



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Seguridad de las tarjetas NFC

NFC card security assessment

Javier Correa Marichal

La Laguna, 13 de junio de 2022

Dña. **Pino Caballero Gil**, con N.I.F. 45.534.310-Z, Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Carlos Rosa Remedios**, con N.I.F. 43.786.084-H, profesor Contratado Laboral Interino adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

C E R T I F I C A N

Que la presente memoria titulada:

"Seguridad de las tarjetas NFC"

ha sido realizada bajo su dirección por D. **Javier Correa Marichal**, con N.I.F. 79.088.385-X.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de junio de 2022.

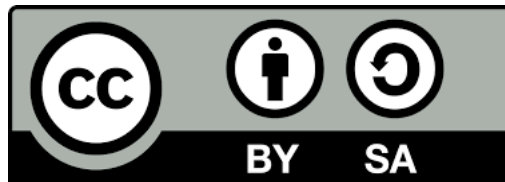
Agradecimientos

A Pino y a Carlos, por ser tutores ejemplares y por la pasión con la que desempeñan su labor; y a Rames, cuya ayuda desinteresada fue un cimiento fundamental para el desarrollo de este trabajo.

A mi familia, por darme la oportunidad, recursos y herramientas para convertirme en la persona que soy hoy en día.

Y, especialmente, a mis amigos, por acompañarme durante cada paso del camino, apoyándome incondicionalmente durante los buenos y malos momentos de mi vida.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional.

Resumen

En este Trabajo de Fin de Grado se ha realizado un estudio de la seguridad de algunas tarjetas de lectura sin contacto que incorporan una interfaz NFC. Esta tecnología es utilizada frecuentemente dentro de las organizaciones como mecanismo de autenticación y control, delimitando el acceso de personal a zonas de alta sensibilidad. También tarjetas que incorporan esta tecnología son, por ejemplo, algunos de los documentos gubernamentales para la identificación de ciudadanos, como por ejemplo la última versión del Documento Nacional de Identidad electrónico o DNIE.

Una brecha de seguridad en cualquiera de estas credenciales podría conllevar un gran impacto para sus usuarios, poniendo en peligro los recursos e información sensible protegidos por los mismos. En el estudio realizado se documenta el funcionamiento de los mecanismos de seguridad implementados en varias etiquetas NFC para su protección contra vectores de ataques comunes. Además, se exploran diversos escenarios prácticos reales donde esos mecanismos son puestos a prueba a través de la realización de auditorías de seguridad.

Palabras clave: NFC, clonación de tarjetas, API hooking, eMRTD

Abstract

In this Final Degree Project, a study of the security status of contactless cards that incorporate an NFC interface has been carried out. This technology is frequently used within organisations as an authentication and control mechanism, limiting the access of personnel to highly sensitive areas. Also cards that incorporate this technology are, for example, some government documents for citizen identification, such as the latest version of the Spanish electronic National Identity Card.

A security breach in any of these credentials could have a major impact on the final users, endangering the resources and sensitive information protected by them. This study documents the operation of the security mechanisms implemented in several NFC tags to protect them against common attack vectors. In addition, various real practical scenarios are explored where these mechanisms are put to the test through security audits.

Keywords: NFC, card cloning, API hooking, eMRTD

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Fases del desarrollo	2
1.4. Estructura de la memoria	2
2. Tarjetas de acceso	4
2.1. Antecedentes	4
2.2. MIFARE Classic	4
2.2.1. Estructura lógica	5
2.2.2. Debilidades criptográficas	6
2.2.3. Tarjetas MIFARE mágicas	7
2.3. Ataques sobre tarjetas de acceso	8
2.3.1. Clonación del bloque del fabricante	8
2.3.2. Clonación de sectores protegidos	11
3. Tarjetas de identidad	15
3.1. Antecedentes	15
3.2. Estandarización	16
3.3. Interfaz de uso	16
3.4. Estructura lógica de datos	17
3.5. Mecanismos de seguridad	18
3.5.1. Basic / Supplemental Access Control	18
3.5.2. Autenticación Pasiva	20
3.5.3. Extended Access Control	20
3.6. Vulnerabilidades conocidas	21
3.7. Investigación práctica	22
3.7.1. Análisis de paquetes APDU en vivo	23
3.7.2. Ataque de fuerza bruta contra PACE	28
3.7.3. Análisis del generador de números pseudoaleatorios	30
3.7.4. Ataque de retransmisión	34
4. Conclusiones y líneas futuras	36
5. Conclusions and future works	38
6. Presupuesto	39
6.1. Costes de personal	39
6.2. Costes de componentes	40

6.3. Coste total	40
A. Artículos enviados a conferencias	41
A.1. 20th International Conference on Security and Management SAM (aceptado)	41
A.2. XVII Reunión Española sobre Criptología y Seguridad de la Información RECSI (enviado)	41

Índice de Figuras

2.1. Esquema de la memoria de una tarjeta MIFARE Classic 1K	5
2.2. Cifrado Crypto1	6
2.3. Pulsera de identificación y Proxmark3 Easy	9
2.4. Información básica de la etiqueta NFC incrustada	10
2.5. Lectura y escritura de la información en una tarjeta mágica	11
2.6. Claves de los sectores de la tarjeta universitaria	12
2.7. Ejecución del <i>hardnested attack</i>	13
2.8. Contenido de la tarjeta universitaria	13
3.1. Ejemplo de estructura de ficheros estandarizada en LDS 2.0 [30]	17
3.2. Vista simplificada de la jerarquía de la PKI del EAC	21
3.3. Interfaz de la aplicación de prueba proporcionada en el SDK	24
3.4. Demostración del proceso de <i>API Hooking</i>	24
3.5. Interfaz gráfica de la herramienta <i>jadx</i>	25
3.6. Visualización de los paquetes intercambiados con el DNIE	28
3.7. Números pseudoaleatorios extraídos del protocolo PACE	33
3.8. Representación gráfica de un ataque de <i>relay</i>	34

Índice de Tablas

3.1. Grupos de datos almacenados en el DNIE 4.0	18
3.2. Algoritmos de seguridad implantados en documentos eMRTD	19
6.1. Presupuesto estimado de personal	39
6.2. Presupuesto estimado de componentes	40
6.3. Presupuesto estimado del trabajo	40

Capítulo 1

Introducción

1.1. Motivación

El uso de la tecnología NFC ha experimentado un notorio auge durante la última década. Hoy en día es cada vez más utilizada en cerraduras inteligentes, tarjetas monedero para el transporte público, tarjetas de crédito como mecanismo de pago, etc. Su extendido uso en una gran variedad de aplicaciones ha hecho de este uno de los estándares más utilizados para la interconexión de dispositivos electrónicos. En consecuencia, se ha convertido también en uno de los más estudiados, puesto que su creciente popularidad e importancia como mecanismo de autorización lo ha convertido en un objetivo ideal para todo tipo de atacantes que deseen acceder a un recurso protegido con dicha tecnología.

En concreto, el uso del protocolo NFC en pasaportes y tarjetas identificativas es cada vez más habitual. En España, el actual Documento Nacional de Identidad electrónico o DNIE incluye un circuito integrado que permite acreditar mediante certificado digital la identidad personal del titular, funcionalidad que es utilizada comúnmente para la firma electrónica de documentos y como mecanismo de autenticación en transacciones telemáticas. A partir de la versión DNIE 3.0, este certificado es accesible de forma inalámbrica a través de NFC, facilitando su lectura mediante *smartphones*, para una mayor comodidad de uso del DNIE.

Asimismo, se ha popularizado el uso de NFC en los sistemas de control de acceso empleados en organizaciones con el fin de identificar a sus usuarios y permitir la entrada a zonas restringidas. La autenticación en estos casos es realizada a través de tarjetas que almacenan la información identificativa del usuario portador, o a través de aplicaciones instaladas en su teléfono móvil. Estos mecanismos de control de acceso son muchas veces preferidos ante otras tecnologías del mercado debido a su bajo coste de producción, durabilidad, versatilidad y sencillez de uso.

1.2. Objetivos

El objetivo de este Trabajo de Fin de Grado es proporcionar un estudio de las tarjetas de control de acceso y de identidad que incorporan una interfaz NFC. Así, esta memoria documenta y recoge información de relevancia para la comprensión del funcionamiento de estos dispositivos, incluyendo la realización de un análisis de la seguridad que ofrecen.

En el caso de las tarjetas de acceso, concretamente se ha investigado el uso de etiquetas que incorporan un chip propietario MIFARE Classic. En este ámbito práctico, se ha llevado a cabo una auditoría de seguridad de sistemas encontrados en entornos reales que utilizan esta tecnología para la operación de sus mecanismos de control de acceso, detallando los distintos recursos y ataques empleados en el proceso.

En la línea de investigación práctica sobre tarjetas de identidad, se ha explorado la implementación de las funcionalidades y mecanismos de seguridad asociados con la última revisión del DNIE. Puesto que se trata de un documento evaluado y certificado por organismos especializados, se ha buscado confirmar la seguridad de este documento sin pretender vulnerarlo, sino más bien comprobar su resiliencia contra los vectores de ataque más comunes.

1.3. Fases del desarrollo

El desarrollo de este trabajo ha sido llevado a cabo en tres fases claramente diferenciadas, con el fin de cumplir con los hitos planteados para este estudio.

En un primer lugar, se ha realizado un proceso exhaustivo de investigación de las tecnologías involucradas en el funcionamiento del estándar NFC y del DNIE. Si bien la información sobre NFC es abundante y existen numerosos recursos bien establecidos para comprender su operación y debilidades, la información sobre el DNIE es relativamente escasa. Se han realizado pocos estudios científicos en materia del funcionamiento y seguridad del DNIE, por lo que ha sido necesario reunir y extrapolar resultados de investigaciones relacionadas con documentos de identidad de otros países europeos. Además, se ha recurrido directamente a la documentación oficial donde se definen los estándares por los que se rigen esos documentos.

Una vez reunida la información necesaria para la comprensión de estas tecnologías, se ha procedido a plantear y desarrollar diversas líneas de investigación para explorar la seguridad de las tarjetas de lectura sin contacto. Este estudio práctico se ha planteado, nuevamente, en dos partes, explorando por separado la realización de ataques conocidos en tarjetas de acceso utilizadas en sistemas en producción y la implementación de los mecanismos de protección incluidos en el DNIE. Concretamente el análisis efectuado sobre el DNIE, una vez concluido, ha sido presentado al organismo responsable del DNIE, la Fábrica Nacional de Moneda y Timbre - Real Casa de la Moneda, como parte de un proceso de revelación responsable de los resultados obtenidos.

Finalmente, se ha procedido a la documentación de la información recopilada y de los hallazgos derivados de los estudios planteados en este trabajo, que se recoge en la presente memoria, así como en dos artículos enviados a congresos científicos. Dichos artículos han sido derivados de este Trabajo de Fin de Grado, pero además desarrollan algunas líneas de investigación que se escapan de los objetivos aquí definidos.

1.4. Estructura de la memoria

El resto de la memoria se estructura como sigue:

- **Capítulo 2. Tarjetas de acceso:** se describe la relevancia de este mecanismo de identificación y se estudian las debilidades de tarjetas MIFARE Classic empleadas en entornos reales.
- **Capítulo 3. Tarjetas de identidad:** se detallan el funcionamiento y mecanismos de seguridad integrados en el DNIE, así como las líneas de investigación desarrolladas para su estudio.
- **Capítulos 4 y 5. Conclusiones y líneas futuras:** se presentan las conclusiones y posibles trabajos futuros a desarrollar a partir de los resultados obtenidos.
- **Capítulo 6. Presupuesto:** se presenta un desglose del presupuesto estimado para la realización de este trabajo.
- **Apéndice A. Artículos enviados a conferencias:** se indica información de relevancia sobre dos artículos derivados de este trabajo, enviados a congresos científicos, uno internacional y otro nacional. El primero ha sido ya aceptado, mientras que el segundo se encuentra en fase de evaluación.

Capítulo 2

Tarjetas de acceso

2.1. Antecedentes

Los sistemas de control de acceso permiten la entrada a zonas de alta sensibilidad utilizando una tarjeta de identificación especial. Las tarjetas de control de acceso funcionan con lectores situados en las entradas de los edificios o en las zonas de alta seguridad de instalaciones. La tarjeta se pasa o se agita delante del lector, que procesa y verifica la información de la misma antes de permitir el acceso. Este proceso es más seguro que la entrada con llaves tradicionales ya que, si una tarjeta de control de acceso se pierde o cae en manos equivocadas, simplemente se puede revocar [1].

Las tarjetas de acceso pueden incorporar distintas tecnologías para la codificación de la información. Por ejemplo, las tarjetas de banda magnética almacenan los datos en una *magstripe*, que el lector decodifica al pasarla. En comparación, las tarjetas de proximidad y *smartcards* sin contacto que utilizan tecnología RFID ofrecen niveles de seguridad más elevados, puesto que pueden incorporar mecanismos criptográficos para la protección de la información almacenada y retransmitida.

Otra de sus ventajas es que, puesto que no es necesario que la tarjeta se inserte en el lector, no se produce ningún desgaste por rozamiento, garantizando una gran durabilidad de los elementos. Las tarjetas de proximidad tampoco pierden su codificación, como podría pasar con las de banda magnética, por lo que su vida útil es indefinida. Además, pueden ser reconfiguradas y reutilizadas para nuevos usuarios siempre que sea necesario [2].

2.2. MIFARE Classic

Las tarjetas MIFARE Classic fueron pioneras en el espacio de las *smartcards*, siendo una de las primeras etiquetas NFC orientadas al ámbito empresarial lanzadas al mercado. Diseñadas en los años 90 por NXP Semiconductors, MIFARE es una familia de productos que comprende cuatro tipos diferentes de tarjetas: Ultralight, Classic, DESFire y SmartMX. De acuerdo con los datos proporcionados por el fabricante, más de 10 billones de estas etiquetas han sido vendidas desde su concepción [3], cubriendo una amplia mayoría del mercado de tarjetas de acceso sin contacto. Aún hoy en día, las tarjetas MIFARE Classic son comúnmente empleadas en sistemas de control de acceso dentro de pequeñas y grandes organizaciones o como billete de transporte válido en estaciones de tren, metro y

autobuses, siendo considerado el modelo de tarjeta RFID criptográfica más utilizado del mundo.

2.2.1. Estructura lógica

La tarjeta MIFARE Classic está compuesta, fundamentalmente, por un chip de memoria EEPROM con los mecanismos necesarios para establecer una comunicación segura. Estas etiquetas se ofertan en dos tamaños de memoria distintos, de 1KB y 4KB. Este espacio de almacenamiento se divide en secciones lógicas denominadas “sectores” y “bloques”, donde es posible realizar operaciones básicas como la lectura y escritura de información o el incremento y decremento de valores almacenados.

Cada sector está a su vez dividido en 4 bloques de 16 bytes cada uno (ver Figura 2.1). El último bloque de cada sector es denominado el “tráiler del sector”. Se trata de un bloque especial donde se almacenan las dos claves secretas asociadas al sector, así como las condiciones de acceso al mismo. Esta división implica que cada uno de los sectores almacenados en la tarjeta puede ser configurado con privilegios de escritura y/o escritura independientes, gestionados con 2 claves de autenticación de 6 bytes diferentes. Para ejecutar una acción en un bloque específico, el lector ha de primero autenticarse utilizando las claves del sector que contiene el bloque objetivo. Las condiciones de acceso determinan, para cada una de las claves empleadas, qué operaciones se le permite realizar al usuario.

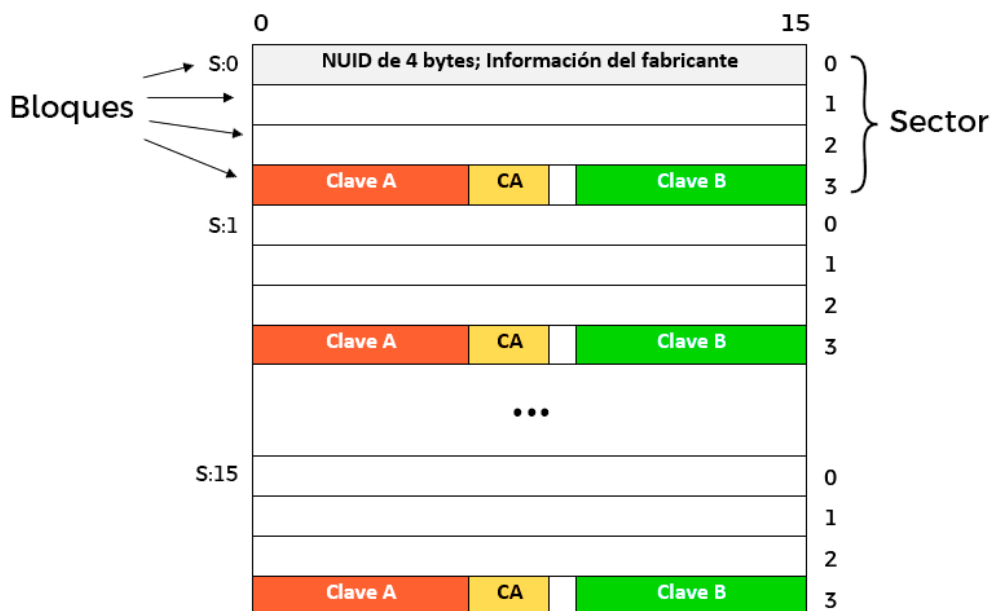


Figura 2.1: Esquema de la memoria de una tarjeta MIFARE Classic 1K

El primer bloque contenido en el almacenamiento (esto es, el bloque 0 del sector 0) es especial, puesto que en él se almacenan datos establecidos por el fabricante durante el proceso de manufacturación. Los primeros 4 bytes conforman el NUID (*Non-Unique Identifier*) utilizado por el protocolo de anticollisión especificado en el estándar ISO/IEC 14443-A; seguido por un byte denominado *Bit Count Check* o BCC, utilizado a modo de

suma de comprobación durante la transmisión del NUID. El resto de bytes almacenan información del fabricante. Este bloque tiene la característica de ser de solo lectura, por lo que no puede ser modificado una vez fabricada la tarjeta.

2.2.2. Debilidades criptográficas

Las tarjetas MIFARE Classic son compatibles con las partes 1 a 3 de la norma ISO/IEC 14443-A, implementando las características físicas, la interfaz de radiofrecuencia y el protocolo de anticollisión allí definidos. Sin embargo, la parte 4 de este estándar, donde se describe el protocolo de transmisión a utilizar, se ve reemplazada por su propia capa de comunicación segura. En ella se utiliza un algoritmo criptográfico propietario denominado Crypto1, para autenticar tanto el lector como la tarjeta MIFARE y proveer un canal seguro para garantizar la confidencialidad de los datos transmitidos. Crypto1 es un cifrado en flujo que consta de un registro de desplazamiento de retroalimentación lineal (o LFSR por sus siglas en inglés, *Linear Feedback Shift Register*) de longitud 48, una función de filtrado no lineal en dos fases utilizada para generar la secuencia cifrante, y otro LFSR que se utiliza durante la fase de autenticación como generador de números pseudoaleatorios para el protocolo desafío-respuesta (ver Figura 2.2).

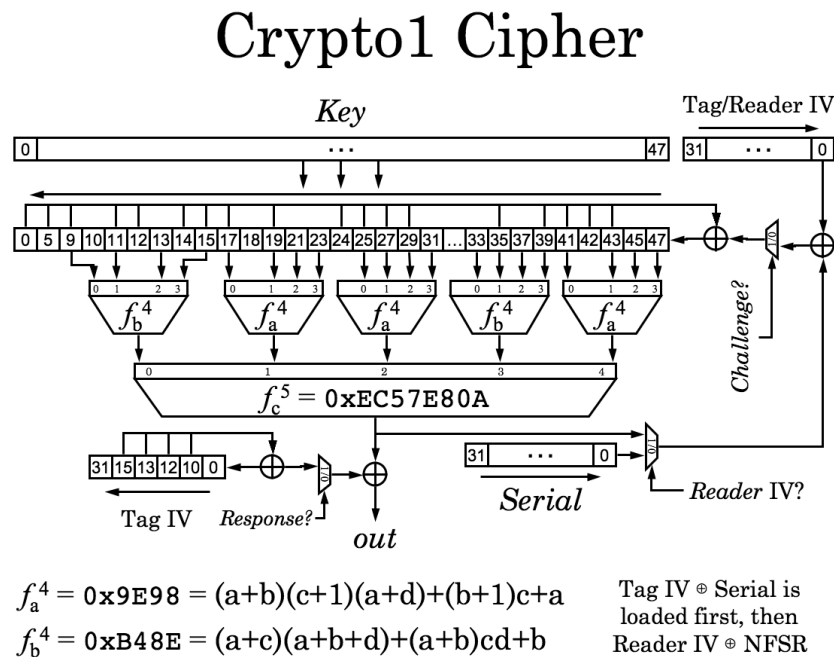


Figura 2.2: Cifrado Crypto1

Incumpliendo el principio de Kerckhoffs, la empresa NXP Semiconductors nunca publicó información relacionada con los detalles del cifrado o sobre la capa de comunicación utilizada por las tarjetas MIFARE Classic. Sin embargo, la criptografía de estas etiquetas fue vulnerada en 2007, cuando un equipo de investigadores utilizó técnicas de ingeniería inversa para determinar el circuito criptográfico utilizado para la implementación del protocolo Crypto1 [4]. Conociendo su funcionamiento, se identificaron serios problemas de seguridad, como debilidades en la generación de números pseudoaleatorios, que sirvieron como base para la concepción de múltiples ataques que permitirían a un adversario

clonar parcialmente una tarjeta, dado acceso a un lector legítimo o a la grabación de una comunicación genuina [5].

Posteriores investigaciones revelaron la posibilidad de realizar ataques que afectan directamente a la implementación de la etiqueta, supliendo la necesidad de interactuar con un lector que almacene las claves secretas. En su lugar, se puede abusar del propio protocolo de autenticación y de debilidades en el cifrado para computar estas claves en tan solo unos pocos intentos de autenticación utilizando tablas pre-computadas [6]; pudiendo clonar una etiqueta por completo tan solo interactuando con ella. Una investigación posterior incluso eliminó la necesidad de emplear estas tablas, rompiendo por completo la seguridad de las tarjetas MIFARE Classic [7]. Puesto que estos ataques se pueden llevar a cabo en un instante en un entorno aislado y no controlado, se consideran una de las amenazas más serias contra la integridad de estas etiquetas, dado su elevado impacto y dificultad de detección.

Debido a las vulnerabilidades discutidas en este apartado, los expertos recomiendan la migración de sistemas que utilicen la tecnología MIFARE Classic a otros sistemas criptográficamente más seguros [8]. Si bien la resiliencia de esas primeras tarjetas, basada en el principio de *seguridad por oscuridad*, ha quedado obsoleta, hoy en día se mantienen como una de las tarjetas de acceso más usadas en multitud de entornos. Por tanto, constituyen un mecanismo habitual de control de acceso que puede ser rápidamente vulnerado con hardware de fácil obtención.

NXP intentó remediar este problema discontinuando las tarjetas MIFARE Classic IC originales y sustituyéndolas por una nueva versión retro-compatible, empleando un generador de números pseudoaleatorios mejorado. Sin embargo, esta versión también fue vulnerada en una investigación publicada en [9], a través de un ataque criptográfico que aprovecha las debilidades inherentes al protocolo Crypto1.

2.2.3. Tarjetas MIFARE mágicas

Como ya se ha mencionado, a pesar de que el diseño empleado por la familia de etiquetas MIFARE Classic es propietario y nunca ha sido publicado por su fabricante, la comunidad de investigadores ha conseguido realizar la ingeniería inversa de los distintos mecanismos criptográficos y protocolos empleados en estas tarjetas de acceso [4]. Esta información ha sido utilizada para la confección de tarjetas MIFARE especiales, conocidas comúnmente como tarjetas “mágicas” o de puerta trasera.

Estas tarjetas incorporan un chip especial de mercado gris que puede emular la estructura de la memoria y la funcionalidad de los circuitos integrados de etiquetas MIFARE reales, pero que, además, implementa características adicionales. Por ejemplo, la mayoría de chips mágicos permiten sobrescribir los datos del bloque establecido por el fabricante como si fuera cualquier otro bloque de usuario. La ventaja significativa de estas tarjetas es la capacidad de cambiar el NUID y los datos de manufacturación, clonando así la información pertinente de una tarjeta MIFARE Classic genuina [10].

Existen dos generaciones de estas etiquetas mágicas, comúnmente denominadas como *gen1a* y *gen2* [11].

El chip de la primera generación *gen1a* incorpora un comando especial de puerta trasera que abre todos los sectores para su lectura y escritura, incluyendo el sector 0. Estas órdenes pueden ser utilizadas para el acceso a cualquier región del almacenamiento, sin necesidad de conocer las claves de autenticación establecidas para cada uno de los sectores. La principal desventaja de estas etiquetas mágicas es que se requiere de hardware especializado para el envío de comandos de puerta trasera, puesto que son enviados al chip después de que este entre en un estado de parada; un comportamiento que no es permitido por la gran mayoría de *smartphones* con capacidad NFC del mercado. Además, algunos lectores buscan chips mágicos emitiendo el comando de puerta trasera y, si el chip responde, se apaga para evitar que una etiqueta posiblemente clonada acceda a la puerta o servicio al que esté conectado el lector.

Las tarjetas de segunda generación *gen2* no tienen ningún comando de puerta trasera pues todos los sectores están simplemente abiertos a la escritura. La ventaja del chip mágico *gen2* es que incluso los teléfonos inteligentes con capacidad NFC pueden simplemente emitir comandos de escritura para cualquier sector, incluido el sector 0. Esto significa que se podría utilizar una aplicación instalada en un *smartphone* para cambiar el NUID del chip junto con todos los datos del bloque de fabricación, sin necesidad de requerir de hardware especializado. Además de esto, los lectores que incorporan contramedidas para ataques de clonación no tienen un procedimiento consistente para determinar si la tarjeta de segunda generación presentada contiene un chip mágico o no, por lo que son más difíciles de detectar que la generación previa.

2.3. Ataques sobre tarjetas de acceso

Durante la realización de este trabajo se han llevado a cabo diversos experimentos involucrando las etiquetas MIFARE Classic previamente descritas. En concreto, se han llevado a cabo dos ataques de clonación haciendo uso de tarjetas de acceso utilizadas en entornos reales como mecanismo de autenticación. A través de los ataques aquí descritos, un adversario podría copiar con rapidez los datos que verifican la identidad de usuarios legítimos del sistema, ganando así acceso a los mismos recursos que estos tienen disponibles. Estos ataques solo requieren de una interacción breve con la tarjeta objetivo, sin necesidad de interactuar con un lector genuino o poseer una comunicación legítima pregrabada.

2.3.1. Clonación del bloque del fabricante

Algunos mecanismos de acceso se basan en la presunción de que hacer uso del bloque establecido por el fabricante es suficiente para garantizar la seguridad del sistema de control de acceso. Si tomamos este bloque como identificador del usuario y tenemos la garantía del fabricante de que nunca se repiten dos identificadores en las tarjetas producidas, podríamos tener un sistema de autenticación resiliente ante ataques de clonación, puesto que se trata de un conjunto de bytes de solo lectura que no pueden ser modificados por un usuario.

Este planteamiento falla completamente ante la concepción de tarjetas MIFARE mágicas que, como se recoge en anteriores apartados, permiten hacer uso de comandos de

puerta trasera para la modificación de cualquier bloque de almacenamiento en el chip. Un atacante que posea las claves de acceso para leer el primer sector de la tarjeta, el sector 0, podría extraer con facilidad los datos identificativos de un usuario a través de una rápida interacción con la etiqueta de este, haciendo uso de un lector NFC cualquiera. Posteriormente, estos datos pueden ser replicados en una tarjeta mágica, siendo necesario modificar únicamente el primer bloque de la misma.

Este ataque fue llevado a la práctica en una organización local que hace uso de pulseras NFC como mecanismo de identificación de clientes. Estas se utilizan tanto a la hora de entrar y salir del recinto como para registrarse en las distintas actividades ofertadas en el centro, por lo que se trata de un recurso frecuentemente utilizado por los usuarios de esta organización. Cabe destacar que la función principal de estas etiquetas en ese caso es facilitar el acceso de los clientes y registrar su actividad en los servicios ofrecidos, más que ofrecer una solución de seguridad para la restricción del acceso a las instalaciones.

Para la realización de esta investigación se ha utilizado una interfaz Proxmark3 Easy para interactuar con la pulsera (ver Figura 2.3). Se trata de una herramienta de implementación abierta utilizada para la investigación y diagnóstico de tarjetas RFID tanto de baja como de alta frecuencia (125 KHz y 13.56MHz, respectivamente). En concreto, para este estudio se ha empleado un firmware conocido como el "*Iceman Fork*", en honor al apodo de su desarrollador. Este implementa una amplia gama de comandos y herramientas de utilidad a la hora de realizar tareas como la identificación de chips, la ejecución de auditorías de seguridad y la propia programación de tarjetas RFID [12].

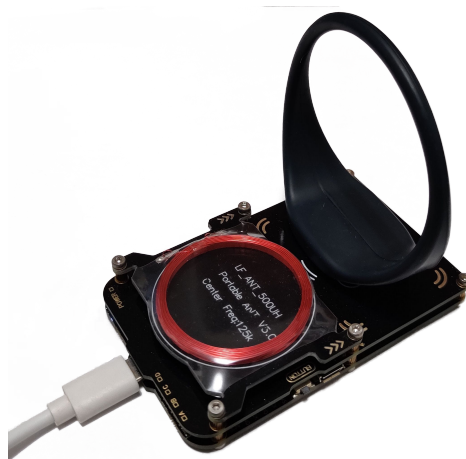


Figura 2.3: Pulsera de identificación y Proxmark3 Easy

El primer paso para el desarrollo de este estudio fue determinar la tecnología utilizada en la etiqueta NFC que incorpora la pulsera objeto de la investigación. Para la interacción entre el equipo y la terminal Proxmark3, se utilizó el cliente Linux incluido en el repositorio del *Iceman fork*. Este programa ofrece una interfaz de uso a través de la línea de comandos que permite enviar órdenes al dispositivo para su interacción con la tarjeta RFID. Una de las utilidades incluidas permite enumerar las etiquetas NFC que son encontradas dentro del rango de la antena de la terminal, listando además información básica sobre el chip detectado.

En la Figura 2.4 se muestran los resultados obtenidos al llevar a cabo esta operación con la pulsera identificativa estudiada.

```
[usb] pm3 --> hf search
      Searching for ISO14443-A tag...
[+]  UID: 77 59 30 7B
[+]  ATQA: 00 04
[+]  SAK: 08 [2]
[+]  Possible types:
[+]  MIFARE Classic 1K
[=]  proprietary non iso14443-4 card found, RATS not supported
[+]  Prng detection: weak
[#]  Auth error
[?]  Hint: try `hf mf` commands

[+]  Valid ISO 14443-A tag found
```

Figura 2.4: Información básica de la etiqueta NFC incrustada

El cliente de la Proxmark3 reportó que se corresponde con una etiqueta MIFARE Classic de 1KB de almacenamiento; tratándose además del modelo IC original, caracterizado por una generación de números pseudoaleatorios (PRNG) débil. Un rápido estudio empleando las utilidades proporcionadas por la herramienta reveló que los 16 sectores de la tarjeta emplean claves de autenticación por defecto, con un valor en hexadecimal de 0xFFFFFFFFFFFF; y que todos los bloques que pueden ser editados por el usuario se encuentran vacíos. La única información restante para identificar al cliente son, por tanto, los 16 bytes establecidos por el fabricante en el bloque 0 de la etiqueta.

Esta información puede ser extraída fácilmente a través de una operación de lectura utilizando la clave del sector 0; y, a continuación, podría ser replicada en una tarjeta MIFARE mágica, realizando así una copia completa del contenido de la pulsera. En la Figura 2.5 se demuestra la ejecución de este procedimiento utilizando la Proxmark3 y una tarjeta con capacidad de puerta trasera.

En disposición de la tarjeta mágica con los nuevos datos cargados, se llevó a cabo una prueba para verificar el éxito de este ataque. Para ello, se utilizó la etiqueta clonada en un lector legítimo del centro, cuyo fin es mostrar al cliente información sobre la actividad que ha realizado dentro de las instalaciones. El lector MIFARE reconoce correctamente la información clonada y muestra los datos relativos al usuario, confirmando así la consecución del objetivo propuesto.

Un adversario malicioso que desee aprovechar las debilidades de este sistema de acceso podría utilizar el propio lector NFC de su *smartphone* para leer los datos de identificación de otros usuarios de la organización, puesto que las claves de acceso al sector 0 son las mismas para todos los clientes de la misma. Con estos, un atacante podría suplantar la identidad de otros usuarios y acceder a información almacenada en el sistema sobre los mismos; así como participar en las actividades ofertadas por el centro en su nombre.

Tal y como se anotaba al comienzo de este apartado, la posibilidad de realizar este ataque no supone como tal un problema de seguridad para esta organización, puesto

```

[usb] pm3 --> hf mf dump
[=] Using `hf-mf-7759307B-key.bin`
[=] Reading sector access bits...
[=] .....
[+] Finished reading sector access bits
[=] Dumping all blocks from card...
[+] successfully read block 0 of sector 0.
[+] successfully read block 1 of sector 0.
[+] successfully read block 2 of sector 0.
[+] successfully read block 3 of sector 0.
[+] successfully read block 0 of sector 1.
[+] successfully read block 1 of sector 1.

```

(a) Extracción de los datos almacenados

```

[usb] pm3 --> hf mf cload -f hf-mf-7759307B-dump.eml
[+] loaded 1024 bytes from text file hf-mf-7759307B-dump.eml
[=] Copying to magic gen1a card
[=] .....

[+] Card loaded 64 blocks from file
[=] Done!
[usb] pm3 --> hf mf rdsc -s 0 -k FFFFFFFFFF

[=] # | sector 00 / 0x00 | ascii
[=] ---+-----+-----+
[=] 0 | 77 59 30 7B 65 | wY0{e
[=] 1 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
[=] 2 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
[=] 3 | 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF | .....

```

(b) Carga de datos en la tarjeta mágica

Figura 2.5: Lectura y escritura de la información en una tarjeta mágica

que existen otros mecanismos para verificar la identidad de los usuarios que acceden a las instalaciones; y esta pulsera es meramente una herramienta que facilita el uso de los servicios ofertados. Sin embargo, este ejemplo demuestra que no solo las tarjetas MIFARE Classic siguen siendo ampliamente utilizadas en el ámbito de las etiquetas de acceso NFC; sino que, además, existen sistemas de acceso que pueden ser directamente vulnerados debido a una implementación que sigue una mala práctica de seguridad, como es el uso de datos de solo lectura como mecanismo de protección ante clonaciones.

2.3.2. Clonación de sectores protegidos

En el siguiente caso de estudio, se analizó la seguridad de las tarjetas proporcionadas por la Universidad de La Laguna para la identificación de sus estudiantes. La tarjeta universitaria es el documento que identifica a todos los miembros de la comunidad académica, y permite acceder a una gran variedad de servicios como los bonos de comedor, el préstamo de libros y el acceso a los colegios mayores y a los aparcamientos de la ULL [13]. Este carné cuenta con un circuito integrado que puede ser utilizado para el acceso a los servicios descritos a través de una interfaz NFC, en la que se basa la investigación detallada a continuación.

Para poder determinar la seguridad de esta tarjeta, fue necesario primero llevar a cabo un análisis inicial para confirmar la tecnología utilizada en esta etiqueta NFC y la estructura de la información almacenada en ella. Para ello, se realizó el mismo procedimiento explicado en la sección anterior, empleando las utilidades ofrecidas por la Proxmark3 para identificar información que podría ser de valor para este estudio.

La herramienta reporta que el carné incorpora una etiqueta MIFARE Classic de 4KB de almacenamiento y con un mecanismo de generación de números pseudoaleatorios fortalecido. Una breve consulta al contenido de la tarjeta revela que 39 de los sectores utilizan la clave de autenticación por defecto y se encuentran ausentes de contenido. Sin embargo, el sector 15 posee claves que no fueron encontradas en el diccionario de la Proxmark3 y que, por tanto, no pudieron ser recuperadas. Los resultados de esta operación se muestran en la Figura 2.6.

```
[usb] pm3 --> hf mf fchk --4k
[=] Running strategy 1
[=] Chunk 2,5s | found 78/80 keys (42)
[=] Running strategy 2
[=] Chunk 1,0s | found 78/80 keys (42)
[=] time in checkkeys (fast) 3,5s

[+] found keys:
```

Sec	Blk	key A	res	key B	res
000	003	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
001	007	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
002	011	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
003	015	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
004	019	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
005	023	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
006	027	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
007	031	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
008	035	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
009	039	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
010	043	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
011	047	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
012	051	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
013	055	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
014	059	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
015	063	-----	0	-----	0
016	067	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1
017	071	FFFFFFFFFFFF	1	FFFFFFFFFFFF	1

Figura 2.6: Claves de los sectores de la tarjeta universitaria

Puesto que el estudio del contenido de este sector podría ser de interés para comprender el funcionamiento del sistema de acceso asociado a esta tarjeta, se utilizó en esta investigación una de las técnicas previamente citadas para la inferencia de claves de autenticación desconocidas, recogida en [9]. Este procedimiento, denominado *hardnested attack*, es un ataque con solo texto cifrado que únicamente requiere de acceso a la tarjeta, puesto que explota un paso crucial y obligatorio en el protocolo de autenticación. Este ataque es generalizable a cualquier tarjeta MIFARE Classic sin importar las medidas de endurecimiento integradas, puesto que depende exclusivamente de las debilidades criptográficas del cifrado de Crypto1.

El único requisito para su ejecución es que el atacante debe de poseer, al menos, una clave válida para alguno de los sectores de la etiqueta. Sin embargo, en la práctica, este requisito es comúnmente satisfecho debido al despliegue masivo de tarjetas que utilizan la clave por defecto para al menos una de estas regiones de almacenamiento, como es el caso de estudio que se presenta.

El *hardnested attack* se encuentra implementado dentro de las herramientas ofrecidas por la Proxmark3. Su ejecución contra la tarjeta universitaria se muestra en la Figura 2.7, recuperando con éxito la clave A para la autenticación en el sector de interés. A partir de este secreto, es posible acceder al resto de datos contenidos en los bloques que componen esta región, mostrados en la Figura 2.8.

```
[usb] pm3 --> hf mf hardnested --blk 0 -b -k FFFFFFFFFF --tblk 60 --ta
[=] Target block no 60, target key type: A, known target key: 000000000000 (not set)
[=] File action: none, Slow: No, Tests: 0
[=] Hardnested attack starting...
[=] -----
[=] Time      | #nonces | Activity                                     | Expected to brute force |
[=] -----+-----+-----+-----+-----
[=] 0         | 0       | Start using 4 threads and AVX SIMD core    |                          |
[=] 0         | 0       | Brute force benchmark: 229 million (2^27,8) keys/s | 140737488355328        | 7d
[=] 10        | 0       | Using 235 precalculated bitflip state tables | 140737488355328        | 7d
[=] 15        | 112    | Apply bit flip properties                  | 105386975232          | 8min
[=] 16        | 224    | Apply bit flip properties                  | 36444532736           | 3min
[=] 17        | 335    | Apply bit flip properties                  | 23595714560           | 2min
[=] 18        | 447    | Apply bit flip properties                  | 23595714560           | 2min
[=] 19        | 558    | Apply bit flip properties                  | 22627538944           | 2min
[=] 20        | 670    | Apply bit flip properties                  | 18605600768           | 81s
[=] 21        | 779    | Apply bit flip properties                  | 18605600768           | 81s
[=] 22        | 891    | Apply bit flip properties                  | 13212351488           | 58s
[=] 23        | 1002   | Apply bit flip properties                  | 11337218048           | 50s
[=] 24        | 1111   | Apply bit flip properties                  | 10391123968           | 45s
[=] 27        | 1223   | Apply Sum property. Sum(a0) = 128         | 5609344512            | 25s
[=] 28        | 1334   | Apply bit flip properties                  | 5609344512            | 25s
[=] 30        | 1446   | Apply bit flip properties                  | 5609344512            | 25s
[=] 31        | 1557   | Apply bit flip properties                  | 5609344512            | 25s
[=] 31        | 1557   | (Ignoring Sum(a8) properties)              | 5609344512            | 25s
[=] 86       | 1557   | Brute force phase completed. Key found: [REDACTED] | 0                      | 0s
[usb] pm3 -->
```

Figura 2.7: Ejecución del *hardnested attack*

```
[usb] pm3 --> hf mf rdsc -s 15 -k [REDACTED]
[=] # | sector 15 / 0x0F | ascii
[=] ---+-----+-----
[=] 60 | 55 4C 4C 20 20 20 20 20 20 20 20 20 20 20 20 | ULL
[=] 61 | 30 30 30 30 30 30 30 31 30 31 32 33 33 35 39 38 | 0000000101233598
[=] 62 | 00 00 00 00 00 00 00 00 52 80 87 01 [REDACTED] | .....R...
[=] 63 | 00 00 00 00 00 00 FF 07 80 69 [REDACTED] | .....i
```

Figura 2.8: Contenido de la tarjeta universitaria

Este sector contiene una estructura de datos común a todas las tarjetas estudiadas, donde se incluyen las siglas de la Universidad, el Número de Identificación Universitaria (NIU) del alumno y un código de uso desconocido. Cabe destacar que los dos primeros datos se encuentran codificados en ASCII, por lo que son fácilmente legibles, mientras que el último está compuesto exclusivamente por dígitos entre el 0 y el 9 cuando es representado en hexadecimal. Es probable que este código se trate de un identificador interpretado numéricamente por el lector, utilizado para identificar inequívocamente las tarjetas producidas por la Universidad. Esta hipótesis es reforzada por los resultados obtenidos en el estudio realizado, donde se comprobó que este, efectivamente, variaba entre distintos estudiantes; e, incluso, entre tarjetas pertenecientes a un mismo alumno, como era el caso de personas que poseían múltiples carnés por pertenecer a distintos roles dentro de la organización.

La copia de esta información a una tarjeta mágica puede realizarse siguiendo el mismo procedimiento utilizado en el ataque de clonación previamente descrito. Puesto que la información de interés se encuentra almacenada en el sector 15 y el resto de los sectores no son utilizados, la copia se pudo realizar sobre una tarjeta mágica MIFARE Classic 1K; a pesar de que la etiqueta original poseía 4KB de almacenamiento. La etiqueta duplicada fue testada contra los lectores NFC ubicados en los aparcamientos de la Universidad, comprobando que la tarjeta clonada es correctamente validada por estos dispositivos y permite el acceso a las instalaciones.

Con el fin de determinar el funcionamiento del sistema subyacente a este mecanismo de acceso, se realizaron posteriores pruebas modificando el contenido de la tarjeta clonada. Puesto que el NIU del estudiante se encuentra incluido dentro de este sector, se propuso la hipótesis de que el lector podría utilizar únicamente esta información para validar la

identidad del alumno. Esto hubiera supuesto un problema de seguridad, puesto que un atacante podría fácilmente editar este valor almacenado en ASCII y reemplazarlo por el de otro usuario conocido del sistema. Sin embargo, se determinó que la única información leída a la hora de validar la tarjeta es el código almacenado en el tercer bloque del sector de interés. Puesto que este es asignado de forma aparentemente aleatoria, no se corre el riesgo de que un adversario pueda suplantar directamente la identidad de otro alumno o de personal universitario en, al menos, este servicio. Sin embargo, una vez conocidas las claves de acceso para este sector, un agente maligno podría intentar leer la información almacenada en la tarjeta universitaria de su víctima, clonando posteriormente los datos a una segunda etiqueta siguiendo los pasos descritos en el estudio realizado.

Capítulo 3

Tarjetas de identidad

3.1. Antecedentes

El Documento Nacional de Identidad (DNI) 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 [15]. 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 implementa 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 [16]. 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.

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 [18].

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 [19]. Una de las características más notables de esa versión es su diseño y 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 [20], la versión actual del DNIE incorpora nuevas medidas de seguridad, tanto visibles como invisibles.

3.2. Estandarización

El DNI electrónico es un documento utilizado tanto como para la identificación nacional de un individuo como documento de viaje válido dentro de la Unión Europea [21]. Como tal, este documento se rige por estándares establecidos por la Organización de Aviación Civil Internacional (ICAO), y en concreto por el estándar ICAO 9303.

Este estándar de 13 partes define los distintos aspectos por los que se debería de registrar un documento de viaje legible automáticamente (*electronic Machine Readable Travel Document* o eMRTD). Aunque estas especificaciones se conciben para aplicarse en particular a los pasaportes electrónicos, se implementan también en documentos nacionales de identidad para garantizar su interoperabilidad entre distintas regiones. En las distintas partes de este estándar, se definen asuntos desde las medidas de seguridad para el diseño, fabricación y expedición de eMRTD hasta la estructura lógica de datos para el almacenamiento de datos en el Circuito Integrado (CI), incluyendo mecanismos de seguridad necesarios [22].

El DNIE se alinea de igual forma con el sistema europeo de reconocimiento de identidades electrónicas (o eIDAS, por sus siglas en inglés de *electronic IDentification, Authentication and trust Services*), establecido en el reglamento (UE) Nº 910/2014 [23]. Este sistema proporciona las bases para la creación de un entorno normativo que permite utilizar sistemas nacionales de identificación electrónica (eID) para acceder a servicios públicos de otros países de la Unión Europea; así como garantiza la validez de servicios como la firma electrónica a través de fronteras, proporcionándoles el mismo estatus legal que los procesos tradicionales basados en papel.

La gestión y protección de los certificados utilizados para la autenticación y firma del ciudadano se convierte bajo este marco legal en una tarea de gran importancia dentro del diseño del DNIE, puesto que es necesario garantizar el no repudio y el compromiso del ciudadano con el contenido de los documentos firmados. Con este fin, se concibe el DNIE como un Dispositivo Seguro de Creación de Firma (DSCF), donde los pares de claves correspondientes a estos certificados son generados en la propia tarjeta y nunca salen del chip; donde se realizan todas las operaciones que requieran de su uso. Con el fin de proteger este mecanismo de seguridad, el DNIE se encuentra certificado de acuerdo con el estándar europeo EN 419211 [24][25] y con las evaluaciones establecidas en el Common Criteria Protection Profile [26].

3.3. Interfaz de uso

El DNIE se encuentra equipado con un chip SLE78CLFX408AP del fabricante Infineon Technologies [27], donde se implementa la funcionalidad digital del documento. Este CI es accesible a través de una interfaz dual que soporta acceso a través de una toma de contacto estandarizada en el ISO/IEC 7816 y de forma inalámbrica por medio de una antena NFC, siguiendo el conjunto de protocolos definido en el ISO/IEC 14443.

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 [28] y de forma inalámbrica por medio de una antena NFC, siguiendo el conjunto de protocolos definido en el estándar

ISO/IEC 14443 [29]. Este 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. La Figura 3.1 muestra un ejemplo de este sistema estandarizado en la estructura lógica de datos (*Logical Data Structure* o LDS) 2.0.

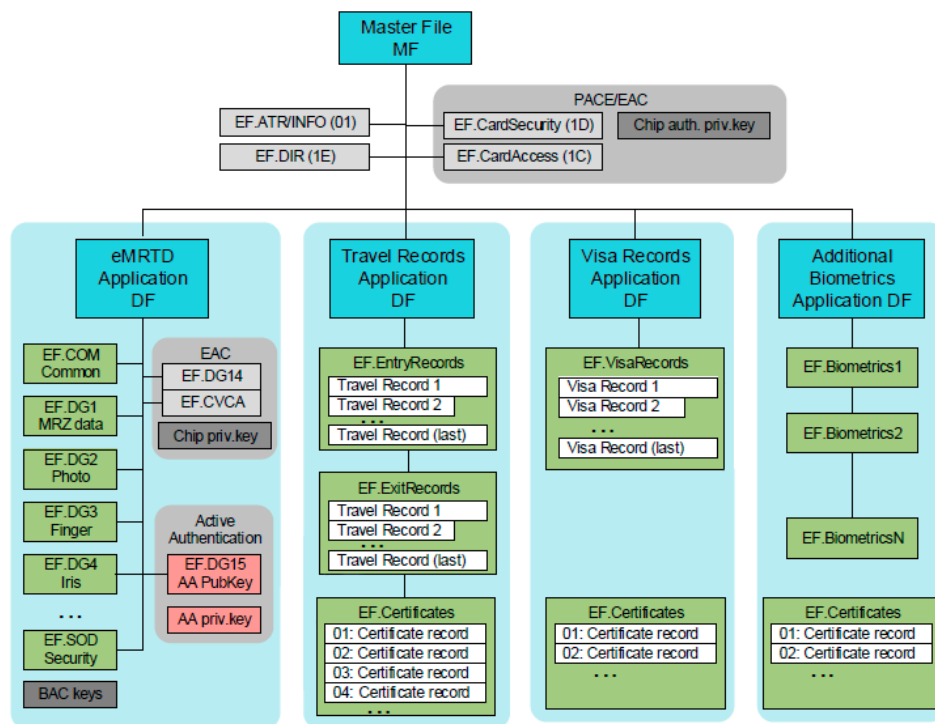


Figura 3.1: Ejemplo de estructura de ficheros estandarizada en LDS 2.0 [30]

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* [31].

3.4. Estructura lógica de datos

Con el fin de asegurar la interoperabilidad internacional de los datos almacenados en el DNIE, la ICAO establece una Estructura Lógica de Datos (o LDS, por sus siglas en inglés) que han de seguir todos los pasaportes electrónicos. La información almacenada en la misma es asignada durante el proceso de creación del DNIE y no puede ser modificada a posteriori, sirviendo como mecanismo de protección contra posibles ataques de manipulación. Estos datos son accesibles a través de la aplicación *ePassport* del DNIE.

De acuerdo con las pruebas realizadas durante esta investigación, el DNIE implementaría la misma versión de LDS que su predecesor, LDS v1.7. Se plantea, sin embargo, que en una próxima actualización de este documento se emplee LDS2 como estructura de datos, extendiendo el uso del DNIE para permitir el almacenamiento seguro de información de viajes realizados o la adición de nuevos datos biométricos del portador.

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. Los datos del portador se almacenan en grupos numerados desde DG1 hasta DG16, aunque no todos son de uso obligatorio. En la Tabla 3.1 se muestran los grupos de datos almacenados en el DNIE, junto a sus formatos de codificación: básica o BER, y de intercambio biométrico o CBEFF.

Grupo de datos	Descripción	Codificación
DG1	Zona legible por máquina	BER
DG2	Datos faciales	CBEFF
DG3	Datos dactilares	CBEFF
DG7	Firma	BER
DG11	Detalles personales adicionales	BER
DG13	Detalles opcionales	BER
DG14	Opciones de seguridad	BER

Tabla 3.1: Grupos de datos almacenados en el DNIE 4.0

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) [32].

3.5. Mecanismos de seguridad

En la parte 11 del estándar ICAO 9303 se define un conjunto de medidas de seguridad para aumentar la resiliencia de los pasaportes electrónicos ante los ataques más comunes contra ese tipo de documentos [33]. En la Tabla 3.2 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.

3.5.1. 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

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
Active Authentication	AA	Desafío-respuesta	Clonación
Chip Authentication	CA	Autenticación	Clonación
Terminal Authentication	TA	Autenticación mediante PKI	Robo de información sensible

Tabla 3.2: Algoritmos de seguridad implantados en documentos eMRTD

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 documento. En particular, se obtienen 3 campos del mismo:

- **Número de documento.** Se trata de un número que identifica de forma única al documento. Está compuesto por hasta 9 caracteres alfanuméricos: en el caso del DNIE 4.0, los tres primeros caracteres son alfabéticos mayúsculas; y, los 6 restantes, numéricos.
- **Fecha de nacimiento.** Fecha de nacimiento del portador, descrita en formato YYMMDD.
- **Fecha de caducidad.** Fecha de caducidad del documento, descrita en formato YYMMDD.

El uso del protocolo BAC como mecanismo de autenticación ha sido objeto de estudio tanto a nivel nacional [34] como internacional [35]. En estas investigaciones, se plantea que la entropía teórica de 61 bits para las claves de sesión derivadas de este proceso puede verse visto reducida a través de ataques de inteligencia de fuentes abiertas (OSINT del inglés *Open Source INTelligence*), conociendo la fecha de nacimiento de la víctima y el horario de apertura de los centros de expedición del DNIE. La seguridad de las claves usadas para el establecimiento de este protocolo se encontraría por debajo del mínimo teórico recomendado.

Con el fin de subsanar las debilidades que presenta el protocolo BAC, se plantea el uso de *Supplemental Access Control* (SAC) como sustituto. Este mecanismo de autenticación utiliza el protocolo *Password Authenticated Connection Establishment* (PACE) para proveer un cifrado fuerte y permitir la compartición de códigos de autenticación de mensaje (*Message Authentication Codes* o MAC) para el establecimiento de un canal seguro.

La clave de sesión utilizada para cifrar la comunicación se deriva a partir de un secreto compartido entre el DNIE y el Sistema de Inspección, que puede ser tanto la contraseña

derivada del MRZ (utilizada por el protocolo BAC), el código PIN, el código PUK o el *Card Access Number* (CAN) asociado al DNIe. Debido a su sencillez, comúnmente se utiliza este último. Se trata de un número impreso en el frontal del documento, de seis dígitos de longitud, asignado de forma pseudoaleatoria durante la creación del documento. A pesar de que este secreto precompartido tiene una baja entropía (de unos 20 bits), permite generar un hash SHA-1 y negociar con este valor, a través de un intercambio Diffie-Hellman, una clave de sesión efímera mucho más fuerte.

3.5.2. 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* (CSCA) reconocida y válida.

3.5.3. Extended Access Control

El trato con 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 lo que 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 [36]. En la Figura 3.2, extraída de [30], se muestra una vista simplificada de la jerarquía de la PKI del EAC. Concretamente, el protocolo EAC consta de tres partes:

1. **Chip Authentication (CA).** El Protocolo de Autenticación del Chip es un protocolo de acuerdo de claves Diffie-Hellman efímero-estático con dos objetivos: proporcionar una comunicación segura con una clave de sesión más fuerte que la negociada inicialmente y proveer de una autenticación unilateral del chip.
2. **Passive Authentication (PA).** El Sistema de Inspección, llegados a este punto, debe llevar a cabo el proceso de Autenticación Pasiva descrito anteriormente; con el fin de confirmar la validez de la clave pública proporcionada por el documento en la fase de Autenticación del Chip.
3. **Terminal Authentication (TA).** El Protocolo de Autenticación del Terminal permite establecer una autenticación unilateral del SI, con el fin de validar la autoridad del mismo y permitir acceso a los datos almacenados en grupos más sensibles. Se basa en un proceso de desafío-respuesta donde se verifica que el certificado almacenado en el SI pertenece a una cadena de certificación firmada por una Autoridad de Certificación de Verificación de país.

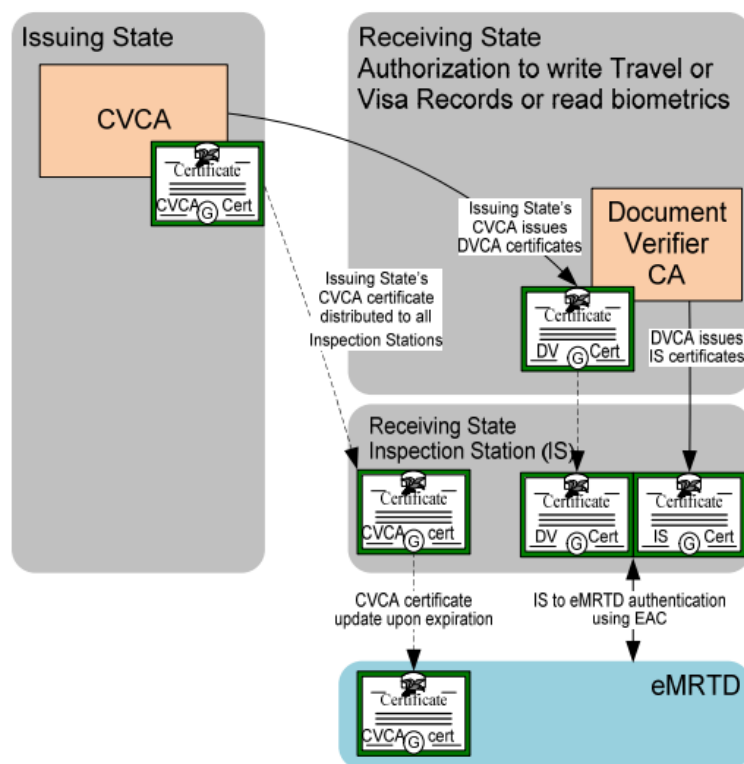


Figura 3.2: Vista simplificada de la jerarquía de la PKI del EAC

3.6. Vulnerabilidades conocidas

En octubre de 2017, se presentó en el *ACM SIGSAC Conference on Computer and Communications Security* el artículo [37] describiendo una vulnerabilidad encontrada en los chips fabricados por Infineon Technologies. El problema se encuentra, en concreto, en una implementación propietaria de la función para la generación de claves RSA incluida en la librería *RSALib* provista por la compañía. Para la seguridad de este proceso se exige la generación pseudoaleatoria de dos números primos suficientemente grandes, una operación que es especialmente costosa en dispositivos con recursos de cómputo limitados, como es el caso de las *smartcards*.

El problema que dio lugar a la vulnerabilidad es que dicho proceso de elección de los dos números primos fue simplificado en la librería con el fin de acelerar la creación de claves RSA. Concretamente los investigadores determinaron que el método usado debilitaba la entropía de las claves generadas, siendo posible realizar un ataque de factorización que permite recuperar la clave privada a partir de su contraparte pública en cuestión de semanas o meses, empleando un único núcleo de los procesadores comerciales más utilizados. Además, señalaron que los pares de claves generados a través de ese proceso poseen una huella digital verificable en cuestión de microsegundos, por lo que las claves vulnerables pueden ser rápidamente identificadas por un adversario. Múltiples implementaciones públicas de ese ataque afectaron especialmente a claves con longitudes 512, 1024 y 2048 bits generadas mediante el uso de *RSALib*.

Esa debilidad pasó a ser conocida bajo el identificador CVE-2017-15361 [38]; o, más comúnmente, bajo el nombre de vulnerabilidad ROCA (acrónimo de *Return Of Copersmith's Attack*). Millones de dispositivos fueron vulnerados mediante ese ataque,

incluyendo chips de Módulos de Plataforma Segura TPM (*Trusted Platform Module*), tokens de autenticación YubiKey y tarjetas inteligentes de todo tipo [39]. Puesto que el DNIE 3.0 incorpora un chip fabricado por Infineon, fue afectado por esta vulnerabilidad: la Dirección General de Policía se vio obligada a revocar 19 millones de certificados digitales en uso por ciudadanos españoles a raíz de este hallazgo.

Las consecuencias de este ataque de factorización son de especial impacto y relevancia en el caso del DNIE, puesto que la recuperación de la clave privada empleada en el certificado digital permitiría a un atacante la firma de documentos en nombre de su víctima o la autenticación en los servicios electrónicos de la Administración Pública. Si bien la revocación de las claves expedidas alivia en cierta medida el problema, este persiste de forma retroactiva en la firma de documentos previo al anuncio de esta vulnerabilidad, como se explica en [40].

Si un adversario tuviera posesión de un documento firmado por la víctima previo al 2017, podría comprobar con facilidad si la firma del mismo se realizó utilizando una clave afectada por ROCA. En caso afirmativo, podría factorizar la clave privada del certificado digital a partir de la clave pública incrustada en el archivo. Si tiene éxito en este proceso, el atacante podría entonces firmar documentos previos a la revocación del certificado digital en 2017, cambiando la hora y fecha de su ordenador para simular condiciones de firma legítimas. Como se indicaba en anteriores apartados de este trabajo, la firma del DNIE posee la misma validez legal que la firma manuscrita de la víctima y tiene un carácter vinculante que garantiza el compromiso del ciudadano con el contenido del documento; por lo que la realización de este ataque con éxito podría suponer un grave problema para la persona afectada.

Otros países de la Unión Europea perjudicados por la vulnerabilidad ROCA, como es el caso de Estonia [41], optaron por reemplazar de inmediato los documentos de identidad afectados. En un primer momento, la respuesta de la Dirección General de Policía fue la de permitir a los ciudadanos seguir utilizando los soportes preexistentes afectados, reduciendo el tamaño de las claves RSA generadas en los nuevos certificados digitales expedidos de 2048 bits a 1920. Esta longitud de clave, mientras que es menos susceptible al ataque descrito en ROCA, reduce significativamente la fortaleza del sistema de firma electrónica del DNIE y se encuentra por debajo de las recomendaciones del Esquema Nacional de Seguridad y de la documentación recogida por el Centro Criptológico Nacional [42] [40].

Por ello, los DNIE que poseen un chip afectado por la vulnerabilidad ROCA han sido paulatinamente reemplazados por nuevos documentos que no poseen esta debilidad y que cuentan con certificados digitales que emplean claves RSA de 2048 bits de longitud. Puesto que la sustitución se realiza por un fallo del sistema y no por una renovación por caducidad, este cambio es gratuito para los ciudadanos; estimándose en un coste aproximado de 134 millones de euros para la Administración Pública [43].

3.7. Investigación práctica

A continuación se recoge la investigación práctica llevada a cabo durante el desarrollo de este trabajo, en cuanto a documentos de identidad se refiere. Este estudio tuvo como

objetivo explorar la implementación de las funcionalidades ofrecidas y mecanismos de seguridad asociados contenidos en la última revisión del DNIE, realizado sobre un documento emitido a finales del 2021. Fueron desarrolladas diversas líneas de investigación para analizar el funcionamiento del DNIE a bajo nivel, con el objetivo de confirmar la resiliencia del DNIE ante vectores de ataques frecuentes en pasaportes electrónicos y dispositivos NFC. El análisis realizado, una vez concluido, fue presentado al organismo responsable del DNIE, siguiendo las buenas prácticas recomendadas de revelación responsable en *hacking* ético.

Este estudio se presenta de forma separada a la investigación realizada sobre tarjetas de acceso, puesto que el DNIE es un documento que cuenta con medidas de seguridad mucho más elaboradas que las de esas primeras etiquetas NFC. Ataques como los de clonación o forja, presentados en anteriores secciones de este proyecto, simplemente no son factibles contra el DNIE. En consecuencia, parte de esta sección se centra en analizar el funcionamiento de los mecanismos de protección integrados en el DNIE, en busca de potenciales vectores de ataque que permitan explorar la seguridad del documento.

3.7.1. Análisis de paquetes APDU en vivo

El primer paso planteado fue determinar una metodología de trabajo factible para la realización de esta investigación. En concreto, se planteó el desarrollo de una plataforma que permita observar los paquetes APDU (*Application Protocol Data Unit*) enviados entre la tarjeta y el lector cuando se lleva a cabo una comunicación legítima. Este análisis permite obtener un mejor entendimiento de los protocolos de acceso descritos en anteriores secciones de este documento, así como plantear nuevas líneas de investigación que requieran de un acceso a bajo nivel a esta conexión.

Una de las restricciones que fueron planteadas en este marco de trabajo es que el estudio debía poder ser replicable utilizando un hardware de fácil acceso; en lugar de utilizar dispositivos especializados para la auditoría de tarjetas RFID, como es el caso de la Proxmark3 usado con anterioridad. Se eligió emplear las capacidades NFC de un *smartphone* Android por su versatilidad y fácil programación; aprovechando además el SDK (*Software Development Kit*) ofrecido por el Cuerpo Nacional de Policía para la programación de aplicaciones que integren el DNIE 3.0 a través de un *middleware* denominado *DNIEdroid* [44]. Este software permite interactuar con el documento de identidad para realizar operaciones como la extracción de datos identificativos del DNIE, la verificación de la mayoría de edad de un ciudadano o la corroboración de la validez de los certificados digitales del mismo (ver Figura 3.3).

El interés principal de este *kit* de desarrollo reside en la posibilidad de analizar las funciones allí implementadas para la interacción del dispositivo con el DNIE: si es posible modificar el comportamiento de la librería *DNIEdroid*, implementar una rutina que muestre las trazas de los paquetes APDU transmitidos debería de ser una tarea relativamente sencilla de llevar a cabo, completando así el primer paso de esta investigación. Por desgracia, esta alteración en el funcionamiento del *middleware* no puede ser realizada directamente, ya que la librería se distribuye ya compilada en el formato de distribución binaria utilizado por Android. Sin la posesión del código fuente original del



Figura 3.3: Interfaz de la aplicación de prueba proporcionada en el SDK

mismo, la modificación estática de esta dependencia puede convertirse en una tarea de gran complejidad que se escapa de los objetivos de esta investigación.

En su lugar, se optó por simplificar esta tarea a través de la modificación dinámica de la ejecución de la librería *DNIEdroid*, utilizando una técnica denominada como *API hooking* (ver Figura 3.4). Este procedimiento permite al investigador interceptar y alterar el comportamiento y el flujo de las llamadas a la interfaz de programación de aplicaciones (*Application Programming Interface* o API) definida en el sistema [45]; permitiendo así la inyección de *scripts* en puntos de interés de la aplicación y la depuración del código propietario utilizado. Esta técnica permite reemplazar cualquier función definida en la librería de *DNIEdroid* por código controlado por el analista y, puesto que estos parches se realizan en tiempo de ejecución, requieren de una menor complejidad de desarrollo que la aproximación estática alternativa.

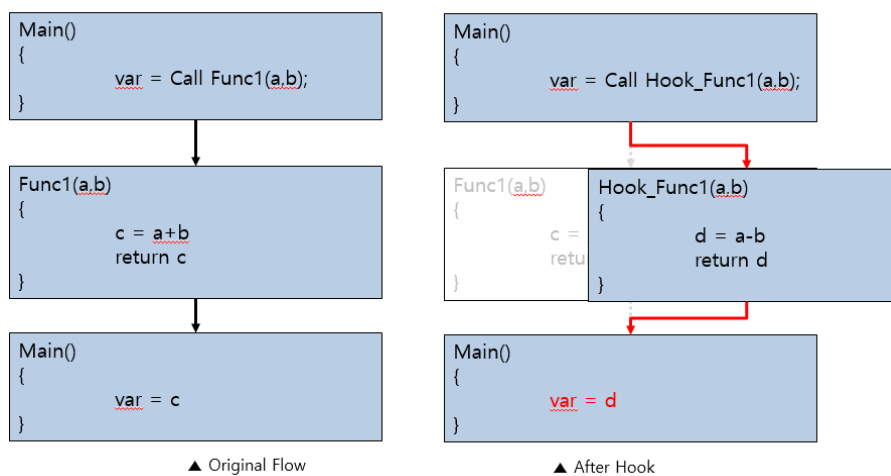


Figura 3.4: Demostración del proceso de *API Hooking*

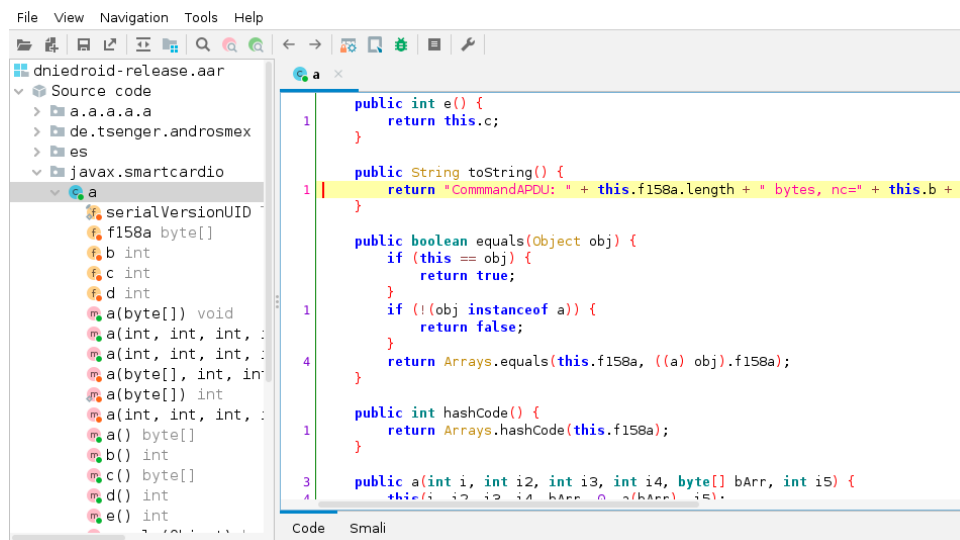
Con este fin, se eligió utilizar la herramienta Frida [46], diseñada específicamente

para la realización de investigaciones como la descrita en esta sección. Esta utilidad es compatible con dispositivos Android, permitiendo así realizar las modificaciones deseadas sobre la librería *DNIEdroid* mientras se ejecuta de forma nativa en el teléfono. Para la ejecución de Frida en un *smartphone*, un requisito indispensable es que se debe poseer acceso como “superusuario” al mismo, requiriendo de la realización de un proceso de *rooting* del dispositivo.

El siguiente paso a completar, una vez decidida la metodología de trabajo, es determinar las funciones que han de ser interceptadas dentro del código de la librería para obtener acceso a los paquetes APDU enviados y recibidos por el lector NFC. Frida requiere que, para enganchar una función específica dentro de una clase, se indique el paquete Java donde esta se encuentra implementada. Puesto que no disponemos del código fuente de la librería para determinar los paquetes en uso, es necesario utilizar técnicas de ingeniería inversa para su extracción.

En esta investigación se ha utilizado *jadx* [47], un decompilador que permite producir código Java a partir del *bytecode* Dalvik empaquetado en una librería binaria de Android dada. Esta utilidad aporta una aplicación GUI a través de la cual es posible navegar a través de los distintos paquetes y recursos incluidos en el fichero en formato AAR, permitiendo al investigador identificar los fragmentos de código relevantes expresados en un lenguaje de alto nivel.

Tras un análisis en profundidad del código decompilado resultante de utilizar esta herramienta sobre la librería *DNIEdroid*, se encontró la presencia de una cadena de texto, resaltada en la Figura 3.5, que revela que la clase en cuestión es la encargada de gestionar el cifrado y envío de los paquetes APDU de comando enviados al DNIE. En el mismo espacio de nombres, se encuentra definida la clase complementaria que se responsabiliza de recibir las respuestas enviadas de vuelta por el documento. Ambas funcionalidades se corresponden con el paquete `javax.smartcardio`, encargado de gestionar la interacción entre aplicaciones Java y *smartcards* [48].



```
File View Navigation Tools Help
-----
dniedroid-release.aar
  Source code
    a.a.a.a.a
    de.tsenger.androsdex
    es
    javax.smartcardio
      a
        serialVersionUID
        f158a byte[]
        b int
        c int
        d int
        a(byte[]) void
        a(int, int, int, ...)
        a(int, int, int, ...)
        a(byte[], int, int, ...)
        a(byte[]) int
        a(int, int, int, ...)
        a() byte[]
        b() int
        c() byte[]
        d() int
        e() int

1 public int e() {
2     return this.c;
3 }

4 public String toString() {
5     return "CommandAPDU: " + this.f158a.length + " bytes, nc=" + this.b +
6 }

7 public boolean equals(Object obj) {
8     if (this == obj) {
9         return true;
10    }
11    if (!(obj instanceof a)) {
12        return false;
13    }
14    return Arrays.equals(this.f158a, ((a) obj).f158a);
15 }

16 public int hashCode() {
17     return Arrays.hashCode(this.f158a);
18 }

19 public a(int i, int i2, int i3, int i4, byte[] bArr, int i5) {
20     +this.f158a = new byte[15];
21 }

Code Smali
```

Figura 3.5: Interfaz gráfica de la herramienta *jadx*

```

1  let commandApuClass = Java.use('javax.smartcardio.a');
2  let responseApuClass = Java.use('javax.smartcardio.b');
3
4  commandApuClass.$init.overload('int', 'int', 'int', 'int', '[B', 'int').implementation
↪ = function (a, b, c, d, e, f) {
5    this.$init(a, b, c, d, e, f);
6    console.log(this.toString());
7    send("{type: 'command'}", new Uint8Array(this._a.value));
8  };
9
10 responseApuClass.$init.implementation = function (a) {
11   this.$init(a);
12   console.log(this.toString());
13   send("{type: 'response'}", new Uint8Array(this._a.value));
14 };

```

La información recabada fue utilizada, a continuación, para la interceptación de los mismos utilizando los mecanismos de instrumentación proporcionados por Frida. En el fragmento de código anterior, se muestra parte del *script* inyectado en el proceso en ejecución en el dispositivo, con el objetivo de recuperar los paquetes APDU involucrados en la comunicación NFC. En primer lugar, se obtiene una referencia a las clases `javax.smartcardio.a` y `javax.smartcardio.b`, correspondiéndose, respectivamente, con la implementación de `CommandAPDU` y `ResponseAPDU` en el paquete oficial. A continuación, se modifica el constructor de cada una de estas para que, cuando un nuevo objeto de estos tipos sea creado, se muestre por pantalla la información más relevante del mismo; recuperada a través de una llamada al método `toString()` de estas clases.

Los *bytes* que componen el cuerpo de los comandos y las respuesta requieren de un mayor procesamiento para su visualización en pantalla. En preparación para estos casos, Frida ofrece un mecanismo de comunicación que conecta el proceso inyectado con el ordenador anfitrión donde se está ejecutando la herramienta. Este puente permite el envío de un objeto JSON a una segunda aplicación enlazada con Frida, en este caso, encargada de realizar el post-procesamiento que sea necesario para la interpretación de los datos recogidos.

Este segundo *script* fue desarrollado utilizando la librería de Frida disponible para Python, diseñada con el fin de automatizar las tareas realizadas con esta herramienta. Primero, el programa se conecta con el servidor de Frida en ejecución en el dispositivo Android, conectado al equipo a través del sistema de depuración USB. A continuación, se acopla al proceso que ejecuta la aplicación de prueba proporcionada en el SDK del Cuerpo Nacional de Policía y, a continuación, carga las modificaciones descritas anteriormente, almacenadas en un fichero JavaScript.

```

1  if __name__ == '__main__':
2    device = frida.get_usb_device()
3    process = device.attach('Sample_Dnie_App')
4    with open('raw_apdu.js') as jscode:

```

```

5     script = process.create_script(jrcode.read())
6     script.on('message', message_handler)
7     script.load()

```

Cuando se ejecuta la función `send()` en el proceso hijo, un mensaje es recibido en el *script* anfitrión. Tras comprobar que esta comunicación no se debe a un error producido, se extraen los datos adjuntos al mismo; correspondiente con los *bytes* que conforman el paquete APDU interceptado. Estos, son mostrados en pantalla utilizando el paquete *hexdump* [49], permitiendo visualizar la información contenida en los mismos en hexadecimal; facilitando así su estudio.

```

1  def message_handler(message, data):
2      if message['type'] == 'error':
3          print('[E]', message)
4      elif message['type'] == 'send':
5          hexdump.hexdump(data)

```

Con estos cambios, es posible recibir en terminal una actualización a tiempo real de los paquetes enviados en una comunicación legítima entre el DNIe y el lector NFC del *smartphone*. Como elemento adicional, se descubrió que en el código de *DNIeDroid* se incluye una clase utilizada para el registro de la información de depuración emitida por la librería. A través de esta funcionalidad, se permite recibir mensajes de estado sobre errores, avisos e información de interés producida durante la ejecución de la aplicación. Puesto que la recolección de estos datos podría ser de utilidad durante el estudio, se incluyen las siguientes adiciones en el *script* inyectado en el proceso hijo, permitiendo su visualización en el equipo del investigador.

```

1  let loggerClass = Java.use('a.a.a.a.a');
2  loggerClass.a.implementation = function (str) {
3      console.log(`[V:${this._b.value}]`, str);
4  }
5  loggerClass.b.implementation = function (str) {
6      console.log(`[E:${this._b.value}]`, str);
7  }
8  loggerClass.c.implementation = loggerClass.d.implementation = function (str) {
9      console.log(`[I:${this._b.value}]`, str);
10 }
11 loggerClass.e.implementation = function (str) {
12     console.log(`[W:${this._b.value}]`, str);
13 }

```

En la Figura 3.6, se muestran los resultados obtenidos al utilizar la herramienta desarrollada. En este caso, se contemplan los paquetes intercambiados por el lector NFC y la tarjeta al acercarla a la terminal. A través de la información textual de

depuración y de los códigos de operación extraídos de los paquetes de comando, se puede comprobar que el primer paso llevado a cabo por el lector es la ejecución del protocolo PACE; necesaria para la autenticación del lector contra el DNIE. El estudio de todas las operaciones consecutivas puede realizarse de la misma forma, proporcionando una visión práctica de los protocolos descritos en anteriores secciones.

```
[H] Attaching to USB device
[H] Attaching to target process: Sample_Dnie_App
[H] Loading JS Frida module
[+] Frida hooked up!
[I:SmartCardMRTDConnection] --NFC type is B--
ResponseAPDU: 2 bytes, SW=9000
00000000: 90 00 ..
ResponseAPDU: 10 bytes, SW=9000
00000000: 31 70 30 0D 06 08 04 00 90 00 1p0.....
ResponseAPDU: 116 bytes, SW=9000
00000000: 31 70 30 0D 06 08 04 00 7F 00 07 02 02 02 02 01 1p0.....
00000010: 01 30 0F 06 0A 04 00 7F 00 07 02 02 03 02 01 02 .0.....
00000020: 01 01 30 12 06 0A 04 00 7F 00 07 02 02 04 02 02 ..0.....
00000030: 02 01 02 02 01 0D 30 12 06 0A 04 00 7F 00 07 02 .....0.....
00000040: 02 04 02 01 02 01 02 02 01 0D 30 12 06 0A 04 00 .....0.....
00000050: 7F 00 07 02 02 04 01 02 02 01 02 02 01 00 30 12 .....0.....
00000060: 06 0A 04 00 7F 00 07 02 02 04 01 01 02 01 02 02 .....
00000070: 01 00 90 00 ....
[I:SmartCardMRTDConnection] --Start of PACE--
[V:Command.MSESetAT] sending command
CommandAPDU: 20 bytes, nc=15, ne=0
00000000: 00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 02 .".....
00000010: 02 83 01 02 ....
ResponseAPDU: 2 bytes, SW=9000
00000000: 90 00 ..
[V:Command.MSESetAT] SW response 0x9000 Normal operation. Operation successful
[V:Command.GeneralAuthenti] sending command
CommandAPDU: 7 bytes, nc=2, ne=0
00000000: 10 86 00 00 02 7C 00 .....|.
ResponseAPDU: 22 bytes, SW=9000
00000000: 7C 12 80 10 93 80 58 B6 09 DA 86 A3 24 CB 7F 88 |.....X.....$...
00000010: 23 11 A4 CE 90 00 #.....
```

Figura 3.6: Visualización de los paquetes intercambiados con el DNIE

3.7.2. Ataque de fuerza bruta contra PACE

En anteriores apartados se discutía el uso del protocolo PACE como mecanismo de protección de la información almacenada en la aplicación de *ePassport* del DNIE, garantizando que el inspector ha mantenido contacto visual con el documento antes de leerlo. Esta medida de seguridad previene la ejecución de ataques de carterismo, evitando que un adversario pueda extraer datos personales de su víctima como su dirección de residencia o su firma manuscrita sin su consentimiento.

En esta línea de investigación, se estudia la viabilidad de realizar un ataque de fuerza bruta contra el DNIE que permita determinar el código CAN asociado al documento. Puesto que este está compuesto por 6 caracteres numéricos, encontrar la clave correcta requeriría de, en el peor caso, 10^6 intentos; una cantidad que podría ser factible en función del tiempo que tarde en completarse cada ejecución del protocolo PACE. Un estudio similar al realizado se encuentra recogido en [34].

Para llevar a cabo este ataque se analizó el código decompilado a través de *jadx* para encontrar la función encargada de llevar a cabo el protocolo PACE. Tras una investigación exhaustiva, se determinó que este comportamiento se incluía en la clase `es.gob.jmulticard.d.b.b`: si bien el nombre de este espacio ha sido ofuscado durante

el proceso de compilación, se determinó a través de una revisión manual del código que a esta clase originalmente le correspondía el nombre `SmartCardMRTDConnection`.

En concreto, la función `a()` de esta clase recibe como argumento una lista de cadenas de texto, cada una de las cuales corresponde con un código CAN a probar contra el DNIe. A través del funcionamiento intencionado de esta aplicación, este argumento siempre corresponde con una única clave que ha introducido previamente el usuario en la interfaz del programa. Sin embargo, a través de la instrumentación ofrecida por Frida, sería posible editar esta lista para que incluyese múltiples códigos que fueran utilizados para intentar autenticar al lector. Esta función finalmente devuelve un *string* que se corresponde con el CAN que ha sido aceptado por el DNIe; por lo que es un objetivo perfecto para la implementación de este ataque.

El código que incorpora esta funcionalidad se encuentra indicado en el siguiente bloque:

```
1 let smartCardClass = Java.use('es.gob.jmulticard.d.b.b');
2 smartCardClass.a.overload('[Ljava.lang.String;',
  ↪ 'de.tsenger.androsmex.d.d.g$d').implementation = function (a, b) {
3   let modifiedA = Java.array('java.lang.String', ['000001', '000002', ...]);
4   let ret = this.a(modifiedA, b);
5   console.log('[+]', ret);
6   return ret;
7 }
```

En primer lugar, se recupera una referencia a la clase `SmartCardMRTDConnection` utilizada por la librería. Se modifica el comportamiento de la función `a()` previamente descrita, cambiando enteramente el argumento proporcionado a la función por una lista de cadenas definidas por el atacante. Utilizando este nuevo parámetro, se ejecuta la función original, almacenando el valor de retorno. Este es mostrado por pantalla y devuelto como resultado, siguiendo el flujo de programa original.

La ejecución de este *script* proporcionando el CAN correcto en una lista de corta longitud, efectivamente, devuelve la clave correcta a los pocos segundos de acercar la tarjeta al lector; accediendo con éxito a los datos almacenados en el documento. Antes de escalar este ataque e incluir todos los valores posibles que este código puede tomar, se decidió medir el tiempo promedio que se tarda en ejecutar el protocolo PACE. Para ello, se empleó el código previamente desarrollado para la visualización de paquetes APDU: entre la información de depuración obtenida de la librería cuando se lee un DNIe se incluye la duración de cada intento de PACE, expresada en milisegundos. Tras la ejecución de varios intentos de autenticación utilizando un código CAN incorrecto, se estimó que cada uno de ellos consumía una media de 1200 ms en llevarse a cabo.

Por tanto, para realizar este ataque con un resultado exitoso haría falta, en el peor caso, de aproximadamente 14 días de contacto ininterrumpido con el DNIe. Este resultado concuerda con el obtenido en [34] y demuestra que la clave CAN, a pesar de poseer una baja entropía, es una medida adecuada para la protección del contenido almacenado en la aplicación de *ePassport* ante ataques de lectura no autorizada, donde el adversario

no posee conocimiento del código impreso en el documento. El uso de esta clave supone un equilibrio entre la seguridad y usabilidad del DNIe, puesto que el CAN es fácilmente legible por inspectores y usuarios legítimos que deseen hacer uso de la funcionalidad sin contacto de este documento.

3.7.3. Análisis del generador de números pseudoaleatorios

El protocolo PACE para la autenticación del lector utiliza secuencias de bits de un solo uso generadas pseudoaleatoriamente, con el fin de asegurar que cada uno de los pasos de la comunicación sea único; impidiendo así la realización de ataques de reproducción y de fuerza bruta offline. El estudio recogido en [34] realiza una investigación de la entropía del generador de números pseudoaleatorios incrustado en el DNIe, determinando que este sistema cumple con las recomendaciones de seguridad establecidas. En esta línea de trabajo, se busca replicar los resultados obtenidos por el equipo de investigadores de la Universidad de Zaragoza, desarrollando una implementación software propia basada en las técnicas discutidas en secciones previas de este documento.

En la primera etapa de la ejecución de PACE, el documento elige un valor utilizando un método de generación criptográfica de números pseudoaleatorios y lo cifra haciendo uso del secreto pre-compartido entre el lector y la tarjeta. Esta información cifrada es almacenada en un paquete ADPU de respuesta y enviada al dispositivo, encargado de descifrarla de vuelta a través del secreto convenido. El valor pseudoaleatorio es utilizado en posteriores pasos para determinar un conjunto de parámetros, empleado para el establecimiento de claves de sesión a través de la ejecución de la variante del protocolo Diffie-Hellman con curvas elípticas. Estas claves son a continuación validadas por ambos dispositivos y utilizadas para cifrar el resto de los paquetes enviados durante la sesión.

La extracción de números pseudoaleatorios puede convertirse en una labor tediosa si ha de esperarse a la finalización del protocolo PACE por cada valor obtenido, puesto que, como se ha discutido en anteriores secciones, este tarda una media de 1.2 segundos en completarse. El proceso de generación de claves efímeras consume una cantidad considerable de este tiempo, debido a los recursos limitados de cómputo que posee el chip Infineon del DNIe: en las pruebas realizadas a través de las herramientas desarrolladas en líneas de investigación paralelas, se estimó que este paso requiere, de media, 500 ms para su ejecución.

Puesto que el número pseudoaleatorio escogido por el chip es enviado en el primer paso de este protocolo, no existe la necesidad de efectuar ningún paso adicional: basta con repetir la ejecución de este primero hasta obtener la cantidad de valores deseada para realizar el análisis de entropía. Con este fin, se debe romper la ejecución PACE esperada y enviar continuamente el código de operación establecido para dar comienzo al mismo, descifrando y grabando las repuestas enviadas por el DNIe en un fichero para su posterior estudio. Si este procedimiento se sigue correctamente, el chip responderá a cada petición con un nuevo número pseudoaleatorio, evitando la ejecución de pasos innecesarios y maximizando la eficacia del mecanismo de extracción.

Utilizando los datos recabado en investigaciones anteriormente desarrolladas, se conoce que la función `a()` implementada en la clase `SmartCardMRTDConnection` es la encargada de llevar a cabo el protocolo PACE. Esta, a su vez, delega la responsabilidad de ejecutar

cada uno de los pasos de este procedimiento a la clase de `de.tsenger.androsmex.e.e`, donde se incluye el código específico encargado de solicitar al DNIe la generación de un número pseudoaleatorio.

Cuando Frida inyecta el *script* en el proceso objetivo, primero localiza y almacena una referencia a esta última clase:

```
1 var paceObj = null;
2 Java.perform(function() {
3   if (paceObj === null) {
4     Java.choose('de.tsenger.androsmex.e.e', {
5       onMatch: (instance) => paceObj = instance,
6       onComplete: () => {}
7     });
8   }
9   // ...
10 }
```

Cuando se acerca el documento al lector NFC del *smartphone*, se ejecuta la función `a()` de `SmartCardMRTDConnection`. Su funcionamiento es alterado, de tal forma que se fuerza el uso del CAN correcto para el DNIe utilizado en la investigación y se inicia un bucle para la ejecución del protocolo PACE. Este se ejecuta indefinidamente hasta que el programa anfitrión notifica al proceso hijo de que el usuario desea detener la operación de extracción.

```
1 let smartCardClass = Java.use('es.gob.jmulticard.d.b.b');
2 smartCardClass.a.overload('[Ljava.lang.String;',
3   ↪ 'de.tsenger.androsmex.d.d.g$d').implementation = function (a, b) {
4   let modifiedA = Java.array('java.lang.String', ['044032']);
5   let ret = null;
6   let condition = false;
7   do {
8     try {
9       ret = this.a(modifiedA, b);
10    } catch (e) {
11      send('{"type": "poll"}');
12      recv(function (outputJson) {
13        condition = outputJson.condition;
14      }).wait();
15    }
16  } while (condition);
17  return ret;
18 }
```

Siguiendo el flujo de ejecución legítimo, se llama a continuación a la función `de.tsenger.androsmex.e.e.a()`, encargada de enviar el paquete APDU de comando

correspondiente a la solicitud de un número pseudoaleatorio. Cuando este comando es recibido, es descriptado por la función de `de.tsenger.androsmex.c.a.a()`: esta es también interceptada a través de Frida, con el fin de recuperar los datos descriptados recibidos de la tarjeta.

```
1 let decryptClass = Java.use('de.tsenger.androsmex.c.a');
2 decryptClass.a.overload('[B', '[B').implementation = function (key, data) {
3     send('{"type": "key"}', new Uint8Array(key));
4     send('{"type": "encNonce"}', new Uint8Array(data));
5     let ret = this.a(key, data);
6     send('{"type": "decNonce"}', new Uint8Array(ret));
7     if (paceObj === null) {
8         throw new Error("Couldn't find de.tsenger.androsmex.e.e");
9     }
10    paceObj.a(0, Java.array('byte', []));
11 }
```

Notablemente, tras recuperar el número pseudoaleatorio descifrado se invoca nuevamente a la función encargada de solicitar este dato al chip del DNIE. Un paquete con un código de error es devuelto por la tarjeta, puesto que solo se permite la generación de un número pseudoaleatorio por cada ejecución del protocolo PACE. Este error genera una excepción en el código, atrapada por el bloque *catch* de la clase `SmartCardMRTDConnection` enganchada. La funcionalidad modificada de este método ignora el error y fuerza un nuevo comienzo del protocolo, solicitando nuevamente un número pseudoaleatorio. Puesto que el DNIE se encontraba en un estado de error que impedía la continuación de la operación en ejecución, acepta esta nueva petición y proporciona el dato solicitado; devolviendo un nuevo valor y dando comienzo a una nueva iteración de este proceso.

Como se mencionaba anteriormente, este procedimiento utiliza nuevamente un *script* anfitrión desarrollado en Python, conectado con la instancia del servidor de Frida ejecutada en el *smartphone*. Similarmente a como se realizó en la aplicación para la visualización de paquetes APDU transmitidos, la función de este *script* es la de mostrar por pantalla los datos recolectados y la de almacenar los números pseudoaleatorios descifrados en un fichero (ver Figura 3.7).

Este método fue empleado para la recolección de 10^5 números pseudoaleatorios devueltos por el DNIE. Estos, fueron posteriormente analizados a través de utilidades criptográficas para determinar el nivel de entropía del generador de números pseudoaleatorios del documento y su resiliencia ante las pruebas definidas en FIPS 140-2. En este estudio, se utilizaron las siguientes herramientas:

- ent [50]:

```
Entropy = 7.999888 bits per byte.
```

```
Optimum compression would reduce the size
of this 1892288 byte file by 0 percent.
```

```

[H] Attaching to USB device
[H] Attaching to target process: Sample_Dnie_App
[H] Loading JS Frida module
[+] Frida hooked up!
[*] Key:
00000000: FE 0C 74 BE A6 10 50 61 B0 55 D8 26 18 62 8D FC ..t...Pa.U.&.b..
[*] Encrypted nonce:
00000000: 06 27 7A BC D5 B0 60 10 A0 BA 31 C0 74 71 FF 7B .'z...'...1.tq.{
[*] Decrypted nonce:
00000000: 74 88 80 3F C5 D3 A5 93 A4 C2 6A 37 7C 8C CF D2 t.?......j7|...
-----
[*] Key:
00000000: FE 0C 74 BE A6 10 50 61 B0 55 D8 26 18 62 8D FC ..t...Pa.U.&.b..
[*] Encrypted nonce:
00000000: 43 BA C6 45 8A AE AC DD 7E 32 5A 6F C7 27 5B BA C..E....~2Zo.'[.
[*] Decrypted nonce:
00000000: B3 DD E9 60 A2 34 F3 36 EC B7 4B 27 F9 E8 26 9B ...`.4.6..K'..&.
-----
[*] Key:
00000000: FE 0C 74 BE A6 10 50 61 B0 55 D8 26 18 62 8D FC ..t...Pa.U.&.b..
[*] Encrypted nonce:
00000000: 33 AC 06 6E 54 85 EA 98 5D 4C 8A 2D C2 D2 4A BD 3..nT...]L.-..J.
[*] Decrypted nonce:
00000000: 8E 61 55 D3 39 67 E7 3B 75 1A 01 26 DB 78 EA C5 .aU.9g.;u..&.x..
-----
[*] Key:
00000000: FE 0C 74 BE A6 10 50 61 B0 55 D8 26 18 62 8D FC ..t...Pa.U.&.b..
[*] Encrypted nonce:
00000000: 9A 76 64 2E EE D2 2A 46 64 F1 78 9F FF AA 4C FC .vd...*Fd.x...L.
[*] Decrypted nonce:
00000000: A3 ED 22 13 6B CF 6D 8F 97 39 5A 8C CE 4F AC DC ..".k.m..9Z..0..
-----
[*] Key:

```

Figura 3.7: Números pseudoaleatorios extraídos del protocolo PACE

Chi square distribution for 1892288 samples is 293.76, and randomly would exceed this value 4.79 percent of the times.

Arithmetic mean value of data bytes is 127.3730 (127.5 = random).

Monte Carlo value for Pi is 3.144044822 (error 0.08 percent).

Serial correlation coefficient is -0.000454 (totally uncorrelated = 0.0).

- rngtest [51]:

```

rngtest: starting FIPS tests...
rngtest: entropy source exhausted!
rngtest: bits received from input: 15138304
rngtest: FIPS 140-2 successes: 756
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.433; avg=7.044; max=9.313)Gibits/s
rngtest: FIPS tests speed: (min=73.078; avg=97.815; max=126.314)Mibits/s
rngtest: Program run time: 151896 microseconds

```

Los resultados obtenidos con ambas herramientas coinciden con los obtenidos en el estudio de referencia, verificándose así que la generación criptográfica de números pseudoaleatorios en el DNIE es buena y cumple con los requisitos establecidos en los estándares de seguridad actuales.

3.7.4. Ataque de retransmisión

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 anteriores secciones. 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 [52] bajo el sobrenombre de “fraude de la mafia” y en [53] 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, incumpléndose así la restricción de distancia teórica máxima para la lectura de tarjetas NFC (ver Figura 3.8).

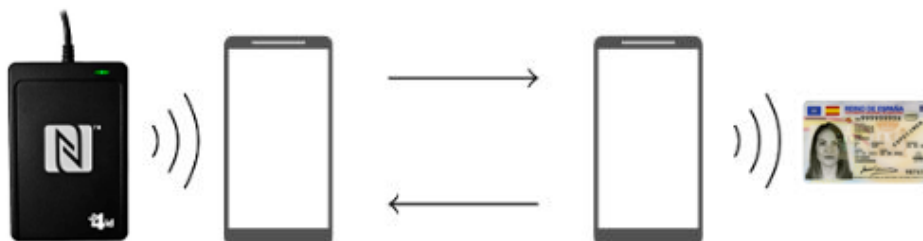


Figura 3.8: Representación gráfica de un ataque de *relay*

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* [54], 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 *rootado* 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 [55], 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 [56].

Capítulo 4

Conclusiones y líneas futuras

En este trabajo se ha presentado un estudio exploratorio de la seguridad de tarjetas de lectura sin contacto que incorporan una interfaz NFC.

Por una parte, a través de las distintas investigaciones teóricas y prácticas recogidas y realizadas en este estudio se ha demostrado que el uso de la tecnología propietaria de MIFARE Classic en tarjetas para el control de acceso es inseguro. Si bien el uso de protocolos propietarios en tarjetas RFID es una práctica común, la implementación de algoritmos criptográficos propios, que a su vez se basa en el principio de seguridad por oscuridad, supone un grave problema en este campo de aplicación. A pesar de que las debilidades de estas tarjetas son conocidas desde hace más de una década, las etiquetas MIFARE Classic siguen siendo empleadas en multitudes de escenarios donde prima la seguridad, tal como se ha demostrado a través de varios casos prácticos de estudios descritos en este trabajo.

Por otra parte, se ha documentado el funcionamiento del DNIE y de los mecanismos de seguridad incorporados para su protección, y se ha confirmado su alto nivel de resiliencia. En particular, la inclusión de una interfaz NFC desde la versión 3.0 de este documento, aunque ofrece una mejor experiencia de uso, expone al DNIE a la posible realización de ataques de retransmisión. Sin embargo, tal como se ha comprobado con diversas pruebas, esa potencial vulnerabilidad está compensada con medidas de seguridad requeridas para el acceso a la información almacenada. Las herramientas desarrolladas y metodología seguida en este trabajo han sido descritas con detalle en este documento de forma que podrán ser reutilizadas para la realización de nuevas investigaciones de exploración de la seguridad del DNIE a través de nuevas perspectivas, abriendo la posibilidad para el desarrollo de nuevas líneas de trabajo en esta área.

En conclusión, la seguridad de las tarjetas NFC depende enteramente de los mecanismos de protección que incluyan para la prevención de posibles ataques. La interfaz NFC de esos documentos implica algunos problemas inherentes a este protocolo, como son la posibilidad de realizar escuchas a escondidas o la exposición ante ataques de corrupción de datos o de retransmisión. Estas debilidades deben ser enfrentadas con las debidas medidas de seguridad integradas en esas tarjetas como, por ejemplo, el cifrado robusto de la comunicación entre el lector y la etiqueta, así como protocolos criptográficos seguros para el establecimiento de claves y la autenticación y control de acceso.

Como ya ha ocurrido muchas veces en el pasado, los principales problemas de ciberseguridad surgen cuando los mecanismos de seguridad no son correctamente validados

antes de la producción en masa de tecnologías, como se resalta en la comparación entre documentos tan distintos como son las etiquetas MIFARE Classic y el DNIe. Antes de realizar el despliegue de una nueva tecnología, es necesario probar y certificar su resiliencia, siguiendo el principio de seguridad por diseño (*security by design*), especialmente, si se trata de un recurso utilizado de forma masiva en escenarios donde la seguridad es un requisito clave.

Capítulo 5

Conclusions and future works

This work has presented an exploratory study of the security of contactless cards incorporating an NFC interface.

On the one hand, through the different theoretical and practical research collected and carried out in this study, it has been shown that the use of proprietary MIFARE Classic technology in cards for access control has proven to be insecure. Although the use of proprietary protocols in RFID cards is a common practice, the implementation of proprietary cryptographic algorithms, which in turn is based on the principle of security by obscurity, is a serious problem in this field of application. Even though the weaknesses of these cards have been known for more than a decade, MIFARE Classic tags are still used in a multitude of security-critical scenarios, as demonstrated by the described case studies analyzed in this work.

On the other hand, the operation of the Spanish eID and the built-in security mechanisms for its protection have been documented, as its high level of resilience has been confirmed. In particular, the inclusion of an NFC interface from version 3.0 of this document, despite offering a better user experience, exposes the eID to possible relay attacks. However, as various tests have shown, this potential vulnerability is offset by the security measures required to access the stored information. The tools developed and work methodology used in this study have been described in detail in this document so that they can be reused for new research that further explores the security of the eID from new perspectives, opening up the possibility for the development of new lines of work in this area.

In conclusion, the security of NFC cards depends entirely on the protection mechanisms they include to prevent possible attacks. The NFC interface of these documents involves some inherent problems, such as the possibility of eavesdropping or exposure to data corruption or retransmission attacks. These weaknesses must be addressed with appropriate security measures built into these cards, such as strong encryption of the communication between the reader and the tag, as well as secure cryptographic protocols for key establishment and authentication and access control.

As has happened many times in the past, the main cybersecurity problems arise when security mechanisms are not properly validated before mass production of technologies, as highlighted by the comparison between documents as different as MIFARE Classic tags and the Spanish National Identity Card. Before deploying a new technology, it is necessary to test and certify its resilience, following the principle of security by design, especially if it is a resource used massively in scenarios where security is a key requirement.

Capítulo 6

Presupuesto

En esta sección se realiza una propuesta del presupuesto estimado para el desarrollo de este Trabajo de Fin de Grado, diferenciando los costes en materia de personal y componentes empleados para su realización.

6.1. Costes de personal

En la Tabla 6.1 se presenta el coste estimado del personal requerido para el desarrollo del estudio presentado a lo largo de este trabajo, asumiendo un pago de 12€ por cada hora empleada:

Tarea	Horas	Coste
Documentación del funcionamiento de NFC	20	240€
Documentación sobre vulnerabilidades NFC	10	120€
Documentación sobre el funcionamiento del DNIE	60	720€
Documentación de las medidas de seguridad del DNIE	60	720€
Investigación de la seguridad de tarjetas de acceso	10	120€
Aprendizaje del uso de la herramienta Frida	10	120€
Ingeniería inversa de la librería DNIEdroid	60	900€
Investigación de la seguridad del DNIE	50	600€
Recolección de resultados y redacción de la memoria	70	840€
TOTAL	350	4380€

Tabla 6.1: Presupuesto estimado de personal

6.2. Costes de componentes

En la Tabla 6.2 se presenta un desglose del costo estimado de los recursos y componentes utilizados durante la investigación práctica desarrollada a lo largo de este trabajo:

Componente	Unidades	Coste
Equipo de desarrollo	1	1000€
Teléfono Android con funcionalidad NFC	1	500€
Proxmark3 Easy	1	50€
MIFARE Classic 1K con puerta trasera	3	5€
Pulsera de acceso NFC	1	7€
Primo Smart Card ID Reader	1	10€
DNIE 4.0	1	150€
TOTAL		1732€

Tabla 6.2: Presupuesto estimado de componentes

6.3. Coste total

En la Tabla 6.3 se recoge el costo total estimado del proyecto:

Presupuesto	Coste
Personal	4380€
Componentes	1732€
TOTAL	6112€

Tabla 6.3: Presupuesto estimado del trabajo

Apéndice A

Artículos enviados a conferencias

A.1. 20th International Conference on Security and Management SAM (aceptado)

Study and security analysis of the Spanish identity card.

Javier Correa-Marichal, Pino Caballero-Gil, Carlos Rosa-Remedios, Rames Sarwat-Shaker.

Proceedings of the 20th International Conference on Security and Management SAM (within the World Congress in Computer Science, Computer Engineering, and Applied Computing CSCE).

Las Vegas, USA. July 25-28, 2022.

Springer Nature.

Indexada en Computing Research and Education (CORE), con ranking C.

Indexada en CS Conference Rankings (0.83).

Indexada en GII-GRIN en Class WiP.

A.2. XVII Reunión Española sobre Criptología y Seguridad de la Información RECSI (enviado)

Un estudio del DNIE y de su infraestructura.

Javier Correa-Marichal, Pino Caballero-Gil, Carlos Rosa-Remedios, Rames Sarwat-Shaker.

XVII Reunión Española sobre Criptología y Seguridad de la Información RECSI.

Santander, España. 19-21 Octubre 2022.

Bibliografía

- [1] "Access Control Cards - Ways to Use ID Cards", AlphaCard. [Online]. Available: <https://www.alphacard.com/learning-center/ways-to-use-your-id-cards/access-control/>. [Accessed: 29-Mar-2022].
- [2] "Tarjetas de proximidad para Control de Accesos", Prevent. [Online]. Available: <https://www.prevent.es/servicios-de-seguridad/control-de-accesos/tarjetas-de-proximidad-para-control-de-accesos>. [Accessed: 29-Mar-2022].
- [3] NXP Semiconductors, "The Value of Genuine MIFARE® Products", 2016.
- [4] K. Nohl, D. Evans, S. Starbug and H. Plötz, "Reverse-Engineering a Cryptographic RFID Tag", USENIX Security Symposium, vol. 28, 2008.
- [5] F. Garcia et al., "Dismantling MIFARE Classic", Computer Security - ESORICS 2008, pp. 97-114, 2008. Available: 10.1007/978-3-540-88313-5_7.
- [6] F. Garcia, P. van Rossum, R. Verdult and R. Schreur, "Wirelessly Pickpocketing a Mifare Classic Card", 2009 30th IEEE Symposium on Security and Privacy, 2009. Available: 10.1109/sp.2009.6.
- [7] M. Roland, "Security Issues in Mobile NFC Devices", T-Labs Series in Telecommunication Services, 2015. Available: 10.1007/978-3-319-15488-6.
- [8] "Security Statement on Crypto1 Implementations", MIFARE, 2015. [Online]. Available: <https://www.mifare.net/en/products/chip-card-ics/mifare-classic/security-statement-on-crypto1-implementations/>. [Accessed: 02-Apr-2022].
- [9] C. Meijer and R. Verdult, "Ciphertext-only Cryptanalysis on Hardened Mifare Classic Cards", Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015. Available: 10.1145/2810103.2813641.
- [10] Jirvin, "The Mifare Classic Chipset Notes", Dangerous Things Forum, 2021. [Online]. Available: <https://forum.dangerousthings.com/t/the-mifare-classic-chipset-notes-wip/12339>. [Accessed: 02-Apr-2022].
- [11] Jirvin, "magic Mifare chips", Dangerous Things Forum, 2020. [Online]. Available: <https://forum.dangerousthings.com/t/magic-mifare-chips/6696>. [Accessed: 02-Apr-2022].
- [12] "The Iceman fork of Proxmark3", GitHub, 2022. [Online]. Available: <https://github.com/RfidResearchGroup/proxmark3>. [Accessed: 03-Apr-2022].
- [13] "Tarjeta universitaria", Universidad de La Laguna, 2022. [Online]. Available: <https://www.ull.es/vive-la-ull/tarjeta-universitaria/>. [Accessed: 03-Apr-2022].

- [14] "Concepto y validez", Ministerio del Interior. [Online]. Available: <http://www.interior.gob.es/web/servicios-al-ciudadano/dni/concepto-y-validez/>. [Accessed: 08-Apr-2022].
- [15] Anuario estadístico del ministerio del interior 2020. 2021, p. 547.
- [16] 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.
- [17] "Qué son las Apps para móviles", Portal del DNI Electrónico, 2022. [Online]. Available: https://www.dnielectronico.es/portaldnie/PRF1_Cons02.action?pag=REF_033. [Accessed: 09-Apr-2022].
- [18] A. Pascual, "El DNI electrónico ha muerto: ilarga vida al DNI 3.0!", El Confidencial, 2013.
- [19] NUEVO DNIE 4.0 – FORMATO EUROPEO. Policía Nacional, 2021, p. 1.
- [20] 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].
- [21] "Travel documents for EU nationals", Your Europe, 2022. [Online]. Available: https://europa.eu/youreurope/citizens/travel/entry-exit/eu-citizen/index_en.htm. [Accessed: 02-Jun-2022].
- [22] Doc 9303 Part 1: Machine Readable Travel Documents, 8th ed. ICAO, 2021.
- [23] "REGLAMENTO (UE) No 910/2014 DEL PARLAMENTO EUROPEO Y DEL CONSEJO", Diario Oficial de la Unión Europea, 2022. [Online]. Available: <https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:32014R0910&from=EN>. [Accessed: 02-Jun-2022].
- [24] "Declaración de prácticas y políticas de certificación v2.7", Cuerpo Nacional de Policía, 2019, p. 17.
- [25] "UNE-EN 419211-1:2016", AENOR, 2016.
- [26] E. Taborda and V. Ramírez, "24 hours Center and new Spanish eID Document 4.0."
- [27] "Descripción del Chip DNIE 3.0", Portal del DNI Electrónico. [Online]. Available: https://www.dnielectronico.es/PortalDNIE/PRF1_Cons02.action?pag=REF_1078. [Accessed: 10-Apr-2022].
- [28] "ISO/IEC 7816-2:2007", ISO, 2007. [Online]. Available: <https://www.iso.org/standard/45989.html>. [Accessed: 10-Apr-2022].
- [29] "ISO/IEC 14443-3:2011", ISO, 2011. [Online]. Available: <https://www.iso.org/standard/50942.html>. [Accessed: 10-Apr-2022].
- [30] "Logical Data Structure (LDS) for Storage of Data in the Contactless IC", ICAO, 2018.

- [31] TR-03110 Part 4: Applications and Document Profiles, 2nd ed. Federal Office for Information Security, 2016, p. 18.
- [32] 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.
- [33] Doc 9303 Part 11: Security Mechanisms for MRTDs, 8th ed. ICAO, 2021.
- [34] 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. Available: 10.1049/iet-ifs.2017.0299.
- [35] G. Avoine, A. Beaujeant, J. Hernandez-Castro, L. Demay and P. Teuwen, "A Survey of Security and Privacy Issues in ePassports", *ACM Computing Surveys*, vol. V, no. N, 2015. [Accessed: 11-April-2022].
- [36] TR-03110 Part 2: Protocols for electronic IDentification, Authentication and trust Services (eIDAS), 2nd ed. Federal Office for Information Security, 2016.
- [37] M. Nemeč, M. Sys, P. Svenda, D. Klinec and V. Matyas, "The Return of Coppersmith's Attack", *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. Available: 10.1145/3133956.3133969.
- [38] "CVE-2017-15361 Detail", CVE, 2017. [Online]. Available: <https://www.cve.org/CVERecord?id=CVE-2017-15361>. [Accessed: 02-Jun-2022].
- [39] "ROCA vulnerability", Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/ROCA_vulnerability. [Accessed: 02-Jun-2022].
- [40] F. Acero, "El DNIE, la vulnerabilidad ROCA y los problemas heredados", *El Radar*, 2020. [Online]. Available: <https://www.elradar.es/el-dnie-la-vulnerabilidad-roca-y-los-problemas-heredados/>. [Accessed: 02-Jun-2022].
- [41] A. Parsovs, "Estonian electronic identity card and its security challenges", *DSPace*, 2021. [Online]. Available: <https://dSPACE.ut.ee/handle/10062/71481>. [Accessed: 02-Jun-2022].
- [42] Centro Criptológico Nacional, "CCN-STIC 807 - Criptología de empleo en el Esquema Nacional de Seguridad", 2022.
- [43] J. Sojo, "CCOO alerta de un error que invalidaría los DNI expedidos entre 2015 y 2018", *El Confidencial*, 2020. [Online]. Available: https://www.elconfidencial.com/espana/2020-07-15/ccoo-error-dni-certificados-digitales-falta-personal_2681807/. [Accessed: 02-Jun-2022].
- [44] "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: 12-Apr-2022].
- [45] "API hooking", Infosec Resources, 2014. [Online]. Available: <https://resources.infosecinstitute.com/topic/api-hooking/>. [Accessed: 03-Jun-2022].
- [46] "Frida • A world-class dynamic instrumentation framework", Frida. [Online]. Available: <https://frida.re/>. [Accessed: 12-Apr-2022].

- [47] "JADX", GitHub. [Online]. Available: <https://github.com/skyloot/jadx>. [Accessed: 04-Jun-2022].
- [48] "Package javax.smartcardio", Docs Oracle, 2022. [Online]. Available: <https://docs.oracle.com/javase/7/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html>. [Accessed: 04-Jun-2022].
- [49] "hexdump", PyPI, 2016. [Online]. Available: <https://pypi.org/project/hexdump/>. [Accessed: 04-Jun-2022].
- [50] J. Walker, "Pseudorandom Number Sequence Test Program", Fourmilab. [Online]. Available: <https://www.fourmilab.ch/random/>. [Accessed: 02-Jun-2022].
- [51] "The rng-tools official repository", GitHub. [Online]. Available: <https://github.com/nhorman/rng-tools>. [Accessed: 02-Jun-2022].
- [52] C. Pomerance, *Advances in cryptology, CRYPTO'87*. Berlin: Springer-Verlag, 1988, pp. 21-39.
- [53] 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. [Accessed: 12-April-2022].
- [54] "GitHub - nfcgate/nfcgate: An NFC research toolkit application for Android", GitHub, 2022. [Online]. Available: <https://github.com/nfcgate/nfcgate>. [Accessed: 12 Apr-2022].
- [55] A. Lotfi, "Could your contactless bank card be vulnerable to virtual pickpocketing?", *The Conversation*, 2016. [Online]. Available: <https://theconversation.com/could-your-contactless-bank-card-be-vulnerable-to-virtual-pickpocketing-55264>. [Accessed: 02-Jun-2022].
- [56] S. Drimer and S. J. Murdoch, "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks", *USENIX security symposium*, Vol. 312, 2007.