Seguridad y Protección de Sistemas Informáticos (2023-24)
Grado en Ingeniería Informática
Universidad de Granada

Prueba fuerza criptográfica a tarjetas RFID

Mario Daniel Castro Almenzar
Rubén De la Torre Vico
Jose Luis Montes Ramón

20 de diciembre de 2023

UNIVERSIDAD DE GRANADA

Índice

Índice de figuras

1. Introducción y contexto

Actualmente, la electrónica se ha integrado tan profundamente en nuestra vida cotidiana que su presencia a menudo pasa desapercibida. La mayoría de las personas utilizan una variedad de chips y circuitos electrónicos de manera inconsciente para realizar una amplia gama de actividades diarias.

Desde el microchip implantado en las mascotas por el veterinario hasta la tarjeta que empleamos cotidianamente en el transporte público, todos estos avances son posibles gracias a la tecnología que exploraremos en las siguientes secciones.

Sin embargo, antes de adentrarnos en el tema, es crucial reconocer la diversidad de tarjetas existentes y entender que no todas operan de la misma manera. Por lo tanto, en el siguiente apartado, realizaremos una distinción técnica para facilitar una comprensión más precisa de los contenidos específicos de esta lectura.

1.1. Introducción a los distintos tipos de tarjetas/chips electrónicos

En primer lugar debemos distinguir entre 3 principales tipos de tarjetas, ya que podrían llegar a confundirnos. Estos son:

- **Banda magnética[8]:** estas tarjetas, introducidas en la década de 1950, representan una de las primeras formas de tarjetas digitales. Su funcionamiento se basa en el almacenamiento de información en una tira de material magnético. La lectura o modificación de datos se realiza mediante un lector o escritor magnético, que aplica un campo magnético sobre la tira para alterar la orientación de los electrones, codificando así la información en forma de bits. Aunque estas tarjetas aún se utilizan, su popularidad ha disminuido frente a alternativas más modernas, que ofrecen mejoras en costos, rendimiento y comodidad.

- **Tarjeta con chip integrado[12]:** estas tarjetas incorporan un microchip embebido, lo que les permite almacenar y procesar datos de manera más segura y eficiente comparadas con las tarjetas de banda magnética. Las tarjetas con chip integrado ofrecen una mayor protección contra el fraude, ya que el chip es capaz de realizar autenticaciones complejas. Son ampliamente usadas en tarjetas bancarias y de crédito, mejorando significativamente la seguridad de las transacciones electrónicas.

- **Tarjetas RFID (Radio-frequency identification)[1]:** estas son las tarjetas más usadas a día de hoy, estas serán las que explicaremos más en profundidad durante el resto del documento. Estas funcionan por radiofrecuencia, como el propio nombre indica. Por lo que no se necesita un contacto con el dispositivo lector o escritor para la comunicación, lo que a como veremos que divide a esto en distintos tipos en función de la frecuencia de las ondas usadas. El circuito integrado en estas tarjetas consiste en un pequeño transpondedor, un recibidor de radio y un transmisor; que esto de primeras puede parecer que es muy complejo pero como se puede ver todo cabe en una pequeña tarjeta, y los costes son muy bajos lo que lo hace perfecto para sistemas de identificación, como las tarjetas de los hoteles, o sistemas de pago, como podría ser el transporte público.

(a) Tarjeta Banda Magnética (b) Tarjeta con Chip (c) Tarjeta RFID

Figura 1: Comparación de Tipos de Tarjetas

1.2. Distinción en RFID

Dentro del marco de los distintos tipos de tarjetas RFID, se encuentran dispositivos que funcionan en el rango de frecuencias de 125kHz hasta 928MHz, además de diferentes capacidades de memoria que rondan desde 1 bit hasta 4 kilobytes. Puesto que enumerar todos los tipos de tarjetas existentes no nos resulta útil, pasamos a enumerar *a grosso modo* los tipos más comunes y conocidos para poder diferenciarlas a la hora de ver el criptosistema.

Tipo	Frecuencia	Ejemplo
LF (Low Frequency)	120-125 kHz	Microchip de Mascotas, Gestión de inventario
HF (High Frequency)	13.56 MHz	NFC, Smart Cards, Tarjeta de transporte
UHF (Ultra High Frequency)	433MHz, 865-868MHz	Pulseras NFC y otros usos militares

Tabla 1: Tipos de tarjetas RFID

2. RFID

2.1. Fundamenos del sistema RFID

Radio Frequency Identification, más conocido como RFID[1] es un sistema utilizado para el almacenamiento y recuperación de datos. Utiliza dispositivos denominados como etiquetas o traspondedores. Su objetivo se basa en la transmisión de la identidad de un objeto a través de ondas de radio.

Las **etiquetas** RFID son pequeños chips, a veces con forma de pegatina, que se adhieren a un producto o ser vivo. Estas contienen unas pequeñas antenas que permiten la recepción y/o emisión de peticiones por radiofrecuencia. Podemos distinguir entre etiquetas pasivas (no necesitan alimentación eléctrica interna) y activas.

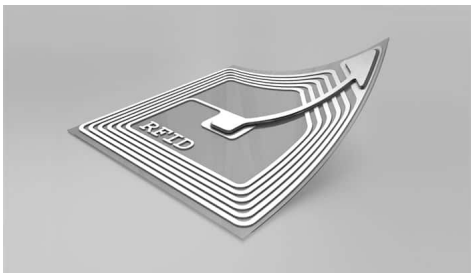


Figura 2: Diseño de Etiqueta RFID

2.2. Introducción a la tecnología NFC

2.2.1. Fundamentos de NFC

Near Field Communication (NFC) es una extensión de la tecnología RFID que permite la comunicación inalámbrica de corto alcance (2 cm o 1 pulgada). Mientras que RFID se utiliza ampliamente para el seguimiento y la identificación, NFC está diseñada para ser una forma segura de realizar transacciones, la apertura de puertas, y simplificar la configuración de dispositivos inalámbricos.

Su diseño cumple con estándares internacionales como la ISO 14443 [6], garantizando la interoperabilidad y fiabilidad de las conexiones. Su integración en dispositivos va más allá de los *smartphones*, incluyendo relojes inteligentes, *tablets*, joyas, *earbuds* y más, gracias a su naturaleza de bajo costo y su capacidad para funcionar sin fuentes de energía propias.

2.2.2. Clasificación de Dispositivos NFC

Los dispositivos NFC se clasifican principalmente en dos tipos:

- **NFC pasivos:** no poseen fuente de energía propia y se activan únicamente cuando entran en el campo de un dispositivo NFC activo. Estos dispositivos son comunes para el uso de aplicaciones como carteles interactivos, anuncios y sistemas de acceso, e incluyen etiquetas NFC y transmisores como **MIFARE Classic**, **MIFARE Ultralight**, y etiquetas de tipos 1 a 5.

- **NFC activos:** pueden enviar y recibir datos de forma independiente, ya que cuentan con su propia fuente de energía. Se usan para la comunicación con otros dispositivos activos o pasivos. Algunos ejemplos son ciertos modelos de *smartphones*, lectores de tarjetas en sistemas de transporte y terminales de punto de venta sin contacto. Usan tecnologías como **MIFARE DESFire** o tarjetas inteligentes con capacidades avanzadas de procesamiento.

Por ejemplo, los *smartphones* admiten tres modos principales de operación con NFC [3]:

- **Modo lector-escritor:** permite al dispositivo leer etiquetas de forma pasiva (actuando como un dispositivo NFC pasivo) y escribir en ellas activamente (funcionando como un dispositivo NFC activo).
- **Modo P2P (peer-to-peer):** habilita el intercambio de datos entre dispositivos NFC, permitiendo la comunicación bidireccional.
- **Modo de emulación de tarjeta:** simula ser una tarjeta NFC pasiva, siendo accesible por lectores NFC externos, como terminales de punto de venta sin contacto.

2.2.3. Comparativa con otras tecnologías

Criterio	**NFC**	**Bluetooth**
Consumo de energía	Bajo	Alto
Alcance	Corto (hasta 10 cm)	Largo (hasta 10 metros)
Velocidad de transferencia de datos	Más lenta (hasta 424 kbit/s)	Más rápida (hasta 2.1 Mbit/s para Bluetooth 2.1 y 1 Mbit/s para Bluetooth de Baja Energía)
Conectividad	Más rápida (menos de 1s para establecer una conexión)	Más lenta (puede requerir emparejamiento manual)
Adecuado para	Dispositivos pasivos como etiquetas y carteles publicitarios	Transferencia de archivos, compartir conexiones con altavoces, y más
Aplicaciones	Pagos móviles (Samsung Pay, Android Pay, Apple Pay)	Compartir archivos, streaming de música, conectar dispositivos

Tabla 2: Comparativa de tecnologías NFC y Bluetooth

2.3. MIFARE Classic

Las tarjetas **MIFARE Classic** están basados en la tecnología RFID y comunicación NFC, y son ampliamente utilizadas a nivel mundial, como por ejemplo en sistemas de tarjetas inteligentes para el transporte público, accesos, etc. Operan en una frecuencia de 13.56 MHz y se basan en la normativa ISO/IEC 14443 [6]. Existen varias versiones, siendo las más comunes MIFARE Classic 1k y 4k, que ofrencen 1 y 4 kilobytes de memoria EEPROM, respectivamente.

2.3.1. Estructura lógica de la tarjeta

La tarjeta **MIFARE Classic** es una tarjeta de memoria con funciones adicionales, dividida en bloques y sectores. Los bloques de datos contienen un identificador único (UID) y otros datos, incluidos los del fabricante. Cada sector incluye un tráiler con claves secretas para la autenticación y condiciones de acceso específicas. Algunos bloques pueden almacenar datos arbitrarios o configurarse como bloques de valor, con un sistema de almacenamiento que involucra la inversión de bits.

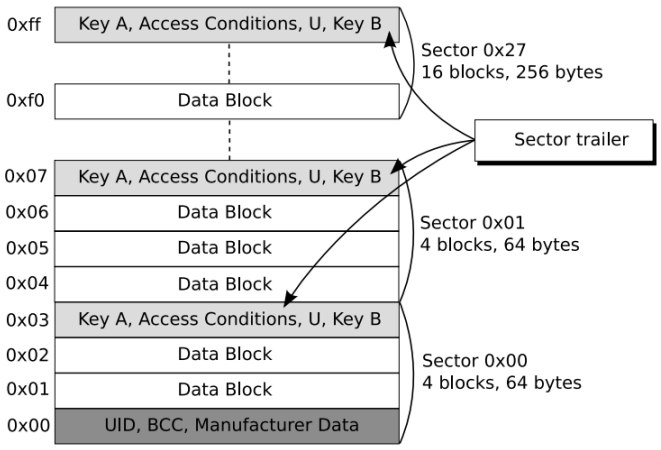


Figura 3: Estructura lógica de la tarjeta

2.3.2. Comandos Básicos

Los comandos de Mifare Classic son limitados y la mayoría están relacionados con un bloque de datos, requiriendo autenticación para el sector que los contiene. Las condiciones de acceso se verifican con cada comando para determinar su permisibilidad. Un bloque de datos puede configurarse como solo lectura y, por ejemplo, un bloque de valor puede solo disminuirse.

- **Leer y Escribir:** manejan un bloque de datos, ya sea un bloque de datos normal o un bloque de valor. El comando de escritura también puede formatear un bloque de datos como bloque de valor o almacenar datos arbitrarios.
- **Decrementar, Incrementar, Restaurar y Transferir:** solo se permiten en bloques de datos formateados como bloques de valor. Los comandos de incremento y decremento modifican un bloque de valor con un valor dado, colocando el resultado en un registro de memoria. El comando de restauración carga un valor en el registro de memoria sin cambios. Finalmente, el registro de memoria se transfiere al mismo bloque o a otro mediante el comando de transferencia.

Authentication

READER	*CARD*	*READER*	*CARD*
60 YY* Using KeyA	4-byte nonce	8-byte response	4-byte response
61 YY* Using KeyB	4-byte nonce	8-byte response	4-byte response

Data

READER	*CARD*	*READER*
30 YY* Read	16 data bytes*	
A0 YY* Write	ACK / NACK	16 data bytes*

Value blocks

READER	*CARD*	*READER*	*READER*
C0 YY* Decrement	ACK / NACK	4-byte value*	Transfer
C1 YY* Increment	ACK / NACK	4-byte value*	Transfer
C2 YY* Restore	ACK / NACK	4-byte value*	Transfer
B0 YY* Transfer	ACK / NACK		

Other

READER
50 00* Halt

YY = block address
* = Followed by two CRC bytes

Card responses (ACK / NACK)

A (1010) ACK
4 (0100) NACK, not allowed
5 (0101) NACK, transmission error

Figura 4: Comandos de MIFARE Classic

3. Seguridad en RFID

Desde que los dispositivos RFID son usados para almacenar información de carácter personal o confidencial como dinero físico, posesiones, o es implantado en animales y personas; obliga a los estándares de esta tecnología implementar mecanismos que aumenten la seguridad de esta. Algunos temas importantes a tratar por los estándares son:

- **Rastreo ilícito:** el hecho de que los chips sean legibles por todo el mundo con un dispositivo activo genera el problema de que información privada pueda ser relevada a una distancia no trivial dependiendo de la frecuencia. O incluso en el caso de realizar transacciones entre dos dispositivos esta podría ser escuchada por un intermediario no autorizado, por esto mismo las que realizan este tipo de operaciones suelen actuar a distancias menores, como la tecnología NFC.
- **Replicación de mensajes:** un atacante podría replicar un mensaje grabado de manera ilícita entre la etiqueta y el lector. Para solucionar esto se usan distintos mecanismos criptográficos como el código evolutivo o protocolos de desafío-respuesta. El uso de código evolutivo trata principalmente de ir cambiando la identificación de cada etiqueta después de cada vez que se pregunta. Mientras que los mecanismos de desafío-respuesta son usados para que la respuesta se encuentre cifrada mediante un cifrado simétrico o asimétrico.

Estos problemas generales se intentan solucionar en distintos con distintos protocolos seguros sugeridos para la comunicación pero depende del tipo de comunicación que se use y la importancia que le dé el administrador del sistema a la seguridad, que muchos casos por desgracia es bastante escaso. Para el caso que nos acontece como ya hemos dicho nos centraremos en dispositivos de alta frecuencia ya que son mucho más flexibles y permiten hacer operaciones más complejas.

3.1. Seguridad en NFC

Para comenzar a comentar cómo el estándar ISO 18092 [5] solventa específicamente los problemas presentados en el apartado anterior es que esté no solventa ninguno de estos problemas. Esto es debido a que al estar las conexiones NFC limitado a unos poco centímetros de distancia para realizar una conexión el atacante debería posicionar una antena a unos pocos centímetros entre el dispositivo pasivo y el activo que realizan la comunicación.

Por lo que para establecer un canal de comunicación seguro se debe usar un protocolo cirptográfico estándar como *TLS*, *IPsec* o cualquier otro que pueda ser usado para securizar el canal de comunicación. Por lo que para establecer mecanismos de seguridad se deja a disposición de lo que el fabricante de las mismas desee implementar.

En nuestro caso vamos a hablar de la seguridad que nos proporciona el modelo **MIFARE Classic** que es probablemente la tarjeta *contactless* más utilizado a nivel mundial y que contempla graves problemas seguridad debido al algoritmo privativo que usa para cifrar la conmunicación.

3.2. Seguridad en MIFARE Classic

Una vez se entiende como funciona la MIFARE Classic y que los mecanismos de seguridad para estos dispositivos son una recomencación, hay que entender los mecanismos de seguridad posee incorporada en ella:

- **UID sólo lectura.**
- **Claves**: como ya se ha visto en la estructura lógica de la tarjeta esta almacena al final de cada sector un par de claves llamadas A y B de 48 bits cada una, usadas para la autenticación para poder realizar cualquier operación con los datos de dicho sector.
- **Protocolo de autenticación**: según la documentación oficial de MIFARE se basan en el estándar ISO 9798-2, pero según investigaciones posteriores esto no parece ser cierto (al ser software privado no se puede saber con certeza comprobando el código de implementación al no estar público). La autenticación se realiza en dos pasos:

1. El lector envía la petición de autenticación para un sector, primero indica el byte `0x60` y `0x61` para autenticarse con la clave A o clave B, respectivamente. Seguido de la dirección del sector en el que autenticarse y los bytes para la paridad.
2. La tarjeta responde con un número de 4 bytes pseudoaleatorios arbitrario n_T (en conceptos criptográficos se conoce como *nonce*).
3. El lector responde con un número de 8 bytes, el cual es su propio reto n_R y la respuesta al reto a_R.
4. Y para finalizar la autenticación la tarjeta responde al reto con un número de 4 bytes, a_T.

A partir del segundo paso toda la comunicación ya se encuentra cifrada por lo que n_T, a_R y a_T, por lo que la comunicación sería segura frente a escuchas no autorizadas desde casi el inicio.

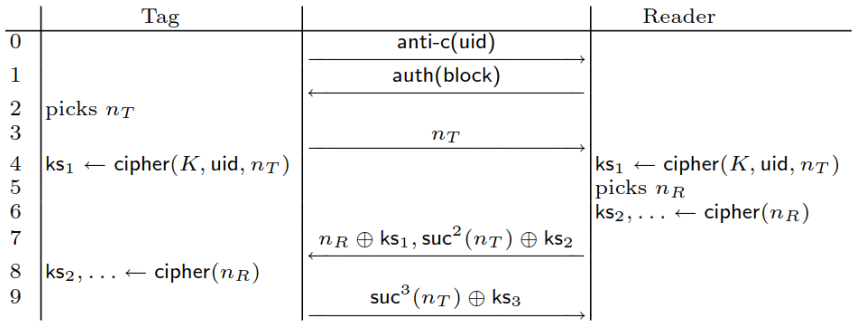


Figura 5: Autenticación en MIFARE Classic

3.3. Debilidad de la MIFARE Classic

Nohl y Plötz, dos investigadores Alemanes expertos en hacking y RFID, mediante el uso de mecanismos de ingeniería inversa consiguieron extraer parcialmente el funcionamiento del criptosistema privado que usa la MIFARE Classic para cifrar las comunicaciones, el conocido como **CRYPTO1**[2], el cual ya ha sido vulnerado de distintas maneras.

3.4. CRYPTO1

Para comenzar hay que entender que este algoritmo toma como entrada un texto plano y lo convierte en un cifrado, mediante la operación XOR entre la entrada y la clave de cifrado. Este tipo de cifrado es muy sencillo y común ya que la operación XOR con la misma clave de cifrado sirve para descifrar.

La generación de esta clave depende de un desplazamiento de registro con retroalimentación lineal (LFSR) u unas funciones de filtrado, pero nosotros nos centraremos en el

LFSR y su uso de generador de números pseudo-aleatorios.

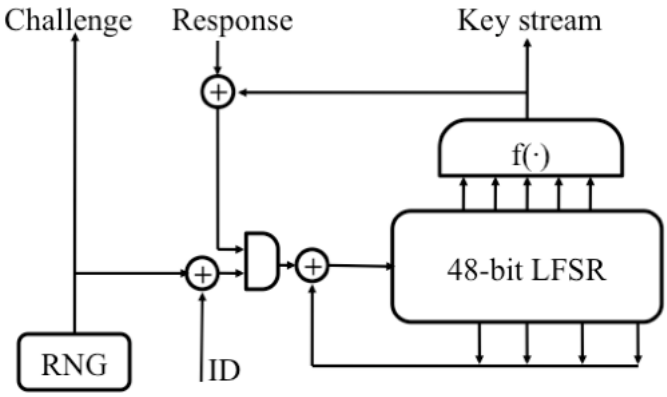


Figura 6: Esquema general de CRYPTO1

El LFSR lo se basa en desplazar todos los bits una posición a la izquierda, descartando el que más a la izquierda se encontraba e introduciendo uno nuevo por la derecha. Este nuevo bit proviene de aplicar una función de transformación lineal de un estado anterior.

Este nuevo bit es el resultado de aplicar una operación XOR entre el *UID* de la tarjeta (en la inicialización) o la respuesta a la anterior comunicación. Y un bit de realimentación es calculado mediante un conjunto de operaciones XOR a determinados bits del estado anterior.

A la posición de los bits escogidos del estado anterior se pueden representar de manera algebraica como un polinomio en $\mathbb{Z}_2$, conocido como el *polinomio generativo* que en este caso es el siguiente: $g(x) = x^{48} + x^{43} + x^{39} + x^{38} + x^{36} + x^{34} + x^{33} + x^{31} + x^{29} + x^{24} + x^{23} + x^{21} + x^{19} + x^{13} + x^9 + x^7 + x^6 + x^5 + 1$.

Este polinomio es escogido ya que cumple la característica de ser irreducible, lo que hace que durante este proceso de realimentación se generen todos los estados posibles antes de que el ciclo comience de nuevo, estos son $(2^{48} - 1)$ estados. Lo que quiere decir que a partir de realizar ese número de iteraciones la secuencia volvería a repetirse.

Una vez se tienen la nueva cadena de bits generada por LFSR se pasa por unas funciones de filtrado que generan la clave de cifrado de 48 bits, este proceso es lo primero que se hace en la autenticación tanto en el lector como en la tarjeta para establecer una clave simétrica que se va actualizando con cada ciclo del reloj interno de la tarjeta y el lector. Lo que hace que la clave vaya cambiando cada 48 bits.

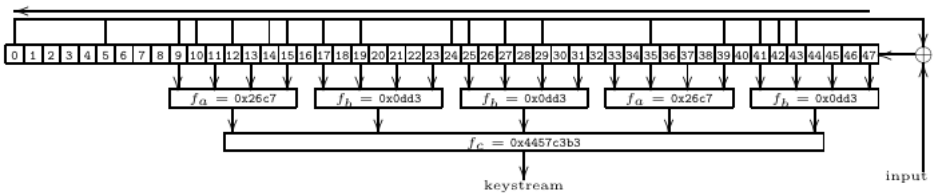


Figura 7: Esquema más en detalle de CRYPTO1

3.4.1. Sobre la inicialización

Una parte interesante y a tener en cuenta es cuando se inicializa el algoritmo, ya que en este tipo de algoritmos para generar un número aleatorio hay que partir de algo en concreto ya que en electrónica y por tanto en informática no existe un verdadero *azar*. Para solucionar esto como ya se ha comentado por encima previamente, se usa como primer número del LFSR la clave privada del sector al que se está intentando acceder.
Además de esto hay que tener en cuenta que el algoritmo al no tener respuesta previa que poder usar como entrada como bit de *feedback* junto al propio del estado anterior, usa como inicio el UID que es un dato conocido. Por lo que sabiendo estos dos datos el resto del cifrado es predecible si se conocen todas las operaciones necesarias. Y por mucho que vaya mutando la clave para generar otras claves, al generarse por realimentación constante cualquier instante del cifrado es predecible.

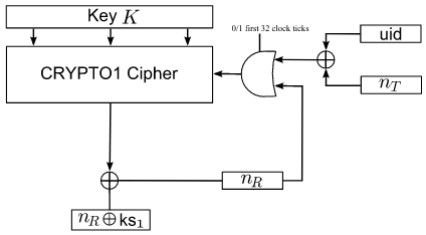


Figura 8: Inicialización de CRYPTO1

3.5. Debilidades de CRYPTO1

- Claves de cifrado de 48 bits son demasiado pequeñas. Vulnerable a fuerza bruta (una clave extraíble en 10 horas).
- El LFSR para generar números pseudo-aleatorios es predecible.
- Si alguien fuera capaz de controlar el tiempo del protocolo, sería capaz de controlar el generador de números aleatorios e intentar recuperar la clave que es el número inicial.

4. Ataques al criptosistema

A la hora de atacar al denominado CRYPTO1, como ya hemos visto que se encarga del cifrado entre la tarjeta y el lector, nos aprovechamos de la debilidad que tiene la tarjeta a la hora de generar números aleatorios. Cuando genera un *nonce* al inicio por un Linear Feed-back Shift Register (LFSR), el cual cambia por cada periodo en el reloj interno.

Existen muchos tipos de ataques a este criptosistema, por lo cual vamos a verlos a partir de software más popular que encontramos ahora mismo. Y luego hablaremos del hardware y realizaremos una pequeña demostración práctica.

4.1. Software

Todo el software que vamos a ver es códgo libre (Open Source) y está a disposición de todo aquel que quiera. Existen otras muchas más implementaciones del mismo ataque de los que vamos a ver a continuación.

4.1.1. Darkside/Mfcuk

Es la implementación del ataque Darkside[9], el cual como ya hemos mencionado previamente se basta del fallo de seguridad de generación de nonces. Fue descubierto por el criptoanálista Nicols Courtosis en 2009 y esta implementación viene dado por cortesia de Andrei Costlin

En el proceso de autenticación cuando se envían los valores n_R y A_R, lo primero que se verifica son los bits de paridad (en caso de ser erróneo se obvia la petición y no se envía respuesta) y si están correctos, pero el A_R es incorrecto devuelve un codigo de error 0x5 (NACK). Todo el codigo de error vienen cifrado, por lo cual si realizamos un XOR entre el valor de error y el texto cifrado obtenemos 4 bits de la clave de comunicación y luego se puede repetirse para el siguiente valor.

Sin embargo este proceso puede demorarse horas para obtener todas las claves. Partiendo de esta técnica junto con otras así obtiene las claves este programa y en ultima instancia realiza fuerza bruta con la parte de clave que haya podido extraer.

4.1.2. Mfoc y Mfoc-hardnested

Realmente el ataque recibe el nombre de "nested attack u offline nested attack"[11] pero fue descubierto por Nijmegan Oaklan también en 2009 y la implementación viene de la mano de Nethemba. Luego nos encontramos otra aportación por parte de Carlo Meijer y Roel Verdult y se agrega la parte "hardenested"[10] que optimiza el ataque haciéndolo mucho más rápido.

En este caso es necesario disponer de al menos una clave conocida previamente para poder realizar el proceso de autenticación y leer el valor de n_T (determinado por LFSR) y después calcular el siguiente valor de n_T en función del tiempo que pasa. De forma que se puede calcular los valores k_{s1}, k_{s2} y k_{s3}, que son las claves desplazadas que se usaban para cifrar durante el protocolo de autenticación, e intentar autenticarse en otro bloque.

Aunque normalmente se suele disponer de al menos las claves de lectura, puesto que las tarjetas se plantean para que se puedan leer por cualquiera y usan claves por defecto. En caso de no disponer de claves se puede combinar con el ataque anterior obteniendo una clave en más tiempo y luego obtener el restor de forma más rápida con este ataque.

4.2. Hardware

El hardware en estos casos varia en función del tipo de tarjeta a usar, puesto que al cambiar de frecuencia se necesitará un tipo de antena diferente. Y si además hablábamos de que tenemos diferentes tipos de protocolos, también implica un soporte por parte del hardware, puesto que como ya hemos hablado antes a partir del estándar cada fabricante impone luego sus propias características.

En el mercado existen multitud de herramientas, tanto los dispositivos que se sirven de esta tecnología para lectura y escritura de las tarjetas de manera estándar, hasta herramientas más avanzadas que tienen innumerables utilidades. En nuestro caso no nos vamos a centrar en la tecnología del propio aparato que lee y escribe de forma habitual como sería por ejemplo, los lectores de los datáfonos a la hora de realizar pagos, lectores que se encuentran en vehículos de transporte publico, sistemas de inventario o tan siquiera su uso con teléfonos móviles. Sino en las herramientas más avanzadas que disponen de más flexibilidad y tienen multitud de utilidades, incluso la de explotar vulnerabilidades en el cifrado.
A continuación enumeramos algunos posibles artilugios disponibles actualmente en el mercado:

- **Lector/Escritor ACR122U** Se trata de un lector y escritor via USB de tarjetas NFC, es un periférico para el ordenador que solo soporta tarjetas de tipo ISO 14443 e ISO/IEC 18092. Pero con el software adecuado es posible usar los scripts mencionados anteriormente para obtener las claves de la tarjeta y modificar los datos.

Figura 9: Lector/Escritor ACR122U

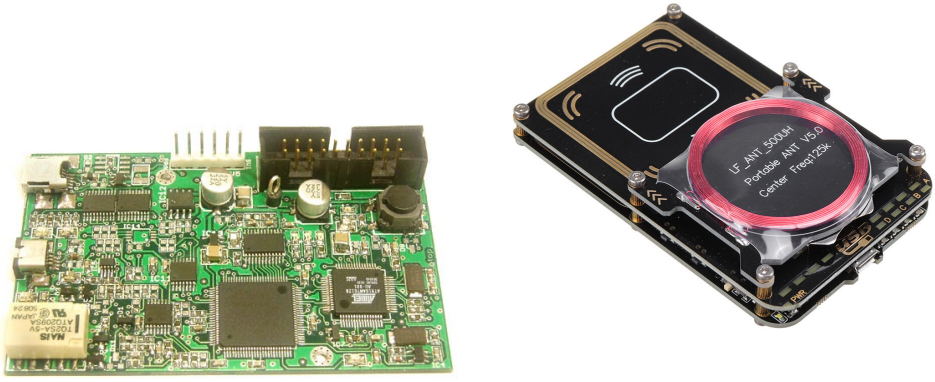
- **ChameleonMini** En este caso está pequeña placa electrónica se trata de un pequeño controlador (el mismo chip con el que cuenta la Arduino Mega) con una antena de RFID que soporta frecuencias de 13.56 MHz y 125 kHz. En este caso solo se soporta la lectura, simulación de tarjeta y escucha de comunicación de otra tarjeta con otro lector.

Figura 10: Chameleon Mini

- **Flipper Zero** Esta herramienta que ha salido recientemente ha tenido mucha popularidad hasta hace poco tiempo como herramienta multiusos. Entre los usos de receptor de señales de radio de baja frecuencia, mando a distancia de infrarrojos, pequeño Keylogger y Wifi Deather. Destacamos su capacidad de emular, clonar y escribir tarjetas NFC de frecuencias de 13.56 MHz y 125 kHz.

Figura 11: Flipper Zero

- **Proxmark 3** Fue en 2009 cuando Jonathan Westhues publicó bajo licencia GPL el documento detallando la construcción de este dispositivo el cual es capaz de tanto leer, escribir, lanzar ataques para extraer las claves, simular tarjetas y captación de comunicación entre tarjeta y lector. Desde entonces se ha ido actualizando tanto el diseño como el soporte a más tipos de tarjetas así como su propio software. En nuestro caso disponemos de la Proxmark 3 RDV3 (Versión 3) y en la actualidad se encuentra la versión 4 que dispone de un diseño modular que permite añadir otros módulos con funcionalidad no tan relacionada con el campo del RFID.



(a) Primera versión (b) Versión RDV3 Easy

Figura 12: Proxmark 3 en sus distintas versiones.

4.3. Demostración práctica con la Proxmark 3

Antes de comenzar a realizar un ataque al criptosistema, vamos a ver el funcionamiento básico. Como se puede observar en la siguiente figura la Proxmark 3 se conecta a través de un cable USB al ordenador de forma que este establece una conexión serial (de manera

parecida a como lo haría un microcontrolador Arduino) para poder enviar ordenes desde el ordenador y recibir respuesta a la orden enviada.

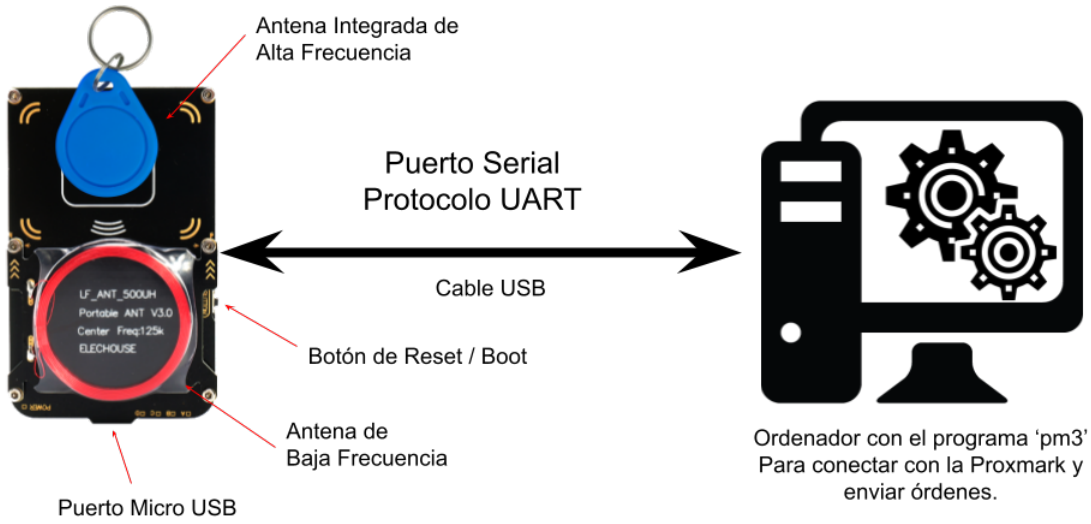


Figura 13: Esquema funcionamiento de la Proxmark 3

Para ello necesitamos del software adecuado, existen dos distribuciones de tanto el propio código para conectar como el firmware del dispositivo.

- **Repositorio oficial**[7]: El cual es el más antiguo y no se encuentra desactualizado, no dispone de soporte a las versiones recientes del hardware.
- **Repositorio de RfidResearchGroup**[4]: Este se refiere a un grupo de personas que contribuyeron a continuar con el código oficial y ayudaron a mantener las actualizaciones. Es el código y firmware que vamos a usar.

Una vez tenemos instalada el software para conectar con la proxmark, la conectramos a traves de usb y ejecutamos el programa. Lo primero que vemos en la terminal es lo que se muestra en la siguiente figura es la consola de comandos para enviarle ordenes a la proxmark.

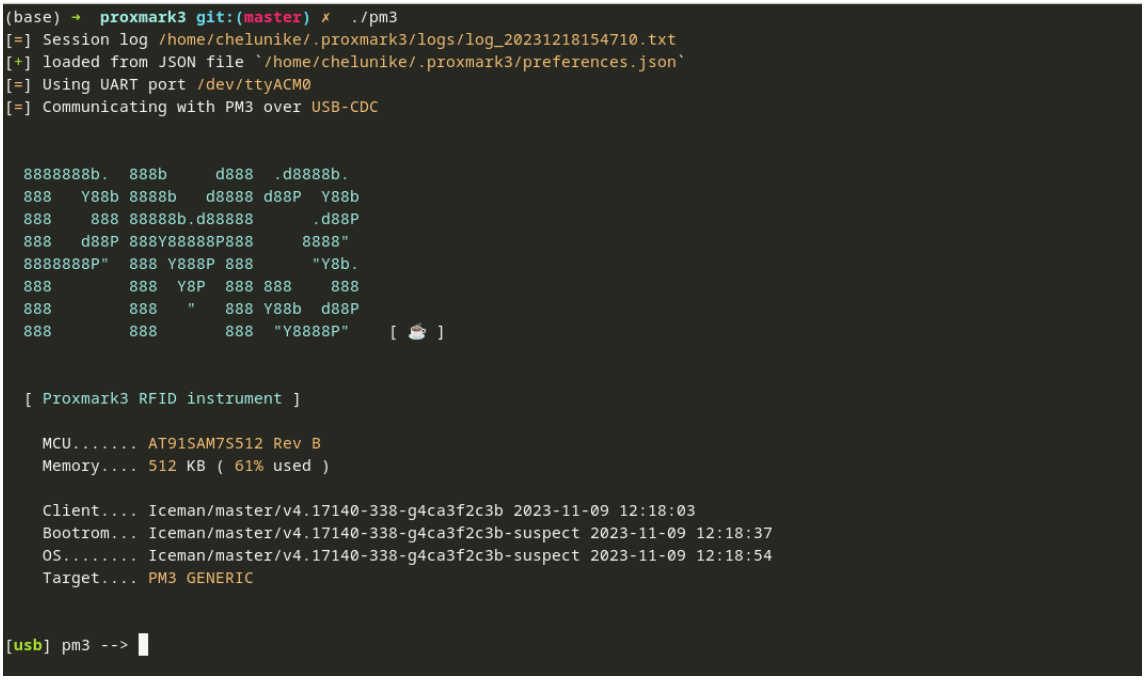


Figura 14: Inico Conexión Proxmark 3

A continuación, usamos el comando help para obtener todas las ordenes de las que dispone. La sección que nos interesa es la `hf`, que hace refenica a High Frecuency. Dentro de esta nos encontramos con `hf mf` que hace referencia a las tarjetas MiFare Classics que estamos estudiando.

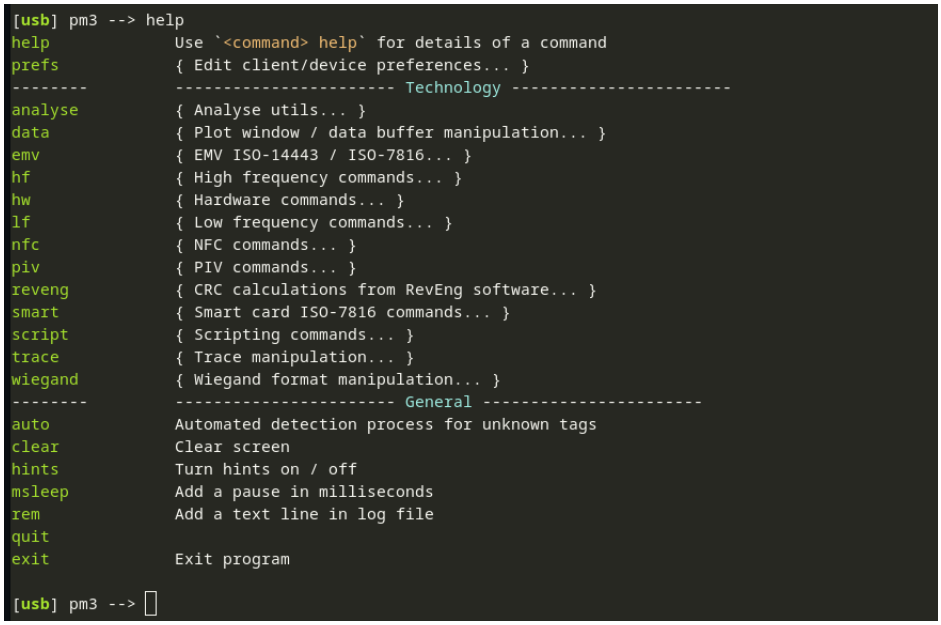
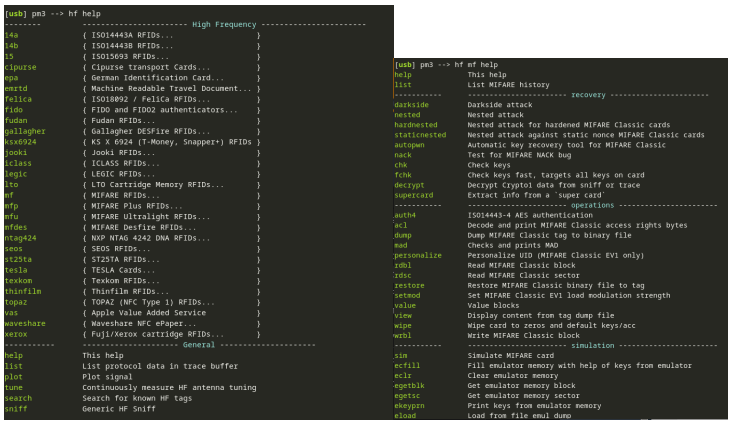


Figura 15: Inico Conexión Proxmark 3

En las siguientes figuras nos encontramos con los comandos de ayuda respecto a las distintas funcionabilidades que ofrece respecto a tarjetas de alta frecuencias, como que posibles ataques dispone para MiFare Classic. Como podemos observar se dispone de los ataques mencionados anteriormente.



(a) Ayuda de alta frecuencia (b) Ayuda respecto a Mifare

Figura 16: Comandos de ayuda.

A la hora de realizar un ataque con la Proxmark 3 se dispone de la opción `autopwn` que automatiza desde la prueba de claves ya conocidas como son las establecidas por defecto normalmente, como desde un diccionario propio, hasta la organización de los ataques.

Probando así el ataque Darkside en caso de no disponer de ningun tipo de clave y luego usando en ataque `nested`(mfoc) o `hardnested` para extraer el resto y así ejecutarse en el menor tiempo posible.

Para ejecutar un ataque a partir de un diccionario propio usaríamos el siguiente comando: `hf mf autopwn -f clavesA.dic`.

Además para complementar un poco respecto a las utilidades de la proxmark vamos a ver como podemos realizar una captación de comunicación entre la Proxmark 3 y un lector/escritor basico hecho con una placa de Arduino Uno [**?**] (Adjuntamos código en el anexo).

El comando para captar dicha información seria el siguiente `hf sniff`.Con ello se pone a la espera de captar información y tras capturarla se pueder guardar con el comando: `data save -f <nombre del archivo>`.

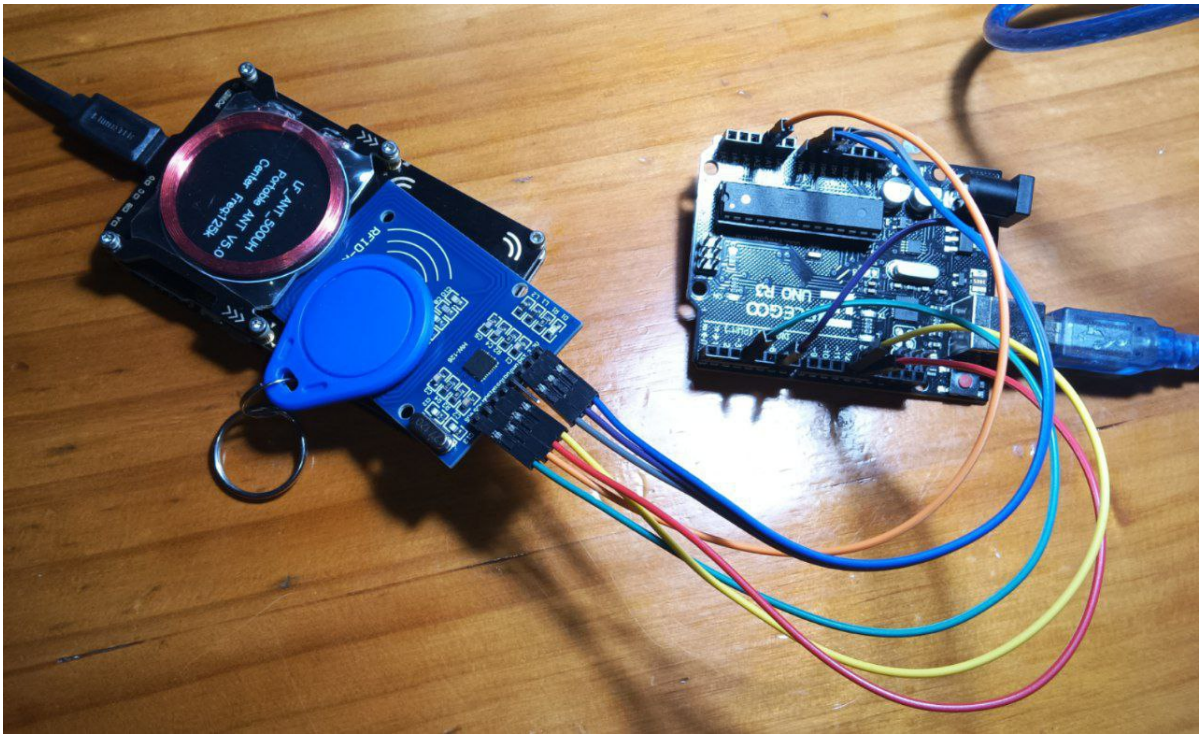


Figura 17: Proxmark captando la comunicación entre el lector de Arduino y la tarjeta.

5. Conclusión

Como conclusión nos gustaría reflexionar sobre lo interesante que nos ha parecido investigar sobre una tecnología que tiene un uso diario pero que ahora además tenemos el conocimiento de como funciona, que más allá del interés divulgativo y personal nos puede servir para entender un poco más que pasa en nuestro alrededor y como existen aún sistemas, como la MIFARE Classic, que son muy usados siguen desactualizados aún siendo bastante inseguros.

Además, queremos hacer hincapié en la importancia de la criptografía y de la liberación del código para proteger sistemas. Como ya hemos visto y se ha demostrado la seguridad por oscuridad no es muy eficaz a la hora de asegurar sistemas, en contra de lo que intuitivamente pudiese parecer.

Por tanto, esto se demuestra una vez más en este trabajo en el apartado criptográfico de CRYPTO1 (3.4) , ya que los algoritmos de cifrado más seguros son los que son ampliamente conocidos y auditados. La seguridad de estos nunca debería residir en la ocultación del proceso sino en otras características como el espacio de claves, aleatoridad, longitud de claves, etc.

Índice alfabético

Referencias

[1] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John wiley & sons, 2010.

[2] Flavio D Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter Van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. Dismantling mifare classic. In *Computer Security-ESORICS 2008: 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings 13*, pages 97–114. Springer, 2008.

[3] GeeksforGeeks. Near Field Communication (NFC). `https://www.geeksforgeeks.org/near-field-communication-nfc/`, 2023.

[4] C. Herrmann, P. Teuwen, O. Moiseenko, M. Walker, et al. Proxmark3 – Iceman repo. `https://github.com/RfidResearchGroup/proxmark3`.

[5] International Organization for Standardization. ISO/IEC 18092:2013 Information technology – Telecommunications and information exchange between systems – Near field communication – Interface and protocol (NFCIP-1). `https://www.iso.org/standard/56692.html`, 2013.

[6] International Organization for Standardization. ISO/IEC 14443-4:2021 Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol. `https://www.iso.org/standard/73598.html`, 2021.

[7] Jonathan Westhues. Código fuente oficial proxmark3. `https://github.com/Proxmark/proxmark3`, 2007.

[8] Magnetic stripe card. Magnetic stripe card — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Digital_card#Magnetic_stripe_card`, 2023.

[9] nfc-tools. Mifare classic tookit - darkside attack. `https://github.com/nfc-tools/mfcuk`, 2019.

[10] nfc-tools. Mifare classic tookit - hardnested attack. `https://github.com/nfc-tools/mfoc-hardnested`, 2019.

[11] nfc-tools. Mifare classic tookit - offline nested attack. `https://github.com/nfc-tools/mfoc`, 2019.

[12] Smart card. Contact smart card — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Smart_card#Contact_smart_cards`, 2023.