Selective Attacks to Mifare Classic Cards

J. Kamlofsky

***Abstract*— Mifare Classic is a proximity card having a chip with memory and cryptography. Cards have a symmetric stream cipher with two keys of 48 bits in each of their 16 sectors. The algorithm is owned by NXP and keeps in secret. From 2007 until today they have been several papers showing cryptanalysis of the cipher.

This paper describes main features of Mifare Classic system, and the main reverse engineering and cryptanalysis works. The more effective attacks were described and implemented and experimental data are shown. Thus, a classification of Mifare Classic systems based on keys configuration is presented and therefore, a more efficient approach to attack selectively Mifare systems.

Keywords— Breaking Mifare Classic security, efficient attacks to a Mifare Classic card, selective attacks to Mifare Classic cards.

I. Introduccion

Mifare es la tarjeta inteligente sin contacto más vendida en el mundo con más de mil millones de tarjetas vendidas. La familia de tarjetas Mifare, producto de NXP Semiconductors (anteriormente Philips) actualmente se compone de los siguientes tipos: Ultralight, Classic, DESfire y SmartMX siendo Mifare Classic la más vendida.

Gracias a tener seguridad criptográfica y tener capacidad de memoria, permite que la validación de las transacciones pueda realizarse directamente en la tarjeta, sin requerir estar conectado con un servidor, manteniendo los saldos y la información contenida en la tarjeta en forma segura. Entonces, la seguridad del sistema depende del sistema criptográfico de la tarjeta, y descansa en él.

Mifare Classic se comunica por radiofrecuencia por lo que se eliminan piezas móviles dentro de los lectores, y se descarta el rozamiento entre lector y tarjeta, lo que disminuye notoriamente los costos de mantenimiento e incrementa la vida útil de ambos: lectores y tarjetas. La frecuencia de radio usada es 13,56MHz. A esta frecuencia puede realizarse una transacción típica de incremento o decremento de saldo en aproximadamente 100mS, tiempo que no es percibido como demora por parte de los usuarios, lo que brinda mejoras ergonómicas.

Así se transformó rápidamente en la tarjeta preferida para aplicaciones de pago del transporte público, estacionamiento y peaje como "título de transporte" de dinero electrónico para pago o para identificación en sistemas de control de accesos a edificios y establecimientos.

Mifare Classic es una tarjeta plástica que contiene embebido un pequeño chip con 1KB de memoria, lógica y criptografía. La comunicación con el chip se realiza por radio-frecuencia a través de una antena en forma de espira embebida en el plástico y conectada al chip. La tarjeta contiene un chip. El chip que actualmente se incluye en Mifare Classic es el NXP MF1ICS50. Según las especificaciones que brinda el fabricante [1], la memoria de la tarjeta está dividida en 16 sectores, cada uno de los cuales está protegido por dos claves de 48 bits para cifrado simétrico cuyo algoritmo es mantenido (aún hoy) en secreto por el fabricante.

Pero esto se contrapone con uno de los principios de la criptografía conocido como "Principio de Kerckhoff" que establece que "*la seguridad de un sistema criptográfico no debe descansar en el secreto del algoritmo en sí, sino en el secreto de la clave*".

Así presentado, quedan expuestas aquí dos debilidades atractivas para el trabajo de investigadores y criptoanalistas:

1. El tamaño de las claves de acceso a cada sector es de 48 bits: hoy es perfectamente posible diseñar un ataque por fuerza bruta contra ese espacio de claves.

2. La violación del principio de Kerckhoff alienta a la búsqueda de debilidades en el algoritmo: Por algo es secreto.

Pasó más de una década desde su creación hasta que a fines de 2007 Karsten Nohl y Henryk Plötz presentaron debilidades en [2], exponiendo así el tema. A inicios de 2008, los mismos autores, junto con David Evans y Starbug presentaron un excelente trabajo [3] donde han descripto un efectivo proceso de ingeniería inversa realizada sobre el chip de la tarjeta Mifare Classic: Lograron regenerar la totalidad del algoritmo de cifrado del chip denominado Crypto1, y luego de realizar un análisis de protocolo basado en la ISO 14443 [4], han encontrado vulnerabilidades del algoritmo y se presentó un ataque.

Si bien el trabajo [3] de Karsten Nohl, David Evans, Starbug y Henryk Plötz no contiene tantos detalles acerca de sus hallazgos como se esperaba, fue de gran inspiración para otros investigadores.

Pocos meses después Gerhard. de Koning Gans et al presentaron un ataque detallado al cifrado del chip [5], que les permitió leer y modificar algunos sectores de memoria sin conocer la clave.

En 2008, Flavio García et al, presentaron otro trabajo [6] donde se detallan dos ataques (a lectores Mifare) basados en debilidades descubiertas en el cifrador Crypto1. En este trabajo se presentaron hallazgos muy interesantes ya que dejó expuesto al criptosistema Crypto1 y a su protocolo, con gran detalle. La repercusión de este trabajo fue enorme.

Cada uno de los ataques enunciados previamente poseen

ciertos condicionamientos que hacen que no sean tan simples de reproducir (ataques a lectores). Así, tanto la empresa NXP, operadores de transporte, como integradores de sistemas han relativizado los trabajos hasta aquí presentados aduciendo a que su reproducción está fuertemente condicionada a que el atacante interactúe con el sistema en infraestructura de propiedad de los operadores, exponiéndose a ser descubierto, o bien, condicionado a disponibilidad de hardware o software especial, no comercial (que no es muy fácil de conseguir o de costo elevado).

Pero es obvio que ningún atacante haría un gran despliegue de equipamiento tecnológico frente a un molinete en la estación más importante del sistema de transporte de la ciudad y así ser fácilmente descubierto, sino que actuará sigilosamente en puntos débiles del sistema, quizás con anuencia de algún empleado que esté a cargo de un punto solitario dentro del sistema, con reducida o nula vigilancia.

Un año más tarde Flavio García et al, presentaron un trabajo [7] donde a partir del descubrimiento de nuevas vulnerabilidades encontradas en el diseño del chip y del cifrador Crypto1 lograron obtener todas las claves de todos los sectores de la tarjeta. Presentaron aquí cuatro ataques (a tarjetas), cada uno de los cuales requiere simplemente de una tarjeta, un lector económico, un computador ordinario y comunicación de radio frecuencia entre tarjeta y lector. Han logrado reproducir todo el contenido de una tarjeta en segundos. Notar que aquí el ataque se realiza sobre tarjetas.

Pocos meses luego, Nicolas Courtois presentó un trabajo [8] en el que con el mismo hardware, y basándose en trabajos previos y en descubrimientos propios, se logra obtener la clave de un sector realizando muy pocas consultas y sin necesidad de tener tablas precalculadas.

Ahora, sin exposición a la vigilancia del operador y con hardware sencillo, rápido, cómodo y en su propio laboratorio, un atacante puede cambiar los saldos de tarjetas de transporte o bien incluso clonar tarjetas y así usurpar identidades en sistemas de seguridad. Esto es definitivamente desbastador para cualquier sistema que posea esta tecnología. Ahora sí, puede considerarse a la seguridad de esta tecnología totalmente quebrada.

El primer desarrollo para la implementación de algunos de los ataques presentados en los trabajos anteriormente mencionados es el código Crapto1[1], creado por David Bolton en Octubre de 2008, que implementa ataques presentados en [6]. Para la implementación de los ataques sobre tarjetas usando los lectores NFC se desarrollaron las librerías Libnfc [9]: las librerías de código abierto para lectores NFC.

Basados en el código Crapto1, usando las librerías Libnfc, hay dos programas que implementan ataques presentados en [7] y [8]: MFOC[2] y MFCUK[3]. Y ambos pueden ser complementarios en la tarea de obtener las claves de una tarjeta.

[1]http://code.google.com/p/crapto1/

[2]http://code.google.com/p/nfc-tools/source/browse/trunk/mfoc/src/mfoc.c?r=977

[3]http://code.google.com/p/mfcuk/

MFOC (MiFare Off line Cracker) implementa el ataque denominado “autenticaciones anidadas” presentado en [7]. Fue creado por el grupo Nijmegan Oakland y codificado por la empresa Nethemba. El mismo, a partir de conocida una de las claves permite obtener rápidamente las 31 claves restantes.

MFCUK (MiFare Classic Universal toolKit) implementa el ataque denominado “lado oscuro” presentado en [8]. Fue creado por Andrei Costin. Permite obtener rápidamente una clave específica de una tarjeta.

Hay diferentes sitios donde se presenta en detalle cómo puede lograrse la instalación de Libnfc, MFOC y MFCUK. Se pueden mencionar el Blog en español de Security Artwork [10], el sitio de Andrei Costin [11] y el blog de la distribución de Linux llamada Backtrack [12].

Una vez realizada la instalación de los componentes, pueden iniciarse los ataques a cualquier tarjeta Mifare Classic.

Pero la obtención de claves de una tarjeta puede demorar desde unos pocos segundos hasta varios días, según qué configuración de claves posea el sistema al que pertenece la tarjeta atacada, y qué ataque se use. En este trabajo se presentan datos experimentales obtenidos de la implementación de diferentes ataques, lo que permitió realizar una clasificación de las tarjetas Mifare Classic (y de los sistemas donde son usadas) según su configuración de claves. Con esto es posible sugerir la conveniencia del uso de los programas MFOC, MFCUK por separado o ambos en conjunto para realizar ataques selectivos eficazmente.

En la Sección II se presentan los Objetivos del trabajo y la Relevancia del tema. En la Sección III se expone acerca del Estado del Arte y Temas relacionados: allí se presentan detalles técnicos de Mifare Classic y un resumen de los ataques más importantes. En la Sección IV se detalla la implementación de varios ataques a tarjetas Mifare Classic. En la Sección V se presenta una clasificación de los sistemas que usan Mifare Classic según la configuración de sus claves. En la Sección VI se proponen ataques selectivos en función de cómo se clasificó al sistema.

II. Objetivo del trabajo y relevancia del tema

A. *Objetivo del trabajo:*

La seguridad de la tecnología Mifare Classic ha sido quebrada gracias a numerosos trabajos presentados por diversos investigadores. También se desarrollaron programas que implementan los trabajos más destacados. En este trabajo se describen los principales trabajos que lograron el quiebre, y se presenta una clasificación de los sistemas que utilizan Mifare Classic según la existencia de claves por default dentro de sus tarjetas. Con esto se sugiere presentar la conveniencia en el armado de un ataque específico al sistema mediante la utilización selectiva de las herramientas de software disponibles.

B. *Relevancia del tema:*

El quiebre de la seguridad de las tarjetas Mifare Classic pone en peligro la recaudación de sistemas de parking, peaje y

transporte masivo dispersos por todo el mundo, así como la integridad de empresas e instalaciones gubernamentales críticas con acceso controlado por esta tecnología. Para evitar sus efectos, se requiere una readecuación tecnológica del sistema tan pronto como sea posible.

En Argentina hay más de diez millones de tarjetas en decenas de sistemas de pago de transporte público, parking y peaje. También hay gran cantidad de sistemas de control de seguridad a edificios que usan Mifare Classic. Algo similar sucede en Brasil, Colombia, Chile, y en diferente medida, en Uruguay y en el resto de los países de la región. Panoramas similares se repiten en todo el mundo.

III. Estado del arte y trabajos relacionados

A. Acerca de Mifare Classic:

Mifare es una familia de tarjetas inteligentes sin contacto de NXP (anteriormente Philips semiconductors). Mifare Classic, también, es el producto más vendido de la familia Mifare. Gracias a tener seguridad criptográfica, comunicarse por radio-frecuencia, y tener capacidad de memoria, se transformó rápidamente en la tarjeta preferida para aplicaciones de pago del transporte público, estacionamiento y peaje como título de transporte de dinero electrónico para pago o para identificación en sistemas de control de accesos. Muchas aplicaciones hacen uso de estas tarjetas a gran escala.

ARQUITECTURA TIPICA DE UN SISTEMA MIFARE:

En una arquitectura básica de un sistema que posea tecnología Mifare Classic, pueden identificarse dos dispositivos:

- **Lector:** Consiste en una antena activa que permanentemente irradia un campo electromagnético de hasta 10cm de efectividad a una frecuencia de 13,56MHz. Posee una electrónica de control y es fácilmente conectable a otros dispositivos por puertos USB o RS232. Se los monta en molinetes, validadoras de buses, equipos de vía de peaje, terminales de entrada-salida y cobro de sistemas de parking, parquímetros, control de accesos y demás.
- **Tarjeta o tag:** Una antena con un chip con memoria y seguridad criptográfica se encuentran embebidos dentro de una tarjeta plástica ISO 7816. Es energéticamente pasiva, y se activa y se alimenta eléctricamente dentro del campo del lector. En la Figura 1 [1] puede observarse un diagrama de un conjunto típico de tarjeta o tag y lector Mifare.

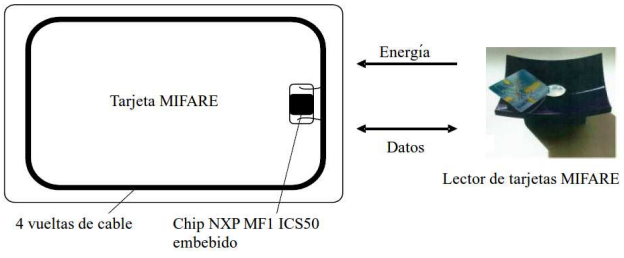

Figura 1. Sistema típico Tarjeta – Lector Mifare.

En [1] pueden encontrarse detalles técnicos acerca de la tecnología, publicados por el fabricante.

LA ESTRUCTURA LOGICA DE LA TARJETA:

Hay diversas versiones de tarjetas Mifare Classic. Sin embargo la más difundida es Mifare Classic 8K. Su EEPROM es de 1KB = 1024 x 8 bits. Está organizada en 16 sectores de cuatro bloques de 16 bytes cada uno. El usuario puede determinar las condiciones de acceso para cada bloque. Cada sector posee dos claves de 48 bits cada una. La figura 2 [1] muestra la organización de la memoria del chip. Cada uno de los 16 bloques posee 4 sectores de 16 bytes cada uno: el cuarto sector denominado “Sector trailer” contiene:

- Los bytes 0 a 5 contienen la clave A de dicho sector.
- El contenido de los bytes 6 a 9 determinan las condiciones de acceso.
- Los bytes 10 a 15 contienen la clave B de dicho sector.

El resto de los sectores pueden contener datos o valores. El primer sector del primer bloque contiene información especial: es el bloque del fabricante. No puede modificarse: es solamente para lectura. El mismo contiene: Los bytes 0 a 4 tienen el número de serie de la tarjeta (UID). El byte 5 tiene un CRC del número de serie, que se obtiene de hacer XOR de los bytes del UID. Los restantes bytes (desde 6 a 15) del sector contienen información diversa del fabricante.

Las operaciones que se pueden realizar sobre un bloque de memoria son: Lectura (read) o escritura (write) de un bloque de datos, decrementar (decrement) o incrementar (increment) el contenido de un bloque de valor (y guarda el resultado temporalmente en un registro interno). Restaurar (restore), mueve el contenido de un bloque al registro interno. Transferir (transfer): escribe el contenido del registro interno en un bloque.

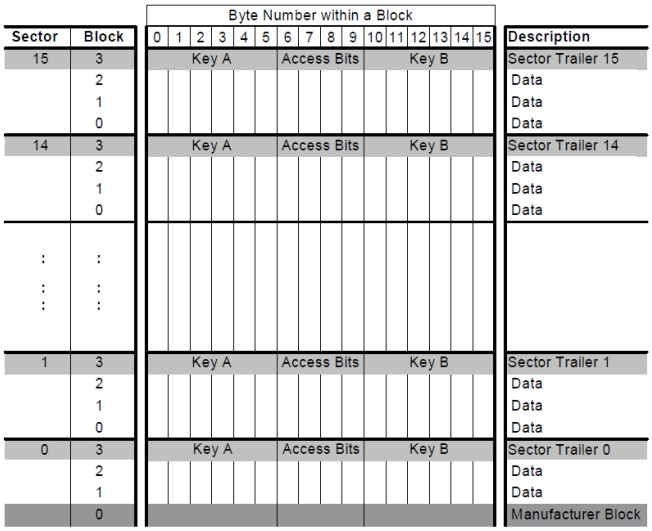

Figura 2: Organización de la memoria de la tarjeta.

COMUNICACIONES ENTRE LECTOR Y TARJETA:

La comunicación entre tarjeta y lector se realiza por radiofrecuencia, a 13,56MHz permitiendo transferencias a una velocidad de 106 kbit/s, logrando típicas transacciones en

menos de 100 ms, tiempo que no causa sensación de demora, cuestión fundamental en sistemas de transporte público masivo.

Según las especificaciones, dependiendo de la geometría de la antena, se pueden hacer transacciones a una distancia de 100mm, pero en la mayoría de los casos la distancia media de operación rara vez supera los 50mm a causa del diseño del hardware: en muchos casos los dispositivos que contienen antenas y lectoras Mifare (molinetes, gabinetes, etc.) se fabrican de acero, lo que produce una disminución de la calidad de señal.

La interfase de Radio frecuencia cumple con los estándares para tarjetas sin contacto ISO/IEC 14443 A. Para comunicación bidireccional hay un bit de inicio en cada trama. Cada byte se transmite con un bit de paridad al final: paridad impar. Primero se transmite el bit menos significativo del byte de menor dirección del bloque seleccionado. La máxima longitud de trama es de 163 bits: 16 bytes de datos + 2 bytes de CRC = 16 * 9 + 2 * 9 + 1 bit de inicio.

El principio de comunicación se basa en un sistema de autenticación en tres pasos según ISO/IEC 9798-2: Anticolisión, autenticación, comunicación encriptada. El proceso anticolisión permite transaccionar de modo seguro con una tarjeta, por más que hubiesen varias de ellas en el campo del lector. El protocolo de autenticación permite asegurar que ambas partes sean quienes dicen ser. A partir de producida la autenticación, la comunicación entre tarjeta y lector se realiza de forma encriptada mediante el cifrador de flujo Crypto1 usando las claves definidas en las condiciones de acceso, que ambas partes conocen.

Para ver cómo son las comunicaciones en Mifare Classic, en [5] se hicieron seguimientos de transacciones entre tarjetas y lectores con un sniffer. Con ello se obtuvo cierta información acerca del protocolo de alto nivel de Mifare Classic. En la figura 3 se presenta un ejemplo detallado de una transacción. La notación usada es en hexa. TAG significa "tarjeta", y PCD significa "Lector".

SEC	Origen	Bytes	
01	PCD	26	Anticolisión
02	TAG	04 00	
03	PCD	93 20	
04	TAG	2a 69 8d 43 8d	
05	PCD	93 70 2a 69 8d 43 8d 52 55	
06	TAG	08 b6 dd	
07	PCD	60 04 d1 3d	Autenticación
08	TAG	3b ae 03 2d	
09	PCD	c4! 94 a1 d2 6e! 96 86! 42	
10	TAG	84 66! 05! 9e!	
11	PCD	a0 61! d3! e3	Incremento y transferencia
12	TAG	0d	
13	PCD	26 42 ea 1d f1! 68!	
14	PCD	8d! ca cd ea	
15	TAG	06!	
16	PCD	2a 2b 17 97	Lectura
17	TAG	49! 09! 3b! 4e! 9e! 5e b0 06 d0! 07! 1a! 4a! b4! 5c b0! 4f c8! a4!	

Figura 3: Un seguimiento de una transacción en Mifare Classic.

CONDICIONES DE ACCESO:

Las condiciones de acceso para todo bloque de datos o sector trailer se definen con 3 bits, los cuales se guardan invertidos y no invertidos en el sector trailer del sector específico. Los bits de acceso controlan los derechos de acceso usando las claves secretas A y B. Las condiciones de acceso pueden alterarse cuando se conocen claves correspondientes y las condiciones de acceso actuales permiten esa operación.

Con cada acceso a memoria la lógica interna verifica el formato de las condiciones de acceso. Si se detecta una violación, todo el sector se bloquea irreversiblemente.

EL CIFRADOR CRYPTO1:

El detalle del mismo fue presentado en [3]. El corazón del algoritmo de cifrado Crypto1 es un simple LFSR de 48 bits. Desde un conjunto fijo de 20 bits, en cada ciclo de reloj se calcula el primer bit del flujo de clave. El LFSR tiene 18 bocas que se combinan linealmente para obtener el primer bit del registro en cada ciclo.

La función filtro *f* no contiene ninguna componente no lineal, lo que por el conocimiento actual en criptografía puede ser considerado como una seria debilidad. Más aún, en [6, 7, 8] se da cuenta que en lugar de ser 5 funciones lineales que alimentan a *f*, son solo 2 que se repiten.

Posee un generador pseudo aleatorio de 16 bits para la generación de los desafíos necesarios para la autenticación, lo cual es muy pequeño.

B. Ingeniería inversa sobre el chip [3]:

El trabajo de ingeniería inversa realizado por Karsten Nohl et al. y publicado en [3] fue sin duda el hito fundamental para el inicio del quiebre de la seguridad del chip, siendo de gran inspiración para varios trabajos posteriores. Pudieron haber visto factible esta campaña viendo que Crypto1 se basa en un espacio de claves muy pequeño (48 bits). Además, pudieron haber desconfiado de su fortaleza basándose en el mantenimiento en secreto del algoritmo por parte de NXP durante tanto tiempo.

Para realizar este trabajo han analizado una gran cantidad de chips. Los mismos los obtuvieron a partir de disolver tarjetas Mifare en acetona. Cada uno de los chips fueron fijados a un soporte, y con lijado micrométrico sucesivo se fueron presentando una a una las 6 capas del chip. En el segundo nivel se encuentra el nivel lógico. Allí se tomó gran cantidad de fotografías con un microscopio de 500x. Lo primero que debieron hacer fue identificar las compuertas básicas. Luego, ayudados con un software de detección de imágenes han podido reconstruir la totalidad del circuito. La figura 4 [3] presenta una imagen obtenida y analizada.

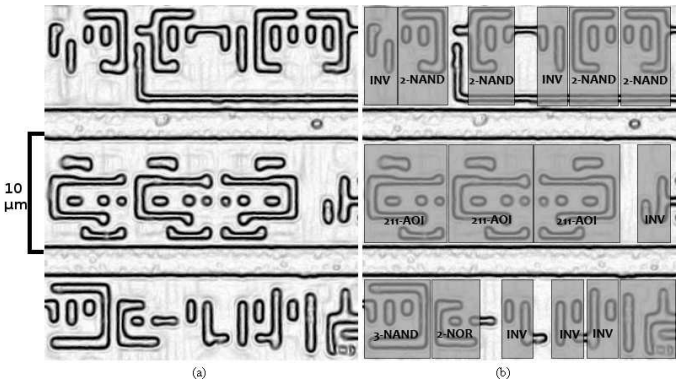

Figura 4: (a) Imagen obtenida del nivel 2. (b) Luego de la detección automática.

Un primer descubrimiento fue que las funciones criptográficas están compuestas por 400 equivalentes de compuertas 2-NAND (400 GE) lo que es muy poco, aún en comparación con aplicaciones comunes de criptografía: por ejemplo el cifrador AES para tags RFID requiere 3400 GE. Pero por otro lado, Crypto1 saca un bit cifrado en cada ciclo de reloj, comparados con los 1000 ciclos requeridos por una operación de AES de 128 bits. Pareciera que quienes diseñaron el chip NXP MF1ICS50 no pudieron resistirse frente a la tentación de la gran velocidad que el Crypto1 brindaba, confiando, quizás, que el algoritmo nunca saldría a la luz, y menos aún, que pudiera ser criptoanalizado tantas veces.

Sabiendo que Crypto1 era un cifrador de flujo de 48 bits, era muy probable que se tratara de un LFSR de 48 bits, por lo que buscaron un registro con 48 lugares. Pudieron deducir el polinomio generador del registro:

$P(x) = x^{48} + x^{43} + x^{39} + x^{38} + x^{36} + x^{34} + x^{33} + x^{31} + x^{29} + x^{24} + x^{23} + x^{21} + x^{19} + x^{13} + x^{9} + x^{7} + x^{6} + x^{5} + 1$.

Se verifica que $P(x)$ es primitivo, por lo que genera claves de longitud máxima [13].

Continuando con el análisis, se pudo encontrar en un rincón del chip un dispositivo que solo tenía salidas. Se pudo deducir que se trataba de un generador de números pseudo-aleatorio basado en un LFSR de 16 bits.

Con un lector OpenPCD se pudieron lograr autenticaciones exitosas con número de tarjetas y claves modificadas, y progresivamente ampliando la búsqueda a mayores cambios, se encontró una serie de combinaciones tales que efectivamente lograron autenticaciones exitosas.

Junto con el análisis de hardware previo, se pudo deducir el detalle completo del cifrador Crypto1. La figura 5 [3] presenta un diagrama completo del cifrador.

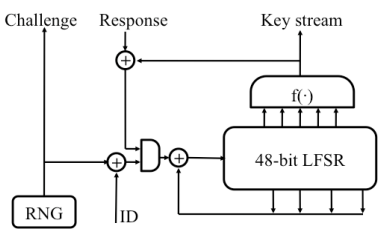

Figura 5: Diagrama del cifrador Crypto1.

El polinomio $P(x)$ es primitivo e irreducible y genera todas las posibles $(2^{48}-1)$ salidas de la sucesión. Se verifico que la salida del cifrador se repite solo luego de $(2^{48}-1)$ etapas, lo que verifica lo anteriormente dicho.

El protocolo de autenticación toma una clave secreta y el número único del tag (uid), como entradas. Al final del proceso de autenticación, las partes han establecido una clave de sesión para el cifrador de flujo, y así, ambas partes están convencidas que la otra parte conoce la clave secreta.

VULNERABILIDADES ENCONTRADAS:

- Los números aleatorios en las tarjetas Mifare Classic se generan a partir de un LFSR con condiciones iniciales constantes. Entonces, cualquier numero aleatorio del flujo de claves, depende de la clave inicial y del número de ciclos desde que la tarjeta se encendió. El LFSR del RNG usa un polinomio primitivo e irreducible de 16 bits:

$R(x) = x^{16}+x^{14}+x^{13}+x^{11}+1$

- Un atacante puede controlar los números generados si controla el sincronismo del protocolo.
- El LFSR empieza siempre en un mismo estado cada vez que se energiza.
- Posibilidad de confeccionar un libro de claves completo para cada uid.

C. Un ataque práctico a Mifare Classic [5]:

En este trabajo se estudió la arquitectura de la tarjeta y el protocolo de comunicación entre tarjeta y lector. Así, luego se llegó a presentar un ataque de bajo costo que permite obtener información secreta de la memoria de la tarjeta. Gracias a una debilidad en el generador claves pseudo-aleatorias, se está en posibilidad de obtener el flujo de claves generado por el cifrador Crypto1.

Explotando la maleabilidad del cifrador se pudieron leer todos los bloques de memoria del sector cero de la tarjeta. Más aún, aquí se está en posibilidad de leer cualquier sector de memoria bajo condición de tener acceso a un bloque dentro del sector.

Los autores declaran que independientemente de los trabajos [2], [3], ellos también encontraron la debilidad en el generador pseudo-aleatorio de claves RNG.

CONTRIBUCIONES DE ESTE TRABAJO:

Este trabajo ha dejado contribuciones muy importantes usadas en trabajos posteriores:

- Usando una debilidad en el generador pseudo-aleatorio de claves RNG, y dado un acceso a una tarjeta, es posible obtener el flujo de clave generado por el cifrador Crypto1, sin conocer la clave.
- Se describe la comunicación entre tarjetas y lectores, con muy buen nivel de detalles técnicos.
- Explotando la maleabilidad del cifrador se pudo leer toda la información del sector 0, sin conocer la clave.
- Si bien el trabajo de Karsten Nohl et al [3] presenta y

describe al cifrador, se le critica la falta de detalles. Este trabajo presenta detalles técnicos únicos: detalles de la arquitectura de la tarjeta, del protocolo, y más aún, el código de instrucciones y comandos elementales usados en lectores y tarjetas.

EL EXPERIMENTO:

Para la realización de este trabajo se utilizó un lector Proxmark III. Su diseño permite que el dispositivo emule tanto a tarjetas como a lectores. Se le incluyeron funcionalidades para dejar rastros de las acciones, con el fin de facilitar el análisis posterior.

Para ver como son las comunicaciones en Mifare Classic se hizo seguimientos de transacciones entre tarjetas y lectores. Un ejemplo se presentó en la figura 3.

Es posible recuperar el flujo de clave de una transacción previamente usada entre una tarjeta y un lector. Como resultado de este ataque se obtiene: texto plano conocido, y detalles acerca de los comandos a nivel byte. Se pueden realizar ciertas operaciones de lectura y escritura sobre la tarjeta sin conocer la clave.

Detalle del proceso:

- Se inicia con el registro y grabado de una transacción legítima mediante Proxmark III.
- Se modifica el texto plano tal que la tarjeta reciba un comando para el cual se conozca el texto plano de la respuesta.
- Para cada segmento de texto plano conocido, se calcula el segmento de flujo de clave.
- Se usa el segmento del flujo de clave calculado para descifrar parte de la transacción grabada.
- Se intenta recuperar más bits del flujo de claves cambiando los comandos.

En la comunicación, al texto plano P_1 se le hace XOR bit a bit con el flujo de clave K, lo cual resulta en el texto cifrado C_1. Si es posible usar el mismo flujo de clave en otro texto plano diferente P_2, y además al menos uno de los textos planos P_1 o P_2 son conocidos, entonces ambos pueden revelarse.

$P_1 \oplus K = C_1$

$P_2 \oplus K = C_2$

$\Rightarrow \; C_1 \oplus C_2 \oplus P_1 \oplus P_2 \oplus K \oplus K \oplus P_1 \oplus P_2$

Pueden cambiarse bits del texto cifrado e intentar modificar el primer comando tal que de otro resultado. Otro resultado brinda otro texto plano. En esto se basa el ataque.

Al ser Crypto1 un cifrador de flujo, los datos se encriptan a nivel bits. Cuando el lector manda o recibe un mensaje, el flujo de clave se corre el número de bits de este mensaje, en ambos lados: en el lector y en la tarjeta. Esto es necesario para que tanto lector como tarjeta se mantengan en sincronismo y ambos usen el mismo bit del flujo de clave tanto para encriptar como para desencriptar.

LECTURA DEL SECTOR CERO:

Se logró leer la totalidad del sector cero sin poseer la clave. En gran parte, con el análisis de la documentación propia de NXP. Como la clave A es secreta, la tarjeta devuelve 000000000000h. Análogamente si la clave B es secreta, la tarjeta devuelve 000000000000h. Muchas veces la clave B es secreta. Y puede conocerse sabiendo las condiciones de acceso, las cuales pueden obtenerse del bloque del fabricante. El byte desconocido (llamado U) normalmente presenta los valores 00h o 69h.

LECTURA DE SECTORES SUPERIORES:

Similarmente a lo hecho para el sector cero, pueden obtenerse los primeros y últimos 6 bytes del texto plano como 000000000000h. El byte U también se lo puede suponer 00h o 69h. Pero quedan 3 bytes (condiciones de acceso) para los cuales nada puede asegurarse, y puede ponerse en compromiso a ese sector.

REVELADO DEL SET DE COMANDOS:

Tanto en la documentación de NXP como en el firmware se refiere a los comandos enviados por el nivel de aplicación al lector. Nada se informa acerca de los comandos enviados desde lectores a tarjetas. Aquí se mostró que estos comandos son exactamente iguales.

Se usaron tarjetas en configuración de transporte con las claves de fábrica y los bloques de datos vacíos, con el fin de hacer mas simple la tarea de revelar los comandos encriptados. Todos los comandos enviados por el lector consisten en un byte de comando, un byte de parámetro y dos bytes CRC. Se hicieron varios intentos para revelar los comandos. La figura 6 [5] muestra el detalle del set de comandos en Mifare Classic.

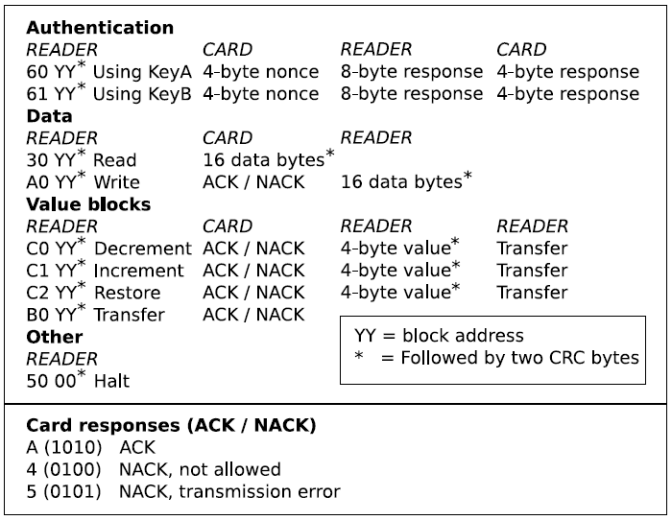

Figura 6: El set de comandos en Mifare Classic.

D. Desmantelando a Mifare Classic [6]:

En el trabajo presentado por Flavio García et al. [6] se logró descubrir completamente al cifrador Crypto1 y a su protocolo. Se describen varias vulnerabilidades descubiertas en los mecanismos de seguridad, las cuales fueron explotadas en dos ataques hacia lectores.

VULNERABILIDADES ENCONTRADAS:

- El generador pseudo-aleatorio RNG para generar desafíos es un LFSR de 16 bits con polinomio generador: $R(x) = x^{16}+x^{14}+x^{13}+x^{11}+1$. Dado que los desafíos del tag n_T tienen 32 bits de longitud y el LFSR tiene solo 16 bits, la primer mitad

de n_T, determina la segunda mitad.

- Se ha notado que si $n_T \oplus uid$ se mantiene constante, entonces el texto cifrado del desafío del lector n_R también se mantiene constante. En particular, al iniciarse nuevas comunicaciones, se verificó que $n_T \oplus uid$ es siempre el mismo.

EL PROTOCOLO DE AUTENTICACION:

La figura 7 [6] describe el protocolo de autenticación.

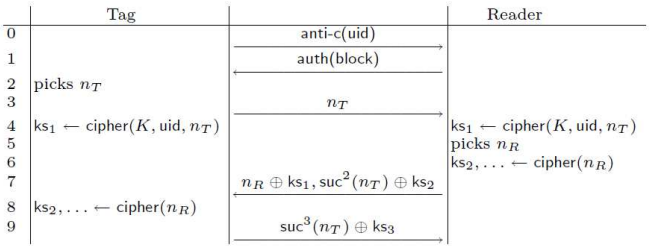

Figura 7: Diagrama del protocolo de autenticación.

RECUPERACION DEL FLUJO DE CLAVE:

Conociendo el protocolo de autenticación, con texto plano conocido, puede obtenerse parte del flujo de claves.

EL CIFRADOR CRYPOl

Se pudo deducir que el estado inicial del LFSR (semilla) es la misma clave de 48 bits. Se determinó que los bits del LFSR que ingresan a la función filtro son: 9, 11, ..., 45, 47: 20 bits que ocupan las posiciones impares a partir del bit 9.

Hay dos funciones lineales o circuitos de primer nivel que se alimentan de estos bits, cada una de ellas, usa cuarto de ellos. Ambas funciones se repiten. Sus resultados (cinco bits) alimentan una función lineal de segundo nivel.

Definición 3.1: La función Filtro.

La función filtro $f: B^{48} \rightarrow B$, $B = \{0,1\}$, *se define por:*

$f(x_0x_1...x_{47}) = f_c(f_a(x_9, x_{11}, x_{13}, x_{15}), f_b(x_{17}, x_{19}, x_{21}, x_{23}), f_b(x_{25}, x_{27}, x_{29}, x_{31}), f_a(x_{33}, x_{35}, x_{37}, x_{39}), f_b(x_{41}, x_{43}, x_{45}, x_{47}))$

donde:

f_a: $B^4 \rightarrow B$ se define por: $f_a(y_0, y_1, y_2, y_3) = ((y_0 \vee y_1) \oplus (y_0 \wedge y_3)) \oplus (y_2 \wedge ((y_0 \oplus y_1) \vee y_3))$

f_b: $B^4 \rightarrow B$ se define por: $f_b(y_0, y_1, y_2, y_3) = ((y_0 \wedge y_1) \vee y_2) \oplus ((y_0 \oplus y_1) \wedge (y_2 \vee y_3))$

f_c: $B^5 \rightarrow B$ se define por: $f_c(y_0, y_1, y_2, y_3, y_4) = (y_0 \vee ((y_1 \vee y_4) \wedge (y_3 \oplus y_4))) \oplus ((y_0 \oplus (y_1 \wedge y_3) \wedge (y_2 \oplus y_3) \vee (y_1 \wedge y_4)))$.

Ya que f solo depende de $x_9, x_{11}, ..., x_{47}$, se nota f como una función $f: B^{20} \rightarrow B$ escribiéndola: $f(x_9x_{11}...x_{47})$.

La figura 8 [6] muestra un esquema de la estructura del cifrador Crypto1.

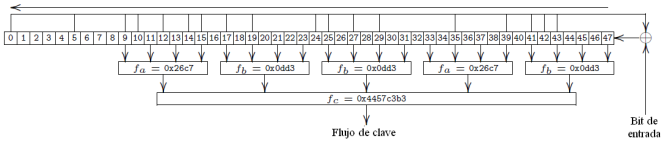

Figura 8: La estructura del cifrador Crypto1.

Definición 3.2: Estados internos de un LFSR.

Si el estado del LFSR en el momento k es: $r_kr_{k+1}...r_{k+47}$, *entonces su estado en el momento k+1 es:* $r_{k+1}r_{k+2}...r_{k+48}$, *donde* r_{k+48} *se obtiene de:*

$r_{k+48} = r_k \oplus r_{k+5} \oplus r_{k+9} \oplus r_{k+10} \oplus r_{k+12} \oplus r_{k+14} \oplus r_{k+15} \oplus r_{k+17} \oplus r_{k+19} \oplus r_{k+24} \oplus r_{k+27} \oplus r_{k+29} \oplus r_{k+35} \oplus r_{k+39} \oplus r_{k+41} \oplus r_{k+42} \oplus r_{k+43}$.

Dado un estado interno del cifrador, conocer el bit siguiente o el anterior permite calcular la función de realimentación o de Rollback de dicho estado interno.

EXPLOTACION DE VULNERABILIDADES:

- Dado un segmento de flujo de clave, es posible obtener un estado del LFSR mediante la inversión de la función filtro (obtenida con o sin una tabla pre-calculada). Mediante una técnica de retroceso del LFSR o LFSR rollback, puede obtenerse el estado inicial del LFSR que es la clave.
- Las entradas impares de la función filtro ayudan a que se pueda obtener su inversa sin necesidad de tener tablas precalculadas.
- El bit de paridad: El protocolo de comunicaciones envía un bit de control de paridad cada ocho bits. Resulta que la paridad no se calcula por sobre el texto cifrado sino que se lo hace sobre el texto plano.

E. Hurto sin cables del contenido de Mifare Classic [7]:

En [6] se presentaron serias debilidades del sistema Mifare, concluyendo con dos ataques contra lectores auténticos. Sin embargo, integradores y operadores de sistemas de tránsito y transporte público han denostado y relativizado la eficacia de dichos ataques: son ataques contra lectores, y se requiere hardware y software no comercial.

Las debilidades aquí presentadas le permiten a un atacante recuperar todas las claves de una tarjeta, con un computador ordinario y un lector de bajo costo. Se presentan cuatro ataques contra tarjetas, los cuales pueden efectuarse sin exposición, en el propio laboratorio del atacante, lo cual resulta desbastador.

VULNERABILIDADES:

Las vulnerabilidades descubiertas y presentadas en este trabajo conciernen al manejo de bits de paridad y a las autenticaciones anidadas o autenticaciones a varios sectores.

- **El manejo de bits de paridad:**

  a.- Mifare Classic envía un bit de paridad por cada byte. Violando el estándar 14443 parte 4, mezcla el nivel de enlace con el nivel de comunicación segura: los bits de paridad se calculan sobre el texto plano.

  b.- Se encripta con el mismo bit del flujo de clave que cifra el primer bit del siguiente byte del texto plano.

  c.- Durante el protocolo de autenticación, si el lector envía paridad incorrecta, la tarjeta detiene la comunicación. Sin embargo, si el lector envía bits de paridad correctos pero datos de autenticación incorrectos,

la tarjeta responderá con un código de error, lo que rompe la confidencialidad del cifrador, permitiendo que un atacante abra un canal paralelo.

- **Autenticaciones anidadas:**
  La memoria de la Mifare Classic está dividida en 16 sectores, cada uno de ellos tienen dos claves secretas. Para realizar una operación a un sector específico, el lector debe primero autenticarse usando la clave correspondiente. Cuando un atacante se autenticó a algún sector, subsecuentes intentos de autenticarse a otros sectores (sin conocer las claves sobre esos sectores), filtran 32 bits de información sobre la clave secreta de esos sectores.

Definición 3.3: Distancia entre desafíos de tag de dos intentos consecutivos.

Sean n_T y n'_T dos desafíos de tag. Se define la distancia entre ambos como: $d(n_T, n'_T) = \min suc^i(n_T) = n'_T$, con $i \in N$.

Si bien la distancia $d(n_T, n'_T)$ es diferente para cada tarjeta, es siempre igual en una misma tarjeta. Y ello es fácil de averiguar: autenticando dos veces consecutivas sobre un mismo sector al que se le conoce la clave.

ATAQUES (A TARJETAS):

- **Ataque por fuerza bruta:** Las vulnerabilidades respecto a los bits de paridad permiten reducir la complejidad de un ataque al espacio de claves de 48 bits por fuerza bruta.
- **Variando el desafío del lector:** Un atacante puede montar un ataque por texto cifrado elegido variando convenientemente la encripción del desafío del lector $\{n_R\}$[4]. Se usa la debilidad de lograr repetir a gusto el desafío del tag desenergizándolo.
- **Variando el desafío del tag:** En este enfoque, el atacante mantiene $\{n_R\}$, $\{a_R\}$ y los bits de paridad constantes, pero en su lugar, cambia $\{n_T\}$. El atacante espera la respuesta de la tarjeta. Cuando sucede, gana conocimiento acerca del estado interno del cifrador. Cuando lo logra, con la función de retroceso o rollback puede lograr el estado inicial del LFSR que es la clave.
- **Autenticaciones anidadas:** Luego que se haya autenticado sobre un sector, una autenticación sobre otro sector, a diferencia del primer caso, se inicia con los desafíos de tag (esta vez) encriptados. Se aprovecha que la distancia $d(n_T, n'_T)$ es constante lo cual facilita los intentos de obtener un conjunto de desafíos adecuados. La componente crucial de este ataque es el hecho de que las entradas de la función filtro son solamente posiciones impares del LFSR. Esto permite calcular por separado todas las posibilidades para las posiciones impares y para las posiciones pares del LFSR que son compatibles con el flujo de clave.

F. El lado oscuro de la seguridad por ocultamiento [8]:

El trabajo presentado por Nicolas Courtois [8] presenta ciertas características acerca de Mifare Classic, así como debilidades. Con ellas, presenta un protocolo con el cual es posible la obtención de una de las claves (A o B) de cualquier sector a elección, en poco tiempo, sin tablas pre-calculadas ni libros de clave. En el mismo, se asegura que el ataque ocupa a lo suno cinco minutos.

En especial, el ataque presentado en este trabajo se diseñó como un complemento más adecuado para el ataque presentado en García et al [7] denominado “auteticaciones anidadas”.

Adicionalmente, en este trabajo, el autor pretende poner en el debate la necesidad de tratar el tema de la vulnerabilidad de la economía del mundo frente a potenciales ataques sobre la infraestructura tecnológica, y el papel que ocupan empresas e investigadores en tecnología y seguridad informática. Se recomienda su lectura. Este tema es el que le da el nombre al trabajo. Más aún, cuando se hace referencia al ataque sobre Mifare aquí presentado, se lo llama “el ataque del lado oscuro”, a pesar que esta parte está separada de la presentación de descubrimientos y del ataque en sí.

HECHOS:

Se presentaron hechos relacionados sobre una debilidad encontrada relacionada con los bits de paridad.

- Si se corre el proceso de autenticación en una tarjeta de la cual no se conoce la clave, ésta responderá con un mensaje de error de 4 bits con probabilidad de 1/256. Este hecho se descubrió independientemente de lo presentado por Flavio García et al [7]
- La tarjeta responde con un mensaje de error encriptado {5} sí y solo sí los bits de paridad sobre el texto plano luego de desencriptar son correctos.
- Cualquiera sea el criptograma C, hay exactamente una elección de los bits de paridad P que genera que la tarjeta responda con un mensaje de error encriptado {5}. Se presentaron dos hechos respecto de la baja variabilidad del flujo de clave.
  - La probabilidad que los 3 bits del flujo de clave generada durante la desencripción de los 3 últimos bits del cuarto byte c_3 del criptograma C no dependa de esos bits de c_3 es de 0,75.
  - Si se fija el desafío del tag n_T y un prefijo de C de 29 bits, los 9 bits del flujo de clave generados en este proceso son constantes simultáneamente para 8 encripciones diferentes de C que comparten el mismo prefijo de 29 bit y varían los últimos 3 bits de c_3, con una probabilidad de 0,75. Se presentó un hecho descubierto relacionado con la propiedad diferencial múltiple.
  - Si se asume que el flujo de clave generado por el cifrador Crypto1 durante la desencripción del cuarto bit de C c_3, es constante y no depende de dicho byte, entonces la diferencia entre el estado del cifrador en cualquier instante del cálculo de ks2 y ks3 es una función lineal que depende solamente en las diferencias en c_3.

[4]Notación: {xx}: xx encriptado.

ALGORITMO:

Se definen los pasos a seguir para perpetrar el ataque.

1. Se realizan en promedio 128 consultas a un desafío de tag n_T fijo o aleatorio, con un criptograma aleatorio C = (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7) que reemplaza a {n_R}, {a_R} y un conjunto de bits de paridad aleatorio hasta obtener una respuesta de 4 bits: {5}.

2. Se mantiene el sincronismo de modo de fijar el desafío de tag n_T. Se mantienen fijos, también, los primeros 29 bits del criptograma C y por supuesto, los 3 primeros bits de paridad. También se mantiene fija la segunda mitad de C, es decir: c_4, c_5, c_6, c_7 . Así, solo son 3 bits los que se pueden variar: 8 casos posibles.

3. Se prueban en orden los 8 posibles casos para {n_R} y los diferentes 5 bits de paridad restantes. Por cada texto cifrado, exactamente uno de los 2^5 = 32 casos, la tarjeta responderá. Luego de $2^3.2^4$ = 128 consultas en promedio, se obtienen 8 respuestas con los 8 posibles valores consecutivos de {n_R}.

4. Las 8 respuestas de 4 bits recibidas brindan 32 bits de información de la clave de 48 bits.

5. Con probabilidad de 0,75 se pueden predecir simultáneamente las diferencias de los estados para las 8 encripciones.

6. Se puede usar el hecho que las funciones booleanas combinadas de Crypto1 reusan la mayoría de los bits de estado luego de 2 pasos. Entonces, exactamente 21 bits de estado determinan los bits del flujo de clave $ks_{3,0}$ y $ks_{3,2}$. Se examinan todos los 2^{21} casos posibles para cada texto cifrado donde la tarjeta ha respondido. Puede dividirse el tamaño de ese espacio por 4. Con 8 respuestas se determinan cerca de 2^5 = 32 posibles valores para los 21 bits.

7. Del mismo modo se obtienen 2^5 posibilidades para los otros 21 bits de estado que determinan los otros bits del flujo de clave $ks_{3,1}$ y $ks_{3,3}$.

8. Se tiene una lista de 2^{10} estados de 42 bits la cual se necesita ampliar a 2^{16} estados de 48 bits.

9. Luego simplemente haciendo vuelta atrás o rollback se obtienen 2^{16} = 65536 posibles claves, y verificando todos los $2^3.2^3$ bits de paridad involucrados en el ataque se permite conocer mejor aún cuál clave es la correcta. O si se encuentra una contradicción en cualquier etapa, significa que el flujo de clave depende de c_3, contrariamente a lo asumido.

10. Si esto falla, se repite todo el ataque. En promedio el ataque se repita 1/0,75 = 1,33 veces.

IV. Implementación de ataques

En este capítulo se presenta un detalle de la implementación práctica de los ataques denominados “lado oscuro” y “autenticaciones anidadas”, con un computador ordinario y un lector de bajo costo.

A. *Equipamiento usado*

Para la implementación de estos ataques se usó un computador portátil y un lector NFC de bajo costo.

El computador usado contiene un procesador Intel Atom CPU Z520 a 1,33GHz con 2Gb de memoria RAM.

En el mismo se instaló una distribución de Linux denominada Back Track versión 5 R3: es una distribución Linux especialmente pensada y diseñada para la seguridad informática. Se basa en Ubuntu 10.04 LTS (Long Time Support) de Cannonical, el cual tiene un núcleo Linux Debian. Puede obtenerme más información en [12].

El lector usado es un NFC ACR122 Touchtag. Es un lector de tarjetas de proximidad compatible con tarjetas de tecnología Mifare, ISO 14443 A y B, NFC y FeliCa.

Funciona a 13.56 MHz y cumple con la norma ISO/IEC18092 de Near Field Communication (NFC). Es compatible con el driver genérico CCID y PC/SC. Por lo que permite una fácil interoperabilidad con los diferentes dispositivos y aplicaciones. La figura 9 presenta la imagen del lector usado.

Figura 9: Imagen del lector NFC ACR122 Touchtag usado.

Se instaló la librería de código abierto Libnfc [9] para el manejo del lector. Sobre Libnfc se instalaron los dos programas que implementan los ataques previamente mencionados:

- **MFCUK (Mifare Classic Universal Toolkit): C**reado por Andrei Costin y Krishan Grupta que implementa el ataque “Lado Oscuro” de Nicolas Courtois [8].
- **MFOC (Mifare Off Line Cracker)**: Es un programa del grupo Nethemba que implementa el ataque de autenticaciones anidadas presentado por Flavio García et al en [7].

Pueden hallarse detalles de la instalación del equipamiento previamente presentado en [10 – 12].

B. *Uso de MFCUK*

MFCUK se usa para poder obtener una clave A o B de un sector a elección. Para lanzarlo basta con ejecutar el comando:

./mfcuk -C -v 2 -R 3:A -M 8

donde:

-C Para realizar la conexión.

-v 2 Modo “very verbose” (muy difuso).

-R 3:A Recuperar la clave A del sector 3.

-M 8 Tipo de tag: Mifare Classic 8K

Si en el modo uno selecciona -v 3, trabajará en modo difuso, pero mientras el programa se encuentra ejecutando presenta la cantidad de consultas, el tiempo transcurrido, y demás información que puede ser de interés observar.

C. *Uso de MFOC*

MFOC permite la obtención del total de las claves de una tarjeta a partir de conocer al menos una de ellas. Para ello, se ejecuta un comando que contiene una de esas claves. Por ejemplo:

./mfoc -k xxxxxxxxxxxx -O fullCard.mfd

En el ejemplo se ingresa como parámetro junto al comando la clave conocida y el contenido de la tarjeta se vuelca sobre un archivo hexadecimal llamado fullCard.mfd

MFOC posee las siguientes opciones:

-h: Imprime la ayuda y sale.

-k: incluye una clave.

-P: cantidad de pruebas por sector. Por default P=20.

-T: Tolerancia del desafío.

-O: Archivo de salida con información del ataque.

Si en lugar de obtener un archivo hexadecimal con la información del ataque se deseara obtener la información en un archivo con formato de texto plano, puede agregarse la correspondiente opción al final del comando. Por ejemplo:

./mfoc -k xxxxxxxxxxxx -O fullCard.mfd > fullCard.txt

V. Clasificación de tarjetas Mifare según la configuración de sus claves

Las tarjetas Mifare pueden presentarse en tres formas posibles según la configuración de sus claves.

A. *Modo transporte (Clase A)*

Cuando las tarjetas salen de fábrica, y hasta que se ponen en funcionamiento dentro del sistema destino, las mismas cuentan con una protección básica consistente en proveer a la tarjeta con claves A no legibles. Generalmente, las mismas pueden obtenerse de un listado de claves provistas por los fabricantes llamado “claves por default”. Las claves B son legibles.

Sin embargo, hay sistemas donde las tarjetas salen a servicio directamente con las claves A provistas por el fabricante, sin modificación alguna.

B. *Contienen al menos una clave por default (Clase B)*

Una vez definido el mapeo de los datos dentro de la tarjeta, muchos implementadores tienen interés solamente en proteger los sectores que contienen información. Así, es común encontrar en campo tarjetas Mifare que mantienen las claves por default provistas por el fabricante en sectores que no tienen información, y en cambio, en los sectores con información relevante (solo en esos), se modifican las claves.

C. *No contiene ninguna clave por default (Clase C)*

Independientemente del mapeo de datos dentro de la tarjeta, suele cambiarse la totalidad de las claves de todos los sectores, independientemente de la relevancia o no de la información contenida en cada sector.

La complejidad en la asignación de claves a cada sector forma parte de la política de seguridad del sistema. Cada tarjeta tiene 32 claves que deben manejarse con seguridad. Entonces, es común encontrar sistemas donde a una tarjeta se le asigne una clave A y una clave B para todos los sectores a los que se le cambie la clave. Asignar más de una clave A y más de una clave B a cada tarjeta puede hacer que la gestión de claves se transforme en una cuestión de difícil manejo. Por ejemplo, el sistema de transportes londinense Oyster usa una sola clave A y una clave B que se repiten en todos los sectores de la tarjeta. Y este formato se puede encontrar en otros sistemas de transporte masivo de Sudamérica.

VI. Atacando tarjetas Mifare según la clasificación de la configuración de sus claves

En este capítulo se atacará a las tarjetas con ambos programas (MFCUK y MFOC). Luego se las clasificará según lo expresado en el capítulo anterior.

A. *Atacando Mifare con MFCUK*

MFCUK se diseñó para obtener una clave de un sector para luego complementarse con MFOC (autenticaciones anidadas) para obtener el resto de las claves.

Sin embargo, es posible lograr la totalidad de las claves de una tarjeta con MFCUK. La figura 10 contiene el resultado del relevamiento (tiempo y cantidad de autenticaciones) realizado durante un ataque a una tarjeta con MFCUK.

Sector	Clave A			Clave B		
	Clave	# Aut.	t (h:m)	Clave	# Aut.	t (h:m)
Sector 00	c24ab16ef53d	7986	0:27'	d41fe65ba32c	23995	1:20'
Sector 01	c24ab16ef53d	22060	1:14'	d41fe65ba32c	22495	1:15'
Sector 02	c24ab16ef53d	288642	16:02'	d41fe65ba32c	21884	1:13'
Sector 03	c24ab16ef53d	5578	0:19'	d41fe65ba32c	44899	2:30'
Sector 04	c24ab16ef53d	40750	2:16'	d41fe65ba32c	77861	4:27'
Sector 05	c24ab16ef53d	21105	1:10'	d41fe65ba32c	9578	0:32'
Sector 06	c24ab16ef53d	5595	0:19'	d41fe65ba32c	4880	0:16'
Sector 07	c24ab16ef53d	2450	0:08'	d41fe65ba32c	4696	0:15'
Sector 08	c24ab16ef53d	29937	1:40'	d41fe65ba32c	10290	0:34'
Sector 09	c24ab16ef53d	8782	0:29'	d41fe65ba32c	10932	0:36'
Sector 10	c24ab16ef53d	20689	1:09'	d41fe65ba32c	25798	1:26'
Sector 11	c24ab16ef53d	8002	0:27'	d41fe65ba32c	14871	0:49'
Sector 12	c24ab16ef53d	6758	0:22'	d41fe65ba32c	10170	0:34'
Sector 13	c24ab16ef53d	7321	0:24'	d41fe65ba32c	32918	1:50'
Sector 14	c24ab16ef53d	16905	0:56'	d41fe65ba32c	19778	1:06'
Sector 15	c24ab16ef53d	18806	1:03'	d41fe65ba32c	44117	2:27'
Total:		511366	28:25'		379162	21:10'

Figura 10: Atacando todos los sectores de Mifare con MFCUK.

En más del 90% de las oportunidades que se inició el programa se obtuvo una clave como resultado.

Sin contemplar interrupción alguna, la obtención de la totalidad de las claves ocupó un tiempo de 49h:35'.

BREVE ANALISIS ESTADISTICO:

Variable de análisis: el tiempo para la obtención de una clave.

- El promedio: X = tiempo total / cantidad de claves.
  $\Rightarrow$ X = 49h:35' / 32 $\Rightarrow$ X = 1h:32'.
- El desvío estándar es dS = 2h:47'
- El coeficiente de variabilidad CV = X / dS = 1,799.

Conclusión: Como CV = 1,799 >> 0,1 se concluye que el tiempo promedio para la obtención de una clave mediante MFCUK no es un indicador de tendencia central adecuado

para la muestra obtenida. Esto significa que es falso afirmar que puede realizarse exitosamente el ataque a una clave de una tarjeta mediante MFCUK en aproximadamente 1h:47' en promedio. Por otro lado, recuérdese que Nicolas Courtois [8] enunció que dicho ataque se realiza en un promedio de 5 minutos, valor muy lejano al valor aquí obtenido y a los resultados presentados en distintos foros ([10] y [12]).

MFCUK Y LA CLASIFICACION DE TARJETAS:

Dada la naturaleza del ataque del lado oscuro implementado mediante MFCUK, es indistinto que la clave del sector atacado sea o no una clave por default. Por consiguiente, realizar un ataque total a cualquier tarjeta Mifare usado MFCUK no será diferente a lo presentado previamente.

B. Atacando Mifare con MFOC

MFOC funciona partir de conocerse una clave de acceso a algún sector, cualquiera. Dado que hay muchos sistemas que dentro de sus tarjetas suelen mantener al menos una clave sin cambiar, puede usarse MFOC solo para obtener la totalidad de las claves de una tarjeta. A continuación se presentan ataques con MFOC a diferentes configuraciones de tarjetas.

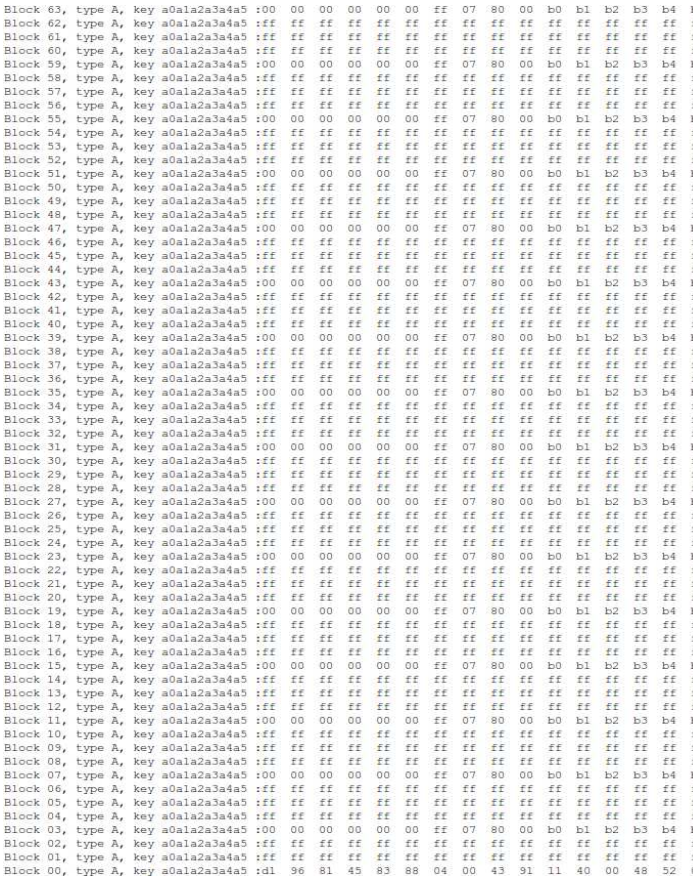

Figura 11: Ataque a una tarjeta Clase A con MFOC.

MFOC EN TARJETAS CLASE A:

Previamente se mencionó que las tarjetas en modo transporte (Clase A según clasificación) contienen todas las claves A por default. Por lo tanto, es de esperar una respuesta positiva como resultado del ataque. Se ejecuta el comando:

./mfoc -O transportCard.mfd

La figura 11 muestra el resultado del ataque con MFOC a una tarjeta en modo transporte. Obsérvese los sectores trailer, en los bytes 6, 7, 8 y 9 se presentan las condiciones de acceso: FF 07 80 00. Puede verificarse en la documentación [1] y en [4] que las mismas corresponde con el modo transporte.

El tiempo promedio para obtener todas las claves con MFOC en una tarjeta Clase A es de 14 segundos.

MFOC EN TARJETAS CLASE B:

Cuando el implementador del sistema define que solo cambiará las claves de los sectores que interesa preservar, dado que contienen información relevante, está dejando al menos una clave por default, lo que deja a la tarjeta indefensa frente a un ataque con MFOC.

La figura 12 muestra información obtenida tras implementar un ataque con MFOC a una tarjeta Clase B.

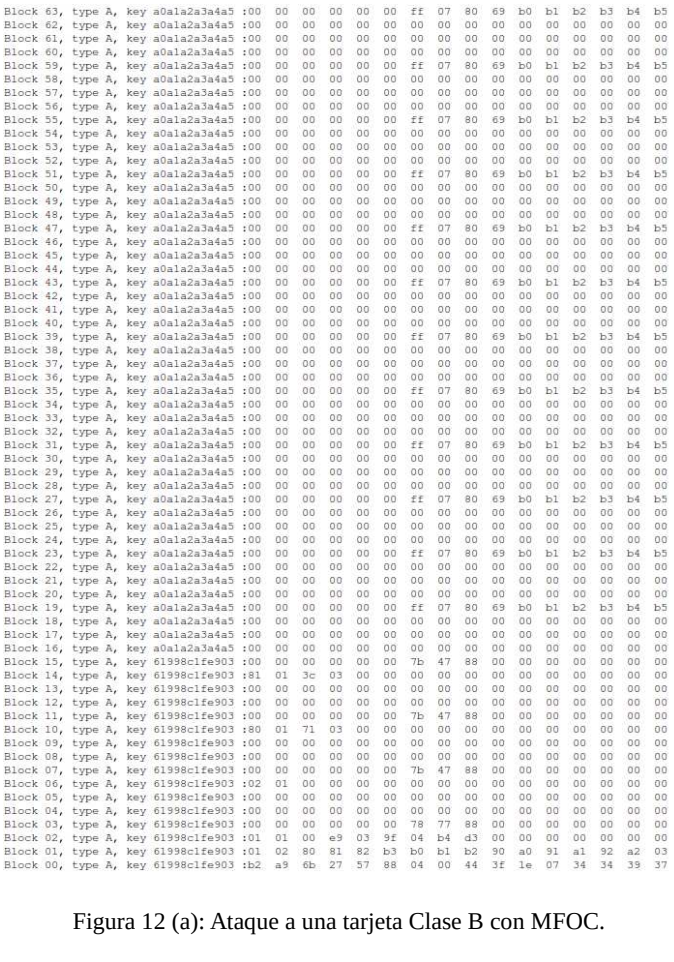

Figura 12 (a): Ataque a una tarjeta Clase B con MFOC.

```
Sector: 00, type B: Found Key: B [99cfd51fe903]
Sector: 01, type B: Found Key: B [99cfd51fe903]
Sector: 02, type B: Found Key: B [99cfd51fe903]
Sector: 03, type B: Found Key: B [99cfd51fe903]
```

Figura 12(b):Ataque a una tarjeta Blase B con MFOC: las claves B.

El tiempo promedio para obtener todas las claves con MFOC en una tarjeta Clase B es de 25 minutos.

MFOC EN TARJETAS CLASE C:

Si el implementador del sistema definese que ninguna clave provista por el fabricante debe continuar, el uso de MFOC sobre tarjetas clase C no permite obtener resultados positivos.

La figura 13 muestra una impresión de pantalla luego del intento de ataque con MFOC a una tarjeta Clase C.

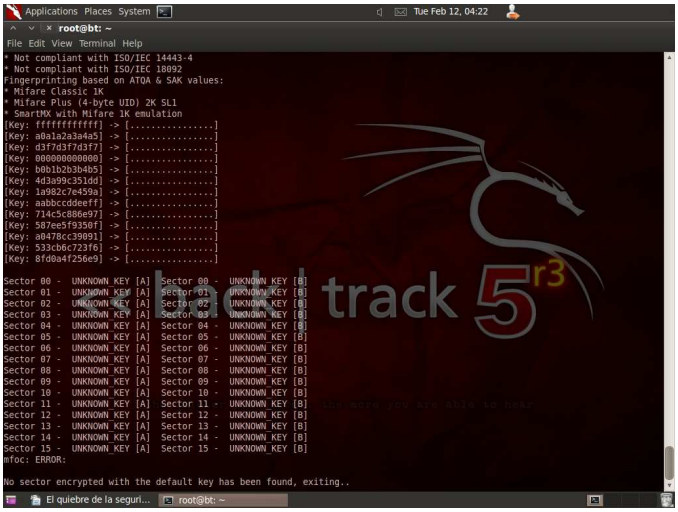

Figura 13: Ataque fallido a una tarjeta clase C con MFOC.

El tiempo promedio para obtener el resultado fallido del ataque con MFOC en una tarjeta Clase C es de 20 segundos.

C. Ataque combinado a Mifare con MFCUK y MFOC

En el caso de atacar a un sistema con tarjetas clase C, el uso de MFOC no permite obtener resultados positivos. Se analiza en este punto, el uso de un ataque que combina el uso de ambos programas.

Para ello, se requiere conocer al menos una de las claves. Para su obtención, ya que no existe ninguna clave ni por default ni otra conocida, se usa MFCUK para la obtención de una de ellas. Luego, se ingresa su valor como parámetro en MFOC para obtener el resto de las claves.

En el ejemplo presentado en la figura 10, tomamos la clave A del sector 00. Para ello, se ejecuta el comando:

./mfcuk -C -v 3 -R 0:B -M 8

El tiempo relevado para la obtención de la clave es de 27 minutos. La clave es: d41fe65ba32c.

Obtenida la clave, se puede ingresar el valor de dicha clave como parámetro en el lanzamiento del programa MFOC, ejecutando el siguiente comando:

./mfoc -k d41fe65ba32c -O fullCard.mfd > fullCard.txt

El tiempo hasta la obtención de la totalidad de las claves a partir de una clave usando MFOC es de 7:30 minutos.

Finalmente, mediante el uso combinado de MFCUK y MFOC se pudo obtener la totalidad de las claves de una tarjeta clase C en 34:30 minutos.

VII. Conclusiones

En honor a los primeros trabajos exitosos [2] y [3] realizados sobre Mifare Classic, que motivaron a muchos investigadores a avanzar sin descanso y así se logró llegar a la realidad actual, de manera preliminar se puede concluir primeramente, que *realizar un trabajo de ingeniería inversa sobre un chip aparentemente seguro y avanzado puede lograrse sin necesidad de contar con un laboratorio con equipamiento costoso*.

El hecho de mantener oculto el algoritmo del criptosistema hasta las últimas instancias, violando los principios de Kerckhoff, parece indicar que el mismo carece de fortalezas necesarias como para ser sometido al escrutinio de los investigadores. Y con sólo un trabajo que presentara algo de información relevante, se inició una catarata vertiginosa de trabajos trascendentes. En efecto, se ocultaba algo: que el chip era muy poco seguro. Entonces, una segunda conclusión puede ser que *si se deseara criptoanalizar un dispositivo, puede ser una buena idea iniciar con aquellos que mantienen oculto sus algoritmos*.

En III.E se resumió el trabajo de Flavio García et al “hurto sin cable” presentado en [7]. En el mismo se explotaron vulnerabilidades en el proceso de autenticación a varios sectores (autenticaciones anidadas), y vulnerabilidades en los bits de paridad, presentes (estas últimas) por el no cumplimiento de la parte 4 de la norma ISO 14443 y en las recomendaciones presentes en [14]. A causa de ello, que logró abrir un canal paralelo de comunicación que permitió perpetrar diversos ataques a tarjetas Mifare con computador ordinario y un lector de bajo costo. Se puede concluir, entonces, que *durante un proceso de criptoanálisis, es buena idea verificar si el dispositivo analizado cumple con las normas, estándares y principales recomendaciones, y elegir aquellos que no cumplan*.

El trabajo presentado en [8] permitió lograr un ataque a las tarjetas con el mismo hardware para obtener una clave a elección rápidamente. Pero la implementación práctica del ataque no se logró reproducir en los tiempos declarados por el autor.

Teniendo en cuenta que la circuitería lógica del cifrador contiene cerca de 400 NE frente a los 3400 NE que tiene el cifrado AES en las tarjetas RFID [3], y teniendo en cuenta que en [8] se declara que funciones idénticas se presentan repetidamente en varias etapas, gastando ineficientemente recursos de lógica, resulta razonable creer que se pretende mostrar al cifrador más fuerte de lo que realmente es, como se declara en [8]. Así, resulta lógico que se hayan encontrado todas las vulnerabilidades presentadas y que finalmente Crypto1 resultara ser mucho más débil de lo que se suponía cuando se comenzó a analizar el chip.

Respecto de la implementación de los ataques, se puede destacar que *pueden realizarse tres combinaciones de ataques diferentes, con las dos herramientas utilizadas (MFCUK y MFOC), según las características de las configuraciones de claves*:

1. *En caso de tratarse de tarjetas en modo transporte*

(clase A), con MFOC pueden obtenerse las claves en 14 segundos.

2. *Si se estuviese en presencia de un sistema con tarjetas con al menos una clave por default (clase B), con MFOC pueden obtenerse la totalidad de las claves en 25 minutos.*

3. *En caso de tratarse de tarjetas que no presenten ninguna de sus claves por default (clase C), es recomendable realizar un ataque mixto entre MFCUK para obtener primero una clave y luego con ella realizar un ataque usando MFOC y la opción -k e ingresando la clave obtenida como parámetro. Teniendo en cuenta que usando MFCUK, para la mitad de los sectores, las claves se obtuvieron en promedio en 28 minutos. Ingresada la clave obtenida como parámetro, usando MFOC, el tiempo para la obtención del resto de las claves promedia los 7 minutos. En estas condiciones, puede decirse que puede obtenerse la totalidad de las claves de una tarjeta Mifare sin claves default en 35 minutos.*

Con el simple análisis de los datos obtenidos del ataque realizado mediante MFCUK, se puede recomendar que en caso de no obtener una clave en 28 minutos, se detenga el programa, y se reinicie el mismo en otro sector.

Finalmente, dado que la configuración de claves y la política de gestión de claves se mantienen constantes dentro de un sistema, para poder atacar a la totalidad del sistema (miles o millones de tarjetas), basta con analizar a una pequeña muestra (con las mismas herramientas). Una vez identificada la política o configuración de claves, puede elegirse la estrategia para realizar los ataques, en función de lo aquí presentado.

VIII. Trabajos Futuros

En la región Sudamericana hay un gran número de sistemas con tarjetas Mifare usadas como medios de pago para diferentes servicios: algunos tienen varios millones de tarjetas e infraestructura importante y otros son muy pequeños y absolutamente elementales. Entre los primeros, podemos mencionar algunos: la tarjeta Capital del sistema de buses rápidos de Bogotá llamado Transmilenio o el Sistema Único de Boleto Electrónico del Estado Nacional de Argentina llamado SUBE. Entre los últimos, podemos mencionar en general a las tarjetas de residentes de varios sistemas de peaje de rutas nacionales, provinciales y corredores viales, sistemas de pago del estacionamiento medido y del sistema de buses de algunas ciudades medianas y pequeñas de la región. Estos últimos son los que más abundan y pueden ser más simples de atacar, debido a que quizás no cuenten con gran infraestructura (ni conocimientos) para una rápida y efectiva defensa.

Una línea de investigación puede presentar un proyecto de varias etapas que analice la robustez de las medidas de seguridad que pueden tomar las empresas operadoras y organismos estatales luego de descubiertas las vulnerabilidades de Mifare Classic:

El primer nivel de defensa que realiza la mayoría de las empresas es el silencio, lo que suele llamarse “Seguridad por ocultamiento”.

El segundo nivel puede ser la implementación de robustos (y caros) sistemas de back office, post-procesamiento, gestión de listas negras y supervisión en línea, incluso con tecnología de comunicaciones de banda ancha móvil. Con ello, se puede supervisar en línea las transacciones de cada tarjeta, por lo que la seguridad del sistema deja de descansar en Crypto1, y Mifare deja de tener sentido.

El tercer nivel de defensa, quizás hoy el más seguro, puede consistir en reemplazar las tarjetas Mifare Classic por Mifare Desfire.

Un primer trabajo pretende realizar una recopilación de los principales sistemas, estudiar su configuración de claves y el mapeo de datos de sus tarjetas, y así derribar la barrera del ocultamiento.

Todos los fabricantes licenciatarios de Mifare Classic aseguran que el número de identificación del chip es único. Es decir, no podrán encontrarse dos tarjetas que presenten el mismo uid. Sin embargo, hay declaraciones en trabajos y en foros relacionados con este tema, que mencionan que se han podido comprar “tarjetas pirata”, cuya uid puede modificarse.

Una tarjeta Mifare Classic con uid modificable es una herramienta perfecta para clonar tarjetas válidas a las que se les han podido copiar sus claves. Una vez logrado esto, las transacciones realizadas con ellas es posible que no puedan diferenciarse de transacciones realizadas con tarjetas auténticas, sin importar las dimensiones del sistema de back office o de supervisión que la empresa disponga. Puede realizarse un trabajo que analice la factibilidad de clonar tarjetas auténticas mediante estas “tarjetas pirata” y usarlas en los sistemas sin ser detectados. Y así derribar la barrera de seguridad mediante sistemas de gestión y supervisión (incluso supervisión en línea).

Existen comentarios en los foros acerca del descubrimiento inminente de vulnerabilidades en las tarjetas Mifare Desfire. Un trabajo de investigación puede analizar las posibles vulnerabilidades y la factibilidad de lograr el quiebre de la seguridad de Mifare Desfire.

Referencias

[1] MF1ICS50 Functional specification. http://www.nxp.com/ acrobat/ other/ identification/ M001053_MF11CS50_rev5_3.pdf, 2008.

[2] Nohl, Karsten y Plötz, Henryk: "Mifare, little security, despite obscurity". 24º Congreso del Caos Computer Club, Berlin, 2007.

[3] Nohl, Karsten; Plötz, Henryk y Evans, David: "Reverse-Engeneering a Cryptographic RFID tag" Usenix Security Symposium 2008.

[4] Identification cards - Contactless integrated circuits cards - Proximity cards (ISO/IEC 14443) 2001.

[5] de Koning Gans, Gerhard; Hoepman, Jaap-Henk and Garcia, Flavio: "A practical attack on the Mifare Classic" Cardis 08, Springer, 2008.

[6] Garcia, Flavio; de Koning Gans, Gerhard; Muijrers, Robert; van Rossum, Peter; Verdult, Roel; Wichers Schreur, Ronny; y Jacobs, Bart: "Dismantling Mifare Classic" European Symposium Esorics, 2008.

[7] Garcia, Flavio; van Rossum, Peter; Verdult, Roel y Wichers Schreur: "Wirelessly Pickpocketing a Mifare Classic card" IEEE Symposium on Security and Privacy, 2009.

[8] Courtois, Nicolas: “The dark side of security by obscurity, and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime”. IACR Cryptology ePrint. Archivo 2009: 137, 2009.

[9] libnfc.org - Public platform independent Near Field Communication (NFC) library. http://www.libnfc.org, 2013.

[10] Amado, Roberto: "Hacking RFID, rompiendo la seguridad de Mifare" Security Art Work: http://www.securityartwork.es, S2 Grupo, 2009.
[11] Krishan Gupta: “Hacking Mifare Classic”, http://www.nicolascourtois.com/MifareClassicHack.pdf, 2013.
[12] Backtrack linux ORG: “RFID cooking with Mifare Classic”. http://www.backtrack- linux.org/ wiki/ index.php/ RFID_Cooking_with_Mifare_Classic, 2013.
[13] Menezes, Alfred; van Oorschot, Paul y Vanstone, Scott: “Handbook of applied Cryptogrphy”, CRC Press 51 edición, 2001.
[14] Krawczyk, Hugo: “The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)” en Advances in cryptology (Crypto '01), Springer, 2001.

Jorge Kamlofsky received a Bachellor in Mathematic from Universidad Abierta Interamericana (UAI) and he finished a SP in cryptography at Information Security from EST (Army Engineering School) and is a student of a Master in IT in the UAI. Actually he is a professor of Discret Mathematics and Physics. He also is a reasercher in the IT Faculty of UAI. Fields of interest: Computer Vision and Simetric cryptography in RFID devices.