereum blockchain.

*2) Sidechains:* Sidechains build a whole new blockchain in parallel to the main blockchain. The assets can flow freely between both networks, however, the consensus mechanism, the tokens and even their security level are different.

Sidechains can interact in many different ways with the main blockchain. Usually, the main use case for them are exchanging assets between blockchains, for instance, implementing exchanges that allow to swap Bitcoin for Ether. However, other use cases are considered when implementing them, such as scalability.

*3) Rollups:* Rollups are a technique that allow to "roll-up" a batch of transactions and put them on the blockchain all together with a proof that the transactions included in the batch are correctly processed.

In all these three variants, there is only a Smart Contract on-chain which has two main tasks: (1) to process deposits and

| Network | Hard drive space | Number of CPU Cores | Amount of RAM | Internet bandwidth | Number of Nodes |
|---|---|---|---|---|---|
| **Bitcoin** | 350GB HDD | 1 | 1GB | 5Mbps | $\approx$ 10.000 |
| **Ethereum** | >500GB SSD | 2-4 | 4-8GB | 25Mbps | $\approx$ 6.000 |
| **Solana** | >1.5TB SSD | >12 | 128GB | 300Mbps | $\approx$ 1.200 |

Table I
COMPARISON OF BITCOIN, ETHEREUM AND SOLANA NETWORK IN TERMS OF HARD DRIVE, CPU CORES, RAM MEMORY AND BANDWIDTH REQUIREMENTS. DATA OBTAINED FROM [12].

| | | **Usability** | | | |
|---|---|---|---|---|---|
| **Scalability solution type** | **Technology name** | **General-purpose script / Turing Complete Machine** | **Separate client or software** | **Supported tokens** | **Native proprietary token** |
| **Payment Channels** | **Lightning Network** | No | Yes | Bitcoin (BTC) | No |
| | **Raiden Network** | Yes, native | Yes | ERC20 tokens | Yes, Raiden Network Token (RDN) |
| **Zero-Knowledge Rollups** | **zkSync** | Yes, in Zinc [18] | Yes | Ether (ETH), ERC20 tokens | No |
| | **Loopring 3.8** | No | Yes | Ether (ETH), ERC20 tokens | Yes, Loopring (LRC) |
| | **Starknet** | Yes, implemented using Cairo [19] | Yes | Ether (ETH), some ERC20, ERC721 tokens | No |
| **Optimistic Rollups** | **Arbitrum** | Yes, through ArbOS [20] (EVM compatible) | Yes | ERC20, ERC721 tokens | No |
| | **Optimism** | Yes, supports Solidity and Vyper [21] | Yes | ERC20, ERC721 tokens | Yes, Optimism (OP) |

Table II
TABLE COMPARING USABILITY

withdrawals and (2) verify *proofs* that everything happening off-chain is following the predefined set of rules.

For rollups, the way these proofs are generated and validated give rise to two different kinds of rollups: Optimistic rollups –which are backed by *fraud proofs*– and zkRollups –which are backed by *validity proofs*–.

There are many differences between fraud proofs and validity proofs. In short, fraud proofs present an evidence that a state transition was *incorrect*, while validity proofs present an evidence that a state transition was *correct*. Therefore, fraud proofs present an *optimistic* point of view, whereas validity proofs present a more pessimistic approach.

Due to the optimistic nature of the fraud proofs, they are not needed for every state transition, they are only required in a possible fraudulent scenario. For this reason, the main advantage is the fact that they require fewer computational resources since proofs are only needed in case a party is trying to cheat the rest of the participants. However, they come with a cost: interactivity and long withdrawal time. There is the need to provide a *challenge period* in which any party in the system can submit a fraud proof invalidating the batch of transactions. The implementation of this challenge period implies interactivity –which forces the node to be *live*– and long withdrawal times –since the challenge period should be long enough for it to be reliable (around 7 days)–.

Validity proofs, on the other hand, represent off-chain computation sent to the main network. The main advantages for this kind of proofs are the fact that the main network always have a correct Layer 2 state, and that this new state can be relied and trusted immediately (unlike fraud proofs). However, they come with the cost that every state transition needs for a proof (which should be both generated and verified), possibly impacting scalability.

## IV. COMPARISON OF LAYER 2 SOLUTIONS

In this section, we present three different tables comparing existing scalability solutions for blockchains in terms of Usability, Security and Cost. Our comparison considers examples of Payment Channel Networks, Zero-Knowledge Rollups, and Optimistic Rollups. In particular, we have chosen the most popular solutions of each kind, except for zkRollups where we have also considered representativity of the different Zero-Knowledge techniques as selection criterion. Our analysis thus includes the Lightning Network (LN) [10] and the Raiden Network [11] as Payment Channel Networks; zkSync [13], Loopring 3.8 [14] and Starknet [15] as Zero-Knowledge Rollups; and Arbitrum [16] and Optimism [17] as Optimistic Rollups.

It is important to note that the Lightning Network aims to scale the Bitcoin blockchain, while all the other solutions are implemented in order to scale the Ethereum network.

### A. Usability

Table II shows the comparison in terms of Usability. This category is intended to illustrate the versatility of the different scalability solutions. For this reason, we have considered

| | | Security | | | | | |
|---|---|---|---|---|---|---|---|
| **Scalability solution type** | **Technology name** | **Security model** | **Cryptographic primitives** | **ZK technique** | **Quantum resistant** | **Separate network** | **Type of network** |
| **Payment Channels** | **Lightning Network** | Inherited from L1 + node always online + censorship-resistant within time $t$ | Hash functions, digital signature | Not applicable | No | Yes | P2P |
| | **Raiden Network** | Inherited from L1 + node always online + censorship-resistant within time $t$ | Hash functions, digital signature | Not applicable | No | Yes | P2P |
| **Zero-Knowledge Rollups** | **zkSync** | Inherited from L1 + CRS always hidden + censorship-resistant within time $t$ | Pairings, KoE, minimal trusted setup | PLONK [22] | No | Yes | Centralised |
| | **Loopring 3.8** | Inherited from L1 + CRS always hidden + censorship-resistant within time $t$ | Pairings and trusted setup | zkSNARK | No | Yes | Centralised |
| | **Starknet** | Inherited from L1 + censorship-resistant within time $t$ | Hash Functions | zkSTARK [23] | Yes | Yes | Centralised |
| **Optimistic Rollups** | **Arbitrum** | Inherited from L1 + based on Game Theory + censorship-resistant within time $t$ | Fraud Proofs (Merkle Trees or SNARK/STARK) | Not applicable | No | Yes | Centralised |
| | **Optimism** | Inherited from L1 + based on Game Theory + censorship-resistant within time $t$ | Fraud Proofs (Merkle Trees or SNARK/STARK) | Not applicable | No | Yes | Centralised |

Table III
TABLE COMPARING SECURITY

if they provide some kind of **general-purpose scripting** (Turing Complete Machine), we have studied if they need an **additional** –separated– **client** or software, which are the **tokens** those solutions can handle and, finally, if they run a **native** –proprietary– **token** in order to interact with the solution.

Let us start with the scripting capabilities. We have found that almost all of the considered solutions present some kind of mechanism that enables the user to implement general-purpose Smart Contracts. However, the Lighting Network does not support this kind of scripting due to the limitations imposed by the Bitcoin Script.

Moreover, all those solutions need a separated –dedicated– client or software. In this sense, none of them are "built-in" or directly integrated into the L1 network.

Regarding the supported tokens, we have found that the LN only supports Bitcoin (BTC), while the rest of solutions implemented on top of the Ethereum Network, in general, support ERC20 tokens and, some of them, also support Ether (ETH) and/or ERC721 tokens. It is worth mentioning that Starknet is currently in an alpha stage and only supports Ether (ETH), some ERC20 and ERC721 tokens, but in the future they plan to support WBTC, USDC, USDT and DAI as well.

Finally, in this topic, some of these off-chain solutions have a native –proprietary– token in order to interact with the auxiliary network. In our listed technologies, the Raiden Network, Loopring and Optimism are the ones that have a native token, while the rest of the solutions inherit the token from the parent network, namely, Bitcoin for the LN and Ether for the rest of solutions.

### *B. Security*

Table III shows the comparison in terms of Security. In this table we present the different security-related features that these solutions have. In particular, we have considered the **security model**, the **cryptographic primitives**, the kind of specific **Zero-Knowledge technique** they implement in each zkRollup, whether they feature **quantum resistance** (i.e. the cryptographic primitive will be still secure once quantum computing is well stablished). Finally, we have studied the needed for independent and **separated network**, as well as, the **type of network** they implement.

The security model in which those technologies rely is very different among the three different scalability solution types.

| | | Cost | | |
|---|---|---|---|---|
| **Scalability solution type** | **Technology name** | **Fees** | **Processing time** | **Withdrawal time** |
| **Payment Channels** | **Lightning Network** | Fees for the funding transaction (+ possible hops) + closing transaction | Near instant | From 1 hour to several days |
| | **Raiden Network** | Similar to Lightning Network fee system | Near instant | Up to 3 hours |
| **Zero-Knowledge Rollups** | **zkSync** | $\approx$ 1/100th of mainnet costs for ERC20 tokens and $\approx$ 1/30th for Ether (ETH) transfers | Near instant | From 10 minutes to 7 hours |
| | **Loopring 3.8** | From 1/30th up to 1/100th of the mainnet costs for ETH and ERC20 tokens | Near instant | From 6 minutes to 2 hours |
| | **Starknet** | Fees only to post to L1, in the future fees also for L2 | Near instant | Not specified |
| **Optimistic Rollups** | **Arbitrum** | Up to 1/10th of the mainnet cost | Near instant | Around 7 days |
| | **Optimism** | L2 execution fee + L1 security fee | Near instant | Around 7 days |

Table IV
TABLE COMPARING COST

In the case of the Payment Channels, the security is granted with the guarantees that L1 provides in addition to the assumptions that the node is always online, and that the network is censorship-resistant[3] within time $t$. Both solutions on this category make use of hash functions and digital signatures as cryptographic primitives.

Regarding the Zero-Knowledge Rollups, they rely on the security inherited from L1 –just like the Payment Channels– but, in this case, they also rely on the validity proofs obtained from the Zero-Knowledge techniques in order to verify that the computation made off-chain is properly done. In particular, for the zkSync, they use PLONK as the zkSNARK that relies on Pairings, Knowledge of Exponent, and a Universal Trusted Setup as cryptographic primitives. The reader can find more information about the security and the cryptographic assumptions that zkSync makes on [24] and [25].

We were not able to find much information about the technical specifications of Loopring but they are using some zkSNARK [26], so we assume that they need Pairings and some kind of trusted setup as cryptographic primitives.

Starknet relies on zkSTARK, which has the constrain requirement of hash functions, presenting the minimum cryptographic requirements among all the Zero-Knowledge solutions studied here. Moreover, given that Starknet *only* relies on Hash functions, it is the only technology –in our list– that is assumed to be Quantum resistant.

Optimistic solutions present a totally different security approach. While they inherit the security from L1, both Arbitrum and Optimism rely on Game Theory when securing their algorithm, and they provide incentives for nodes that detect frauds to uncover them. A single party executing and validating transactions is thus enough to detect a fraud.

Finally, when looking at the actual implementation of the different solutions, we found that all the studied solutions implement a separated network. To be precise, both the Lightning Network and the Raiden Network use a Peer-to-peer (P2P) network, while the rest of solutions have a centralised approach. Nonetheless, zkSync, Starknet and Arbitrum have plans for decentralising their network [27], [28], [29].

[3]This is, there exists a high enough fee threshold such that the transaction will be mined onto L1 in a block within a specific amount of time.

### C. Cost

Table IV shows the comparison in terms of Cost. In this Table, we have considered two different approaches for the transaction cost: **fees** and **time**. We have considered these two approaches since we find that they are the main concerns regarding cost for the scalability solutions.

The fees systems used in those solutions vary in a wide range. The Lightning Network uses a fee system composed by some on-chain fees to pay for the funding and closing transactions; and some off-chain fees nodes may charge to use their channels for multi-hop payments. The Raiden Network fee system is similar to the Lightning Network model.

For the zkRollup approaches and for Optimism the table summarizes the claims the projects make in their official websites [30], [14], [28], [31] (zkSync, Loopring, Starknet and Optimism, respectively); for Arbitrum, we include an estimation based on external resources [32].

Finally, regarding the processing time cost on L2, we found that all the solutions have a near instant processing time, only limited by the hardware that actually performs the operation on the L2 and communication delays. However, when considering the withdrawal time[4], we can see a wide variety of time ranges. In the case of the Lightning Network, the withdrawal time window is up to 1 hour in case of cooperative closing (6 block confirmations in the Bitcoin blockchain), and may vary from 1 hour to several days in case of fraudulent closing. A similar approach applies for the Raiden Network, however, in this case, the time windows is shrinked to up to 3 hours in the worst case [33].

zkRollups are considered to be fastest in terms of withdrawal time, with an average withdrawal time between 6 and

[4]The time required to take the funds from L2 back to L1.

10 minutes and with a maximum time window of 7 hours.

Optimistic Rollups have bad withdrawal times since they rely on fraud proofs that take a considerable amount of time (around 7 days) due to the challenge window.

## V. Conclusions and Future Work

In this paper, we have presented some of the state-of-the-art existing solutions implemented as scalability techniques for blockchain. These solutions can be categorised in two different categories: Payment Channel Networks and Rollups.

We have first introduced the scalability problem in blockchains. Then, we have presented a brief introduction to the different types of scalability techniques considered in this review and we have compared them among different dimensions (Usability, Security, and Cost). Several insights can be drawn from the result of this review.

Firstly, from a usability standpoint, we can see that both rollup approaches excel at providing a wide variety of compatible tokens, as well as Smart Contract support in most cases.

Secondly, from a security point of view, we can see that generally Layer 2 solutions inherit their security model from the underlying Layer 1, and tend to add additional security assumptions. Moreover, most zkRollups require the usage of complex cryptographic primitives (pairings), whereas the other approaches are based only on signatures and hash functions.

Thirdly, considering the costs of using those solutions, we can see that, theoretically, zkRollups are the best ones in terms of both fees and time constraints, in the sense that they are the ones that present less fees when transacting and interacting between Layers, and they provide a reasonable amount of withdrawal time. However, since they are centralized, they may be prone to censorship, less privacy preserving than PCN (where L2 transactions are only seen by the sender and the receiver), and susceptible to classic single point of failure attacks.

As future work for this paper, we plan to extend this article by deploying the actual considered solutions and performing experiments on those in order to benchmark the capabilities of them. To be precise, we want to review and classify the actual capabilities for the general-purpose scripting those solution offer, we want to detail better the Zero-Knowledge techniques zkRollups are using and, finally, perform experiments regarding the fee cost and the processing time these solutions offer.

## Acknowledgements

This work is partially supported by the Spanish Government under grants RTI2018-095094-B-C22 “CONSENT” and PID2021-125962OB-C31 “SECURING”.

## References

[1] NAKAMOTO, Satoshi. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 2008, p. 21260.

[2] KHAN, Kashif Mehboob; ARSHAD, Junaid; KHAN, Muhammad Mubashir. Secure digital voting system based on blockchain technology. *International Journal of Electronic Government Research (IJEGR)*, 2018, vol. 14, no 1, p. 53-62.

[3] SULLIVAN, Clare; BURGER, Eric. Blockchain, digital identity, e-government. *En Business Transformation through Blockchain*. Palgrave Macmillan, Cham, 2019. p. 233-258.

[4] Decentralized applications (DAPPS). *ethereum.org [online]*. [Accessed 23 May 2022]. Available from: https://ethereum.org/en/dapps/

[5] Decentralized finance (DEFI). *ethereum.org [online]*. [Accessed 23 May 2022]. Available from: https://ethereum.org/en/defi/

[6] Non-fungible tokens (NFT). *ethereum.org [online]*. [Accessed 23 May 2022]. Available from: https://ethereum.org/en/nft/

[7] Top blockchain games. *DappRadar [online]*. [Accessed 23 May 2022]. Available from: https://dappradar.com/rankings/category/games

[8] Yadav, Jyoti & Shevkar, Ranjana. (2021). Performance-Based Analysis of Blockchain Scalability Metric. *Tehnički glasnik*. 15. 133-142. 10.31803/tg-20210205103310.

[9] Vecteezym.com, Bitcoin elements thin line and pixel perfect icons [online]. [Accessed 4 June 2022]. Available from: https://www.vecteezy.com/vector-art/681001-bitcoin-elements-thin-line-and-pixel-perfect-icons

[10] Lightning network. Lightning Network [online]. [Accessed 30 May 2022]. Available from: https://lightning.network/

[11] Fast, cheap, Scalable token transfers for Ethereum. Raiden Network [online]. [Accessed 30 May 2022]. Available from: https://raiden.network/

[12] STARKWARE. Redefining scalability. Medium [online]. 1 December 2021. [Accessed 4 June 2022]. Available from: https://medium.com/starkware/redefining-scalability-5aa11ffc5880

[13] ZkSync - rely on math, not validators [online]. [Accessed 4 June 2022]. Available from: https://zksync.io/

[14] Loopring [online]. [Accessed 4 June 2022]. Available from: https://loopring.io/#/

[15] StarkNet [online]. [Accessed 4 June 2022]. Available from: https://starknet.io/

[16] Arbitrum One Portal [online]. [Accessed 4 June 2022]. Available from: http://arbitrum.support/

[17] Optimism [online]. [Accessed 4 June 2022]. Available from: https://www.optimism.io/

[18] LABS, Matter. ZKEVM FAQ. zkSync Documentation [online]. [Accessed 4 June 2022]. Available from: https://docs.zksync.io/zkevm/#general

[19] Hello, cairo. Hello, Cairo - StarkNet documentation [online]. [Accessed 4 June 2022]. Available from: https://starknet.io/docs/hello_cairo/index.html#hello-cairo

[20] Arbos: The Arbitrum Operating System · Offchain Labs Dev Center. · Offchain Labs Dev Center [online]. [Accessed 4 June 2022]. Available from: https://developer.arbitrum.io/docs/arbos

[21] Developing smart contracts on optimism. Optimism Docs [online]. [Accessed 4 June 2022]. Available from: https://community.optimism.io/docs/guides/smart-contract-devs/#

[22] GABIZON, Ariel; WILLIAMSON, Zachary J.; CIOBOTARU, Oana. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *Cryptology ePrint Archive*, 2019.

[23] BEN-SASSON, Eli, et al. Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive*, 2018.

[24] LABS, Matter. Security. zkSync Documentation [online]. [Accessed 4 June 2022]. Available from: https://docs.zksync.io/userdocs/security/#cryptography-used

[25] MATTER-LABS. Zksync/protocol.md at master Matter-Labs/zksync. GitHub [online]. [Accessed 4 June 2022]. Available from: https://github.com/matter-labs/zksync/blob/master/docs/protocol.md#assumptions

[26] Loopring open sources its ZKSNARK Circuit code. [online]. 29 October 2019. [Accessed 4 June 2022]. Available from: https://blogs.loopring.org/loopring-open-sources-its-zksnark-circuit-code/

[27] LABS, Matter. Decentralization. zkSync Documentation [online]. [Accessed 4 June 2022]. Available from: https://docs.zksync.io/userdocs/decentralization/

[28] StarkNet terms of use. StarkNet [online]. 6 December 2021. [Accessed 4 June 2022]. Available from: https://starknet.io/starknet-terms-of-use/

[29] Inside arbitrum · Offchain Labs Dev Center. · Offchain Labs Dev Center [online]. [Accessed 4 June 2022]. Available from: https://developer.offchainlabs.com/docs/inside_arbitrum

[30] LABS, Matter. Overview. zkSync Documentation [online]. [Accessed 4 June 2022]. Available from: https://docs.zksync.io/userdocs/intro/#introduction

[31] Transaction fees – optimism. [online]. [Accessed 4 June 2022]. Available from: https://help.optimism.io/hc/en-us/articles/4411895794715-Transaction-fees

[32] L2fees.info. L2Fees.info [online]. [Accessed 4 June 2022]. Available from: https://l2fees.info/

[33] NETWORK, Raiden. How do I withdraw tokens to my on-chain account? Medium [online]. 5 October 2021. [Accessed 4 June 2022]. Available from: https://medium.com/raiden-network/how-do-i-withdraw-tokens-to-my-on-chain-account-dfb27771829e

# Quantum Random Number Generator based on Vertical-cavity Surface-emitting Lasers

Marcos Valle-Miñón
Universidad de Cantabria-CSIC
Instituto de Física de Cantabria (IFCA)
39005 Spain
marcos.valle@alumnos.unican.es

Ana Quirce
Universidad de Cantabria-CSIC
Instituto de Física de Cantabria (IFCA)
39005 Spain
quirce@ifca.unican.es

Angel Valle
Universidad de Cantabria-CSIC
Instituto de Física de Cantabria (IFCA)
39005 Spain
valle@ifca.unican.es

Jaime Gutiérrez
Universidad de Cantabria
Department of Applied Mathematics
and Computer Science
39005 Spain
jaime.gutierrez@unican.es

***Resumen*—We report a quantum random number generator based on a vertical-cavity surface-emitting laser. In our experiment the current applied to this laser is periodically modulated in a such a way that the linearly polarized modes of the device are randomly excited. We collect a sufficient number of random bits in order to evaluate if a standard statistical test suite for random number generators is passed. We consider Von Neumann's and several linear post-processing methods for reducing the bias found in this type of random number generators. We show that the post-processed random bits pass all tests in the standard statistical test suite provided by the National Institute of Standards and Technology. We compare the results obtained with different post-processing methods. We show that $[n, k, d]$ linear codes with large values of $n$ and $k$ offer improved randomness and throughput.**

***Index Terms*—TRNG, QRNG, Semiconductor lasers, post-processing**

## I. Introduction

Random number generators (RNGs) are extensively used in cryptography and secure communications that require trusted and fast random numbers [1]. There are several ways of obtaining random numbers. A pseudorandom number generator (PRNG) generates a sequence of random numbers by using a deterministic algorithm and a provided seed. True random number generators (TRNGs) produce the random sequences by using physical entropy sources. Quantum random number generators (QRNGs) stand out from TRNGs because their randomness stems from quantum processes. Using QRNGs is a necessary security requirement for quantum key distribution systems [2].

Most of the existing QRNGs are based on quantum optics [1]. Several types of light emitting sources are used in QRNGs, ranging from single-photon sources to multi-photon sources like LEDs or semiconductor lasers. QRNGs based on semiconductor lasers offer high speed random bit generation that can go beyond Gbps rates [3], [4]. Semiconductor lasers can be classified in two types: edge-emitting lasers and vertical-cavity surface-emitting lasers (VCSELs). In QRNGs based on edge-emitting lasers the current that is applied to the device is modulated in such a way that the random bits are obtained from the fluctuations of the phase of the generated optical wave [3], [4].

Very recently QRNGs based on VCSELs have been proposed [5]. In contrast to edge-emitting lasers, VCSELs are characterized by the possibility of emission in two linearly polarized modes. This means that the lasing electrical field can oscillate along two orthogonal directions in the plane perpendicular to the laser beam. In [5] the current applied to the VCSEL is periodically modulated in such a way that the linearly polarized mode that is preferably excited in each period is random. Random excitation of the VCSEL polarizations can be considered as a quantum entropy source because it is triggered by the spontaneous emission that is quantum mechanical in nature and can be considered as quantum noise [6].

Both linearly polarized modes can be excited with similar probabilities by adjusting the amplitude of the current modulation and the sampling time at which the random bit is obtained [5]. In this way we could, in principle, obtain a low bias generator, that is with similar probabilities for "0" and "1" bits. However raw outputs of TRNGs show deviations from the mathematical ideal of statistically independent and uniformly distributed bits [1], [7], [8], [9], This happens also in VCSEL-based QRNGs because unavoidable slight variations of the laser temperature or of the current modulation parameters appear resulting in bias in the random bit generator. To address this problem an additional post-processing step is added to decrease bias in the bit stream and to increase the bit entropy. However this may come at the expense of throughput, for instance the Von Neumann processed bit stream is almost one quarter as long as the raw bit stream [8].

In our previous work [5] the number of generated random bits was not large enough in order to fully pass all tests in a standard statistical test for the validation of random number generators for cryptographic applications. We considered the suite provided by the National Institute of Standards and Technology (NIST) [10]. Our initial results showed that statistical tests requiring small number of bits were passed but much more data would have to be collected in order to pass all the

tests [5].

In this work we generate a large set of random bits in order to obtain significant statistical results. This set has an appreciable bias value so post-processing of these raw bits must be performed. We consider different post-processing methods: a set of linear BCH compression codes and the nonlinear Von Neumann's code. Linear compression codes are interesting because it has been shown that they can achieve much better throughput than Von Neumann compression, while achieving practically good level of security [8]. We show that our post-processed random bits fully pass all the NIST tests. We finally compare the results obtained with different post-processing codes in order to determine the optimum performance in terms of randomness, evaluated by the obtained results in the NIST tests, and throughput.

## II. Experimental Results

We show our experimental set-up in Fig. 1. A complete description of the experiment and of the VCSEL's characteristics can be found in [5].

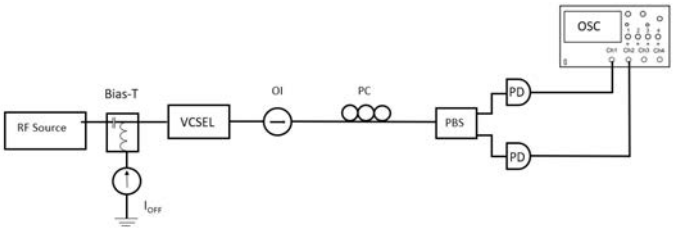



Figura 1. Experimental set-up

The main elements of the set-up are the current modulation part, the VCSEL, and the detection part. Current pulses are applied to the VCSEL by using a pulse pattern generator as RF source, a Bias-T and a direct current source ($I_{\text{off}}$). The two signals corresponding to each linearly polarized mode are separated by using a polarization controller (PC) and a polarization beam splitter (PBS). Both signals are detected by using two high-speed photodetectors (PD) and an oscilloscope (OSC). A squared wave voltage is used to modulate the current applied to the VCSEL with a 100 MHz frequency. This current changes between two values, below and above the laser threshold current, in order to generate the laser pulses. The temperature of the VCSEL is fixed at 25$^o$C during all the measurements.

Fig. 2 shows the temporal waveforms of the signals corresponding to both linearly polarized modes ($x$ and $y$) signals measured at the oscilloscope, $V_x(t)$ and $V_y(t)$, using its high resolution mode. These signals are proportional to the power of the $x-$ and $y-$linearly polarized modes, respectively. The VCSEL switchs-off in all pulses (twelve consecutive pulses are shown) in such a way that there is a random excitation of both linearly polarized modes after the large value of the current is applied. Our data have been obtained with a sampling rate of 0.4 GSa/s so four values of the signals are obtained each modulation period. In this way a well defined maximum in the sum of both signals is always identified, as shown in Fig. 2.

Random bits are obtained by comparing the $x$ and $y$ signals at the times at which the sum of both signals is maximum,

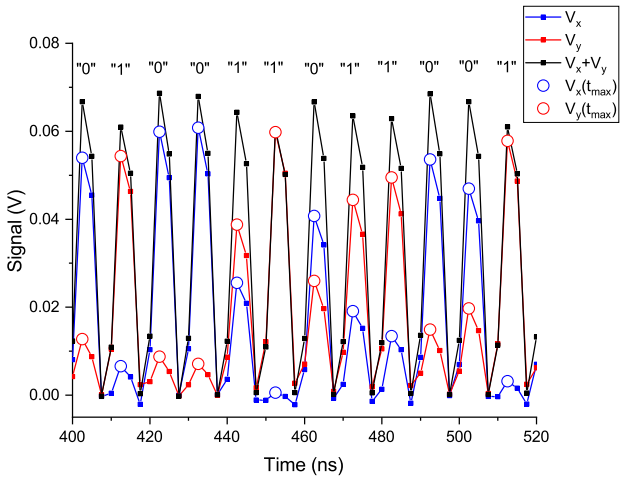



Figura 2. Signals corresponding to the $x-$ and $y-$polarizations.

$t_{max}$. We illustrate this process in Fig. 2. A "0" ("1") bit is assigned when $V_x(t_{max}) > V_y(t_{max})$ ($V_x(t_{max}) \leq V_y(t_{max})$). Then, the probability of obtaining a "0" bit, $p(0)$, is the probability of excitation of the $x-$polarization at $t_{max}$. In our experiment we have collected $2{,}05 \times 10^7$ values of $V_x(t)$ and $V_y(t)$ in each data file recorded at the oscilloscope. Taking into account that we have recorded 788 files we have obtained $4{,}0385 \times 10^9$ raw random bits. If we define the bias, $e$, of the generator by $e = p(0) - 1/2$, the bias obtained for the complete set of raw bits has a large value, $e = 1{,}54 \times 10^{-2}$.

Apart from the $e$ value corresponding to the complete raw bits sequence we can also wonder how the bias evolves as the measurements are performed. We show in Fig. 3(a) the values of the probability of obtaining a "0" bit for each of the sequences extracted from each recorded file (shown in temporal order). Each sequence has $5{,}125 \times 10^6$ bits. Initial values are around 0.5 that is close to the 0.52 value that was obtained in [5] under very similar experimental conditions. $p(0)$ fluctuates around 0.5 until the sequence #75 after which it suddenly increases to 0.61. $p(0)$ begins to decrease until the sequence #170 after which $p(0)$ stabilizes, fluctuating again around 0.5. The previous behaviour repeats three more times. Results in Fig. 3(a) can be explained when considering the way in which we performed the measurements. These took five sessions in such a way that all the equipment was completely turned-off after each session. The beginning of the second, third, fourth and fifth sessions correspond to the numbers of the sequences at which the sudden increases are observed. Taking into account that the recording of each file at the oscilloscope takes around four minutes, the stabilization of $p(0)$ is only achieved after a time around five hours. We think that this long stabilization time is due to a non optimum control of the temperature of the VCSEL in our setup, a situation that could be improved by considering another laser mount. Since the values of $e$ are large, post-processing of the raw bits is required in order to get bits with reduced bias that can pass the statistical tests.

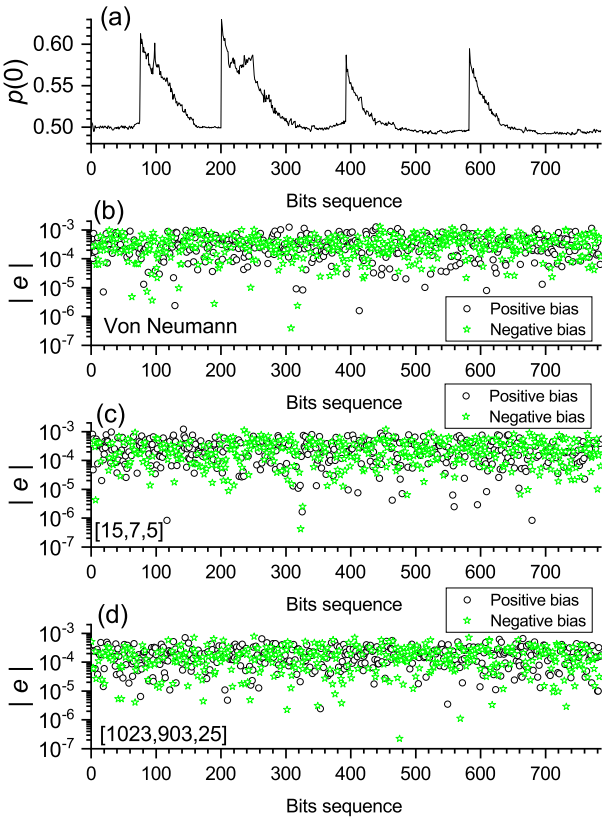



Figura 3. (a) $p(0)$ for the raw bits and (b-d) $\mid e \mid$ when postprocessing .

### II-A. *Post-processing methods and results*

The first post-processing algorithm that we consider is the nonlinear Von Neumann's method described in [8]. This algorithm is very efficient for reducing the bias with a modest throughput, that is the accepted bit rate is $0{,}25 - e^2$ [8]. We obtain around $10^9$ post-processed bits for which $e = -3{,}1 \times 10^{-5}$. Fig. 3(b) shows the absolute value of the bias obtained when post-processing each of the sequences of Fig. 3(a). In Fig. 3(b) we distinguish with different colours the sign of the bias of each sequence, that has around $1{,}27 \times 10^6$ bits. Fig. 3(b) shows that Von Neumann's post-processing significantly decrease the bias values obtained in Fig. 3(a).

We now consider the post-processing techniques using linear corrector codes [9]. They are based on the following result:

*Theorem 1:* [9] Let $G$ be a linear corrector mapping $n$ bits to $k$ bits. Then the bias of any non zero linear combination of the output bits is less or equal than $2^{d-1}e^d$, where $d$ is the minimal distance of the linear code constructed by the generator matrix $G$.

As suggested in [8] we use the efficient $[n, k, d]$-BCH codes defined over the finite field $GF(2)$ and where $n + 1$ is a power of 2. For the raw input bits $(x_{n-1}, \ldots, x_0)$, the output $(y_{k-1}, \ldots, y_0)$ is obtained as:

$$\begin{pmatrix} g_{n-k} & \ldots\ldots\ldots & g_0 & 0\ldots\ldots 0 \\ 0 & g_{n-k} & \ldots\ldots\ldots g_0 & 0\ldots\ldots 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0\ldots\ldots & 0 & g_{n-k} & \ldots\ldots g_0 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ x_{n-2} \\ \vdots \\ x_0 \end{pmatrix} = \begin{pmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_0 \end{pmatrix} \quad (1)$$

and $g(x) = g_{n-k}x^k + \cdots + g_1 x + g_0$ is the cyclic generator polynomial of the $[n, k, d]$-BCH code.

For instance the BCH code with parameters $[15, 7, 5]$ has as generator cyclic polynomial $x^8 + x^7 + x^6 + x^4 + 1$. Another example is the BCH code with parameters $[1023, 1003, 5]$ that has generator cyclic polynomial $x^{20} + x^{15} + x^{13} + x^{12} + x^{11} + x^9 + x^7 + x^6 + x^3 + x^2 + 1$ (see [11] for several properties of those practical linear codes).

Using appropriate values of $k$ and $n$, the throughput $(k/n)$ can be much larger than that obtained with Von Neumann's method, around $0{,}25$, while maintaining a very efficient bias reduction $2^{d-1}e^d$, as stated in Theorem 1. The choice of a $k$ value slightly smaller than $n$ results in reduced bias with high throughput. For instance applying the [1023,1003,5] method to the complete set of raw bits results in $e = 8{,}4 \times 10^{-6}$ with $k/n \sim 0{,}980$. Fig. 3(c) and Fig. 3(d) show $\mid e \mid$ when post-processing each of the sequences of Fig. 3(a) with [15,7,5] and [1023,903,25], respectively. Both figures show that these linear corrector codes are also very effective in reducing the bias.

### II-B. *Results of statistical tests*

We now discuss the results obtained with the NIST statistical tests applied to the bits resulting from the two different types of post-processing techniques described in the previous section. Each test is performed using 1000 sequences of $10^6$ bits each with a statistical significance level, $\alpha$ = 0.01. We show in Fig. 4 the P-value$_{\mathrm{T}}$ and the proportion of sequences passing the tests using as input the bits obtained after Von Neumann post-processing method.

P-value$_{\mathrm{T}}$ gives an idea of the uniformity of the distribution of the P-values [10]. For tests that return multiple P-value$_{\mathrm{T}}$

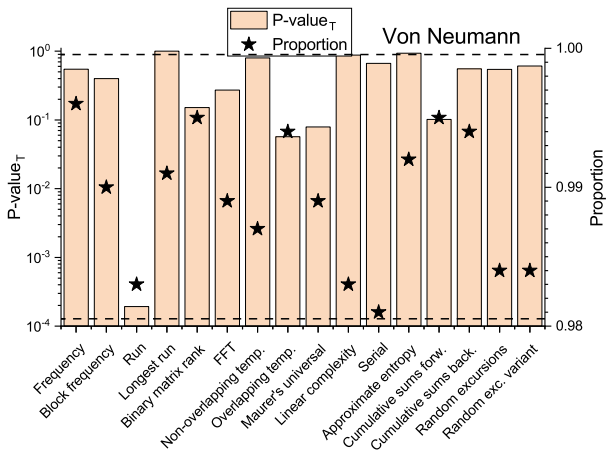



Figura 4. NIST test results when using Von Neumann's post-processing.

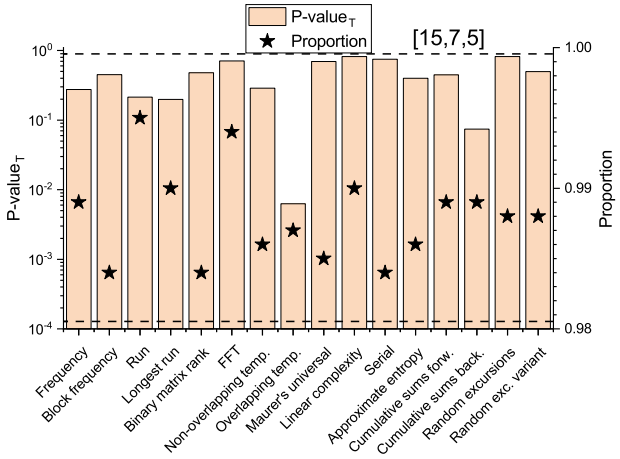



Figura 5. NIST test results when using [15,7,5] post-processing.

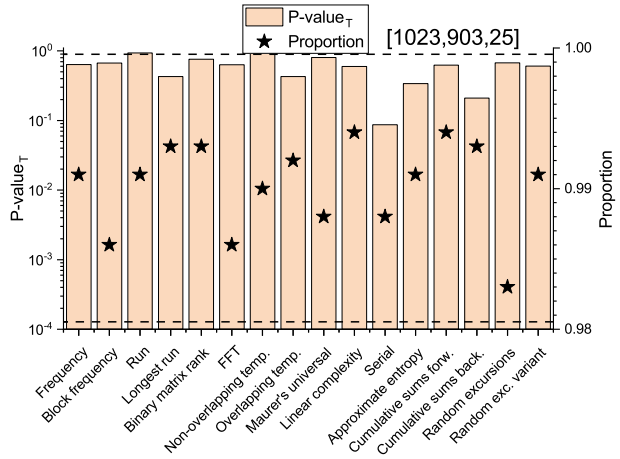



Figura 6. NIST test results when using [1023,903,25] post-processing.

and proportions, the more representative case, that is the one having a P-value$_{\rm T}$ closest to the median of the set of P-value$_{\rm T}$, has been plotted. Two criteria are used in these tests for "success": i) the P-value$_{\rm T}$ must be larger than $10^{-4}$, and ii) the proportions must be in the (0.9805607,0.9994393) confidence interval [10]. These values have been included in Fig. 4 using horizontal dashed lines. Results shown in Fig. 4 confirm that the bits sequences obtained from Von Neumann's post-processing are sufficiently random for passing the statistical tests of NIST.

We now consider the results of the NIST tests applied to the bits obtained by using several $[n, k, d]$ linear post-processing codes. Fig. 5, Fig. 6 and Fig. 7 show the results corresponding to post-processings with BCH codes with parameters [15,7,5], [1023,903,25] and [1023,1003,5], respectively. Again, results shown in these figures confirm that the post-processed bit strings pass all NIST tests.

The performance of the different post-processing methods are now compared by using the results included in Table I. This table shows the output bias and the accepted bit rate for the post-processed data used in Fig. 4 to Fig. 7. Some other linear codes with intermediate values of $n$, like [255,107,45] and [511,484,7], are also included in the table. In order to quantify in a summarised way the results of NIST tests we have also calculated <P-val$_{\rm T}$ > and <Pr>, that are defined as the averages over the 16 tests of P-value$_{\rm T}$ and proportions,

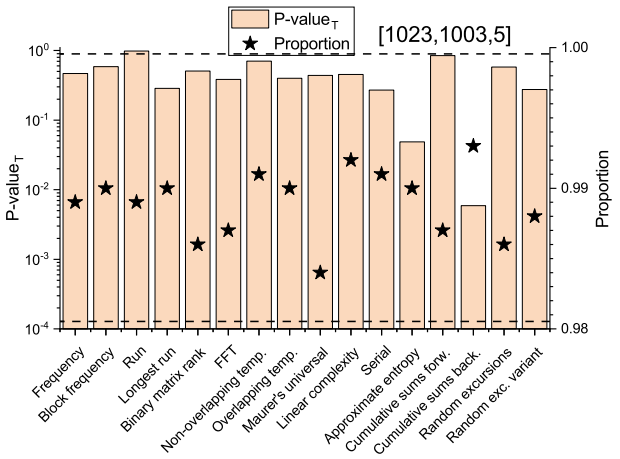



Figura 7. NIST test results when using [1023,1003,5] post-processing.

respectively. The dispersion of the proportions around <Pr> is also quantified by including the standard deviation of the proportions over the tests, $\sigma_{\rm Pr}$. All these quantities have been also included in Table I.

All the considered post-processing techniques are very efficient to reduce the bias, as shown in Table I. The minimum value of $\mid e \mid$ has been obtained using the BCH code with parameters [15,7,5]. The accepted bit rate changes from the value close to 0.25 corresponding to Von Neumann's method to the value 0.980 obtained for [1023,1003,5]. One of the methods with large values of $n$ and $k$, [1023,903,25], gives the best uniformity of the distributions of P-values because the obtained <P-val$_{\rm T}$ > is the largest in table I. This table also shows that <Pr> is close to 0.99, as expected for a good RNG using $\alpha = 0{,}01$. The smallest values of the standard deviation of the proportions are obtained with the BCH codes with parameters [511,484,7] and [1023,1003,5]. Our results show that BCH codes with large values of $n$ and $k$ parameters are the best choice to obtain large values of bit rate and <P-val$_{\rm T}$ > with a small $\sigma_{\rm Pr}$.

Efficient hardware implementation of linear corrector functions based on BCH codes can be performed by using XORing and shift registers [8]. A field-programmable gate array (FPGA) implementation utilizes more resources as the values of $n$ and $k$ increase. However, our results show that there are methods, like [15,7,5], using less resources than those with large values of $n$ and $k$, that permit to obtain similar <P-val$_{\rm T}$ > to that found with Von Neumann's method, but with

Tabla I
POST-PROCESSING AND NIST TEST RESULTS FOR DIFFERENT POST-PROCESSING METHODS

| Post-proc. | Bias | Rate | <P-val$_{\rm T}$ > | <Pr> | $\sigma_{\rm Pr}$ |
|---|---|---|---|---|---|
| Von Neumann | -3.1 $10^{-5}$ | 0.248 | 0.474 | 0.989 | 0.005 |
| [15,7,5] | -1.4 $10^{-8}$ | 0.467 | 0.445 | 0.988 | 0.003 |
| [255,107,45] | -2.2 $10^{-5}$ | 0.420 | 0.420 | 0.989 | 0.004 |
| [511,484,7] | 8.2 $10^{-6}$ | 0.947 | 0.410 | 0.991 | 0.002 |
| [1023,903,25] | 9.0 $10^{-6}$ | 0.883 | 0.587 | 0.990 | 0.003 |
| [1023,1003,5] | 8.4 $10^{-6}$ | 0.980 | 0.451 | 0.989 | 0.002 |

a much larger bit rate and smaller $\sigma_{\mathrm{Pr}}$.

### *II-C. Discussion of the experimental results*

The raw bit rate that we have considered in our experiment is not very fast, 100 Mbps. This rate can be significantly increased with our VCSEL because its current modulation bandwidth is close to 4 GHz. In fact, operation for extracting random bits at 200 Mbps was already shown in [5]. However the maximum rate in our experimental setup is limited by the maximum current modulation frequency (600 MHz) of the laser mount. We note that the random bit rate obtained using our technique can increase significantly because the VCSEL's modulation bandwitdhs can go beyond 35 GHz. Anyway we remark that the main intention of our work has been to collect a sufficient number of data to fully pass the NIST statistical tests. Further work to confirm the randomness of our data using another test batteries (Dieharder, TestU01 and AIS 31) will be performed in the future.

Our results indicate that several simple post-processings of large bias raw bits are enough to pass the NIST tests. This also means that in practice our proposed QRNG would not be affected by small variations of the experimental conditions, provided that an appropriate post-processing is considered.

## III. Conclusions

In conclusion, we have reported a quantum random number generator based on the random excitation of the linearly polarized modes of a VCSEL subject to current modulation. We have collected a large set of raw bits that have been post-processed with different techniques, ranging from the Von Neumann's compression to the family of linear BCH codes. These offer a large pool of codes to choose from in which it is possible to trade-off different aspects of a RNG performance, like output bias and throughput. We have shown that the post-processed random bits fully pass the batteries of the NIST SP800-22 statistical tests. We have compared the results obtained with different post-processing methods. We have shown that $[n, k, d]$ BCH codes with large values of $n$ and $k$ offer improved throughput and randomness.

## Acknowledgements

Ministerio de Ciencia e Innovación. RTI2018-094118-B-C22 MCIN/AEI/10.13039/501100011033/FEDER "Una manera de hacer Europa". J. G. is partially supported by grant PID2019-110633GB-I00 funded by MCIN/AEI/10.13039/501100011033.

## Referencias

[1] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators", *Reviews of Modern Physics* **89**, 015004, 2017.

[2] T. K. Paraiso, R. I. Woodward, D. G. Marangon, V. Lovic, Z. Yuan and A. J. Shields, "Advanced laser technology for quantum communications (tutorial review)", *Advanced Quantum Technologies* **4**, 10, 2100062, 2021.

[3] C. Abellán, W. Amaya, M. Jofre, M. Curty, A. Acín, J. Capmany, V. Pruneri and M. Mitchell, "Ultra-fast quantum randomness generation by accelerated phase diffusion in a pulsed laser diode", *Optics Express* **22**, 2, pp. 1645–1654, 2014.

[4] Z. Yuan, M. Lucamarini, J. Dynes, B. Fröhlich, A. Plews, and A. Shields, "Robust random number generation using steady-state emission of gain-switched laser diodes", *Applied Physics Letters*, **104**, 261112, 2014.

[5] A. Quirce and A. Valle, "Random polarization switching in gain-switched VCSELs for quantum random number generation", *Optics Express,* **30**, 7, pp. 10513–10527, 2022.

[6] R. Loudon, "The quantum theory of light", (OUP Oxford), 2000.

[7] M. Dichtl, "Bad and good ways of post-processing biased physical random numbers", in *International Workshop on Fast Software Encryption,* (Springer), pp. 137–152, 2007.

[8] S.-H. Kwok, Y.-L. Ee, G. Chew, K. Zheng, K. Khoo, and C.-H. Tan, "A comparison of post-processing techniques for biased random number generators", in *IFIP International Workshop on Information Security Theory and Practices,* (Springer), pp. 175–190, 2011.

[9] P. Lacharme, "Post-processing functions for a biased physical random number generator", in *International Workshop on Fast Software Encryption,* (Springer), pp. 334–342, 2008.

[10] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert *et al.*, "NIST special publication 800-22: a statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications", NIST Special Publication, **800**, 22, 2010.

[11] F. J. MacWilliams and N. J. A. Sloane, "The theory of error correcting codes", vol. 16 (Elsevier), 1977.

# Lista de Autores

*Octubre, 2022*

La Reunión Española sobre Criptología y Seguridad de la Información (RECSI) es el congreso científico referente español en el tema de la Seguridad en las Tecnologías de la Información y Comunicación, donde se dan cita de forma bienal los principales investigadores españoles en el tema, así como invitados extranjeros de reconocido prestigio. En estos encuentros se muestran los avances de los grupos de investigación que presentan comunicaciones y fomentan la participación de los jóvenes investigadores.

Este libro recoge los resúmenes de las conferencias plenarias junto con los trabajos presentados en la XVII RECSI celebrada en Santander del 19 al 21 de Octubre de 2022 organizada por el grupo de investigación AMAC (Algorithmic Mathematics And Cryptography) de la Universidad de Cantabria.

The Spanish Meeting on Cryptology and Information Security (RECSI) is the Spanish reference scientific congress on the subject of Information Technology Security, where the main Spanish researchers on the subject, as well as foreigners of recognized prestige, meet every two years. These meetings show the progress of the research groups that present communications and encourage the participation of young researchers.

This volume collects the summaries of the plenary conferences together with the papers presented at the XVII RECSI held in Santander, 19-21 October in 2022 organized by AMAC Research group (Algorithmic Mathematics And Cryptography) from Universidad de Cantabria.

RECSI

2022

## AUTORES

Adrià Torralba-Agell, Adrian Tobar Nicolau, Aitor Urbieta, Alba Cruz Torres, Aleksander Styrmoe, Àlex Miranda-Pascual, Alexandre Viejo, Amador Jaume Barceló, Amparo Fúster-Sabater, Ana Isabel Gómez, Ana Quirce, Andrés Caro Lindo, Andrés Marín, Andrew Tirkel, Angel Longueira-Romero, Angel Valle, Anis Fellah-Touta, Antonio Muñoz, Balint Zoltan Teglasy, Branislav Petrovic, Candelaria Hernández-Goya, Cándido Caballero-Gil, Carles Anglés-Tafalla, Carles Ventura, Carlos Andres Lara-Nino, Carlos Rosa-Remedios, Carmen Sánchez Ávila, Consuelo Martínez López, Cristina Pérez-Solà, Cristóbal Arellano, Cristòfol Daudén-Esmel, Dane Flannery, Daniel Collins, David Arroyo, David Balbás, David Megías, Diego José Abengózar Vilar, Domingo Gomez, Fabian Ricardo Molina Gomez, Fernando Román-García, Francesc Sebé , Francisco Martín-Fernández, Gerardo Fernández, Guillermo Azuara, Helena Rifà-Pous, Iñaki Garitano, Jaime Gutiérrez, Jaume Ramis Bibiloni, Javier Alonso Díaz, Javier Correa-Marichal, Javier Parra-Arnau, Jeimy Cano, Jesús A. Manjón, Jesús Díaz, Jezabel Molina Gil, Joan Amengual Mesquida, John Livieratos, Jordi Castellà-Roca, Jordi Forné, Jordi Serra-Ruiz, Jorge Wallace, José Andrés Armario, José Antonio Onieva, Jose Carlos Sancho Nuñez, José Daniel Escánez-Expósito, José Ignacio Bengoechea-Isasa, José Ignacio Sánchez García, Jose L. Muñoz-Tapia, Jose Luis Flores, José Luis Salazar, Jose Ruiz-Mas, Josep Domingo-Ferrer, Josep-Lluís Ferrer-Gomila, Juan Hernández-Serrano, Juan Manuel Martínez, Julian Fernandez-Navajas, Julián Salas, Lilian Bossuet, Llorenç Huguet-Rotger, Luis Hernández Encinas, M. Francisca Hinarejos, Macià Mut-Puigserver, Magdalena Payeras-Capellà, Manuel Ruiz, Marcel Fernández Muñoz, Marcos Valle-Miñón, Margarita Robles-Carrillo, Miguel Ángel González de la Torre, Miquel Soriano, Miquel-Àngel Cabot-Nadal, Mohammad Hossein Homaei, Moti Yung, N. Mohanapriya, Néstor Álvarez-Díaz, Oriol Alàs, Oscar Esparza, Óscar Mogollón Gutiérrez, Pablo Pérez, Patricia Guerra-Balboa, Pedro García-Teodoro, Phillip Gajland, Pino Caballero-Gil, Rafael Genés-Durán, Rames Sarwat-Shaker, Raúl M. Falcón, Rodrigo Román, Ronan Egan, Rosa Iglesias, Rosa Pericas-Gornals, Rubén Ríos, Sara D. Cardell, Sebastià Martín Molleví, Sergi Simón, Sergio Chica, Slobodan Petrovic, Sokratis Katsikas, Sonia Díaz-Santos, Tanya Koohpayeharaghi, Thorsten Strufe, Urko Zurutuza, V. Aparna, Verónica Requena, Victor Garcia-Font, Xabier Saez de Camara

UC UNIVERSIDAD DE CANTABRIA

ISBN 978-84-19024-14-5 0 €

www.editorial.unican.es

THEMA: GPJ, PDM, URY, UNKD