Workshop Introduction to RFID

1

Join the following WiFi network

SSID: OctoBox

PSK: RFIDWorkshop

2

Register on the CTF platform

http://score.rfidwork.shop

flag{XXXX}

3

Download the workshop files

http://files.rfidwork.shop

Workshop: Introduction to RFID

Download the workshop files:
http://files.rfidwork.shop

Join the following WiFi network:

SSID: OctoBox

PSK: RFIDWorkshop

Version 2024.07

Vinnie Vanhoecke

IT Security Consultant

Belgium

Radio Amateur (ON3RCE)

Bachelor Thesis about researching different real-life RFID implementations and its weaknesses.

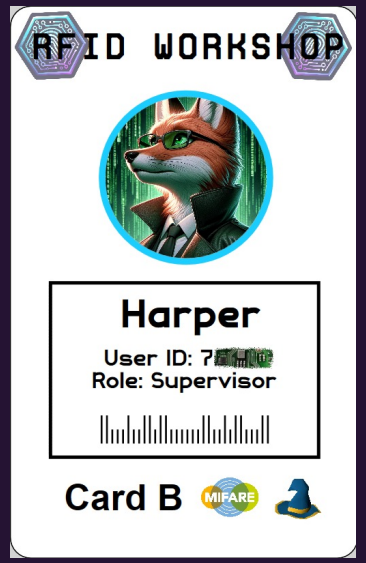
RFID Workshop in Brucon 2017 and DEFCON 2018

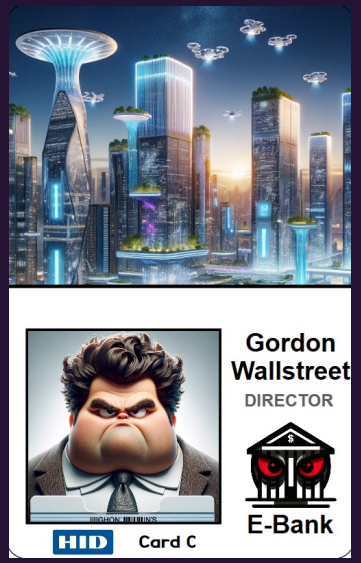
https://github.com/VinnieV

Workshop Use Cases

- Spread Awareness
- Red Teaming
- Legitimate Reasons
- Fun

Workshop Cards





Zebra ZXP Series 3

Booster Packs

MPC®
MAKEPLAYINGCARDS.COM

https://www.makeplayingcards.com

Workshop Bundles

http://files.rfidwork.shop

Challenge Bundle
Contains all challenges without solutions

Solutions Bundle
Contains all challenges with solutions

Slides
Workshop presentation

Mifare keys
Wordlist used in one of the challenge

Theory Bundle
Setup instructions & RFID theory

OctoBox

OpenScad

PrusaSlicer

Raspberry Pi + Touchscreen + RF Module
Software written in Python3 + Kivy

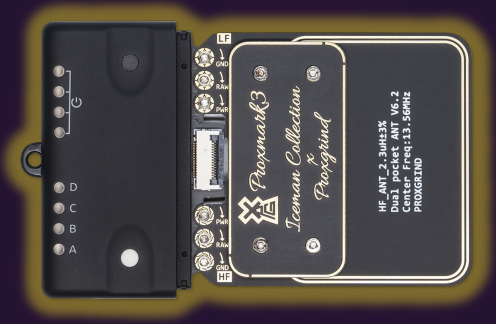
Workshop Readers

ACR122U

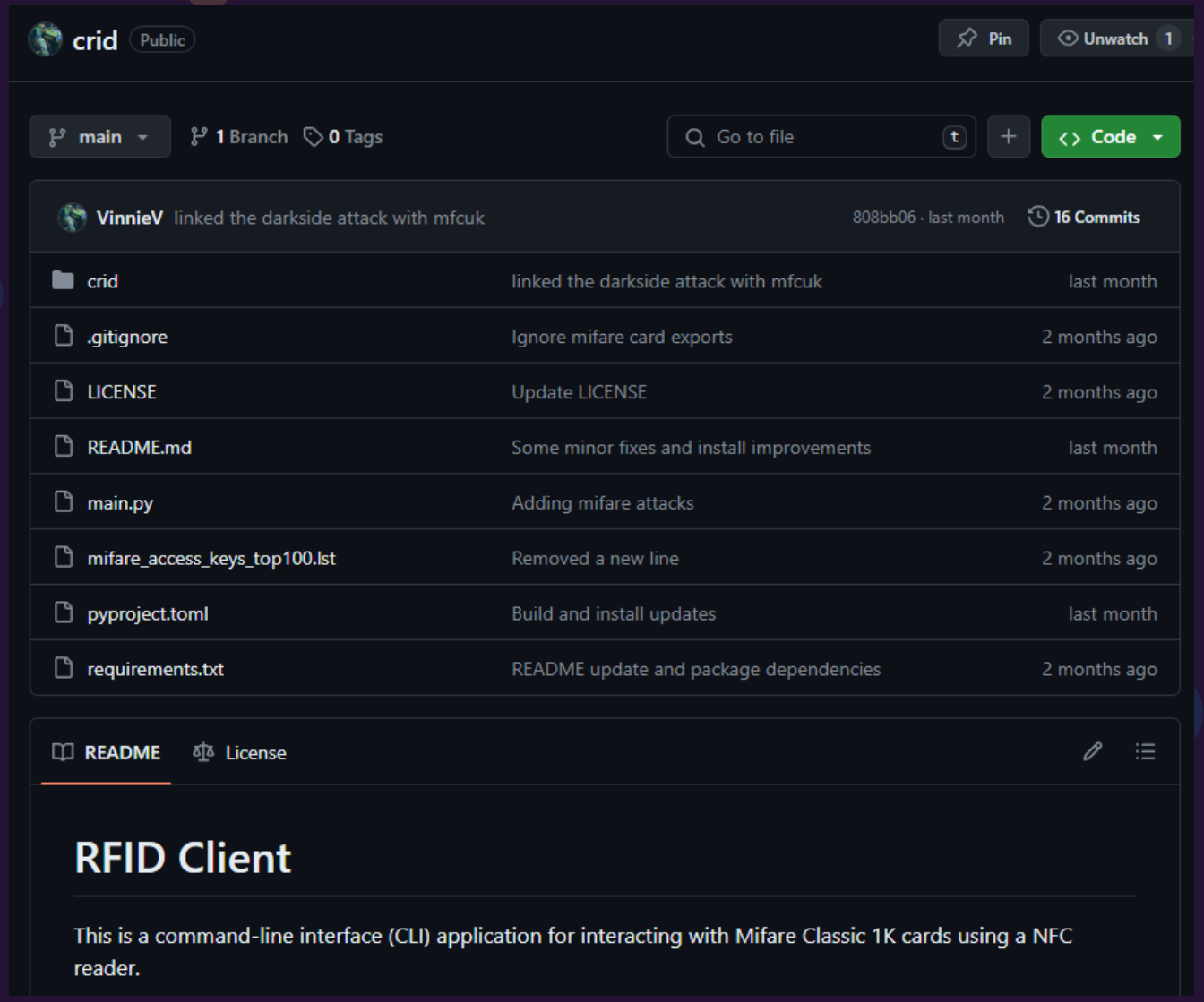
NFC Compliant
Not expensive 30-50 EUR
Good library support
Uses NXP PN532 chip

Flipper Zero

Proxmark3



CRID



https://github.com/VinnieV/crid
https://pypi.org/project/crid/

Python3 required

Install instructions

```
sudo apt-get update
sudo apt-get install libpcsclite-dev
libpcsclite1 pcscd pcsc-tools

pip3 install crid
```

Workshop USB-stick

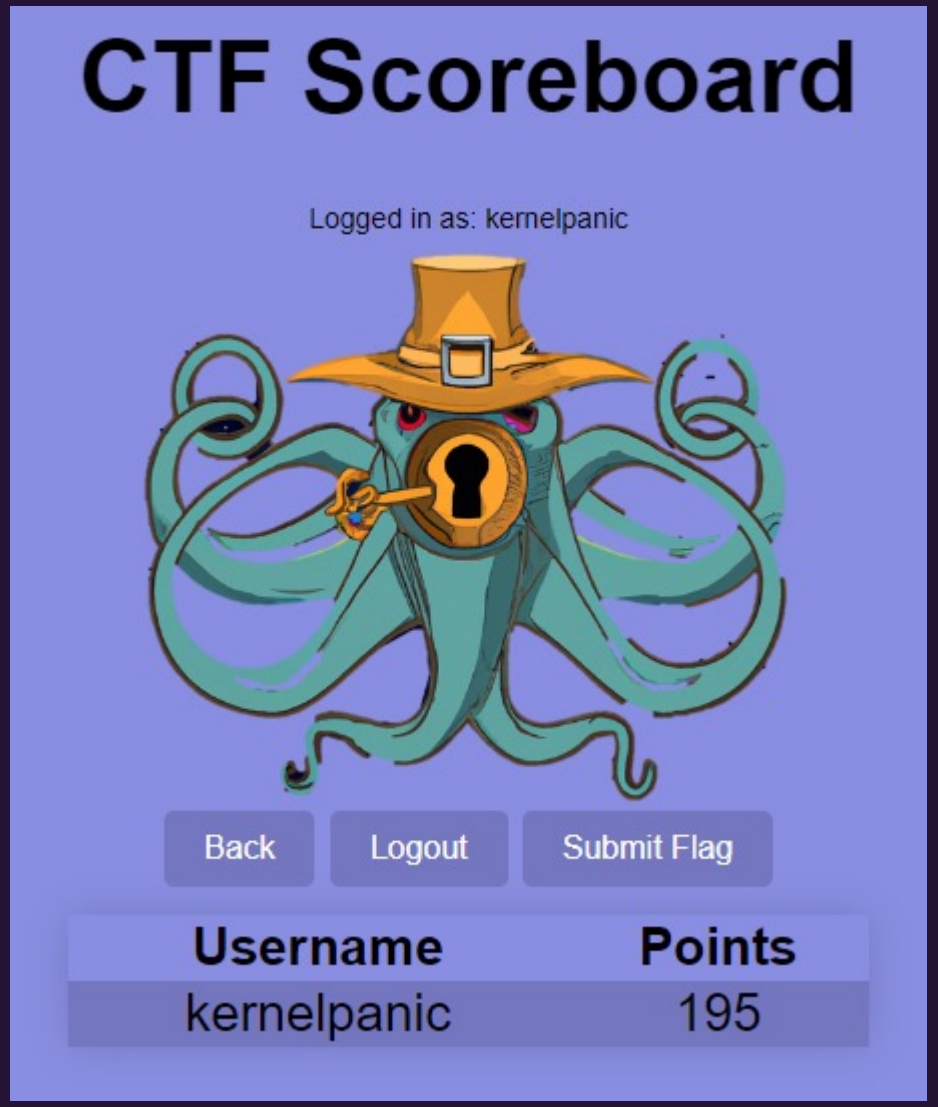
Cubic
https://github.com/PJ-Singh-001/Cubic

Linux Live boot USB

Ubuntu 22.04 LTS

Workshop tools installed

Scoreboard



On the Octobox wifi network:
http://score.rfidwork.shop

Register your username
(alphanumeric and max 16 characters)

Flag Format:
flag{XXXX}

There are 17 flags to find/obtain

Intro and Basics

🕒 20 min

Setup and first challenge

🕒 30 min

Mifare Theory & Practical Challenges

🕒 1-2 hours depending on your skill level and tools

*If you are experienced with Mifare Classic, you could follow your own pace

🏆 The Challenges 🏆

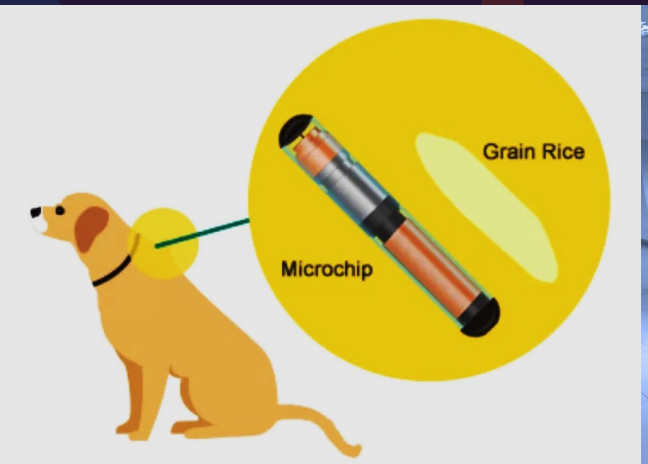
Some challenge might award you a challenge coin.

Introductory challenges

Harder challenges

What is RFID?

Radio Frequency Identification



Different Forms

A couple of examples

Sticker/Paper **Bio Tags** **Key Fobs** **Cards**

Secure?

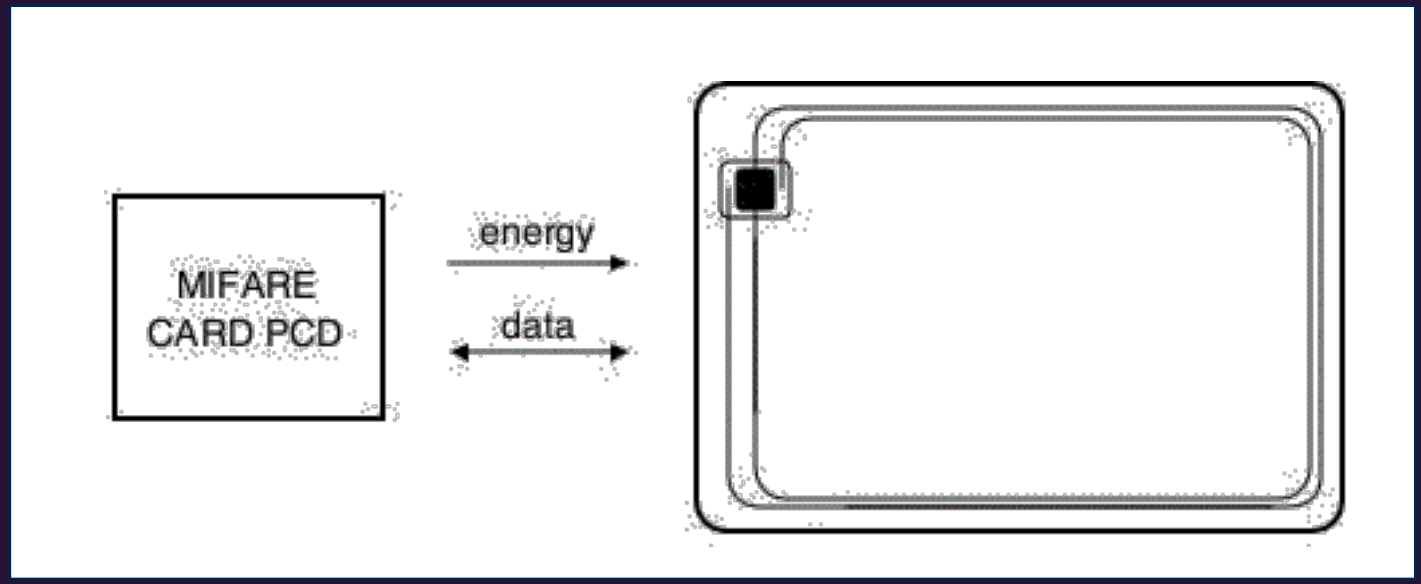
- Used Personal Identifier
- Provides physical access (office buildings, concerts, schools, hotels, ...)
- Financial operations (EMV)
- Festival tickets/balance cards

How?

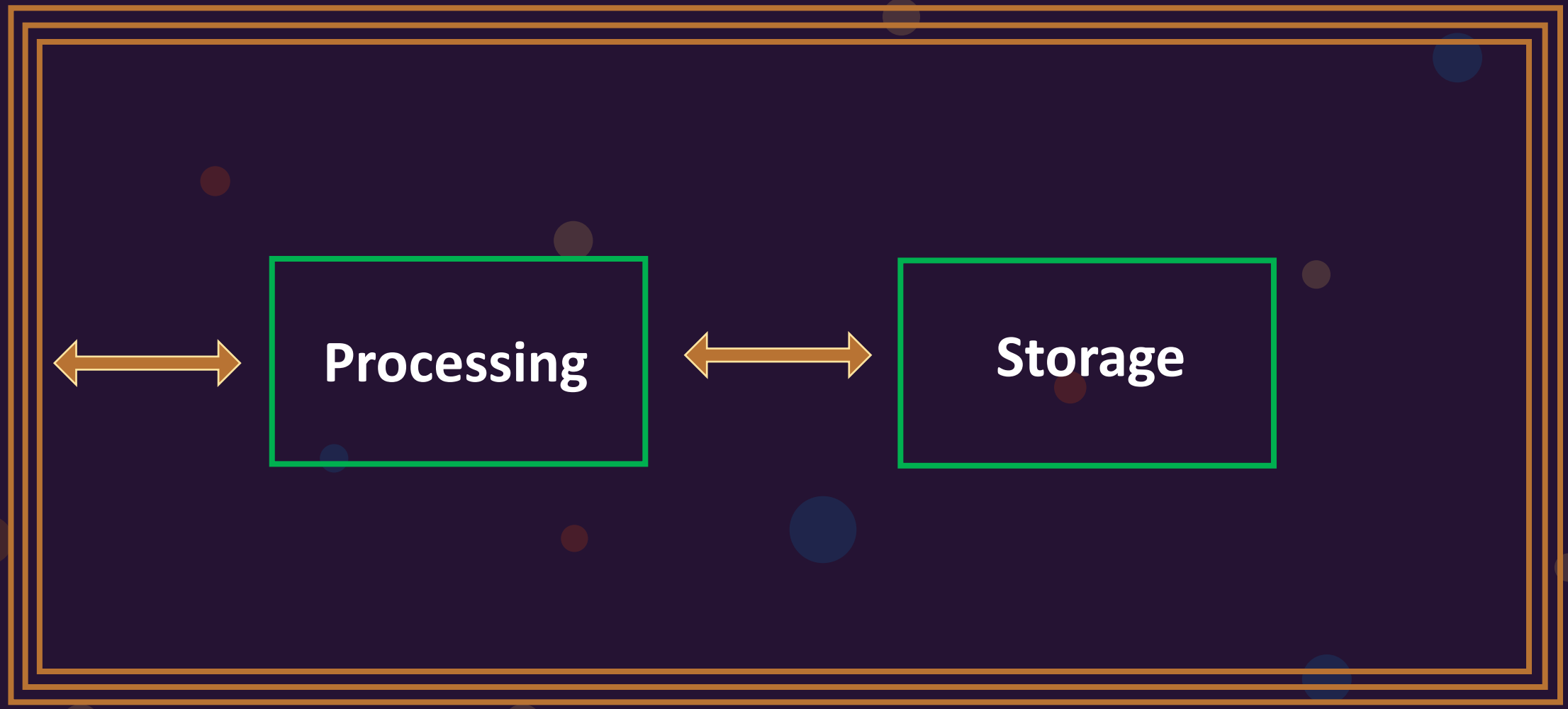
Magic 🧙

Radio Electro Magnetic Waves

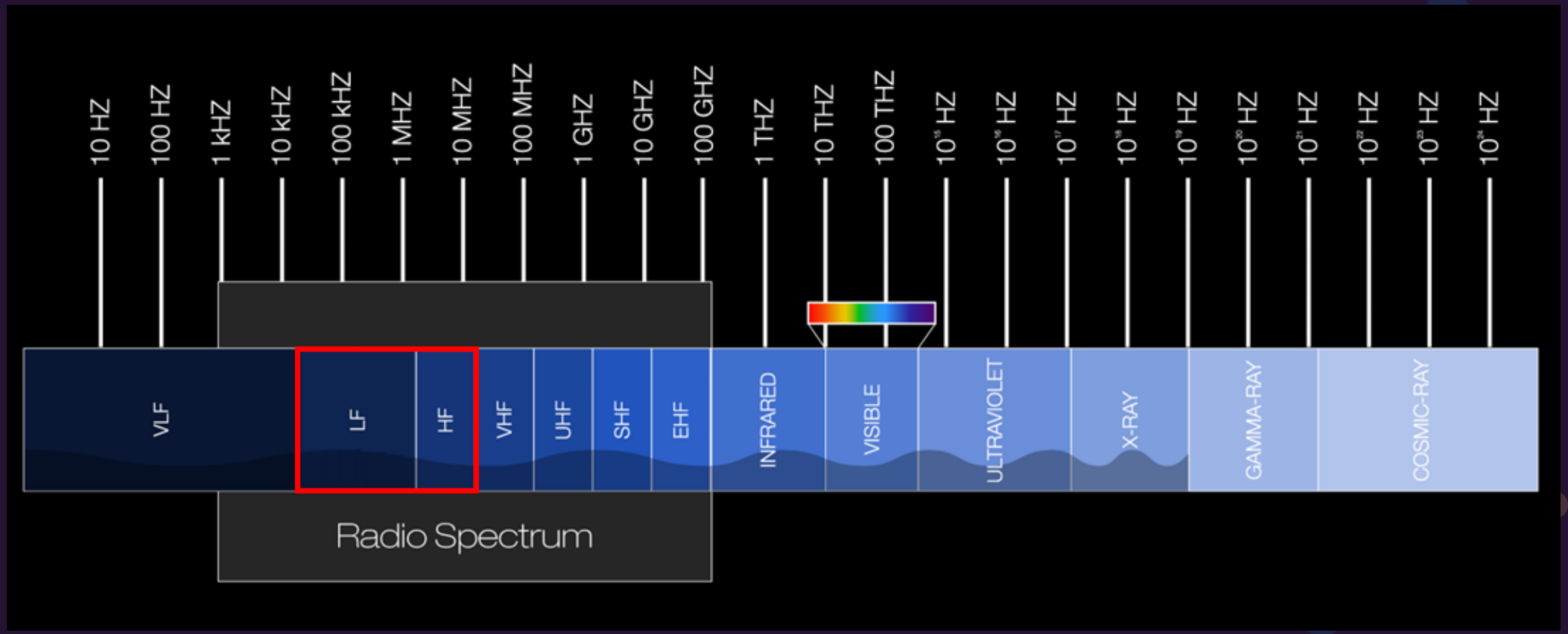
Wireless Power



Mifare Internals

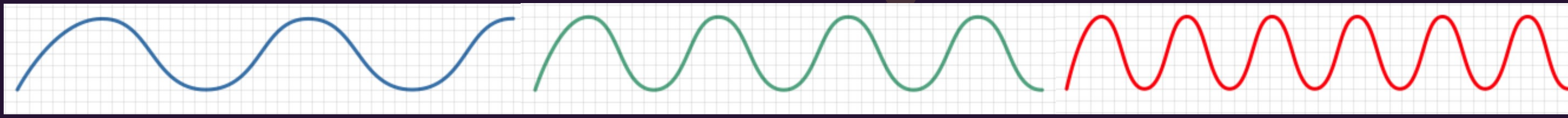


Radio Spectrum



Different frequencies

These are the most common configurations, deviations can exist.



Low Frequency

- 125 kHz or 134 kHz
- Low Read Distance
- Usually less secure

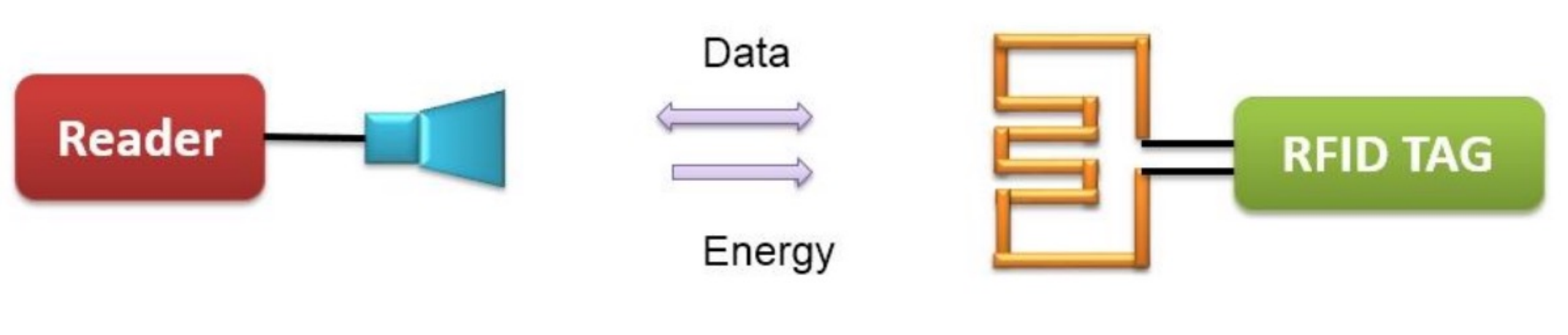
High Frequency

- 13,56 MHz
- Low Read Distance
- Higher transfer rates

Ultra High Frequency

- 860 MHz – 960 MHz
- >10m Read Distance
- Used in logistics

Data?



Mifare Classic Block 0 Read

```
[usb] pm3 --> hf mf list
[=] downloading tracelog data from device
[+] Recorded activity (trace len = 188 bytes)
[=] start = start of start frame end = end of frame. src = source of transfer
[=] ISO14443A - all times are in carrier periods (1/13.56MHz)

     Start |        End | Src | Data (! denotes parity error)                                                  | CRC | Annotation
-----------+------------+-----+--------------------------------------------------------------------------------+-----+-----------------
         0 |        992 | Rdr |52(7)                                                                           |     | WUPA
      2100 |       4468 | Tag |04  00                                                                          |     |
      7040 |       9504 | Rdr |93  20                                                                          |     | ANTICOLL
     10548 |      16436 | Tag |69  cd  1e  50  ea                                                              |     |
    122240 |     132768 | Rdr |93  70  69  cd  1e  50  ea  30  97                                              |  ok | SELECT_UID
    133812 |     137332 | Tag |08  b6  dd                                                                      |  ok |
    139520 |     144224 | Rdr |60  00  f5  7b                                                                  |  ok | AUTH-A(0)
    145716 |     150452 | Tag |26  6c  ca  e9                                                                  |     | AUTH: nt
    159872 |     169184 | Rdr |2e  80  6c  da  f0! 50  b6  f9                                                  |     | AUTH: nr ar (enc)
    170292 |     174964 | Tag |c4! f2  e6  2b                                                                  |     | AUTH: at (enc)
    180864 |     185568 | Rdr |1d! 57  31! 99                                                                  |     |
           |            |  *  |                                                  key FFFFFFFFFFFF prng WEAK    |     |
           |            |  *  |30  00  02  A8                                                                  |  ok | READBLOCK(0)
    186676 |     207476 | Tag |b6! c9! 24! 0b! d2  81  4b! ec  67! 01  44! 61  71! 13! de  8c! 73! 96          |     |
           |            |  *  |69  CD  1E  50  EA  08  04  00  02  2E  D7  66  E2  94  75  1D  E7  08         |  ok |
    220416 |     225120 | Rdr |ff  14  91  2d                                                                  |     |
           |            |  *  |50  00  57  CD                                                                  |  ok | HALT
```

Mifare Classic Sector 0 Read

Load key

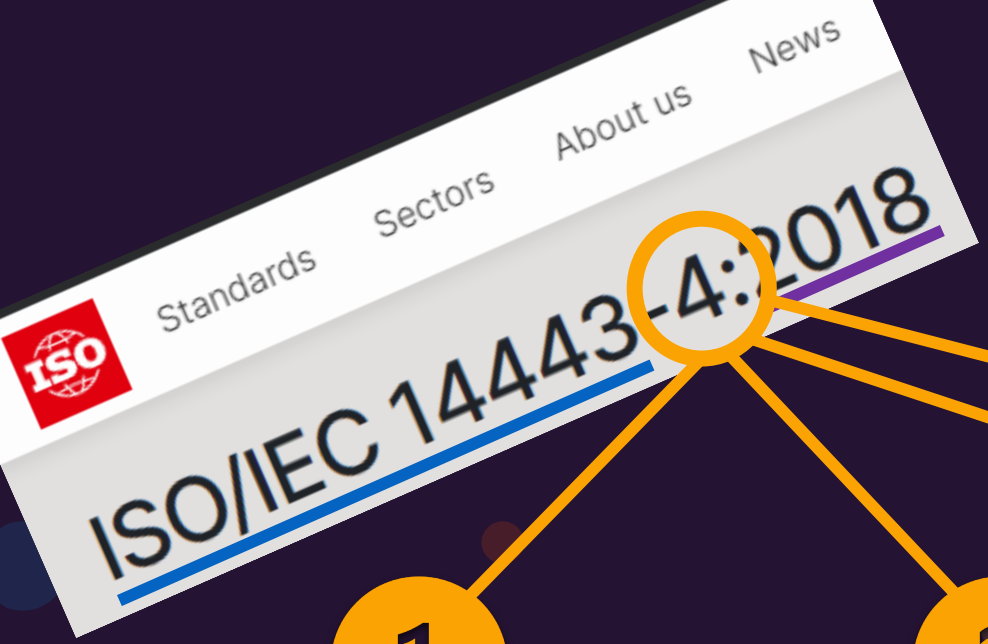
Auth block 0 with key A

Read the four blocks

```
; ------------------------------------------------------------
; Read sector 0
; ------------------------------------------------------------

; [1] Load (Mifare Default) key in reader (key location 0)
FF 82 00 00 06 FF FF FF FF FF FF

; [2] Authenticate sector 0, Block 0 with key A(60) at location 0
FF 86 00 00 05 01 00 00 60 00

; [3] Read the full 16 bytes from Sector 0, Block 0
FF B0 00 00 10

; [4] Read the full 16 bytes from Sector 0, Block 1
FF B0 00 01 10

; [5] Read the full 16 bytes from Sector 0, Block 2
FF B0 00 02 10

; [6] Read the full 16 bytes from Sector 0, Block 3
FF B0 00 03 10
```

APDU commands

Load Authentication Keys APDU Command										
Example: FF82000006FFFFFFFFFF										
Length is 11 Bytes:										
1	2	3	4	5	6	7	8	9	10	11
Class	INS*	Key Structure	Key Location	Lc	Key (6 bytes)					
FF	82	00	00|01	06	XX XX XX XX XX XX					



RFID Layers

ISO 14443 Parts

1

Physical characteristics

2

Radio frequency power and signal interface

3

Initialization and anticollision

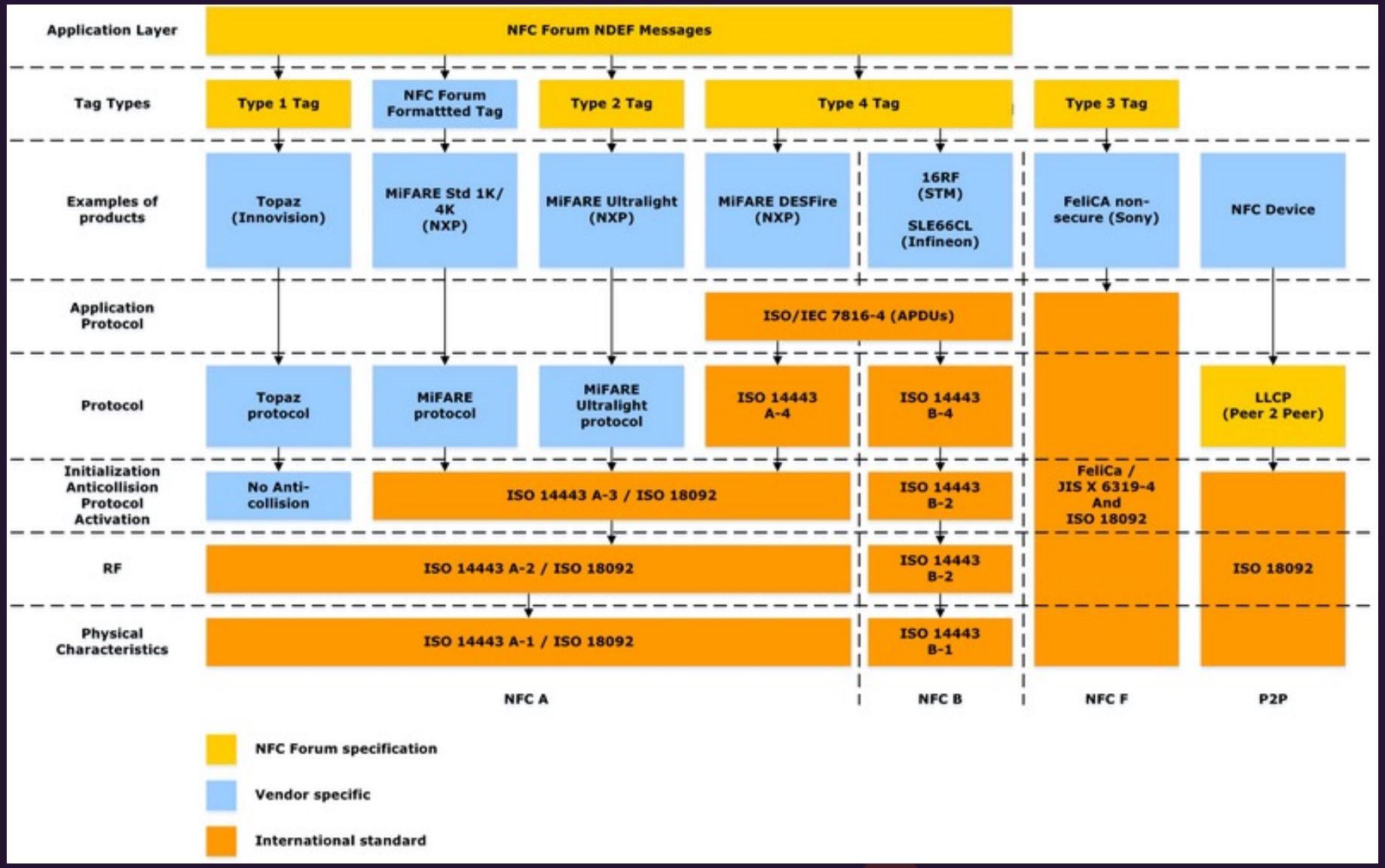
4

Transmission protocol

RFID Protocols

RFID Protocol	Date of Invention	Frequency
Low Frequency		
EM4100	1970s	125 kHz
Indala	1980s	125 kHz
HID Prox	1991	125 kHz
Hitag1, Hitag2	1994	125 kHz
Temic T55x7	Not specified	125 kHz
Unique	Not specified	125 kHz
High Frequency		
FeliCa	1987	13.56 MHz
Legic	1992	13.56 MHz
Mifare	1994	13.56 MHz
ISO 15693	2001	13.56 MHz
NFC (Near Field Communication)	2002	13.56 MHz
Ultra-High Frequency		
EPCglobal UHF Class 1 Gen 2	2004	860-960 MHz
ISO 18000-6A, 6B, 6C	Various	860-960 MHz

NFC


Application Layer
NFC Forum NDEF Messages
Tag Types
Type 1 Tag
NFC Forum Formattted Tag
Type 2 Tag
Type 4 Tag
Type 3 Tag
Examples of products
Topaz (Innovision)
MiFARE Std 1K/ 4K (NXP)
MiFARE Ultralight (NXP)
MiFARE DESFire (NXP)
16RF (STM)
SLE66CL (Infineon)
FeliCA non-secure (Sony)
NFC Device
Application Protocol
ISO/IEC 7816-4 (APDUs)
Protocol
Topaz protocol
MiFARE protocol
MiFARE Ultralight protocol
ISO 14443 A-4
ISO 14443 B-4
LLCP (Peer 2 Peer)
Initialization Anticollision Protocol Activation
No Anti-collision
ISO 14443 A-3 / ISO 18092
ISO 14443 B-2
FeliCa / JIS X 6319-4 And ISO 18092
RF
ISO 14443 A-2 / ISO 18092
ISO 14443 B-2
ISO 18092
Physical Characteristics
ISO 14443 A-1 / ISO 18092
ISO 14443 B-1
NFC A
NFC B
NFC F
P2P
NFC Forum specification
Vendor specific
International standard

The Tools

Proxmark is the best option

Proxmark RDV4

Read, write, emulate and intercept all RFID protocols

Opensource Community based project

Versatile tool with the correct knowledge

€ 100 - 400

Flipper Zero

Can do much more than just RFID protocols, infrared, sub-GHz, bad usb, …

Handheld device with interface

Versatile tool with the correct knowledge

€ 160 - 200

Chameleon

Can emulate different cards easily

Useful for daily usage due to storage capabilities and compactness

€ 50 - 200

Cheap readers

Limited to one frequency usually

Limited drivers/software

€ 20 - 100

Proxmark

https://github.com/RfidResearchGroup/proxmark3

100% opensource

Jonathan Westhues in 2007

Community Driven and managed by iceman

The real RFID experts are in the following discord:

Flipper Zero

https://flipperzero.one/

Portable Multi-Tool

Has wide arrange of features

ACR122U

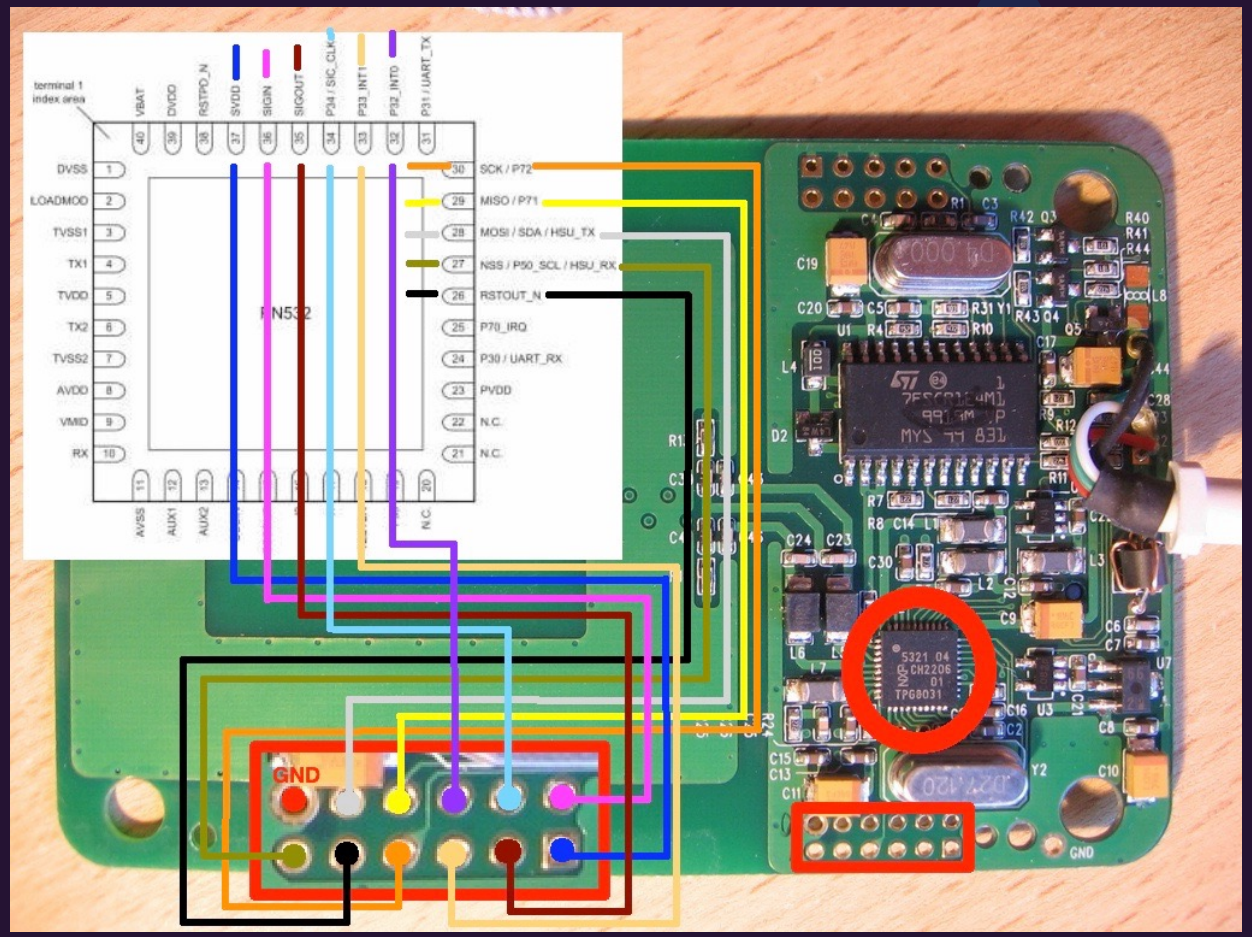
https://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/

NFC Compliant

Not expensive 30-50 EUR

Good library support

Uses NXP PN532 chip



CRID

https://github.com/VinnieV/crid

Windows/Linux/MacOS tool

Created for the Workshop

Usage/Examples

```
# Read the UID of the card
crid --read_uid

# Identify the card to the user
crid --identify

# Read a block from the card using key type A and key value ffffffffffff
crid --read_block 2 --key_type A --key_value ffffffffffff

# Write "Hello, World!" to block 3 using key type B and key value a1b2c3d4e5f6
crid --write_block 3 --data "48656C6C6F2C20576F726C6421" --key_type B --key_value a1b2c3d4e5f6

# Read sector 1 from the card using key type A and key value ffffffffffff
crid --read_sector 1 --key_type A --key_value ffffffffffff

# Read all 16 sectors from the card using key type B and key value a1b2c3d4e5f6
crid --read_full --key_type B --key_value a1b2c3d4e5f6

# Find keys for a Mifare Classic card (requires mfoc binary)
crid --find_mifare_keys

# Brute force keys for block 18 using the key list file keys.txt
crid --brute_force_keys 18 --key_list mifare_access_keys_top100.lst

# Mute the buzzer on the reader
crid --mute

# Provide the flag option
```

Methodology

What is our approach?

01 **Identify the RFID Protocol**

02 **Obtain access to the data**

03 **Process the data**

04 **Use the data**

Methodology

01 **Identify the RFID Protocol**

02 Obtain access to the data

03 Process the data

04 Use the data

- Information on the card/reader
- Using different readers

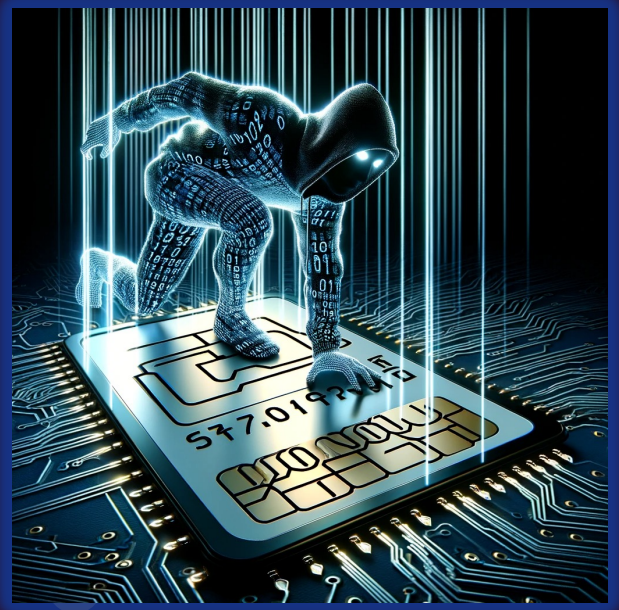
Methodology

01 Identify the RFID Protocol

02 **Obtain access to the data**

03 Process the data

04 Use the data



- Data protection?
- Brute force? Exploitation?

Methodology

01 Identify the RFID Protocol

02 Obtain access to the data

03 **Process the data**

04 Use the data

- Reverse engineer data
- Compare the data
- Different data formats and encodings
- Checksum?

Methodology

01 Identify the RFID Protocol

02 Obtain access to the data

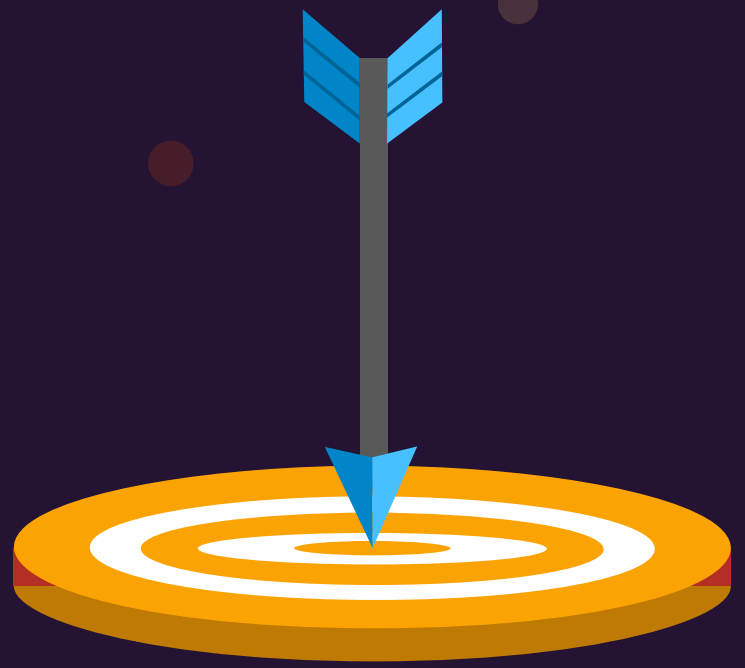
03 Process the data

04 **Use the data**

- Create a copy or emulate
- Modify values on the card (e.g. money)

Device Setup

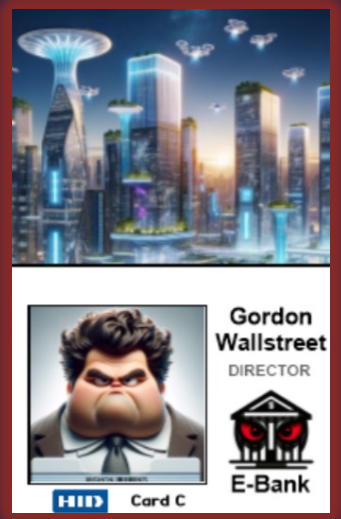
Challenge 0: Identify



Three Cards

Covers almost all RFID systems

T5577 Card C



Low Frequency

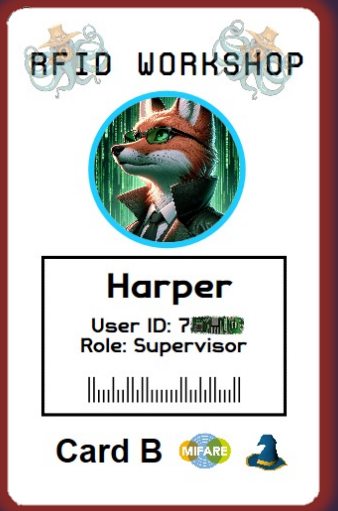
HID

Mifare Classic Card A

High Frequency

Mifare Classic

Mifare Classic Card B



High Frequency

Mifare Classic

HID

Example Methodology

(S)HID?!

01 02 03 04

Identify the RFID Protocol

Obtain access to the data

- Low Frequency (125kHz)
- Readonly
- No access controls

Process the data

- Card only contains a number
- Two possible formats

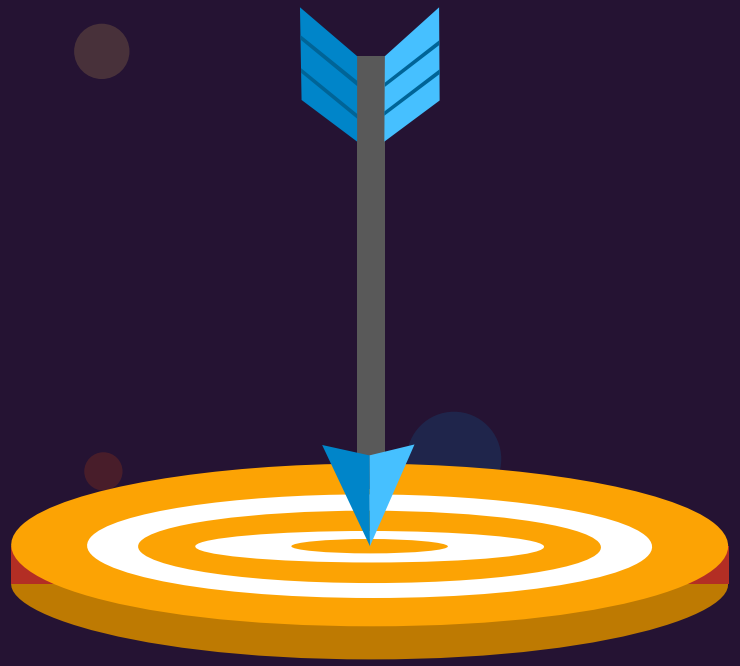
Use the data

Clone card on unlocked HID cards like the T5577

HID

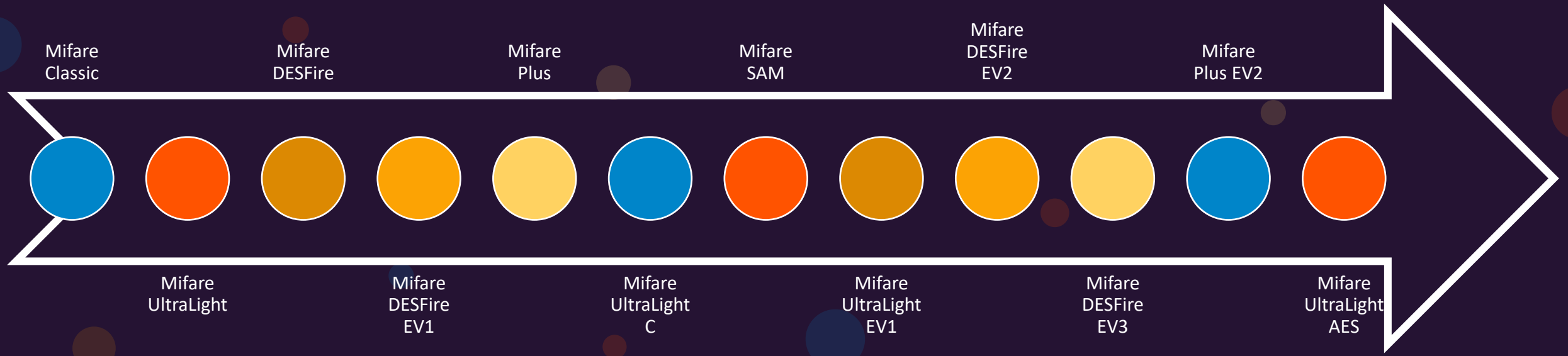
Format	Layout
HID 26-Bit format	<Parity bit> <8-bit site code> <16-bit credential> <Parity bit>
HID 37-Bit format	<Parity bit> <16-bit site code> <19-bit credential> <Parity bit>

Challenge 1: HID Clone



Mifare

Mifare Version History



From 1994 - now

Mifare Classic

Different Tyoes

Mifare Classic Mini

Bruto Data Storage	320 Bytes
Sectors	5
Netto Data Storage	224 bytes

Mifare Classic 1K S50

Bruto Data Storage	1024 Bytes
Sectors	16
Netto Data Storage	752 bytes

Mifare Classic 4K S70

Bruto Data Storage	4096 Bytes
Sectors	40
Netto Data Storage	3440 bytes

Mifare Internals

Mifare Memory Layout

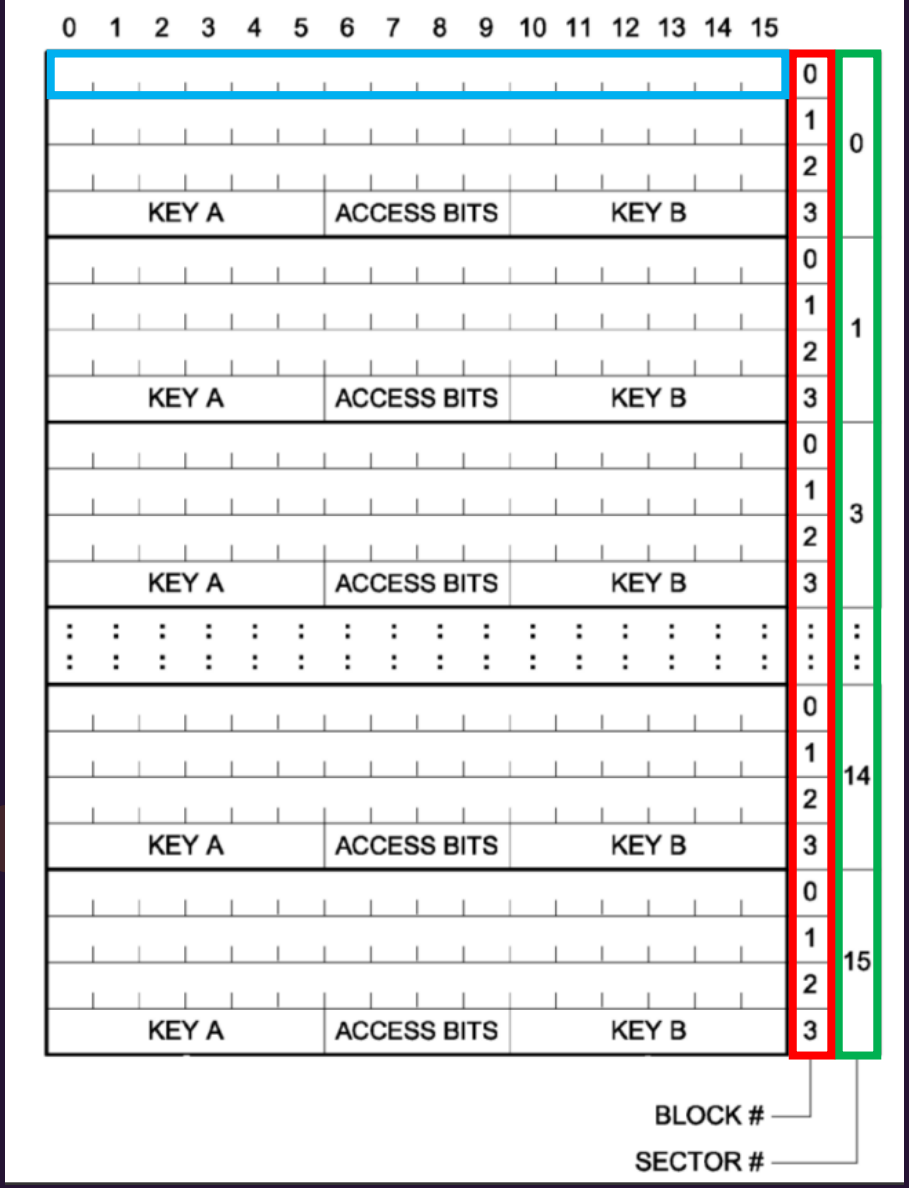
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
																0
																1
																2
KEY A						ACCESS BITS				KEY B						3

x 16

Block Number

Each cell is one byte, so from 0-255
or 0x00-0xFF

Mifare Memory Layout



Mifare Classic 1K

- Sector 0 Block 0 → UID
- 16 sectors
- 4 blocks/sector
- 16 bytes/block

- 16 x 3/4 x 16 = 768/1024 bytes

Access Bits

6 bytes

4 bytes

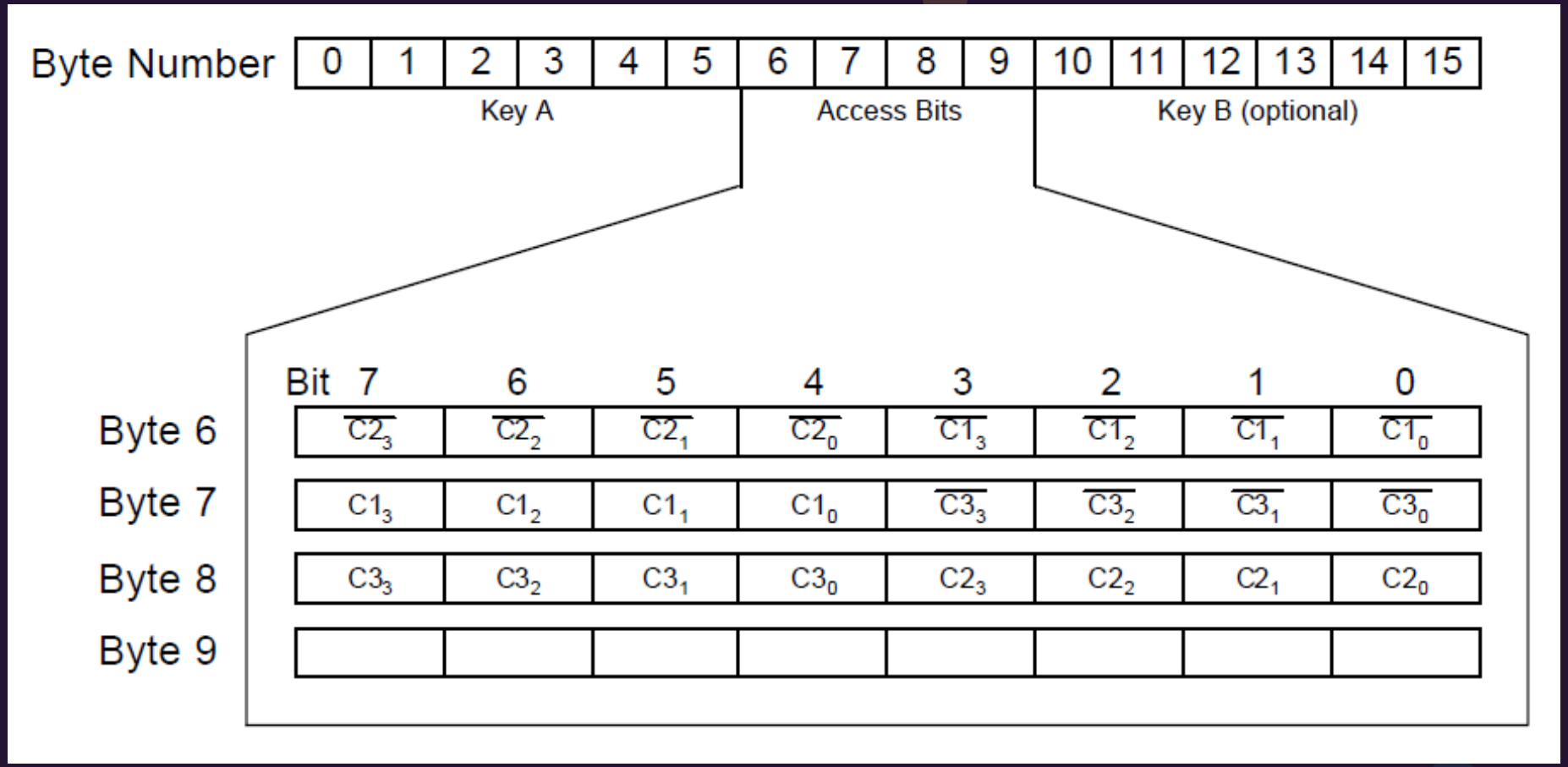
7B 47 88 00

6 bytes

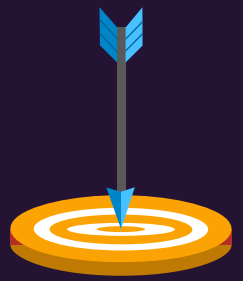
Possible configurations for each block:

$C1_0$	$C2_0$	$C3_0$	read	write	increment	decrement, transfer, restore
0	0	0	key A|B^1	key A|B^1	key A|B^1	key A|B^1
0	1	0	key A|B^1	never	never	never
1	0	0	key A|B^1	key B^1	never	never
1	1	0	key A|B^1	key B^1	key B^1	key A|B1
0	0	1	key A|B^1	never	never	key A|B^1
0	1	1	key B^1	key B^1	never	never
1	0	1	key B^1	never	never	never
1	1	1	never	never	never	never

Access Bits

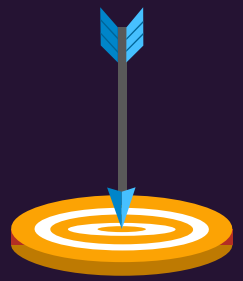

Byte Number
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Key A
Access Bits
Key B (optional)
Bit 7 6 5 4 3 2 1 0
Byte 6
$\overline{C2_3}$ $\overline{C2_2}$ $\overline{C2_1}$ $\overline{C2_0}$ $\overline{C1_3}$ $\overline{C1_2}$ $\overline{C1_1}$ $\overline{C1_0}$
Byte 7
$C1_3$ $C1_2$ $C1_1$ $C1_0$ $\overline{C3_3}$ $\overline{C3_2}$ $\overline{C3_1}$ $\overline{C3_0}$
Byte 8
$C3_3$ $C3_2$ $C3_1$ $C3_0$ $C2_3$ $C2_2$ $C2_1$ $C2_0$
Byte 9

Challenge 2: Basic Flag



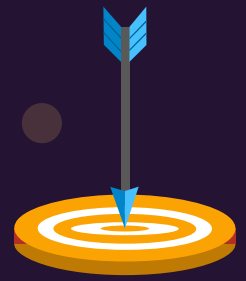
25 points

Challenge 3: Username



75 points

Challenge 4: Access Bits

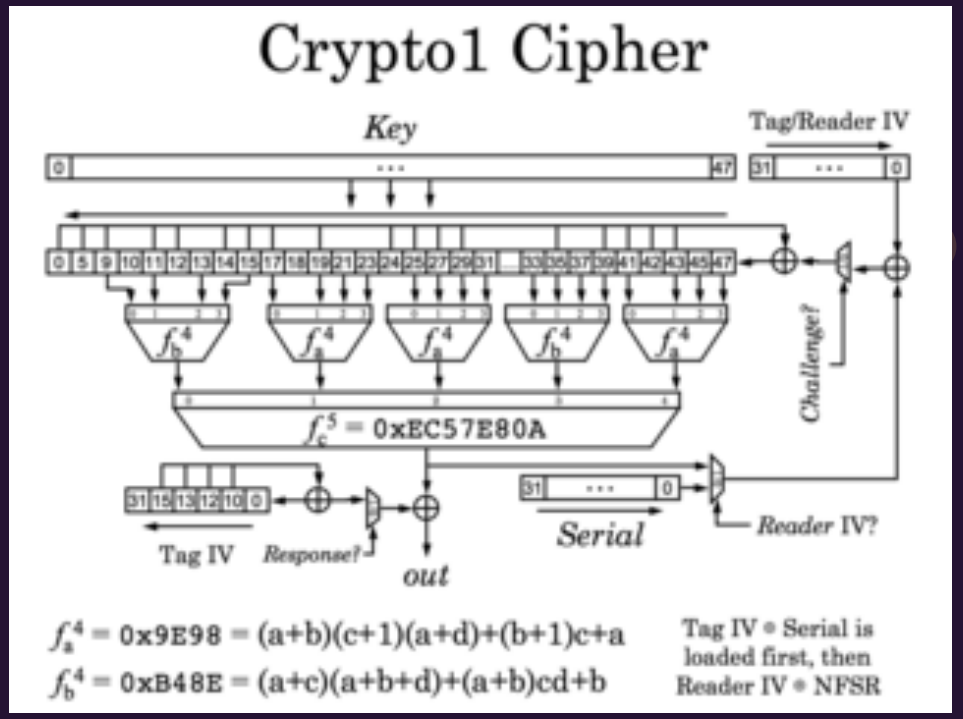


75 points

Key Recovery

Unknown key A && || key B?

Crypto1 Cipher



Security through Obscurity

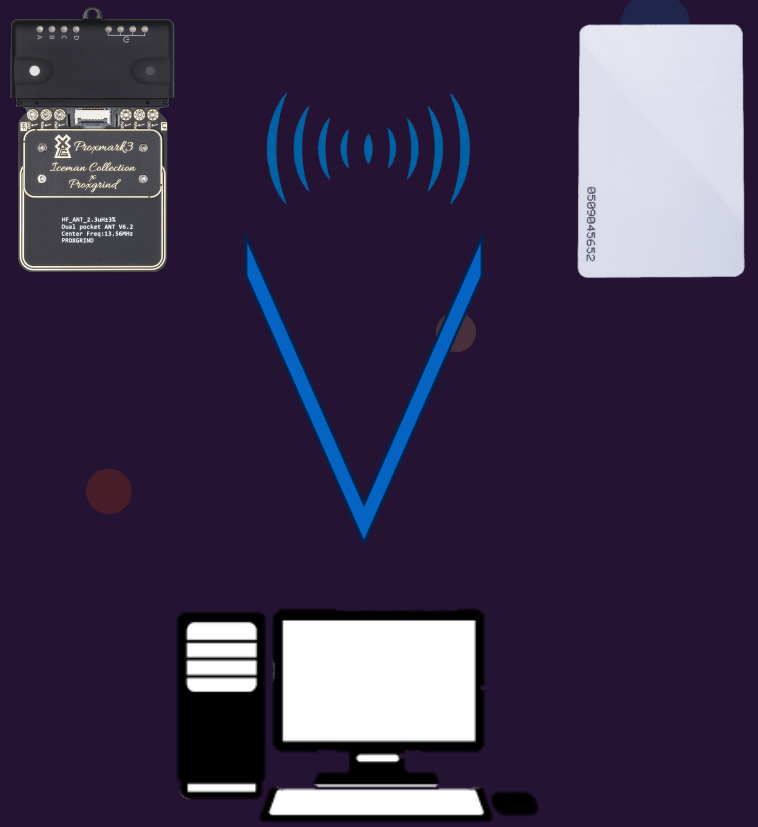
Brute Force (48 bit key)

Bad Random Number Generator

YouTube 25c3: Analyzing RFID Security →

Brute Force?

Slow



Fast

Vulnerabilities

(Hard)Nested Attack

- Requires a valid key
- Exploits the linear feedback shift register (LFSR)
- Timing based attack
- Collects many nonces/challenges to obtain the keys through some cryptographic magic

DarkSide Attack

- During authentication the tag responds with some encrypted data
- The encrypted data is a static error code
- Can be used to recover the key

Vulnerabilities

- Proxmark:
  - Nested: hf mf nested
  - Hardnested: hf mf hardnested
  - Staticnested: hf mf staticnested
  - Darkside: hf mf darkside

OR

hf mf autopwn

- Flipper:
  - Nested: Mfkey32 application
  - Hardnested: FlipperNested firmware?
  - Staticnested: FlipperNested firmware?
  - Darkside: Not implemented

- ACR122U
  - Nested: mfoc tool
  - Hardnested: miLazyCracker
  - Darkside: mfcuk

Challenge 5: Key Recovery

100 points

Challenge 6: Nested

125 points

Challenge 7: Hardnested

150 points

Challenge 8: Darkside

150 points

Magic Mifare 🧙

- UID should be readonly
- Modified cards that have special commands to overwrite the UID
- Currently different implementation because it can be detected by readers as well
- There are cards with a 4-byte or a 7-byte UID

Challenge 9:

Vault

150 points

Challenge 10:

Employee Card

150 points

Challenge 11:

Vending Machine

250 points

Challenge 12:

Hotel Rooms

350 points

Challenge 13:

Clone Speedrun Challenge