Trabajo Fin de Grado

Ataques de retransmisión inteligente en protocolos de pago NFC

Intelligent relay attacks on NFC payments protocols

Departamento de Informática e Ingeniería de Sistemas

Grado en Ingeniería Informática

Autor: Guillermo Cebollero Abadías

Director: Ricardo J. Rodríguez

Ponente: José Merseguer Hernáiz

UNIZAR

Septiembre del 2018

Ataques de retransmisión inteligente en protocolos de pago NFC

RESUMEN

La inclusión de nuevas medidas de seguridad en las tarjetas de pago a crédito o a débito ha mejorado notablemente la seguridad con la que se realizan estas transacciones. De realizar operaciones usando la banda magnética de las tarjetas (siendo una tecnología muy insegura) se pasó a usar un chip electrónico, con las llamadas tarjetas EMV o tarjetas *chip-and-PIN* (tecnología algo más segura). Estas tarjetas implementan el protocolo EMV (de donde toman su nombre), que permite autorizar las transacciones realizadas con la tarjeta mediante la introducción de un código numérico personal (PIN) por parte del titular de la tarjeta. Dichas tarjetas EMV han sido recientemente actualizadas con la adición de un chip que permite la comunicación inalámbrica. Estas nuevas tarjetas, denominadas tarjetas *contactless*, permiten realizar pagos aproximando únicamente la tarjeta al punto de venta, es decir, sin necesidad de insertarlas en un lector de chip. Además, si la cuantía del pago no supera cierto límite se elimina la obligatoriedad de autorizar la transacción mediante un PIN.

La comunicación inalámbrica de las tarjetas *contactless* se basa en el protocolo de comunicación en el campo cercano o Near Field Communication (NFC), que usa la banda de alta frecuencia de Radio Frequency IDentification (RFID). Debido a que NFC no introduce seguridad ni cifrado en la realización de las comunicaciones, la adopción de esta tecnología en las tarjetas permite investigar nuevas formas de atacar a las transacciones realizadas. Por ejemplo, pueden realizarse ataques *eavesdropping* realizando escuchas en las perturbaciones del campo electromagnético, así como ataques *man-in-the-middle* permitiendo la inserción, manipulación o corrupción de los datos. Este proyecto se centra en los riesgos introducidos relativos a los ataques de retransmisión, permitiendo realizar una comunicación entre una tarjeta de crédito o débito y un punto de venta malicioso separados por una distancia arbitrariamente grande.

Paralelamente a la realización de estos experimentos, se detallan las mejoras introducidas a la aplicación Android `NFCLeech`. Esta aplicación es un prototipo inicialmente desarrollado en 2014 para probar los ataques de retransmisión y que ahora permite realizar las pruebas de concepto relativas a los ataques de retransmisión extendidos en este trabajo, así como una mejor depuración de la retransmisión de los mensajes y el protocolo utilizado para realizar los pagos.

Índice general

1. Introducción **1**
1.1. Motivación 1
1.2. Objetivo 2
1.3. Trabajos relacionados 2
1.4. Organización 3

2. Contexto **5**
2.1. Conexiones inalámbricas: Bluetooth y NFC 5
2.2. Normas ISO/IEC 14443 e ISO/IEC 7816 7
2.3. UML 8
2.4. EMV y tarjetas de crédito/débito 8

3. Ataque de retransmisión inteligente **11**
3.1. Detección de tramas a retransmitir 12
3.2. Protocolo de interacción 14

4. Estrategias para retraso de comunicaciones en NFC **17**
4.1. Trama WTX 18
4.2. Trama NAK 19
4.3. Cacheo de tramas 20

5. Implementación: Extensión de `NFCLeech` **21**
5.1. Estado inicial de `NFCLeech` 21
5.2. Mejoras introducidas 22
5.2.1. Sistema de registro 22
5.2.2. Cacheo de mensajes 23
5.2.3. Mejoras de estabilidad 24
5.2.4. Refactorización y aplicación de técnicas de desarrollo del software 26
5.2.5. Retransmisión Bluetooth 29

6. Experimentos y discusión **31**
6.1. Evaluación de tiempos 31
6.2. Otros escenarios de ataques: Eavesdropping, DoS 33

7. Conclusiones y trabajo futuro **35**
7.1. Trabajo futuro 36

Bibliografía **36**

A. Extensión temporal del proyecto **41**

B. Traza de ejecución **43**

Glosario **47**

Índice de figuras

2.1. Configuración del PCD y PICC [9, página 13] 7
2.2. Comando APDU 8
2.3. Respuesta de un comando APDU 8

3.1. Ataque de retransmisión en sistemas de pago *contactless*, extraída de [22] 12
3.2. Diagrama de secuencia con PayWave (VISA) [4] 13
3.3. Diagrama de secuencia con PayPass (MASTERCARD) [4] 13
3.4. Diagrama de secuencia de la comunicación en un *relay* inteligente 15

4.1. Bloque de transmisión de protocolo 18

5.1. Ejemplo de interacción con VISA 23
5.2. Ejemplo de interacción con MASTERCARD 23
5.3. Diagrama de secuencia de la detección de AIDs y cacheado 25
5.4. Diagrama de clases inicial para la implementación de la comunicación. . . 27
5.5. Diagrama de clases final para la implementación de la comunicación. . . . 28

A.1. Diagrama de Gantt. 41

Índice de tablas

4.1. Codificación de los bits b8-b7 del Protocol Control Byte (PCB) 18
4.2. Codificación de un bloque tipo S 18
4.3. Codificación del bloque de información para el Waiting Time eXtension (WTX) 19
4.4. Codificación del WTX antes del Cyclic Redundancy Check (CRC) 19
4.5. Codificación de un bloque tipo R 20
4.6. Codificación del Not AcKnowledge (NAK) 20

6.1. Tiempos con la versión inicial mediante WiFi 32
6.2. Tiempos tras la implementación del cacheado mediante WiFi 32
6.3. Tiempos tras la implementación del cacheado mediante Bluetooth 33

Capítulo 1

Introducción

1.1. Motivación

Actualmente, el pago mediante tarjetas de crédito o débito se está convirtiendo en el método primario de pago debido a la comodidad y la seguridad que ofrece al usuario. Aunque aumentan los pagos con tarjetas, todavía no son el mayor método de pago usado [5]. Aún así, países como Dinamarca, Suecia o Finlandia están fomentando que el pago mediante tarjeta se convierta en el único método de pago, reduciendo así los costes asociados para el Estado de poner una moneda física en circulación y dificultando además el blanqueo de capitales (dado que quedaría registro de toda transacción realizada).

Tradicionalmente, los pagos se realizaban gracias a la información que se leía de la banda magnética presente en la parte posterior de las tarjetas. Aunque este método de pago era cómodo para el usuario, esta información se podía copiar fácilmente. Buscando añadir más seguridad, se incorporó un chip que contiene toda la información necesaria para realizar el pago y que obliga a introducir un código numérico personal para autorizar las transacciones. Estas tarjetas de chip (llamadas tarjetas *chip-and-PIN* o tarjetas EMV por su principal promotor) supusieron un gran avance en cuanto a mecanismos de seguridad para el usuario. Sin embargo, diversos tipos de vulnerabilidades y ataques se han ido haciendo públicos a lo largo de los años [2, 11, 23, 26]. Recientemente se han puesto en circulación un nuevo tipo de tarjetas que permiten realizar los pagos a través de NFC, conocidas como tarjetas sin contacto o *contactless*. Para ello, utilizan la banda de alta frecuencia de RFID estableciendo una comunicación inalámbrica entre el punto de venta y la tarjeta simplemente aproximando la tarjeta al terminal.

Según un informe publicado por el Consejo Económico y Social de España [5], el 90,8 % de las personas ha realizado alguna vez un pago con tarjeta de débito y cerca del 15,2 % ha pagado con tarjetas *contactless*. Aunque la cuota de mercado es todavía relativamente baja, el 52,4 % del público encuestado conoce el método de pago *contactless*. Dada la facilidad y rapidez de pago en los comercios mediante estas tarjetas, se prevé un aumento de esta cuota en los años venideros.

La creciente cuota de mercado de las tarjetas de pago sin contacto abre nuevas preguntas sobre la seguridad de estos pagos, debido a que el ya desfasado modelo clásico

del robo de crédito basado en el clonado o robo de tarjetas, ahora queda relegado a un segundo lugar. Las tarjetas *contactless* abren nuevas vías de ataque, como realizar pagos no autorizados a distancia (siendo especialmente peligrosas en zonas con un gran flujo de personas como podría ser el transporte público), el robo de información de la tarjeta (como puede ser el número de la tarjeta y la fecha de caducidad) o captar la información generada al realizar un pago de forma remota.

Además, actualmente existen más de 400 teléfonos móviles [27] que incorporan la tecnología NFC y la mayoría de los teléfonos nuevos que salen a mercado también son compatibles. Este aumento de dispositivos compatibles hace que su uso y acceso a esta tecnología esté en continua expansión. El creciente uso de NFC y las tarjetas *contactless* ha permitido que aparezcan nuevos riesgos de seguridad, permitiendo realizar ataques como *eavesdropping* (realizar escuchas en las perturbaciones del campo electromagnético), ataques *man-in-the-middle* (insertar, manipular o corromper parte de la comunicación) o ataques de retransmisión como el desarrollado en este proyecto.

1.2. Objetivo

Para la realización de este trabajo, se han marcado los siguientes tres objetivos:

El primer objetivo consiste en verificar si es posible realizar una transacción correcta sin necesitar retransmitir todos los mensajes entre la tarjeta y el punto de venta. De este modo, se puede reducir así el tiempo necesario para completar el pago, evitando que el punto de venta rechace la transacción por requerir más tiempo del permitido. Para ello, se ha utilizado la aplicación Android `NFCLeech` [3] como base inicial para la implementación de dicho sistema de retransmisión.

En segundo lugar, investigar si es posible aumentar el tiempo máximo de transacción permitido por el protocolo EUROPAY, MASTERCARD y VISA (EMV), mediante la introducción de mensajes artificiales en el canal de comunicación usando diferentes estrategias.

En tercer y último lugar, hacer más estable la aplicación `NFCLeech` y aumentar la viabilidad del ataque en un entorno real, reduciendo la cantidad de pasos necesarios para configurar el ataque, así como añadirle nuevas características que permitan convertirla en una herramienta para depurar cualquier comunicación a través de NFC en dispositivos móviles. Para ello, se ha mejorado la información mostrada por el sistema de registro actual de los mensajes así como la posibilidad de utilizar diferentes sistemas de registro.

1.3. Trabajos relacionados

Aunque los ataques de retransmisión han sido estudiados en numerosos artículos (como [7, 12, 24, 26], por citar algunos), es difícil encontrar trabajos relacionados con la retransmisión inteligente en el protocolo de pago EMV contactless que proporcionen alguna prueba de concepto fácilmente utilizable.

La posibilidad de retrasar la comunicación usando mensajes propios del protocolo se describe inicialmente en [20], detallando también un ataque de retransmisión (en parte)

inteligente. Sin embargo, la implementación realizada en dicho trabajo está basada en hardware específico y personalizado, lo que no permite generalizar conclusiones a entornos como los usados en este trabajo (dispositivos Android sin modificar). La estrategia seguida finalmente en este trabajo para retrasar la comunicación, consistente en cachear los mensajes, también ha sido estudiada en la literatura. En concreto, la distinción entre los mensajes susceptibles de ser previamente cacheados fue introducida en [4]. Sin embargo, no se proporcionó una prueba de concepto funcional de la misma.

A diferencia de los trabajos anteriores, en este trabajo se proporciona una prueba de concepto de los ataques de retransmisión inteligente y se experimenta sobre la posibilidad de retrasar la comunicación usando hardware convencional y al alcance de cualquiera (en concreto, dispositivos Android sin modificar). La existencia de una herramienta como prueba de concepto de estos ataques utilizando únicamente dispositivos móviles facilita realizar estudios de costes de retransmisión de una transacción usando diferentes canales de comunicación, conociendo así la viabilidad real de estos ataques para proporcionar mecanismos de defensa adecuados.

Por cuestiones éticas, la herramienta desarrollada en este trabajo no se ha liberado a la comunidad, siendo proporcionada únicamente a los investigadores de la academia o profesionales interesados bajo petición motivada.

1.4. Organización

Esta memoria se ha estructurado de la siguiente manera. En el Capítulo 2 se introducen las tecnologías, normativas y conceptos necesarios para la realización de este proyecto. A continuación, en el Capítulo 3 se detalla el concepto de retransmisión inteligente, detallando cómo se ha realizado la prueba de concepto así como la interacción que realizan las tarjetas de crédito y el punto de venta para llevar a cabo una transacción. En el Capítulo 4 se detallan las estrategias seguidas para intentar reducir el tiempo de retransmisión, así como los problemas encontrados. Después, en el Capítulo 5 se habla del estado inicial de la aplicación `NFCLeech` y se detallan los cambios, mejoras y correcciones aplicadas. A lo largo del Capítulo 6 se presentan los resultados obtenidos así como otros posibles escenarios de ataque. Para finalizar, en el Capítulo 7 se comenta una breve conclusión del proyecto así como futuras vías de investigación que pueden realizarse y su interés.

Adicionalmente, se incluyen como apéndices el diagrama de Gantt, una traza de la ejecución y un glosario con los términos y abreviaciones utilizadas a lo largo del proyecto para facilitar la lectura y comprensión del documento.

Capítulo 2

Contexto

En esta sección, se introducen brevemente los conceptos, normativas y protocolos utilizados durante la realización de este trabajo. En primer lugar, se detallan las tecnologías utilizadas para realizar la comunicación de forma inalámbrica. Luego, se describen el conjunto de normativas utilizadas a lo largo de toda la realización del proyecto, así como una breve introducción al lenguaje de modelado utilizado para la generación de los diagramas. Para finalizar, se describirá cómo son y qué características tienen las tarjetas de crédito/débito en la actualidad.

2.1. Conexiones inalámbricas: Bluetooth y NFC

Bluetooth

Bluetooth fue inventado por *Ericsson* en 1994 como un intento de unificar las comunicaciones inalámbricas entre ordenadores y dispositivos móviles. Como curiosidad, su nombre se debe al rey danés *Harald Blåtand*, traducido al inglés como *Harald Bluetooth* [25], quien unificó las tribus noruegas y sueco-danesas en el siglo X.

Bluetooth es un estándar de comunicación inalámbrica, dentro de la banda de *Ultra Alta Frecuencia*, que permite el intercambio de información en el rango de los 2.4 a los 2.485 Ghz, usando para ello la banda de uso libre Industrial, Científica y Médica (ISM), por lo que no es necesario disponer de licencias para su uso. Para realizar la comunicación, se utilizan 79 canales diferentes de 1 Mhz cada uno, realizando 1600 "saltos"por segundo, reduciendo así las interferencias y aportando un leve matiz de seguridad al protocolo contra terceros que intenten hacer escuchas. Estos canales proporcionan una comunicación *half-duplex* (únicamente un dispositivo es capaz de transmitir información al mismo tiempo) entre los dispositivos, permitiendo además que existan múltiples esclavos. El alcance en Bluetooth normalmente está limitado a 10 metros (en interior), aunque es posible realizar comunicaciones de hasta 100 metros de distancia si se utilizan receptores y emisores especiales.

Actualmente, se ha publicado el estándar *Bluetooth 5* introduciendo numerosas mejoras como Bluetooth Low Energy, que permite duplicar la velocidad (hasta 2Mbit/s) o

multiplicar por cuatro el rango (hasta aproximadamente 200 metros en exterior).

NFC

La comunicación de campo cercano o Near Field Communication (NFC) es un estándar de comunicación inalámbrica de punto a punto basado en la tecnología de identificación por radiofrecuencia (en inglés RFID). En concreto, usa la banda de alta frecuencia (más exactamente en los 13.56 Mhz), permitiendo que se realice su uso sin ningún tipo de restricción o licencia.

NFC se basa en el principio físico de la inducción para realizar el intercambio de la información. Para ello, se hace circular una corriente a través de unas espiras generando un campo electromagnético para posteriormente modular la información como describe la Figura 2.1. Esto permite realizar las comunicaciones hasta una distancia máxima teórica de 10 cm, con tasas de transferencia de hasta 424 kbit/s. Dependiendo del papel que realice el dispositivo en la comunicación, se distinguen dos tipos de dispositivos:

- **Activo**, donde ambos dispositivos inducen dicho campo electromagnético.
- **Pasivo**, donde el dispositivo que inicia la comunicación es el encargado de generar el campo y el otro dispositivo aprovecha la modulación de la carga para recibir la información.

Dependiendo de las capacidades de los dispositivos involucrados, la comunicación puede realizarse de tres formas diferentes:

- **Lector/Escritor**, donde el dispositivo activo realiza operaciones de lectura y/o escritura sobre el elemento pasivo, comúnmente llamado etiqueta o *TAG*.
- ***Peer-to-Peer***, donde dos dispositivos activos realizan la comunicación creando una red *ad-hoc*.
- ***Host-card-emulation***, compuesto por dos dispositivos activos, donde uno de ellos simula ser un dispositivo pasivo.

Para los sistemas de pago sin contacto, la comunicación se realiza con un dispositivo funcionando en modo activo y otro en pasivo. El punto de venta (datáfono) o Proximity Coupling Device (PCD) trabajará de forma activa y es el encargado de iniciar la comunicación. Por otro lado, la tarjeta *contactless* o Proximity Integrated Circuit Card (PICC) trabajará en modo pasivo y utilizará el campo electromagnético para modular la información.

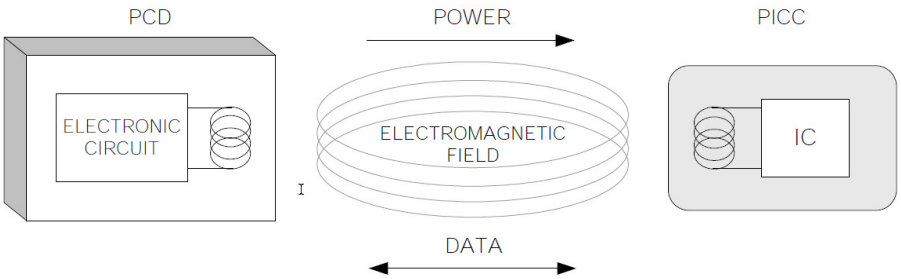

Figura 2.1: Configuración del PCD y PICC [9, página 13]

2.2. Normas ISO/IEC 14443 e ISO/IEC 7816

Norma ISO/IEC 14443

Multitud de organizaciones a lo largo del globo han intentado definir los estándares RFID según sus intereses y aplicaciones. Entre todas las propuestas, es de especial interés el ISO/IEC 14443 [16–19], debido a que es un estándar internacional relativo a las tarjetas de proximidad e identificación gestionado conjuntamente por International Organization for Standardization (ISO) e International Electrotechnical Commission (IEC).

Este estándar consta de 4 documentos o partes y su ultima revisión es del año 2016. El estándar especifica dos tipos de tarjetas (tipo A y tipo B), utilizando diferentes tipos de modulación, métodos de inicialización y codificación, pero siempre trabajando en la misma frecuencia de 13.56 Mhz.

Para la realización de este trabajo, es de interés su cuarta parte [18], ya que especifica el protocolo de transmisión usado entre dos dispositivos.

Norma ISO/IEC 7816

Este estándar internacional desarrollado conjuntamente por ISO, IEC y editado por Comité Técnico Conjunto 1 y el subcomité 17, se reparte en un total de 15 documentos donde se recogen las especificaciones de las tarjetas inteligentes. En esta norma, se especifica la unidad de comunicación entre un lector y la tarjeta inteligente, también conocido como Application Protocol Data Unit (APDU). Existen dos tipos de APDU, los comandos y las respuestas.

Los comandos (véase la Figura 2.2) tienen una cabecera de carácter obligatorio de 5 bytes y pueden transmitir un datagrama de información de hasta 255 bytes. Del mismo modo, las respuestas (véase la Figura 2.3) pueden tener hasta 256 bytes de información de la respuesta generada y obligatoriamente contienen una terminación de 2 bytes donde se recoge el estado del procesamiento del comando anteriormente enviado.

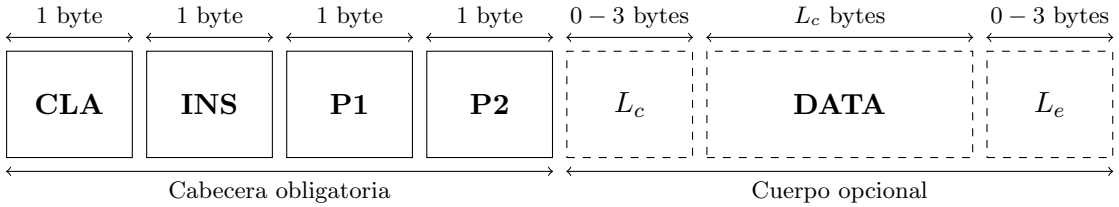

Figura 2.2: Comando APDU

Figura 2.3: Respuesta de un comando APDU

2.3. UML

El lenguaje unificado de modelado o UML por sus siglas en inglés, es un lenguaje de propósito general enfocado en el modelado de un sistema en la rama de la ingeniería del software. UML busca facilitar la visualización, especificación, diseño y documentación de un sistema a través de un lenguaje gráfico utilizando dos tipos de diagramas.

Los **diagramas estructurales** muestran la estructura estática de un sistema y sus objetos. A lo largo de este documento, de los diagramas estructurales, sólo se utilizan los diagramas de clases. Un **diagrama de clases** representa las clases con forma de caja, mostrando sus atributos en la parte superior y sus métodos en su parte inferior. Los diferentes tipos de flechas interconectan las clases e indican el tipo de relación existente entre las clases.

El segundo tipo de diagrama son los **diagramas de comportamiento**, que muestran las acciones entre los distintos elementos del sistema. En este documento sólo se utilizan los diagramas de secuencia. Los **diagramas de secuencia** tienen una columna por elemento a modelar (objeto de una clase) del sistema, llamada línea de vida. Las interacciones entre los elementos se representan mediante flechas, así como una breve descripción de la acción (normalmente expresada como la llamada a un método o función concreto del objeto representado).

2.4. EMV y tarjetas de crédito/débito

EMV

EMV es una marca registrada (comúnmente tratado como un estándar) que define la interoperabilidad de las PICC y los PCD para pagos seguros mediante tarjetas de crédito y débito. El nombre proviene del acrónimo EMV, debido a las tres compañías

que inicialmente colaboraron en su desarrollo (EUROPAY, MASTERCARD y VISA). Actualmente, otras compañías como JCB, Discover y Union Pay forman parte también de EMV.

El propósito de EMV es ofrecer una interoperabilidad a nivel mundial así como mejorar la seguridad en los pagos respecto al pago clásico mediante banda magnética. Según *The Nilson Report* [13], durante el año 2016 el fraude relativo a las tarjetas de crédito creció un 4.4 % respecto al 2015, alcanzando un total de 22.800 millones de dólares; es decir, de cada transacción de 100 dólares, $0.0775 iría a manos de terceros mediante la utilización de alguna técnica de fraude.

El aumento de seguridad ofrecido por EMV viene dado gracias a la utilización de algoritmos de cifrado como DES, RSA o algoritmos de *hashing* como SHA, así como la inclusión de un código numérico personal (PIN). Un código PIN se compone de 4 a 6 dígitos que únicamente conoce el titular de la tarjeta y que debe ser introducido en el terminal de pago a la hora de realizar una transacción, sirviendo de autorización del pago.

La mayor parte de la documentación sobre el protocolo EMV está de forma pública en su página web, aunque llegar a tener una visión general del protocolo puede suponer un reto debido a la gran extensión de sus documentos, ya que las especificaciones sobre las tarjetas EMV (tanto en la versión de contacto como *contactless*) tienen una longitud de aproximadamente 1200 páginas con un detalle técnico muy elevado.

Tarjetas de crédito/débito

Las tarjetas de crédito ó débito son tarjetas bancarias, normalmente de plástico y con unas dimensiones de 8,5 x 5,3 cm conforme al estándar ISO/IEC 7810 [15]. Son emitidas por un banco o entidad financiera y autorizan al titular a utilizarla como método de pago en los negocios adheridos. Como mínimo, siempre cuentan con el nombre del titular, la fecha de expiración así como el Primary Account Number (PAN) en el anverso y con una banda magnética y un código Card Verification Value (CVV) en el reverso. En las nuevas tarjetas también suele venir incluido un chip electrónico que almacena los datos de la tarjeta y proporciona una conexión inalámbrica mediante NFC. Estas tarjetas son conocidas como tarjetas *contactless*.

Un establecimiento que disponga de un punto de venta compatible con las tarjetas *contactless* reducirá el tiempo entre cobros, ya que si no se alcanza una cantidad mínima, dicho pago no requiere de verificación por parte del usuario introduciendo su PIN. Por cuestiones de seguridad, se puede definir una cuantía máxima por la cual no es necesario introducir el PIN o firmar la transacción, siendo de hasta 25 euros en la mayoría de los países de la Unión Europea [1] (en España está reducido a 20 euros), aunque existen otros países en los cuales no existe dicho límite. El número de pagos consecutivos realizables mediante *contactless* sin confirmación mediante PIN también se encuentra superiormente acotado para reducir posibles fraudes.

Capítulo 3

Ataque de retransmisión inteligente

Un ataque de retransmisión o ataque de *relay* es un tipo de ataque estrechamente relacionado con ataques *man-in-the-middle* (donde el atacante tiene conexiones con ambas víctimas, envía mensajes y reenvía mensajes entre ambos, haciéndoles creer que existe una conexión directa entre ellos) y ataques de *replay* (donde el atacante reenvía información válida interceptada anteriormente).

Los ataques de *relay* se basan en la idea introducida por Conway [6] en 1976, conocida como *"Chess Grandmaster problem"* donde una niña pequeña reta a dos grandes maestros a una partida de ajedrez por correspondencia. Conway expone que la niña se limita únicamente a retransmitir los movimientos recibidos entre los dos grandes maestros. De esta manera, la niña sólo puede ganar o empatar aún desconociendo las reglas del ajedrez.

En los ataques de *relay* modernos, el emisor original puede ser o no consciente de haber enviado el mensaje al atacante y en caso de que lo sea, el emisor tendrá la impresión de estar enviando la información al receptor original.

En los sistemas de pago *contactless*, de la misma manera que la idea de Conway, únicamente se reenvían los mensajes del PCD al PICC y viceversa, haciendo creer a ambas partes que están comunicándose directamente. Un escenario susceptible a realizar un ataque de retransmisión se puede observar en la Figura 3.1, donde existe una tarjeta *contactless* (PICC) en un punto geográfico A y un punto de venta (PCD) en otro punto geográfico B. Introduciendo mediante dos actores artificiales, se consigue realizar una retransmisión y una comunicación efectiva entre ambos puntos.

El actor artificial que se comunica con el PICC y actúa como un falso PCD recibe el nombre de `Mole`. Del mismo modo, el actor artificial que se comunica con el PCD y actúa como un falso PICC se llama Proxy. Entre estos elementos puede existir una distancia arbitrariamente grande, siendo posible incluso la retransmisión a través de Internet.

Por ejempo, en [26] se demostró que era posible retransmitir la información incluso a más de 5700 km (desde Madrid a Nueva York). Además, se demostró que estos ataques son factibles usando dispositivos moviles Android sin modificar. La prueba de concepto

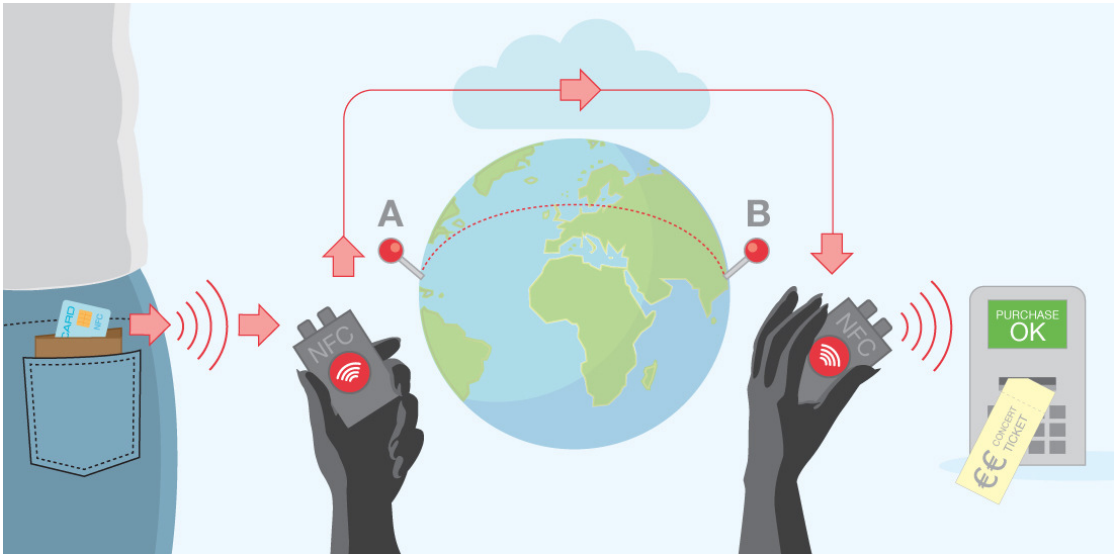

Figura 3.1: Ataque de retransmisión en sistemas de pago *contactless*, extraída de [22]

mostrada en [26] exigía retransmitir todos los mensajes entre los dispositivos, añadiendo retardo a la comunicación.

Los principales inconvenientes de este tipo de ataques son principalmente este retardo introducido en la comunicación debido a la retransmisión de los mensajes y la latencia del canal de retransmisión. Si el receptor o el emisor exceden el tiempo máximo para la recepción de los mensajes, el protocolo EMV terminará automáticamente la comunicación.

El principal objetivo de este trabajo consiste en implementar un *relay inteligente* para reducir dicho retardo. Esto supone no enviar en bruto toda la información generada, sino en su lugar, gracias al conocimiento del protocolo, no enviar todos los mensajes generados realizando una precarga o cacheo de algunos de ellos. Adicionalmente, se pueden introducir mensajes artificiales en la comunicación que generen un entorno más favorable para que se produzca el pago sin exceder el tiempo máximo permitido.

A continuación, se va a detallar el proceso que se ha seguido para la implementación del *relay* inteligente, identificando qué mensajes son enviados y cuál es su papel en la interacción entre la tarjeta de crédito/débito y el punto de venta.

3.1. Detección de tramas a retransmitir

Debido a que EMV nació como un conjunto de normas y estándares de codificación de los pagos con tarjeta pero sin especificar el protocolo a seguir para realizar dichos pagos, cada entidad financiera debe implementar su propio sistema de pago compatible.

Los protocolos PayPass y PayWave son las implementaciones propuestas por MASTERCARD y VISA, respectivamente. Aunque ambos protocolos son muy parecidos,

existen diferencias a la hora del envío de los mensajes a la entidad bancaria, así como el momento en el que empieza el cifrado de la comunicación. Estas diferencias entre el envío de mensajes de ambos protocolos se muestran en los diagramas de secuencia mostrados de la Figura 3.2 y la Figura 3.3 respectivamente.

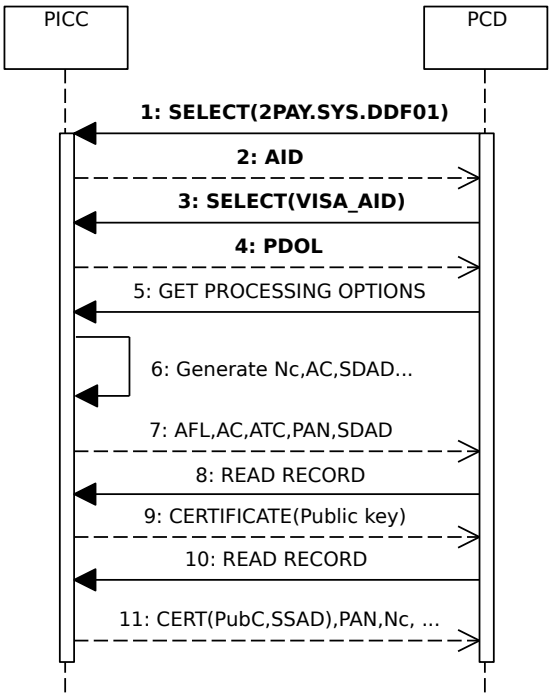

Figura 3.2: Diagrama de secuencia con PayWave (VISA) [4]

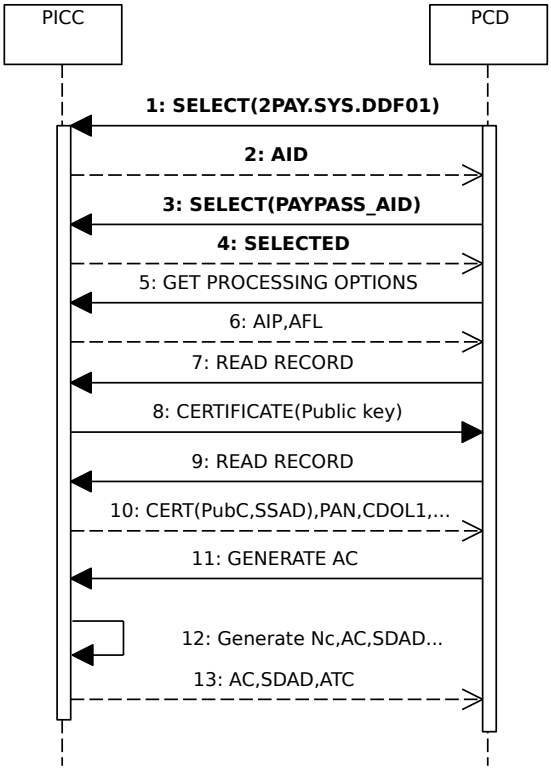

Figura 3.3: Diagrama de secuencia con PayPass (MASTERCARD) [4]

Durante la realización del pago, se generan numerosas tramas que contiene una o más etiquetas. Cada etiqueta está formada por unos pocos bytes y determina la información que retransmite dicha trama. El orden, así como el contenido de la trama, es dependiente de la implementación concreta.

Como se puede observar, ambos protocolos (PayPass y PayWave) realizan la identificación de la tarjeta en primer lugar, correspondientes a los mensajes 1-3 en la Figuras 3.2 y 3.3. A continuación, ambos protocolos iniciarán la obtención de la información necesaria de la tarjeta, aunque realizando secuencias distintas. Para ello, utilizan el comando APDU *Get Processing Options (GPO)* (mensaje 5 en ambas Figuras 3.2 y 3.3), obteniendo los métodos de pago disponibles en la tarjeta. Posteriormente, se procede a leer los registros de información de la tarjeta y cifrar la comunicación. Como punto de diferenciación, el protocolo PayWave genera el criptograma necesario para cifrar la comunicación en el mensaje 6, mientras que en el protocolo PayPass esto no sucede hasta el mensaje 12, provocando así que se envíen un mayor número de mensajes sin cifrar.

Una trama será susceptible de ser cacheable únicamente cuando sus datos no sean dinámicos (es decir, que su información cambie en cada transacción) ni estén firmadas digitalmente. Las tramas que son susceptibles a ser precargadas han sido marcadas en negrita en las Figuras 3.2 y 3.3.

Dado que ambos protocolos inician la comunicación seleccionando la aplicación *Entorno del sistema de pago por proximidad* (mensajes número 1) para posteriormente seleccionar el Application IDentifier (AID) correspondiente, se abre la posibilidad de eliminar los retardos de retransmitir las primeras cuatro tramas si dichos mensajes se han cacheado previamente.

3.2. Protocolo de interacción

Una vez identificadas las tramas que son susceptibles a ser cacheadas y cuáles es necesario retransmitir, el nuevo protocolo de interacción del *relay* se compone de tres partes. La Figura 3.4 muestra el protocolo de retransmisión inteligente implementado.

En primer lugar está la fase de cacheado de los mensajes entre el PICC, la `Mole` y el `Proxy`. Debido a que no interviene en la comunicación el PCD, esta fase se realiza fuera del proceso de pago por lo que su duración temporal queda excluida del tiempo total de la transacción.

En segundo lugar está la inicialización del protocolo EMV seleccionando el pago *contactless* e identificando qué AIDs admite el PICC. En esta fase únicamente interviene el `Proxy` y el PCD y corresponde a los mensajes del 7 al 10 del diagrama de secuencia de la Figura 3.1. Esto permite que el PCD pueda identificar el tipo concreto de tarjeta es, así como determinar si pertenece a VISA o MASTERCARD.

La tercera y última fase corresponde a la retransmisión del resto de tramas dinámicas o firmadas digitalmente. Desde el envío del GPO, correspondiente al mensaje número 11 en la Figura 3.4, todas las tramas generadas por el PCD tienen que ser retransmitidas al PICC y viceversa.

El sistema de retransmisión propuesto no comienza el cacheado de los mensajes hasta que se detecte la aproximación de una tarjeta *contactless* (mensaje 1 en la Figura 3.4). Cuando haya una tarjeta *contactless* disponible, se envían de forma iterativa un mensaje de selección de un AID por cada uno de los disponibles en la prueba de concepto desarrollada (mensajes 2-3). Si la tarjeta admite ese entorno de pago, esta información será retransmitida al `Proxy` mediante dos mensajes: el AID soportado por la tarjeta (mensaje 4) y la respuesta obtenida (mensaje 5).

Una vez que hayan comprobado todos los AIDs disponibles, el sistema de retransmisión notifica al `Proxy` (mensaje 6) que ya puede empezar la comunicación con el PCD. A partir de este punto, la comunicación no puede superar el tiempo máximo permitido por el protocolo NFC (por defecto, 500ms).

Al iniciar la comunicación con el PCD, éste empezará a solicitar a la tarjeta los AIDs para identificar qué protocolo de pago va a realizarse (mensajes 7-10). En la implementación inicial de la aplicación usada en este proyecto (`NFCLeech`, explicada en más detalle en el Capítulo 5), estos mensajes eran retransmitidos al `Mole` para permitir el procesado

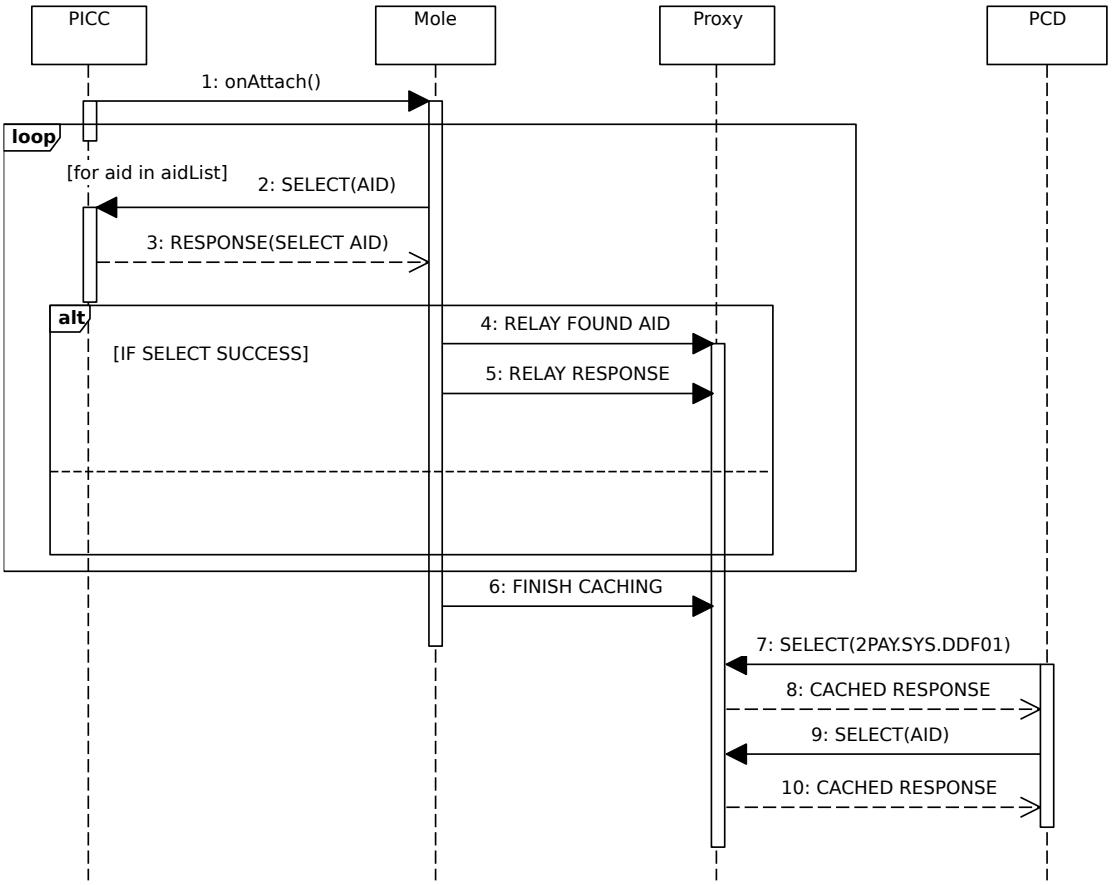

Figura 3.4: Diagrama de secuencia de la comunicación en un *relay* inteligente

de la tarjeta y posteriormente retransmitir el resultado. Con esta nueva implementación, se elimina el tiempo de retardo introducido, dado que no se realiza retransmisión de estos mensajes. A partir de este punto, todos los mensajes restantes deben ser retransmitidos y se consideran por tanto los retardos de transmisión.

Capítulo 4

Estrategias para retraso de comunicaciones en NFC

Debido a la introducción de retardos por la retransmisión de las tramas, es posible que la transacción falle si el tiempo de procesamiento y el tiempo de retransmisión excede el tiempo máximo permitido. Según el protocolo EMV [8, página 91], el tiempo de transacción no debería superar los 500 *ms*. Aunque ha habido experimentos [4] donde se han realizado transacciones de más de 600 *ms*, este máximo sigue siendo un factor limitante para realizar correctamente el ataque de retransmisión.

En el momento de iniciar la comunicación, se define el *Frame Waiting Time (FWT)*, que será el tiempo de espera máximo de espera por parte del PCD para recibir una respuesta. Dicho tiempo se define según la ecuación 4.1, donde Frame Waiting time Integer (FWI) codifica un valor entero utilizado para definir el FWT y f_c representa la frecuencia a la que trabaja el PCD.

$$FWT = 256 \cdot \left(\frac{16}{f_c}\right) \cdot 2^{FWI}, 0 \leq FWI \leq 14 \tag{4.1}$$

El protocolo EMV ofrece la posibilidad de enviar tramas especiales, que no transmiten información de la transacción, sino que permiten modificar ciertos parámetros de la comunicación entre PICC y PCD. Según dicho protocolo [9, páginas 195-203], los bloques de transmisión del protocolo están divididos en tres partes, el *PCB*, el *INFormation field (INF)* y el *Error Detection Code (EDC)*, como se observa en la Figura 4.1. En los dos bits más significativos del PCB se codifica el tipo de bloque y determina la función de la trama. Los tipos de bloques, su funcionalidad y la codificación se resumen en la Tabla 4.1.

A continuación, se detalla cómo son y qué papel tienen en la comunicación las dos tramas que permiten alargar el tiempo de comunicación.

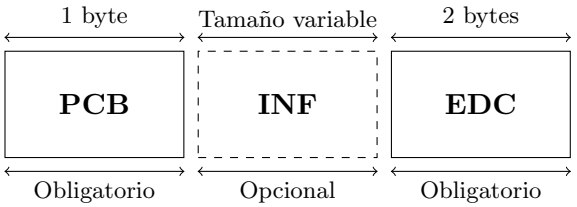

Figura 4.1: Bloque de transmisión de protocolo

b8	b7	Significado	Uso
0	0	Bloque tipo I	Transmitir información a la capa de aplicación.
0	1	No permitido	No aplicable (N/A)
1	0	Bloque tipo R	Transmitir la recepción o no recepción del mensaje.
1	1	Bloque tipo S	Intercambio información de control.

Tabla 4.1: Codificación de los bits b8-b7 del PCB

4.1. Trama WTX

Si el PICC está realizando tareas de elevado coste computacional, como por ejemplo el cifrado de la comunicación, es posible solicitar al PCD una extensión del tiempo máximo del estipulado previamente en el FWT. Para ello, se puede mandar una petición *WTX* al PCD aumentando así el tiempo disponible para procesar el bloque recibido.

La trama WTX es retransmitida en un bloque de tipo S, por lo que primero es necesario codificar del bloque PCB como se muestra en la Tabla 4.2. Además, es necesario incluir un bloque INF de un 1 byte donde se codifica el valor de extensión de la comunicación.

A continuación, se muestra cómo se codifican los campos de información y de detección de errores, así como el resultado final de la trama.

Codificación del bloque de información

La estructura del bloque de información está compuesta por 2 bits relativos al *Power level* y el *Waiting Time eXtension Multiplier (WTXM)* tal y como se recoge en la Tabla 4.3 Los dos bits que estipulan el *Power level* tienen el valor (00)b por defecto (no obstante, es posible que existan implementaciones del PCD con otros valores). Para la realización de los experimentos, se considerará siempre el valor por defecto. Una vez

b8	b7	b6	b5	b4	b3	b2	b1
1	1	x	x	0	0	x	0

x	Significado
0	*DESELECT*
1	WTX

Tabla 4.2: Codificación de un bloque tipo S

bits	Significado
8-7	Power level indication
6-1	WTXM

Tabla 4.3: Codificación del bloque de información para el WTX

	Byte(s)	Decimal	Hexadecimal
PCB	1 1 1 1 0 0 1 0	242	F2
INF	0 0 1 1 1 0 1 1	59	3B
EDC_1	1 0 0 1 0 0 0	72	48
EDC_2	1 1 0 1 1 1 1 0	222	DE

Tabla 4.4: Codificación del WTX antes del CRC

definido este valor servirá como factor multiplicativo respecto al FWT.

Con los 6 bits restantes, se codifica el WTXM. Dicho WTXM viene definido con un rango desde el valor 1 al 59, siendo el valor 1 el mínimo y el valor 59 el máximo, tratando como error cualquier otro valor. Para extender la comunicación lo máximo posible, se codificará el valor 59.

Codificación del epílogo

Para codificar el EDC, es necesario calcular la verificación por redundancia cíclica (en inglés, CRC) de 16 bits según el ISO-13239 [14] con un valor inicial `0x6363` y sin la inversión lógica después de realizar el cálculo.

Construcción de la trama

Tras todo lo expuesto anteriormente, el tamaño de trama será $WTX_{size} = PCB_{size} + INF_{size} + EDC_{size} = 1B + 1B + 2B = 4Bytes$

Con estos valores, el nuevo valor FWT_{new} viene calculado por:

$$FTW_{new} = FWT \cdot WTXM$$

Durante la realización de los experimentos, se hicieron varias pruebas de inclusión de dicha trama con los valores arriba expuestos y con otros valores para comprobar la viabilidad de esta estrategia. Lamentablemente, debido a la implementación interna del terminal punto de venta usado en los experimentos, dichos mensajes eran considerados *"Protocol Error"* produciendo un reinicio de la comunicación por parte del PCD. Debido a estos problemas, esta estrategia fue descartada como solución.

4.2. Trama NAK

Otra posible estrategia para alargar el tiempo máximo de la comunicación es enviar tramas NAK para introducir errores artificiales en la comunicación, buscando que el

b8	b7	b6	b5	b4	b3	b2	b1
1	0	1	x	0	0	1	0

x	Significado
0	*ACK*
1	*NACK*

Tabla 4.5: Codificación de un bloque tipo R

	Byte(s)	Decimal	Hexadecimal
PCB	1 0 1 1 0 0 1 0	178	B2
EDC_1	0 1 1 0 0 1 1 1	103	67
EDC_2	1 1 0 0 0 1 1 1	199	C7

Tabla 4.6: Codificación del NAK

PCD extienda el tiempo de procesamiento.

El protocolo EMV define la trama NAK para expresar el no reconocimiento de la respuesta obtenida, ya sea por un error de la comunicación o por un valor no esperado. Una vez dicha trama es recibida, se inicia el protocolo de recuperación, siendo éste en la mayoría de los casos volver a enviar la respuesta. Los NAK están contenidos dentro de las tramas tipo R (las encargadas de transmitir la recepción positiva o negativa de los datos esperados), como se observa en la Tabla 4.5. Según la definición del protocolo EMV de las tramas tipo R [9, páginas 196-197], jamás llevan el campo opcional de información, por lo que trama resultante será la concatenación del PCB tipo R y el CRC calculado de igual forma que en el apartado anterior como se observa en la Tabla 4.6

Una vez codificada la trama NAK (con valor hexadecimal `0xb267c7`) se realizó una modificación de la aplicación para introducir los errores de forma aleatoria. Lamentablemente, se observó que el único resultado era el reenvío del último comando enviado sin extender el tiempo máximo, por lo que esta estrategia también se descartó.

4.3. Cacheo de tramas

El PCD identifica la tarjeta que se ha aproximado durante la fase de inicialización para realizar cualquier transacción. Además, obtiene de ella los datos necesarios para la autenticación de la tarjeta para posteriormente iniciar una comunicación cifrada y procesar la transacción. Como es de esperar, la información de la tarjeta es inmutable. Por lo tanto, esta información puede ser almacenada directamente en el dispositivo que va a realizar la comunicación con el PCD, eliminando de esta manera el coste asumido por el procesamiento de la petición por parte del PICC y su retransmisión.

De las estrategias comentadas, ésta ha sido la que ha dado mejores resultados. En la Sección 5.2.2 se comentará con mayor detalle cómo se ha realizado la implementación de la estrategia de cacheo, así como del comportamiento en cada dispositivo.

Capítulo 5

Implementación: Extensión de `NFCLeech`

En las siguientes secciones se va a detallar el estado inicial y final de la aplicación Android `NFCLeech` [3], que servirá como prueba de concepto donde aplicar todo lo anteriormente descrito.

5.1. Estado inicial de `NFCLeech`

Antes de iniciar esta prueba de concepto, la aplicación `NFCLeech` se desarrolló en [26] como prueba de concepto de los ataques de retransmisión en dispositivos Android sin modificar. Originalmente, esta aplicación contaba con casi 2000 líneas de código y realizaba una retransmisión del pago a través de una red WiFi con una latencia baja. En concreto, permitía utilizar una red WiFi para retransmitir la comunicación entre dos puntos geográficos muy distantes entre sí, sabiendo previamente las direcciones IP de ambos dispositivos. De hecho, su capacidad de retransmisión a distancia se demostró en la conferencia RFIDSec'15 haciendo un pago en Madrid desde Nueva York (más de 5700 km) [26]. Como contrapartida, la aplicación requería estar en un entorno con una red disponible y con una vía de comunicación auxiliar para intercambiar las direcciones IP entre el `Proxy` y la `Mole`.

Adicionalmente, la aplicación incluía la opción de utilizar como canal de comunicación una red WiFi P2P (también conocido como *WiFi Direct*), donde uno de los dos dispositivos (`Proxy` o `Mole`) es el encargado de generar una red WiFi y actuar como enrutador. Esto permite realizar la comunicación hasta un máximo teórico de 100 metros. Aunque en algún momento del anterior desarrollo esta funcionalidad se ejecutaba correctamente, en la versión de la herramienta usada inicialmente en este trabajo resultaba imposible comunicar correctamente los dos dispositivos utilizando esta opción.

Anteriormente, para utilizar correctamente la aplicación, era necesario seleccionar el canal de comunicación (WiFi o WiFi Direct) y el rol (`Proxy` o `Mole`) en la primera ventana de la aplicación. Después, en la segunda ventana era necesario introducir la dirección IP del otro dispositivo, incluyendo además el número de puerto en la `Mole`.

Una vez terminada la configuración, se abría la ventana con el registro de la aplicación, donde era necesario pulsar el botón de *"Start"* para poder iniciar la retransmisión. Finalmente si se aproximaba la `Mole` al PICC y el `Proxy` al PCD se retransmitía la comunicación. Este modo de funcionamiento era poco utilizable para el usuario, además de que facilitaba el error del usuario durante el manejo de la aplicación, resultando en errores irrecuperables (la aplicación carecía de una buena gestión de excepciones).

5.2. Mejoras introducidas

A lo largo de esta sección, se describen brevemente las mejoras que se han implementado a lo largo del desarrollo del proyecto. En concreto, estas mejoras han consistido en un sistema de registro (permitiendo un parseo dinámico de los mensajes recibidos), la inclusión de un sistema de cacheo de mensajes, la mejora de la estabilidad de la aplicación, una refactorización del código y la inclusión de un nuevo canal de comunicación para la retransmisión (Bluetooth).Como resultado final, ahora la aplicación cuenta con más de 3500 líneas de código. A continuación se describen cada una de las mejoras en más detalle.

5.2.1. Sistema de registro

Durante la toma de contacto inicial se observó que el sistema de procesamiento y representación de los mensajes APDU en ocasiones era insuficiente, puesto que únicamente mostraba la información recibida representando el datagrama recibido en hexadecimal.

Para facilitar el desarrollo, se decidió implementar un nuevo sistema de procesamiento de los mensajes que mostraba más información relativa al comando o respuesta recibida. Entre las mejoras se encuentran:

- **Visualización del código de estado:** Se incluyeron los códigos de estado que faltaban a la hora de procesar la respuesta recibida, así como una breve descripción. Adicionalmente, este nuevo sistema de registro expone funciones para extraer y procesar únicamente el estado de la petición, permitiendo acceder a dichas funciones desde cualquier parte de la aplicación, mejorando la legibilidad del código y evitando errores.
- **Nuevos modos de impresión:** En esta nueva versión es posible decidir cómo va a ser la información que se va a mostrar por pantalla, pudiéndose elegir mostrar el comando APDU, su representación en hexadecimal, ASCII o cualquier combinación entre los tres modos anteriores de forma dinámica.
- **Formateado de las tramas:** Se definió un nuevo formato para mostrar cada trama recibida, mostrando además el destinatario y remitente con el objetivo de facilitar los procesos de depuración e incluyendo además una leyenda de colores. Así, ahora se identifican los mensajes enviados por el PCD con el color verde, los mensajes enviados por el PICC con el color azul y los mensajes introducidos la aplicación con el color naranja.

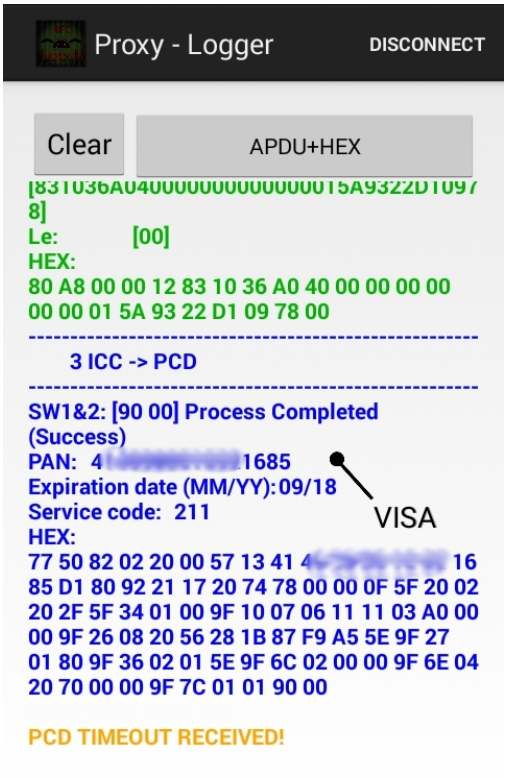

Figura 5.1: Ejemplo de interacción con VISA

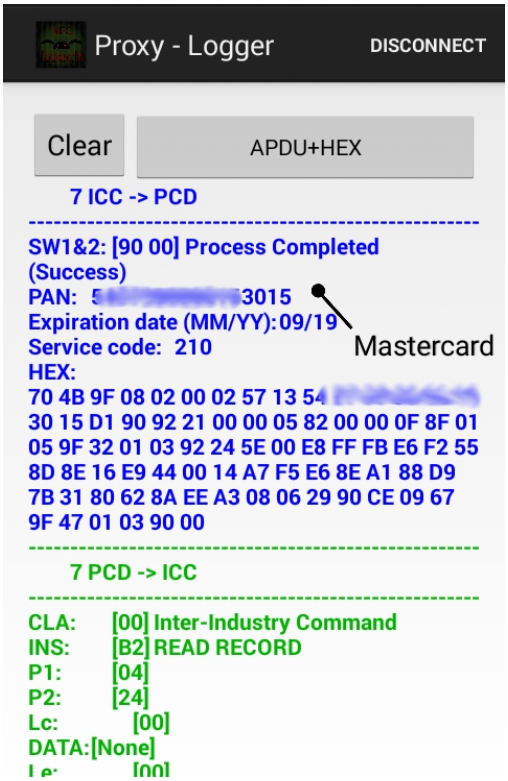

Figura 5.2: Ejemplo de interacción con MASTERCARD

- **Lectura de los datos bancarios:** Para finalizar el proceso de mejora, se implementó la lectura de los datos proporcionados por la tarjeta en el momento de procesar la solicitud (PAN, fecha de expiración de la tarjeta y código del servicio), para mostrarlos por pantalla como se observa en las Figuras 5.1 y 5.2.

Como resultado, ahora la aplicación es capaz de mostrar una mayor información por pantalla de una forma más clara y precisa, lo que permite realizar pruebas de una manera mucho más rápida y eficaz. En el Apéndice B se muestra un ejemplo de una traza de ejecución de la aplicación capturada durante la retransmisión de una transacción exitosa.

Adicionalmente, debido a la creación del nuevo sistema de registro se abre la posibilidad de la creación futura de un nuevo sistema de registro genérico permitiendo la depuración de cualquier tipo de comunicación realizada mediante NFC.

5.2.2. Cacheo de mensajes

Una parte crítica del *relay* inteligente es dotar al `Proxy` de la información necesaria para emular ser el emisor original. Para ello, es necesario conocer la lista de AIDs que soporta la tarjeta antes de empezar a procesar el pago. De este modo, se pueden descartar

las tramas de inicialización de la comunicación, evitando introducir este retardo en la retransmisión.

Para obtener la lista de AIDs soportados por la tarjeta, en primer lugar se buscó el listado completo de AIDs soportados por VISA y MASTERCARD. Dicho listado es de dominio público y fácilmente accesible en Internet.

Tal y como se muestra en la Figura 5.3, una vez que se produce una aproximación entre el PICC y la `Mole`, se inicia un proceso de comprobación de forma iterativa de todos los AIDs disponibles. Si el código de estado de la respuesta APDU es de éxito, se envía al `Proxy` el AID correspondiente, así como la respuesta generada. Una vez que haya terminado la comprobación de todos los AIDs disponibles, se notifica al `Proxy`, permitiendo así que acepte las comunicaciones del PCD e iniciando el proceso de retransmisión.

5.2.3. Mejoras de estabilidad

La versión inicial de `NFCLeech` permitía hacer el proceso de retransmisión de los pagos, a costa de sufrir errores críticos durante su ejecución ante alguna situación no esperada o algún error durante la realización de la transacción. Los errores en la captura de excepciones, el desacople de la tarjeta durante la retransmisión, así como problemas derivados del canal de transmisión, producían un error crítico que provocaba la terminación forzosa del evento responsable de la acción en Android, terminando de forma abrupta la comunicación y en muchos casos, llegando al bloqueo del canal. Esto provocaba la necesidad de terminar la aplicación en ambos dispositivos, volver a iniciar y configurar la conexión, haciendo su uso difícil en un entorno real.

Estos problemas de estabilidad se han solventado aumentando la información mostrada al desarrollador durante su uso en modo depuración. Al ser capaz de obtener más información durante la ejecución, se localizaron los errores para posteriormente solucionarlos, permitiendo además realizar una captura de excepciones más eficiente. Estas mejoras tuvieron impacto directo en el rendimiento de la aplicación y sobre todo en su estabilidad.

Adicionalmente, se detectó que se producían numerosos errores a la hora de configurar la aplicación, ya que en ocasiones el elevado número de pasos necesarios para establecer el canal de comunicación hacía imposible el uso de la aplicación. Inicialmente, después de haber elegido el rol (`Proxy` o `Mole`) del dispositivo y de elegir como canal de comunicación el WiFi, era necesario introducir la dirección IP y puerto de cada dispositivo en su contrario. Después de aceptar dichos parámetros, se mostraba la pantalla del registro de la transacción, donde antes de empezar la comunicación había que pulsar el botón "*Start*" primero en el `Proxy` y posteriormente en la `Mole`, respetando siempre este orden.

Todo este sistema de configuración fue mejorado gracias a la funcionalidad que ofrece Android con la clase *SharedPreferences*, donde mediante atributos clave-valor, permite guardar los parámetros de configuración en la memoria persistente del dispositivo. De esta forma, al iniciar la aplicación, el último canal de comunicación utilizado se establece automáticamente, así como la última dirección IP introducida y su puerto.

Posteriormente se modificó el código del programa para, una vez que estableciese el canal, el parámetro relativo al puerto a utilizar se fuese incrementando automáticamente,

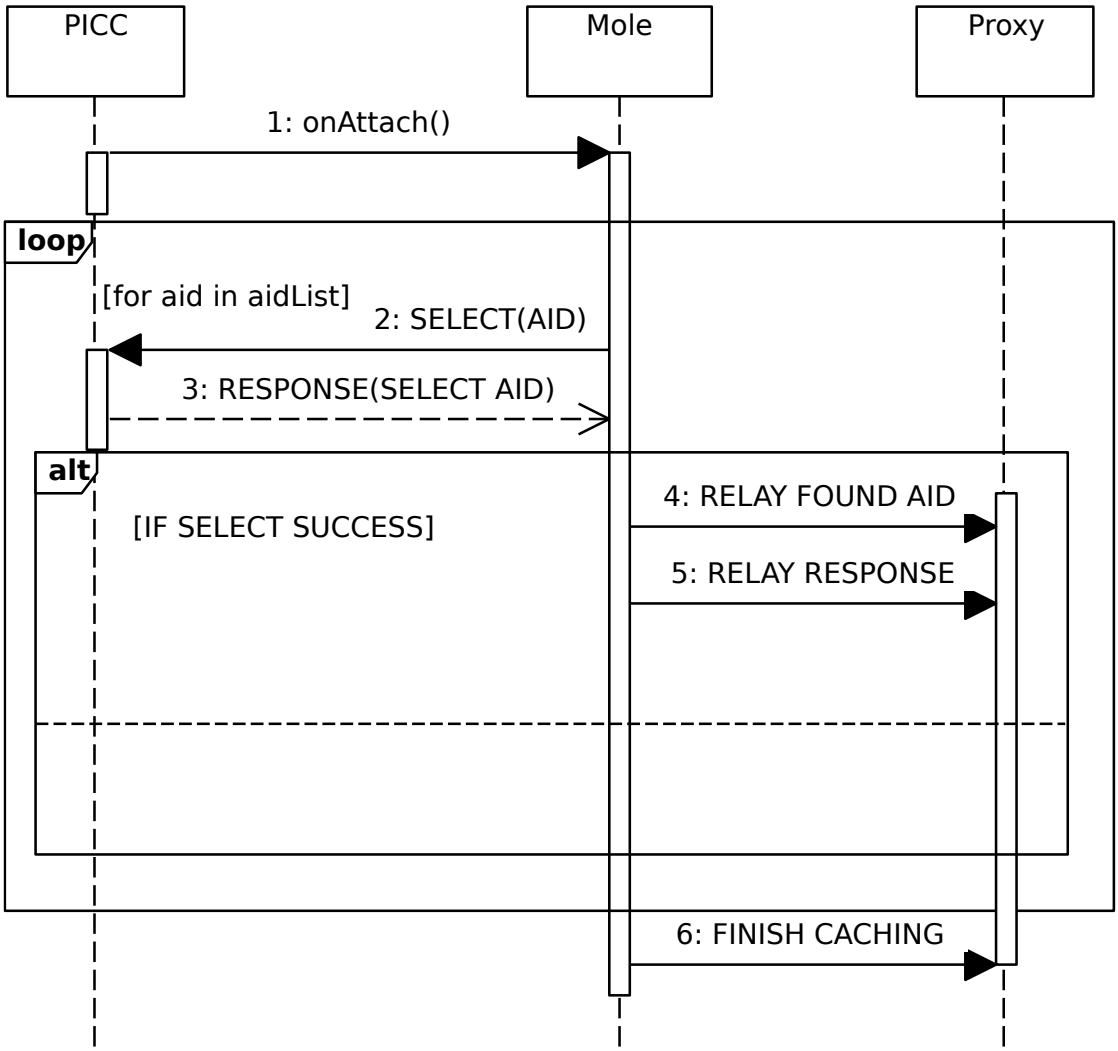

Figura 5.3: Diagrama de secuencia de la detección de AIDs y cacheado

evitando así los errores relativos a intentar establecer la comunicación con un canal cerrado incorrectamente.

Como última mejora introducida orientada a facilitar la inicialización de la aplicación, se incorporó la funcionalidad de iniciar automáticamente la retransmisión una vez que el sistema detectaba que la conexión mediante sockets se había realizado correctamente, eliminando la necesidad de incluir el botón de inicio en la pantalla del registro.

Todas estas mejoras no sólo aportaron estabilidad y facilidad de uso, sino que el tiempo necesario para iniciar un experimento se redujo drásticamente, facilitando en primer lugar las tareas de desarrollo y depuración y en segundo lugar, mejorando la posibilidad de realizar el ataque con éxito en un escenario real.

5.2.4. Refactorización y aplicación de técnicas de desarrollo del software

Uno de los principales problemas encontrados a la hora de extender la funcionalidad de la aplicación ha sido la escasa modularidad así como el fuerte tipado del código. Para facilitar la extensión de las funcionalidades, se han refactorizado la interfaz ChannelP2pI y se ha añadido una nueva interfaz SocketPeer2Peer.

Implementación inicial

Inicialmente, la interfaz ChannelP2pI se utilizaba como plantilla para definir los dos únicos canales existentes, WiFi y WiFiP2P. Esta implementación inicial utilizaba la clase SocketP2P, que actuaba como envoltorio de la clase nativa Socket. Esto obligaba a que las direcciones para realizar conexión debían ser una dirección IP, por lo que era incompatible con la inclusión de Bluetooth como un nuevo canal de comunicación. Se puede observar el diagrama de clases correspondiente en la Figura 5.4.

Adicionalmente, el código presentaba dependencias circulares, lo que complicaban la depuración así como la posible incorporación de nuevos canales o tipos de *sockets*. Para mejorar el diseño de este módulo, en primer lugar se han eliminado o sustituido los parámetros dependientes de una implementación del canal, como usar *InetAddress* que permite únicamente utilizar direcciones IP. Para solucionarlo, se han cambiado todas las direcciones a *String*, haciendo una interfaz mucho más genérica, a pesar de que sea necesaria una conversión posterior al tipo de dato correspondiente.

Para continuar, se ha implementado una nueva interfaz SocketPeer2Peer, donde se expone una colección de métodos necesarios para realizar correctamente la interfaz relativa al canal. Adicionalmente y para mejorar la organización del código, se han creado dos paquetes adicionales que agrupan los códigos fuente dependiendo de si son relativos a la implementación del canal de configuración o al Socket de conexión. El diagrama de clases posterior a la refactorización se puede observar en la Figura 5.5:

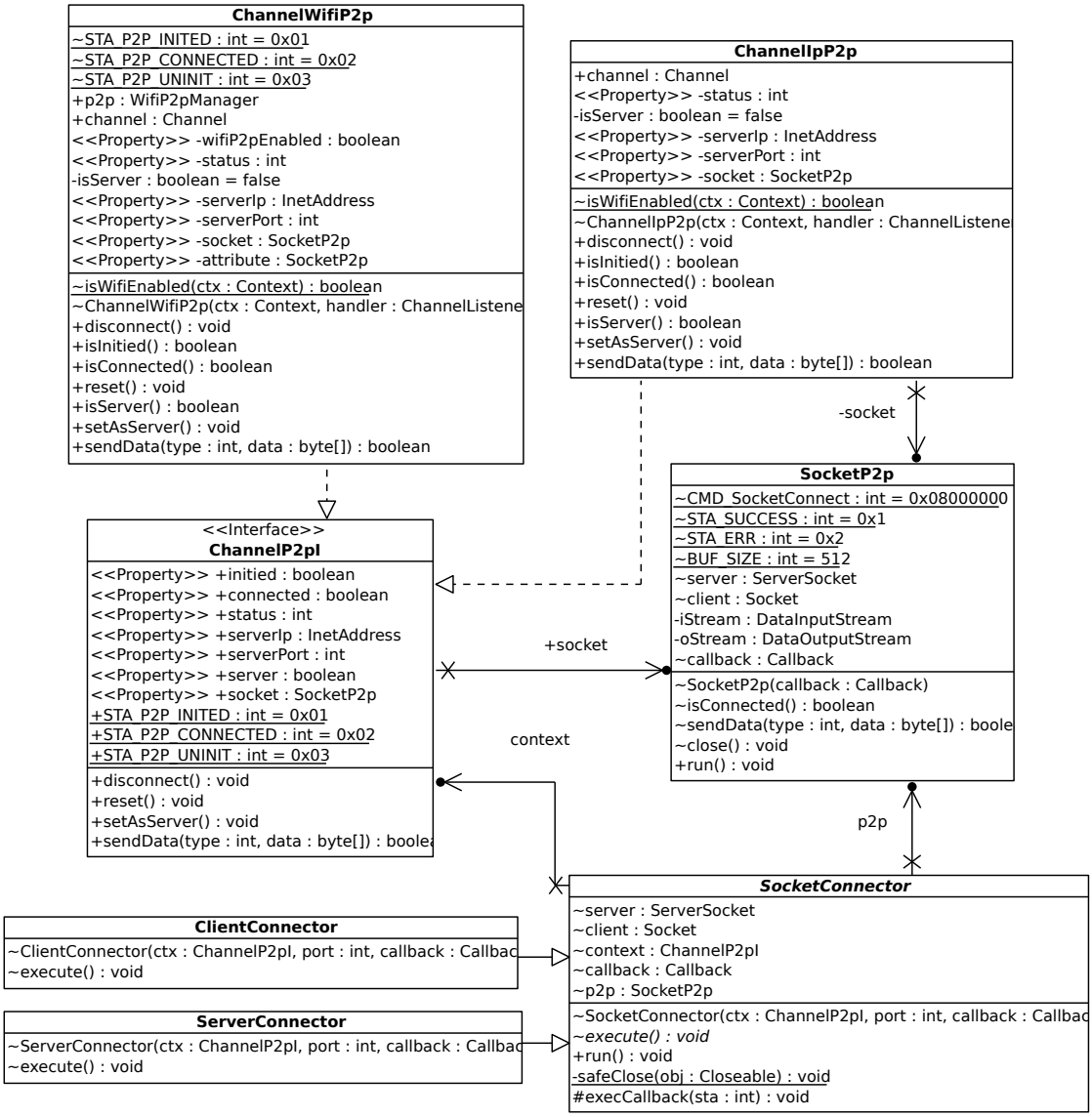

Figura 5.4: Diagrama de clases inicial para la implementación de la comunicación.

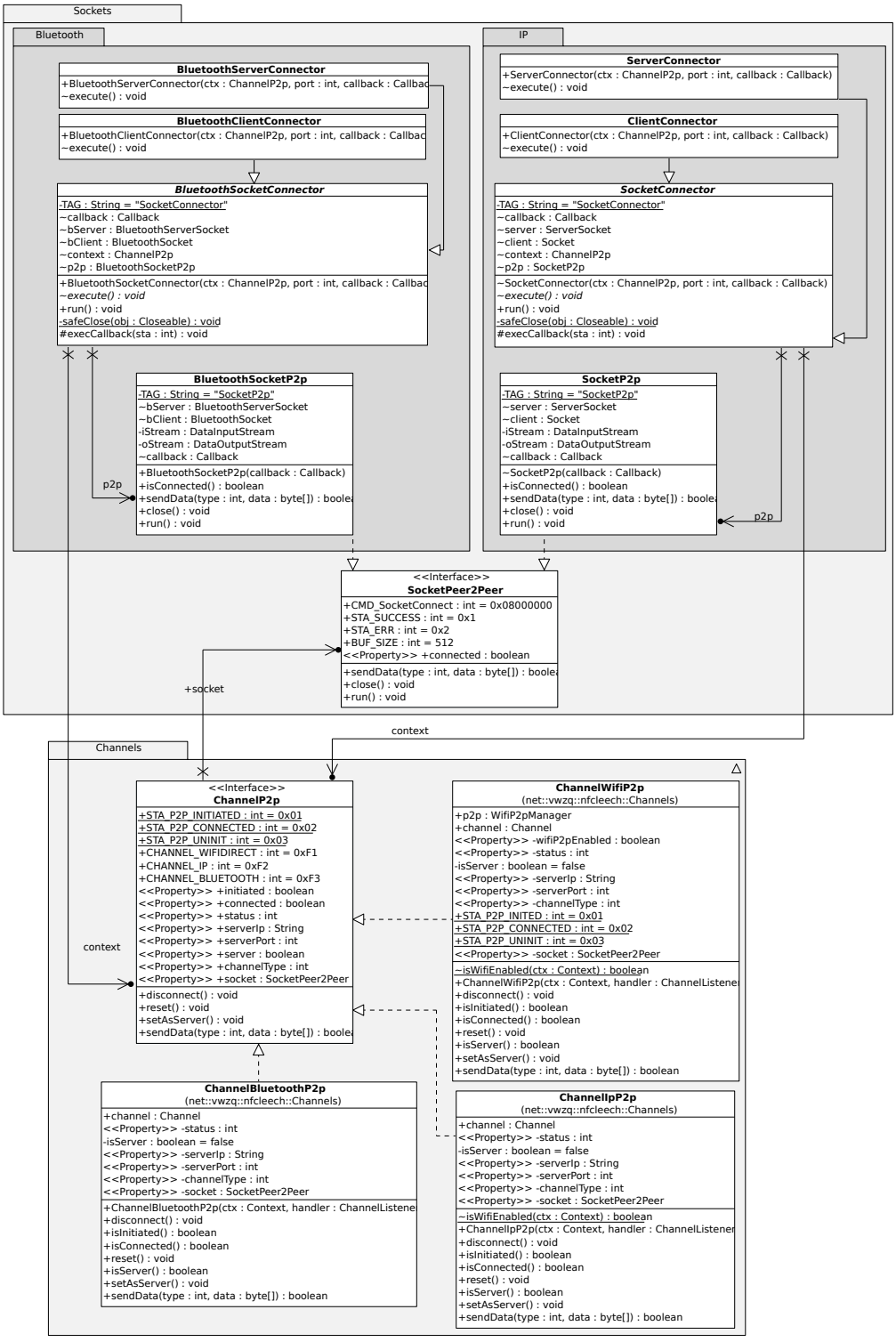

Figura 5.5: Diagrama de clases final para la implementación de la comunicación.

5.2.5. Retransmisión Bluetooth

Una de las mejoras introducidas ha sido la posibilidad de utilizar la tecnología Bluetooth como canal de comunicación entre el `Proxy` y la `Mole`. Esto elimina la necesidad de realizar los experimentos en lugares con conectividad WiFi, pero reduce la distancia máxima a la hora de realizar los ataques.

Para poder incluir la capacidad de utilizar Bluetooth, es necesario incluir los permisos correspondientes en la aplicación. Para ello, se ha incluido en el *AndroidManifest.xml* el permiso **BLUETOOTH**. Adicionalmente, se ha incluido el permiso **BLUETOOTH_ADMIN**, que permite a la aplicación modificar el estado del adaptador, así como crear nuevos dispositivos vinculados.

Para ofrecer la comunicación mediante el uso de Bluetooth, ha sido necesario implementar una nueva ventana donde mediante una lista se pueden ver los dispositivos previamente emparejados, así como los dispositivos visibles dentro del alcance. Para implementar esta funcionalidad, se ha utilizado las clases nativas Android **BlueetoothAdapter** (que representa el adaptador físico del dispositivo) y **BluetoothDevice** (representando un dispositivo remoto), explicadas a continuación.

BluetoothAdapter

Esta clase representa el adaptador Blueetooth del dispositivo local, permitiendo realizar las operaciones básicas como iniciar el descubrimiento de dispositivos, obtener la lista de dispositivos enparejados o iniciar una nueva conexión con un dispositivo remoto conociendo previamente su dirección MAC.

Adicionalmente, es posible realizar la comunicación a través de Bluetooth Low Energy utilizando esta misma clase. Aunque esta tecnología pueda ofrecer características importantes como el ahorro de batería, finalmente no se ha incorporado al proyecto debido a las limitaciones relativas a la velocidad de transferencia así como el rango de acción, siendo dos componentes críticos para el correcto funcionamiento del sistema de retransmisión buscado.

BluetoothDevice

Esta clase representa el dispositivo remoto permitiendo obtener información de dicho dispositivo, como el nombre, su dirección hardware, el estado de la conexión o crear una conexión con él. De esta forma, debido a que esta clase actúa como una representación de una dirección hardware Bluetooth remota, se puede generar una conexión socket mediante Blueetooth.

Gracias a la refactorización detallada en la sección 5.2.4, se ha podido implementar una clase BluetoothSocketP2p que cumple con la especificación de métodos definidos en la interfaz a la vez que extiende a la clase nativa *Thread*, permitiendo que todas las acciones se realicen de forma asíncrona y en segundo plano, liberando así el hilo principal de aplicación de forma que la interacción sea más fluida.

Capítulo 6

Experimentos y discusión

A lo largo de este capítulo se exponen las medidas experimentales del tiempo necesario para completar una retransmisión de la transacción utilizando los diferentes canales de comunicación así como la utilización de la retransmisión inteligente. Posteriormente se introducen otros posibles escenarios de ataques relativos a las tarjetas *contactless*.

6.1. Evaluación de tiempos

Una vez introducidos todos los cambios y mejoras en la aplicación `NFCLeech` se han realizado una serie de experimentos evaluando el rendimiento en función el tiempo necesario para realizar un pago. Para ello se han realizado varios pagos con la versión inicial de la aplicación (sin cacheo y utilizando WiFi como canal de comunicación) y la versión final(con cacheo y utilizando Bluetooth y WiFi) para realizar la retransmisión de las tramas, diferenciando entre tarjetas VISA y MASTERCARD.

Una vez obtenidos los resultados, se ha calculado la media aritmética $\overline{x}$, la desviación estándar $\mathbf{s}$ (véase la ecuación 6.1, donde n tiene valor 3 para los experimentos sin retransmisión inteligente y 5 mediante retransmisión inteligente) y el coeficiente de variación C_v (véase la ecuación 6.2).

$$\mathbf{s} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_1 - \overline{x})^2} \tag{6.1}$$

$$C_v = \frac{\mathbf{s}}{|\overline{x}|} \cdot 100 \tag{6.2}$$

Los resultados obtenidos se muestran en las Tablas 6.1, 6.2 y 6.3, donde se puede apreciar que el protocolo PayPass de MASTERCARD es ligeramente más lento a la hora de realizar el pago que el protocolo PayWave de VISA.

Adicionalmente, donde mejores resultados se observan respecto al tiempo medio para realizar un pago es en la retransmisión por Bluetooth. No obstante si se utiliza WiFi como canal de comunicación, los tiempos medios son muy parecidos, en torno a 20 *ms* más lento. Además, utilizando Bluetooth como medio de transmisión, se observa una

VISA	
Nº Exp.	**Tiempo (*ms*)**
1	961
2	1135
3	1043
$\overline{x}$	1041.52
s	87.05
C_v	8.36 %

MASTERCARD	
Nº Exp.	**Tiempo (*ms*)**
1	1221
2	1001
3	1138
$\overline{x}$	1112.45
s	111.10
C_v	9.99 %

Tabla 6.1: Tiempos con la versión inicial mediante WiFi

VISA	
Nº Exp.	**Tiempo (*ms*)**
1	698
2	740
3	619
4	679
5	759
$\overline{x}$	695.43
s	54.96
C_v	7.90 %

MASTERCARD	
Nº Exp.	**Tiempo (*ms*)**
1	803
2	710
3	696
4	707
5	837
$\overline{x}$	746.30
s	64.69
C_v	8.67 %

Tabla 6.2: Tiempos tras la implementación del cacheado mediante WiFi

desviación estándar más pequeña por lo que se puede asegurar que la comunicación es más estable y se ve menos afectada por perturbaciones externas.

Para finalizar, se observa una reducción notable (en torno a 400 *ms*) si se utiliza el cacheado previo de los mensajes (comparando los resultados obtenidos utilizando WiFi), por lo que se puede afirmar que dicha estrategia contribuye de manera efectiva a realizar con mayor éxito un ataque de retransmisión.

No se ha podido realizar un mayor número de experimentos debido a la amplitud del proyecto y falta de tiempo puesto que cada experimento debe realizarse de forma manual.

VISA	
Nº Exp.	**Tiempo (*ms*)**
1	665
2	653
3	676
4	697
5	667
$\overline{x}$	671.28
s	16.39
C_v	2.44 %

MASTERCARD	
Nº Exp.	**Tiempo (*ms*)**
1	704
2	694
3	755
4	766
5	693
$\overline{x}$	721.05
s	35.26
C_v	4.89 %

Tabla 6.3: Tiempos tras la implementación del cacheado mediante Bluetooth

6.2. Otros escenarios de ataques: Eavesdropping, DoS

A continuación, se detallan dos posibles riesgos de seguridad adicionales presentes en NFC o en las tarjetas *contactless* que aunque no tengan relación directa con los ataques de retransmisión, son de posible interés.

Eavesdropping

Eavesdropping, es un término en inglés utilizado en términos de seguridad cuya traducción al español significa *"escuchar secretamente"*, es decir, obtener información mediante ataques de escuchas a un medio que puede estar cifrado o no. En términos de comunicaciones mediante NFC, *eavesdropping* implica el uso de una antena de alta ganancia para capturar, visualizar y demodular la información contenida en las perturbaciones del medio. Dentro del problema de *eavesdropping*, se puede distinguir entre el ataque pasivo (el atacante escucha una comunicación legítima entre un datáfono y una tarjeta contactless) y el ataque activo (el atacante interactúa con la tarjeta).

La distancia máxima reportada en la literatura para un ataque pasivo es de 18 metros [10], conseguida mediante los armónicos de tercer orden y en unas condiciones de laboratorio ideales (sin apantallamiento entre la antena de escucha y la tarjeta). La distancia máxima conseguida para un ataque activo ha sido de 50 cm [11], pero de nuevo en condiciones muy controladas: la tarjeta se encontraba colocada bajo el arco formado por la antena del atacante y el más mínimo movimiento desajustando la orientación de la tarjeta inhabilitaba la comunicación.

Independientemente del tipo de ataque, este problema pone en manifiesto la clara necesidad de añadir una capa más de cifrado adicional para evitar que un tercero sea capaz de obtener la información transmitida simplemente escuchando el canal de comunicación.

Denegación de servicio

Los ataques de denegación de servicio (o DoS, por sus siglas en inglés), es un tipo de ataque a un sistema informático que imposibilita el acceso al servicio o recurso. En términos de los pagos con tarjetas *contactless*, una denegación de servicio sería la imposibilidad de utilizar dicha tarjeta para hacer un pago o retirada de efectivo.

Una denegación de servicio en una tarjeta de crédito/débito se puede conseguir mediante la introducción incorrecta del número PIN de forma consecutiva. Este hecho bloquea la tarjeta, inhabilitándola para su uso y obligando al titular a acudir a su banco para solicitar el desbloqueo.

La mayoría de las tarjetas *contactless* suelen incluir también el chip para realizar los pagos con contacto. En dichas tarjetas se puede configurar las funcionalidades que provee cada interfaz. Algunos estudios [21] han demostrado que en tarjetas de Inglaterra y Holanda existe una mala configuración permitiendo el uso del comando APDU *VERIFY*. Este comando APDU es el encargado de realizar la comprobación del PIN en la tarjeta.

Esto permite usar este comando de forma mal intencionada para provocar tres intentos de verificación fallidos, resultando en el bloqueo de la tarjeta y el cese del servicio legítimo.

Capítulo 7

Conclusiones y trabajo futuro

En conclusión, las tarjetas de crédito *contactless* incrementan su cuota de utilización cada día. Aunque el protocolo EMV ha introducido las bases para una mejor interoperabilidad así como la autorización por parte del titular introduciendo el PIN, las cuestiones relativas a la seguridad del protocolo recaen finalmente en la entidad financiera o bancaria que desarrolle el software de las tarjetas.

Adicionalmente, la posibilidad de realizar la comunicarse mediante NFC ofrece nuevas posibilidades de realizar nuevos vectores de ataques. Si es cierto que la mayoría de los riesgos de seguridad que existen en las tarjetas *contactless* ya estaban presentes en las tarjetas de chip, el permitir realizar la comunicación a través de NFC ha supuesto una reducción del coste económico necesario originado por la eliminación de la necesidad de adquirir hardware específico: cualquier dispositivo móvil actual incluye un chip NFC. Esta reducción del coste conjuntamente con la posibilidad de introducir dispositivos entre la comunicación legítima abre la posibilidad de estudiar el paso de mensajes del protocolo.

El protocolo EMV está accesible públicamente a través de Internet y no aporta por defecto las medidas de seguridad necesarias para evitar este estilo de ataques de retransmisión. A lo largo de la realización de este trabajo, ha surgido la necesidad de realizar una herramienta que aporte más información al desarrollador, lo que en definitiva facilita realizar más pruebas de auditorías sobre el protocolo NFC. Además, se ha desarrollado una prueba de concepto de ataques de retransmisión inteligente que ha permitido reducir el coste de la retransmisión de mensajes, haciendo este tipo de ataques más real.

Aunque la viabilidad de estos ataques en entornos reales se ve gravemente influenciada debido al pequeño rango de acción del NFC y a las interferencias que se pueden originar durante la comunicación, los riesgos de seguridad relativos a los pagos y a la información privada del titular de la tarjeta pueden convertirse en un riesgo real con la adquisición de hardware más específico.

7.1. Trabajo futuro

Aunque la aplicación `NFCLeech` ha mejorado en estabilidad y en madurez, existen multitud de mejoras a introducir en el futuro. Por ejemplo, añadir la posibilidad de realizar ataques de *replay* realizando de esta manera un clonado de la tarjeta introduciendo cargos falsos en un PCD. Otra posible línea de investigación futura puede ser aprovechar las características que ofrece un dispositivo móvil Android y realizar un ataque distribuido para adivinar el CVV de la tarjeta permitiendo realizar pagos en ciertos servicios *e-commerce* o pasarelas de pago en multitud de páginas webs, de manera similar al proceso descrito en [2].

Bibliografía

[1] Límites de pago sin necesidad de PIN en tarjetas contactless. [Online; `https://en.wikipedia.org/wiki/Contactless_payment#Floor_limit`]. Accedido: Junio 2018.

[2] M. A. Ali, B. Arief, M. Emms, and A. van Moorsel. Does the Online Card Payment Landscape Unwittingly Facilitate Fraud? *IEEE Security Privacy*, 15(2):78–86, March 2017.

[3] J. V. Bausili. Ataques de relay en NFC con dispositivos Android. techreport, Universidad de Zaragoza, 2014.

[4] T. Chothia, F. D. Garcia, J. de Ruiter, J. van den Breekel, and M. Thompson. Relay Cost Bounding for Contactless EMV Payments. In R. Böhme and T. Okamoto, editors, *Financial Cryptography and Data Security*, pages 189–206, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[5] Consejo Económico y Social. Nuevos hábitos de consumo, cambios sociales y tecnológicos. Technical report, Gobierno de España, Apr. 2016.

[6] J. H. Conway. *On Numbers and Games*. AK Peters Ltd./CRC Press, 2nd edition, Dec. 2000.

[7] S. Drimer and S. J. Murdoch. Keep your enemies close: Distance bounding against smartcard relay attacks. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 7:1–7:16, Berkeley, CA, USA, 2007. USENIX Association.

[8] EMVCo. *EMV Contactless Specifications for Payment Systems. Book A, Architecture and General Requirements*, 2016.

[9] EMVCo. *EMV Contactless Specifications for Payment Systems. Book D, EMV Contactless Communication Protocol Specification*, 2016.

[10] M. Engelhardt, F. Pfeiffer, K. Finkenzeller, and E. Biebl. Extending ISO/IEC 14443 Type A Eavesdropping Range using Higher Harmonics. In *Smart SysTech 2013; European Conference on Smart Objects, Systems and Technologies*, pages 1–8, June 2013.

[11] R. Habraken, P. Dolron, E. Poll, and J. Ruiter. An RFID Skimming Gate Using Higher Harmonics. In *Revised Selected Papers of the 11th International Workshop on Radio Frequency Identification - Volume 9440*, RFIDsec 2015, pages 122–137, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[12] G. Hancke. A Practical Relay Attack on ISO 14443 Proximity Cards. Technical report, University of Cambridge, Jan. 2005.

[13] HSN Consultants, Inc. The Nilson Report, Oct. 2017.

[14] International Organization for Standardization. *ISO/IEC 13239:2002: Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures.* Geneva, Switzerland, July 2002.

[15] International Organization for Standardization. *ISO/IEC 7810:2003: Identification cards – Physical characteristics.* Geneva, Switzerland, Nov. 2003.

[16] International Organization for Standardization. *ISO/IEC 14443-2:2016: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal interface.* Geneva, Switzerland, July 2016.

[17] International Organization for Standardization. *ISO/IEC 14443-3:2016: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision.* Geneva, Switzerland, June 2016.

[18] International Organization for Standardization. *ISO/IEC 14443-4:2016: Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol.* Geneva, Switzerland, June 2016.

[19] International Organization for Standardization. *ISO/IEC 14443-1:2018: Cards and security devices for personal identification – Contactless proximity objects – Part 1: Physical characteristics.* Geneva, Switzerland, Apr. 2018. `https://www.iso.org/standard/73596.html`.

[20] W. Issovitsy and M. Hutter. Weaknesses of the ISO/IEC 14443 protocol regarding relay attacks. In *Proceedings of the IEEE International Conference on RFID-Technologies and Applications (RFID-TA) 2011*, pages 335–342, 2011.

[21] E. P. Jordi van den Breekel, Diego A. Ortiz-Yepes and J. de Ruiter. EMV in a nutshell. techreport, KPMG, IBM Research Zurich, Radboud University Nijmegen, 2016.

[22] N. Klaus. Q&A about NFC cards – security of NFC cards in 2016. [Online; `https://www.nixu.com/nl/node/109`], 2016. Accedido el 19 de septiembre de 2018.

[23] H. S. Kortvedt and S. F. Mjølsnes. Eavesdropping Near Field Communication. In *The Norwegian Information Security Conference (NISK)*, volume 27, page 5768, 2009.

[24] L. Sportiello and A. Ciardulli. Long distance relay attack. In M. Hutter and J.-M. Schmidt, editors, *Radio Frequency Identification*, pages 69–85, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[25] B. Technology. Origin of the Bluetooth Name. [Online; `https://www.bluetooth.com/about-us/bluetooth-origin`]. Accedido el 19 de septiembre de 2018.

[26] J. Vila and R. J. Rodríguez. Practical experiences on NFC Relay Attacks with Android: Virtual Pickpocketing Revisited. In *Revised Selected Papers of the 11th International Workshop on Radio Frequency Identification - Volume 9440*, RFIDsec 2015, pages 87–103, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[27] N. World. NFC phones: The definitive list, June 2018. `https://www.nfcworld.com/nfc-phones-list/`.

Apéndice A

Extensión temporal del proyecto

El tiempo necesario para la realización de este proyecto ha sido más extenso de lo normal debido a que durante la gran parte del periodo de realización he estado trabajando así como evaluándome de las asignaturas que tenía pendientes, por lo que no se pudo tener una dedicación diaria regular. Adicionalmente, durante la primera fase del proyecto, el estudio de documentación se ha prolongado debido a la gran cantidad de conceptos nuevos y términos que dificultaban la comprensión de los artículos.

Durante la realización del proyecto, el proceso también se ha visto alargado debido a la necesidad de familiarizarse con el código obtenido y el estudio de la viabilidad de las estrategias estudiadas en el capítulo 4.

En la Figura A.1 se puede ver una representación gráfica de la extensión temporal.

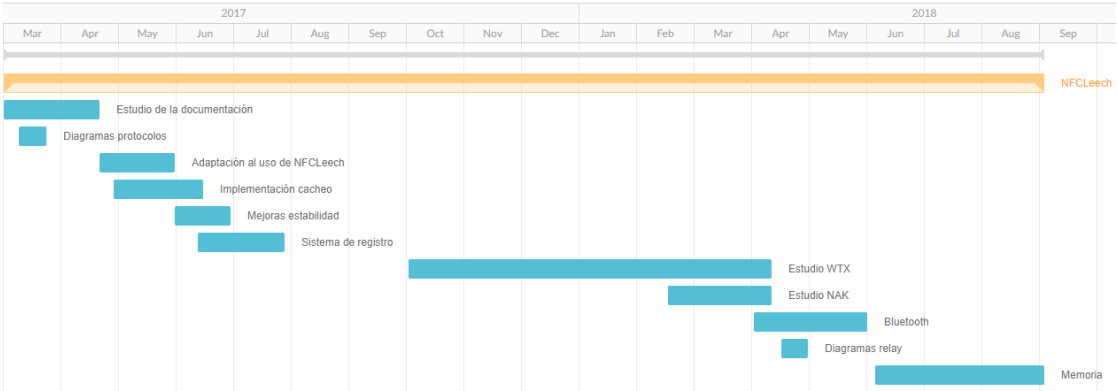

Figura A.1: Diagrama de Gantt.

Apéndice B

Traza de ejecución

```
INFO Messenger registered.
INFO Socket stablished.
INFO Waiting for remote tag...
INPUT Remote tag attached.
INFO Waiting for PCD request...
AID0 received from ICC: Proximity
    Payment System Environment -
    PPSE (2PAY.SYS.DDF01)
AID1 received from ICC: MasterCard
    Card Manager
AID2 received from ICC: MasterCard
    Credit/Debit (Global)
RELAY SEARCH FINISHED, 3 AIDs FOUND
```

```
  0 PCD -> ICC
```

```
CLA: [00] Inter-Industry Command
INS: [A4] SELECT
P1:  [04]
P2:  [00]
Lc:  [0E]
DATA:[325041592E5359532E4444463031]
Le:  [00]
HEX:
00 A4 04 00 0E 32 50 41 59 2E 53 59 53
2E 44 44 46 30 31 00
```

```
  1 RELAY -> PCD
```

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
6F 40 84 0E 32 50 41 59 2E 53 59 53 2E
44 44 46 30 31 A5 2E BF 0C 2B 61 29 4F
07 A0 00 00 00 04 10 10 50 10 44 45 42
49 54 20 4D 41 53 54 45 52 43 41 52 44
87 01 01 9F 0A 08 00 01 05 01 00 00 00
00 90 00
```

```
  1 PCD -> ICC
```

```
CLA: [00] Inter-Industry Command
INS: [A4] SELECT
P1:  [04]
P2:  [00]
Lc:  [07]
DATA:[A0000000041010]
Le:  [00]
HEX:
00 A4 04 00 07 A0 00 00 00 04 10 10 00
```

```
  2 RELAY -> PCD
```

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
6F 38 84 07 A0 00 00 00 04 10 10 A5 2D
50 10 44 45 42 49 54 20 4D 41 53 54 45
52 43 41 52 44 87 01 01 BF 0C 15 9F 6E
07 07 24 00 00 30 30 00 9F 0A 08 00 01
05 01 00 00 00 00 90 00
```

```
  2 PCD -> ICC
```

```
CLA: [80] Proprietary Command
INS: [A8] GET PROCESSING OPTIONS
P1:  [00]
P2:  [00]
Lc:  [02]
DATA:[8300]
Le:  [00]
HEX:
80 A8 00 00 02 83 00 00
```

```
  3 ICC -> PCD
```

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
77 0E 82 02 19 80 94 08 10 01 01 01 20
01 04 00 90 00
```

```
  3 PCD -> ICC
```

```
CLA: [00] Inter-Industry Command
INS: [B2] READ RECORD
P1:  [01]
P2:  [14]
Lc:  [00]
```

```
DATA: [None]
Le:   [00]
HEX:
00 B2 01 14 00
```

4 ICC –> PCD

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
70 81 87 5F 25 03 18 06 25 5F 24 03 19
09 30 9F 07 02 FF C0 5A 08 54 27 38 00
66 16 30 15 5F 34 01 03 8E 0E 00 00 00
00 00 00 00 00 42 03 1E 03 1F 03 9F 0D
05 B4 50 84 00 00 9F 0E 05 00 00 00 00
00 9F 0F 05 B4 70 84 80 00 8C 27 9F 02
06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A
03 9C 01 9F 37 04 9F 35 01 9F 45 02 9F
4C 08 9F 34 03 9F 21 03 9F 7C 14 8D 0C
91 0A 8A 02 95 05 9F 37 04 9F 4C 08 5F
28 02 07 24 9F 4A 01 82 90 00
```

4 PCD –> ICC

```
CLA:  [00] Inter-Industry Command
INS:  [B2] READ RECORD
P1:   [01]
P2:   [24]
Lc:   [00]
DATA: [None]
Le:   [00]
HEX:
00 B2 01 24 00
```

5 ICC –> PCD

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
70 81 B4 9F 46 81 B0 55 0F 5B 5C 48 7F
AF 8B 34 B1 7D AA BB 82 C1 FE B1 94 43
30 B3 93 FB CD CB 37 2F CA AE 22 C8 D5
4F F9 31 FE 83 3D A7 51 5A A2 68 AD BE
79 62 10 BE 9F 17 47 AE 69 E6 69 F1 CD
1C 98 77 F7 B1 6C FF 1D 52 85 9B C5 E8
33 E1 94 B6 EE 40 E4 C8 C5 85 6D 8C B8
69 CC C0 2E 20 DB EB 35 E6 63 63 99 B8
D2 FA 9C BE FD 67 0E 4F D9 FD B6 93 27
82 BF 9E CB F0 2B 68 1C 9A 0A 0D ED 18
75 66 CB AD ED 05 02 F7 9B 78 78 BD 6F
31 53 95 2B 34 51 8C 43 E4 CC 2F B0 96
86 26 F6 2C E3 B3 87 F9 5D 85 0C 64 E5
95 4E C9 F9 35 BD 2C A9 88 46 DE 76 C0
60 90 00
```

5 PCD –> ICC

```
CLA:  [00] Inter-Industry Command
INS:  [B2] READ RECORD
P1:   [02]
P2:   [24]
Lc:   [00]
DATA: [None]
Le:   [00]
HEX:
00 B2 02 24 00
```

6 ICC –> PCD

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
70 09 9F 42 02 09 78 9F 44 01 02 90 00
```

6 PCD –> ICC

```
CLA:  [00] Inter-Industry Command
INS:  [B2] READ RECORD
P1:   [03]
P2:   [24]
Lc:   [00]
DATA: [None]
Le:   [00]
HEX:
00 B2 03 24 00
```

7 ICC –> PCD

```
SW1&2: [90 00] Process Completed
    (Success)
PAN: XXXXXXXXXXXXXXXX
Expiration date (MM/YY): 09/19
Service code: 210
HEX:
70 4B 9F 08 02 00 02 57 13 XX XX XX XX
XX XX XX XX D1 90 92 21 00 00 05 82 00
00 0F 8F 01 05 9F 32 01 03 92 24 5E 00
E8 FF FB E6 F2 55 8D 8E 16 E9 44 00 14
A7 F5 E6 8E A1 88 D9 7B 31 80 62 8A EE
A3 08 06 29 90 CE 09 67 9F 47 01 03 90
00
```

7 PCD –> ICC

```
CLA:  [00] Inter-Industry Command
INS:  [B2] READ RECORD
P1:   [04]
P2:   [24]
Lc:   [00]
DATA: [None]
Le:   [00]
HEX:
00 B2 04 24 00
```

8 ICC –> PCD

```
SW1&2: [90 00] Process Completed
    (Success)
HEX:
70 81 B3 90 81 B0 4E 75 5C D9 46 8F EA
55 9B 90 3F 72 30 4D 70 86 BB 67 39 24
5B FA 83 BD 6E 31 5B 9B 70 4B 96 E5 36
26 22 DE B4 23 97 9F 4E 02 C6 B3 C6 FC
64 F8 26 E9 C0 4D BB 1C B3 70 FA 80 4A
```

```
13 55 2E AF 57 B5 B4 F7 2D 86 30 4B CD
6F E4 D6 5E 19 13 30 17 3E 91 C4 34 51
EC D4 20 2C 9B 42 EA D3 F7 AD 2B AB 66
F5 07 AF 1F 17 B2 6F 11 82 A5 28 25 A9
3C 74 08 1D 3A 78 0A DA 7C CD 83 D4 8C
1F 2C D5 AA 75 C2 74 75 8E D5 95 C3 38
A0 FC 86 8A 97 60 83 BE 39 1C 03 42 58
E2 0A 22 50 BC D5 F0 4E 84 97 49 0A 35
E4 F7 47 B7 55 81 EE 44 30 B6 D8 ED A4
90 00
```

8 PCD –> ICC

```
CLA: [80] Proprietary Command
INS: [AE] GENERATE APPLICATION
     CRYPTOGRAM
P1:  [90]
P2:  [00]
Lc:  [42]
DATA:[000000000001000000000000072400000
0800009781809200009 9F366222000000000000
000000001F030219242500000000000000000000
00000000000000000000]
Le:  [00]
HEX:
80 AE 90 00 42 00 00 00 00 00 01 00 00
00 00 00 00 07 24 00 00 00 80 00 09 78
18 09 20 00 09 9F 36 62 22 00 00 00 00
00 00 00 00 00 00 1F 03 02 19 24 25 00
00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00
```

9 ICC –> PCD

```
SW1&2: [90 00] Process Completed
     (Success)
HEX:
77 81 A2 9F 27 01 80 9F 36 02 00 54 9F
4B 81 80 1D C3 2F 46 92 6B 15 86 79 42
0F 8D 64 00 84 F4 1B 15 61 ED 5E 39 54
10 D9 60 85 F1 AF 31 EB 8C 9D 2D F5 C3
6D 69 2D C3 37 3B 42 F6 4D 16 D0 6A D9
7A BD C8 33 2D FE 4A 2F B1 CA DF 4B 87
CB AB CC 91 2F A7 41 9B F3 8B 13 4E A9
35 AD AE 25 34 28 A8 36 62 7B 26 52 45
AF AE 6B 15 97 26 C2 D3 36 B9 79 D2 3C
0F FD 0B 68 DA 71 38 A5 54 FE 5F EE 9E
44 8C B9 41 37 C0 F9 D1 18 06 E8 72 A6
C0 9F 10 12 01 14 A0 40 03 22 00 00 00
00 00 00 00 00 00 00 00 FF 90 00
Relay time: 799 ms.
PCD TIMEOUT RECEIVED!
```

Glosario

AID Application IDentifier 14, 23, 24
APDU Application Protocol Data Unit 7, 13, 22, 24, 34

BLE Bluetooth Low Energy 5, 29

CRC Cyclic Redundancy Check 19, 20
CVV Card Verification Value 9, 36

DES Data Encryption Standard 9

EDC Error Detection Code 17, 19, 20
EMV EUROPAY, MASTERCARD y VISA 2, 8, 9, 12, 14, 17, 20, 35

FWI Frame Waiting time Integer 17
FWT Frame Waiting Time 17–19

GPO Get Processing Options 13, 14

HCE Host-card-emulation 6
HF alta frecuencia 6

IEC International Electrotechnical Commission 7
INF INFormation field 17–19
ISM Industrial, Científica y Médica 5
ISO International Organization for Standardization 7

JTC Comité Técnico Conjunto 7

N/A No aplicable 18
NAK Not AcKnowledge 19, 20

NFC Near Field Communication 1, 2, 6, 9, 14, 23, 33, 35

PAN Primary Account Number 9
PCB Protocol Control Byte 17–20
PCD Proximity Coupling Device 6, 8, 11, 14, 17–20, 22, 24, 36
PICC Proximity Integrated Circuit Card 6, 8, 11, 14, 17, 18, 20, 22, 24

RFID Radio Frequency IDentification 1, 6, 7
RSA Rivest, Shamir y Adleman 9

SHA Secure Hash Algorithm 9

UHF Ultra Alta Frecuencia 5

WTX Waiting Time eXtension 18, 19
WTXM Waiting Time eXtension Multiplier 18, 19