Mifare Classic analysis

in Czech Republic / Slovakia

Ing. Pavol Lupták, CISSP, CEH
Lead Security Consultant

Legal disclaimer

- Nethemba s.r.o. is not responsible for a public misuse of Mifare Classic cards in Czech or Slovak republic
- this presentation is supposed to be Mifare Classic security analysis in Czech / Slovak environment, not a manual that can be misused for commiting crimes

Contents

- Background
- Mifare Classic basics & security
- Mifare Classic attacks in theory
- Available hardware tools & software implementations
- Vulnerabilities in Slovak Mifare Classic cards
- Our Mifare Classic Offline Cracker

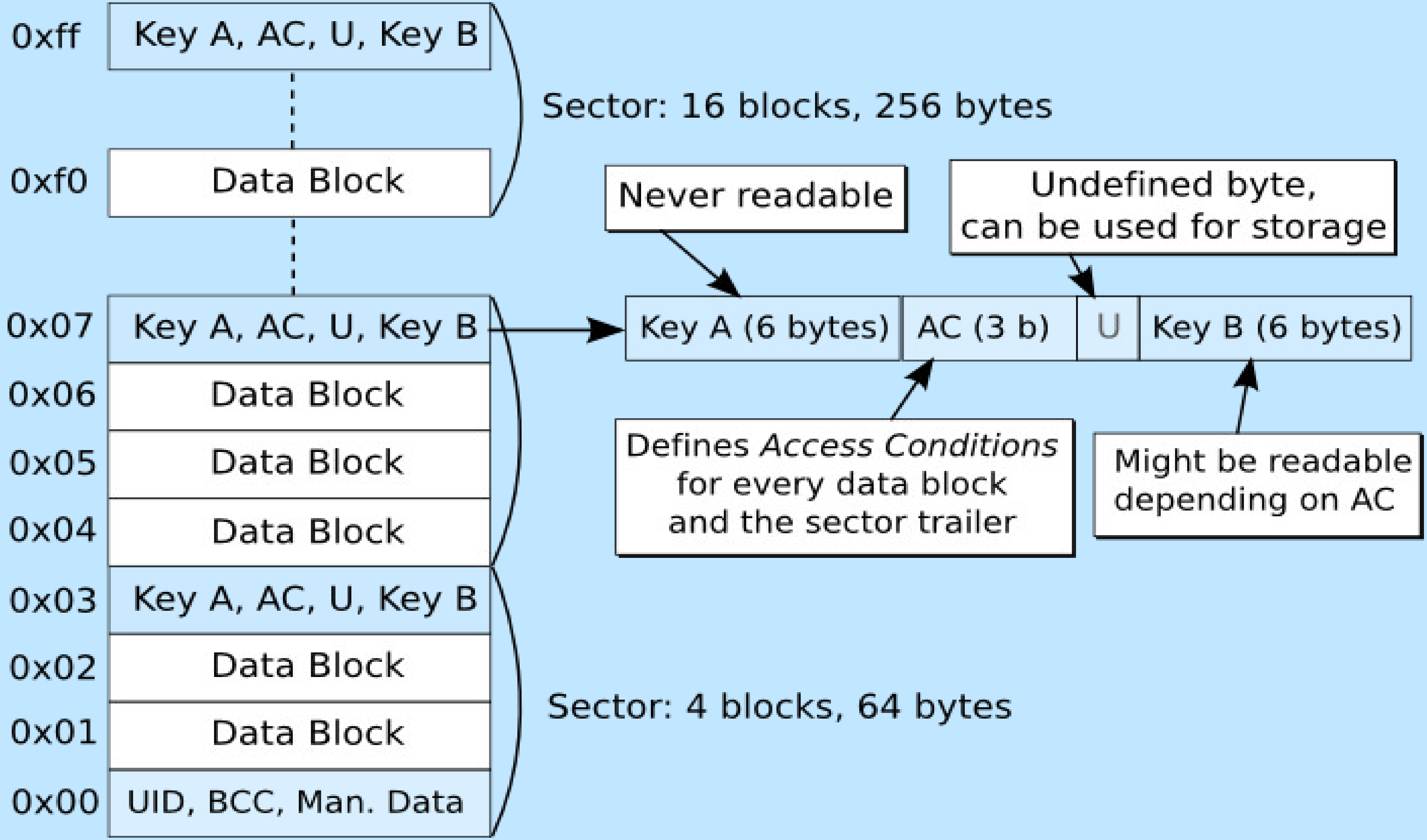
Mifare Classic technology

- one of the most used RFID card (more than 1 billion smart card chips are used)
- is based on ISO/IEC 14443 Type A, 1kB or 4kB
- uses 13.56 Mhz contactless smartcard standard
- uses a proprietary CRYPTO1 with 48 bits keys
- had a lot of security problems in the past but nobody cares :-)
- it's cheap (about 1 €)

Usage in Czech/Slovak republic

- all University/ISIC/Euro26 cards
- public transport ID (“električenka”) in Bratislava
- Slovak Lines, Slovak railways cards
- parking cards
- for the current list see http://www.emtest.biz/sk/

Mifare Classic structure



0xff	Key A, AC, U, Key B
0xf0	Data Block
0x07	Key A, AC, U, Key B
0x06	Data Block
0x05	Data Block
0x04	Data Block
0x03	Key A, AC, U, Key B
0x02	Data Block
0x01	Data Block
0x00	UID, BCC, Man. Data

Sector: 16 blocks, 256 bytes

Sector: 4 blocks, 64 bytes

Key A (6 bytes)	AC (3 b)	U	Key B (6 bytes)

Never readable

Undefined byte, can be used for storage

Defines *Access Conditions* for every data block and the sector trailer

Might be readable depending on AC

Mifare Classic security

- read-only Unique Identifier (UID)
- mutual authentication between reader and writer and encrypted communication
- CRYPTO1 non-public algorithm implementation
- obfuscated parity information
- hardware implementation only

Mifare Classic commands

- **authenticate**
- **read, write, increment, decrement** – always sent in encrypted session
- **transfer** – writes the result of decrement, increment/restore to non-volatile memory
- **restore** – prepares the current value of a block for being rewritten to non-volatile memory

Mifare Classic default keys

- a lot of publicly used cards (even in Czech Republic / Slovakia) use at least one block encrypted with default keys:

0xffffffffffff **0xa0a1a2a3a4a5**

0xb0b1b2b3b4b5 **0x4d3a99c351dd**

0x1a982c7e459a **000000000000**

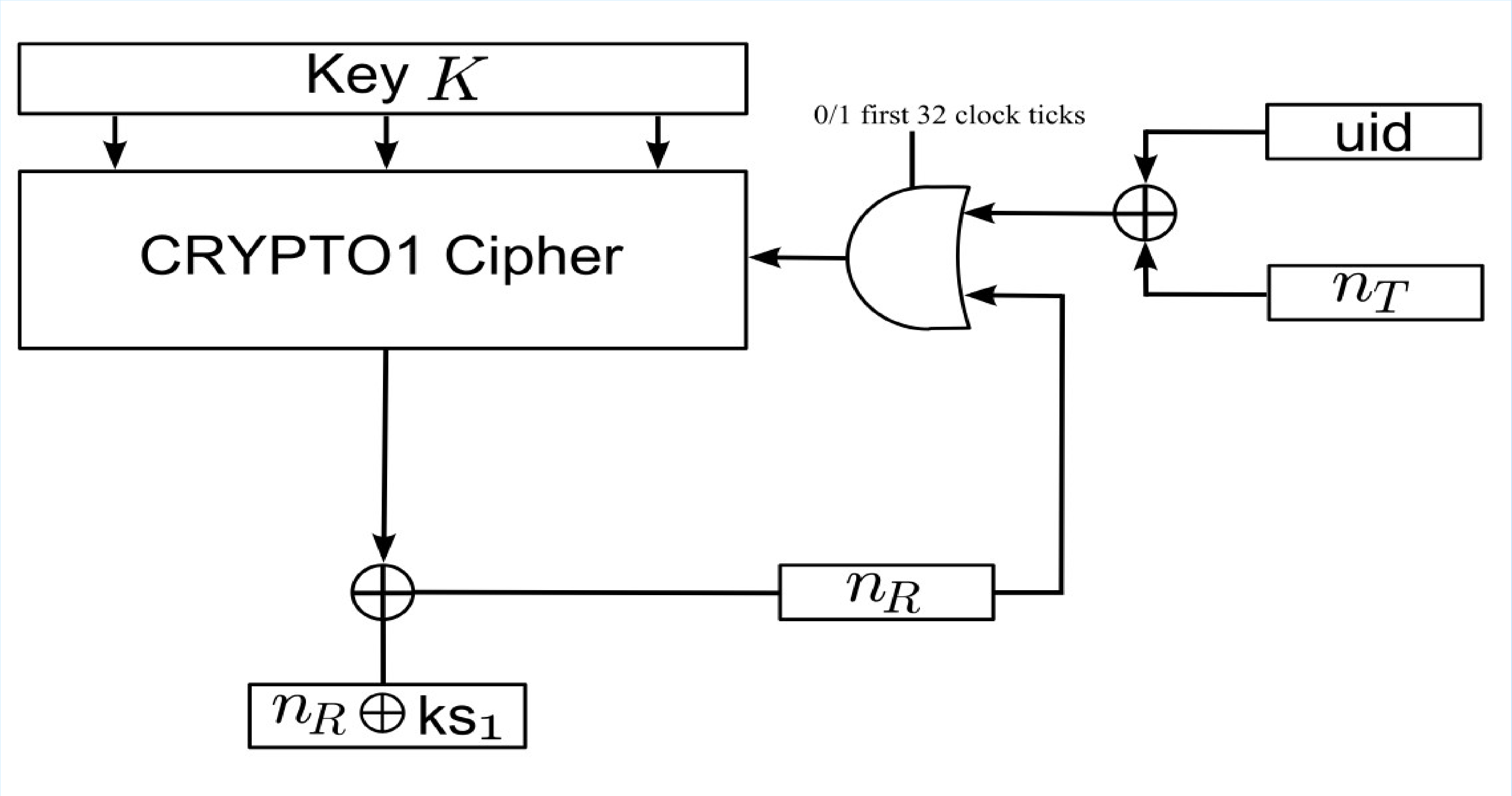
0xd3f7d3f7d3f7 **0xaabbccddeeff**

Linear Feedback Shift Register (LFSR)

- pseudo random generation defined by the polynomial **x^16 + x^14 + x^13 + x^11 + 1**
- length is 32 bits, but it has only 16 bits entropy!
- L16 = x0 XOR x11 XOR x13 XOR x14 XOR x16
- Ar = suc2(Nt), At = suc3(Nt)
- generated nonces can be predicted in the time

CRYPTO1 Cipher initialization

- No non-linear feedback



Authentication process

Step	Sender	Hex	Abstract
01	Reader	26	req type A
02	Tag	04 00	Answer req
03	Reader	93 20	select
04	Tag	c2 a8 2d f4 b3	uid, bcc
05	Reader	93 70 c2 a8 2d f4 b3 ba a3	select(uid)
06	Tag	08 b6 dd	MIFARE 1k
07	Reader	60 30 76 4a	auth(block 30)
08	Tag	42 97 c0 a4	Nt
09	Reader	7d db 9b 83 67 eb 5d 83	Nr XOR ks1,Ar XOR ks2
10	Tag	8b d4 10 08	At XOR ks3

Tag	Reader
picks Nt and sends to reader	ks1 <- cipher(K, uid, Nt), picks Nr
ks1 ← crypto1(K, uid, Nt)	ks2, ks3 .... ← cipher(K, uid, Nt, Nr) and sends to tag Nr XOR ks1, suc2(Nt) XOR ks2
ks2, ks3.. ← cipher(K, uid, Nt, Nr)	
sends to reader suc3(Nt) XOR ks3	Ar = suc2(Nt)

Authentication process with timeout – how to recover ks2, ks3

Ghost	Reader
picks Nt and sends to reader	ks1 <- cipher(K, uid, Nt), picks Nr
	ks2, ks3 .... ← cipher(K, uid, Nt, Nr) and sends to tag Nr XOR ks1, suc2(Nt) XOR ks2
Wait for timeout	
	Reader sends to the tag halt XOR ks3

“timeout” Attack in practice

- computing offline LFSR state table (for 2^36 entries) LFSR state from 0 to 0xffffffff and adequate ks2 ks3, it takes 4-8 hours
- computing online Nt table (for 2^12) entries from 0 to 0xfff0 and adequate ks2 ks3 → there is one Nt producing LFSR for a given ks2 ks3, it takes 2-14 minutes
- rolling back Nr, Nt XOR uid and the result key

Nested Attack

1. Authenticate to the block with default key and read tag's Nt (determined by LFSR)
2. Authenticate to the same block with default key and read tag's Nt' (determined by LFSR) (this authentication is in an encrypted session)
3. Compute “timing distance” (number of LFSR shifts)
4. Guess the Nt value and authenticate to the different block

Other Mifare Classic mistakes

- reader-side accepts invalid frame-lengths
- the parity bit is encrypted, but the internal state will not shift → the first bit of the next byte will be encrypted by the same keystream bit
- only 20 bits are used or keystream bit
- statistical bias in the cipher
- influence of bits is not balanced

Cloning

- when all keys are gained, every card can be easily cloned
- we can make 99.6% clone (except 0.block in 0.sector that contains read-only UID)
- all blocks (including UID!) can be 100% emulated by Proxmark3
- **protection against cloning** – make whitelist of allowed UIDs, or always use it in card content encryption

Restoring Credit

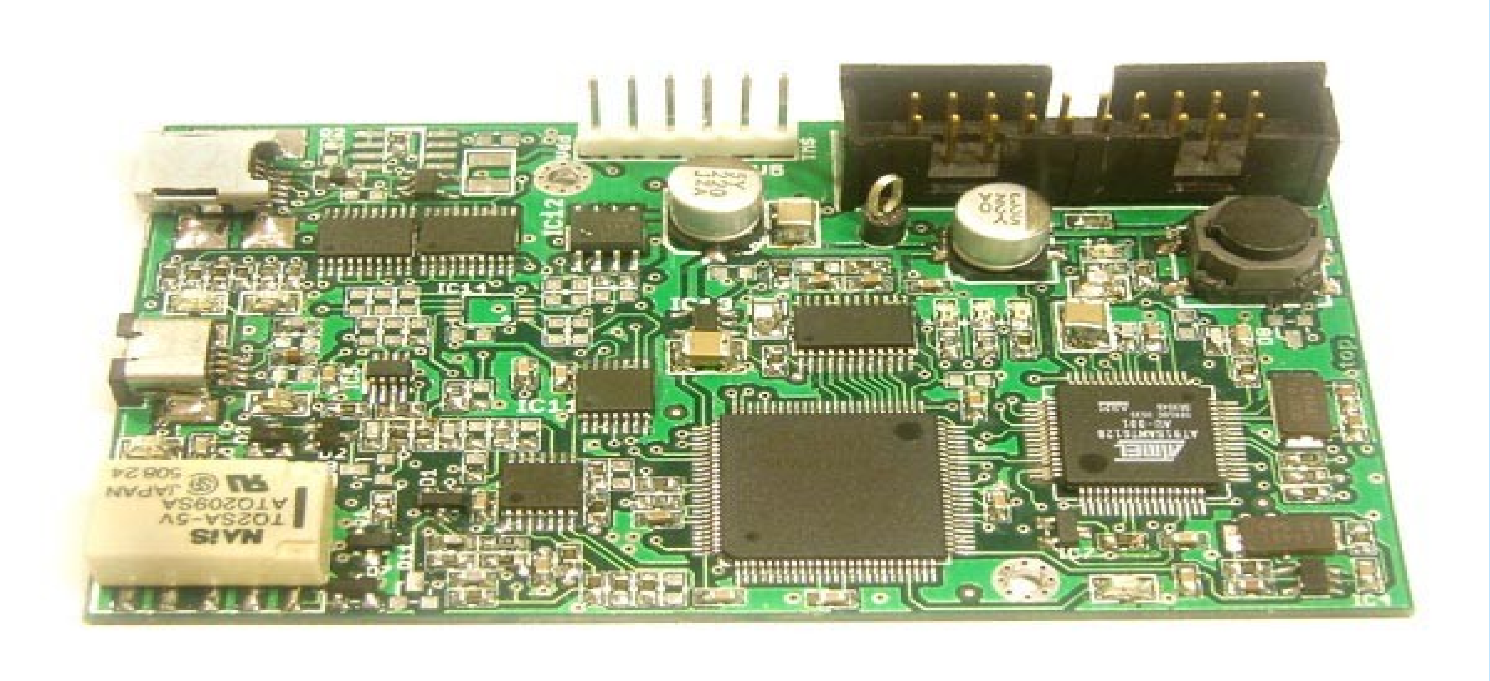
- Anti-cloning protection does not work against dumping the whole card - when you decide to charge your card and restore the dump with original credit (UID remains the same)
- **Countermeasure #1** – use safer cards (Mifare Plus/DESFire or other)
- **Countermeasure #2** – use “decrement counter” protection (it's only “workaround”)
- **Countermeasure #3** – use online checking

Crapto1

- open implementation of attacks against the CRYPTO1 cipher
- can be used for cracking Mifare Classic initial authentication handshake
- our “nested offline” card attack is based on crapto1 libraries

Proxmark3

- general-purpose RFID tool designed to snoop, listen and emulate everything from LF (125kHz) to HF (13.56Mhz) tags, universal hacking RFID tool :-)



Tikitag / Touchatag

- very cheap (30 EUR), NFC-based RFID reader/w

Slovak Mifare Classic vulnerabilities

- all tested cards use the same keys (!!!) for the first 1024 bytes (first 16 keys are the SAME!)
- there is always at least one sector encrypted with default key! (possibility of nested attacks)
- the name of passenger/owner is always stored in 0xd block – imagine what can happens with strong antenna :-)
- no protection against cloning or modification!
- can be easily cloned and modified!!!

Mifare Classic binary analysis

- we have done binary difference analysis between new bought card, after its activation and charging credit
- 0xd block – passenger/user name
- 0x24 block - “električenka” expiration date
- 0x81 block – student's university number
- 0x82 block – student's name

Attacker's costs

- 30 € – tikitag / touchatag RFID reader/writer (sufficient for reading / cracking / writing / cloning Mifare Classic cards)
- $ 449 – Proxmark 3 (just for advanced RFID playing :-)
- 1 € for blank 4kB Mifare Classic (can be bought on ebay.com from Thaiwan/China :-)

Solution

- use safer technology + strong cryptography, bind user identity with card's read-only UID + use UID in card content encryption
- **partial workaround:** bind user identity with card's read-only UID, use UID in card content encryption, use UID whitelists, use “decrement counter” solution

“Decrement-counter” workaround

- replacing all Mifare Classic cards to safer ones is very expensive and time-consuming – is it possible to use insecure Mifare Classic layer with “secure” implementation???
- “decrement counter” (initially set to 0xffffffff), keys A/B have permissions only for decrementing counter and cannot be changed, content of card (with passenger credit) is encrypted/hashed with card UID, decrement counter and private key

Our Mifare Classic Offline Cracker

- the first public disclosure of Mifare Offline cracker based on “Nested Attack” already published by Dutch researchers
- we want to demonstrate that massively used Mifare Classic cards can be easily cracked / dumped
- can be found here https://www.nethemba.com/research/
- use it, improve it and let us know the bugs

Mifare Classic Key Recovery Tool

- “Dark side” paper attack implementation by Andrei C
- recovers at least one key for a card that can be used with our MFOC Nested Attack
- http://code.google.com/p/tk-libnfc-crapto1/
- wait for MFOC integration!

What's next?

- wait for our hitag analysis! (most of Czech/Slovak “badge” cards are affected, and yes – it's also used in Renault / Opel / Peugeot/ Citroen / … car keys :-)
- playing with GSM, see & support http://reflextor.com/trac/a51 project, all GSM communication can be cracked!

References

- http://nickholdren.com/wp-content/uploads/2009/07
- https://har2009.org/program/attachments/123_[HAR
- http://www.cs.ru.nl/~flaviog/publications/Talk.Mifare
- http://www.cs.ru.nl/~petervr/papers/grvw_2009_picl
- http://code.google.com/p/crapto1/
- http://www.touchatag.com/
- http://proxmark3.com/