RFID Hacking

Who's this?

jof

- jof
- Telecoms and networking geek
- Infrastructure and Security

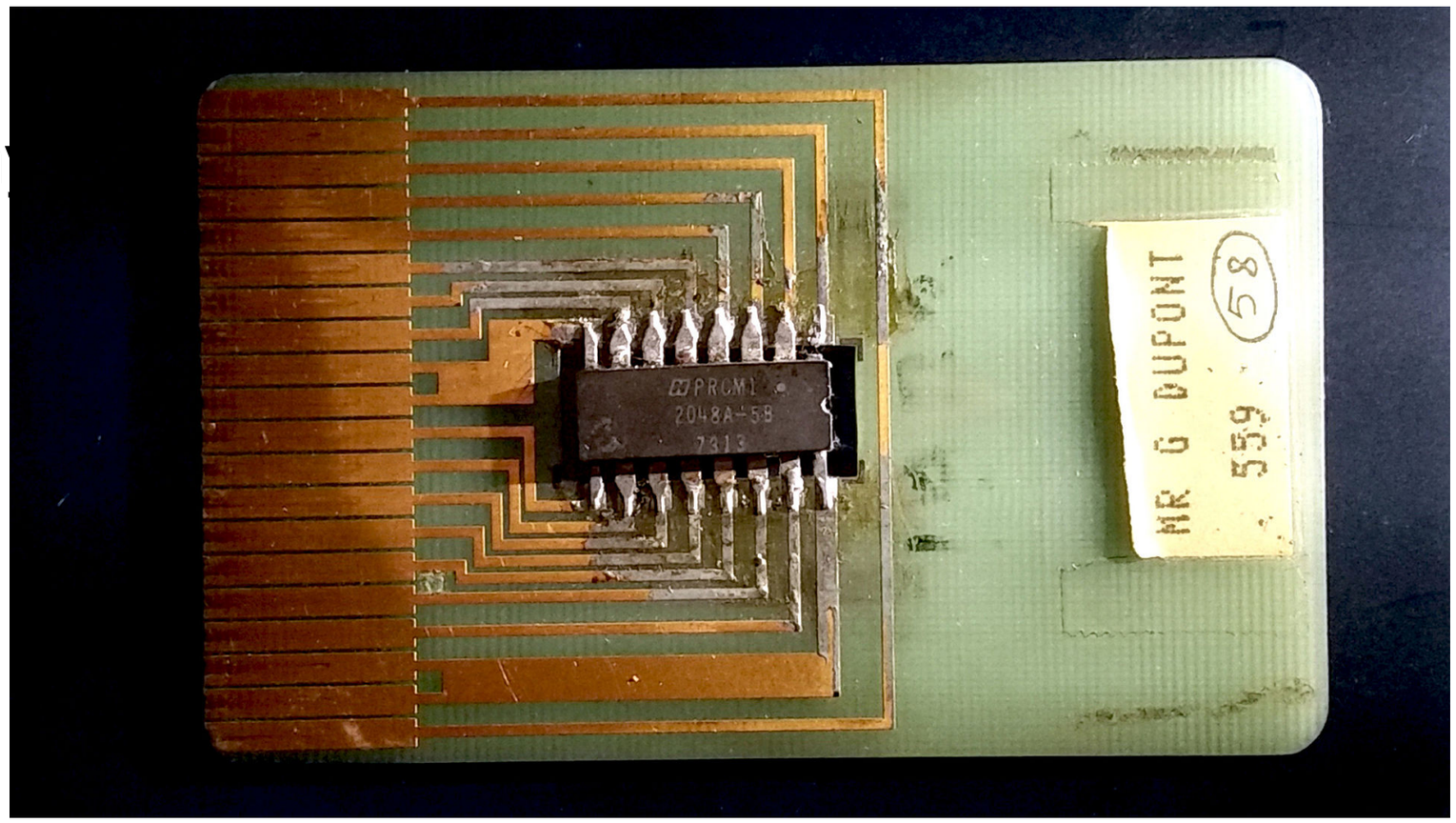
- First time workshop
- RFID-curious for ~3 years
- Good tools and visibility made all the difference
- Today's motivation:
  - Give more people information and tools to follow their own curiosity

RFID

- Contactless cards using RF energy to communicate
- With some identifying number
- Radio Frequency, IDentification

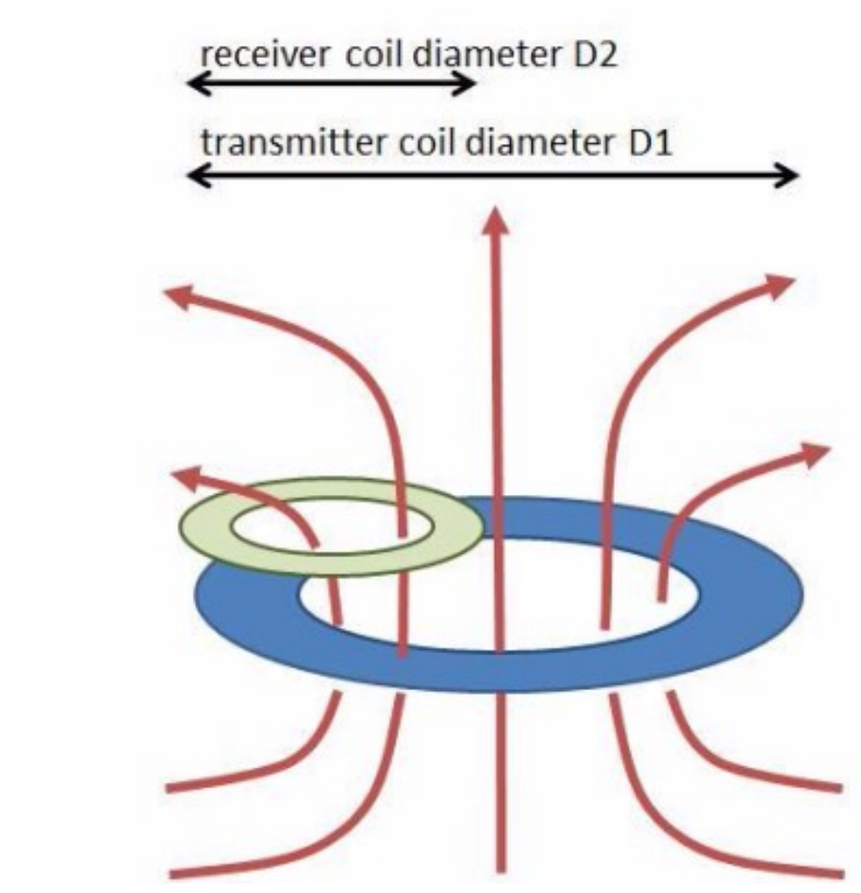
History

- IC Cards
- A bit of memory that can move around
- Microprocessors
- (Weak/DIY) Cryptography
- Better Cryptography



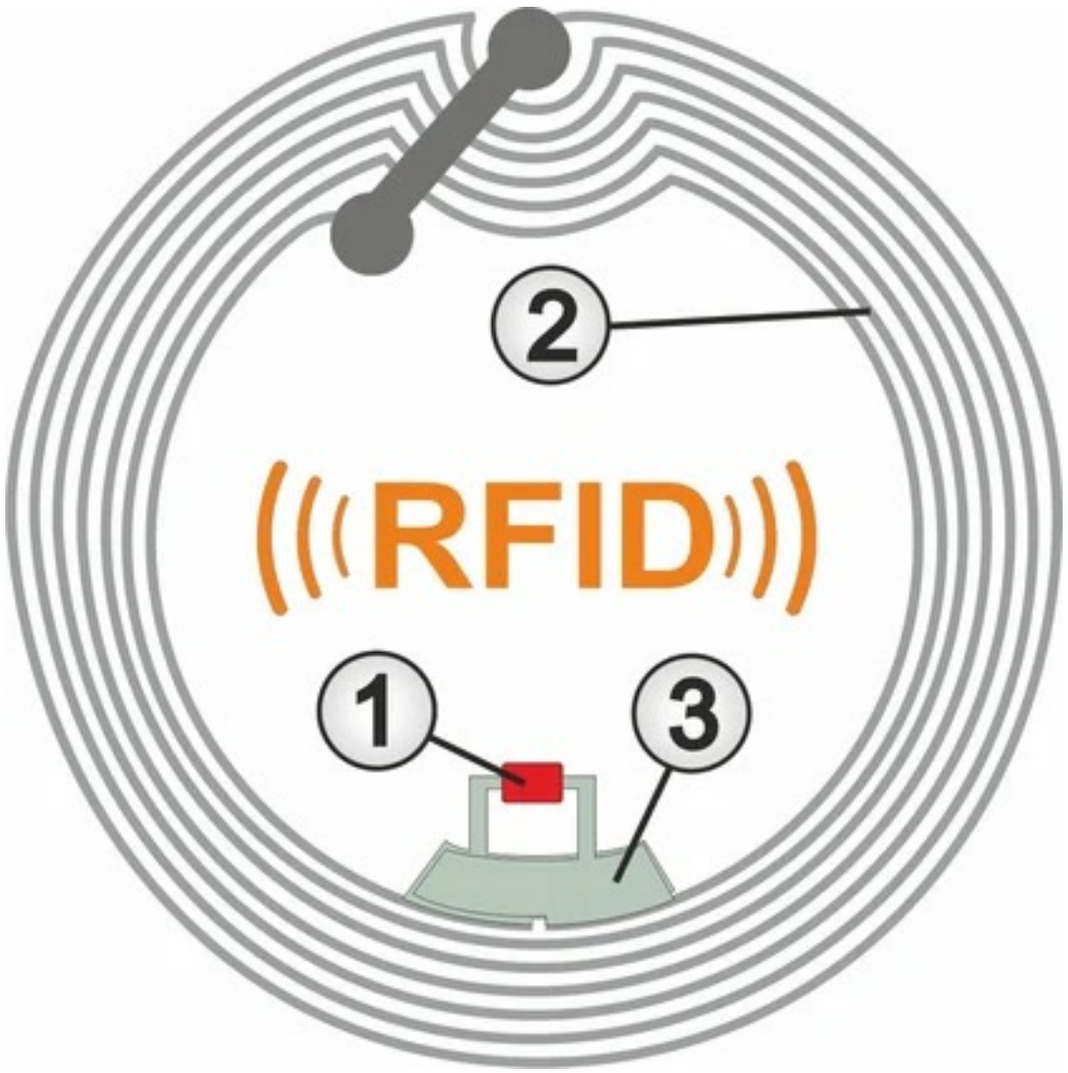
Physics

- LC systems: a capacitor and a coil/inductor



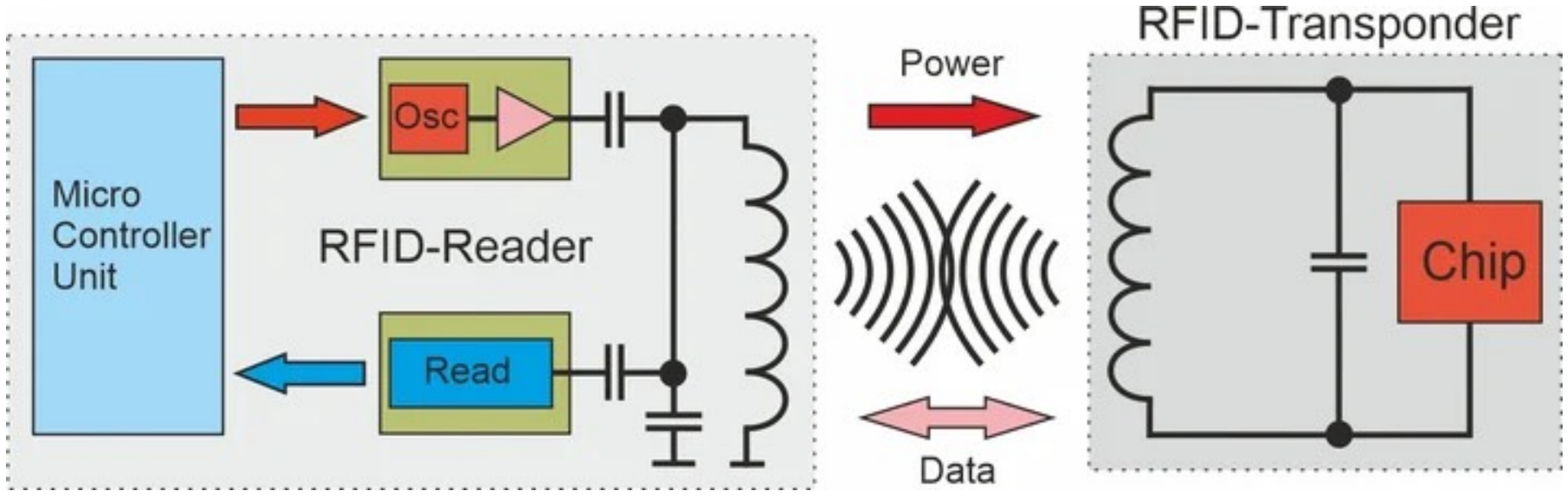
- Dipole RF Antenna

Physics



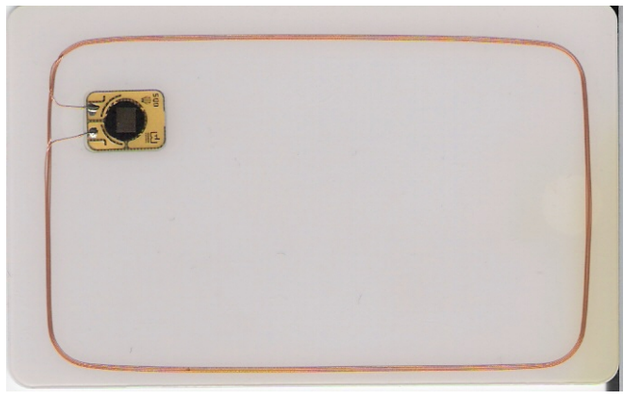
Block Diagram

- A “reader” (standards: “PCD”)
  - powers a coil with an alternating carrier wave, sometimes modulated with an signal with amplitude shifts
  - detects and demodulates signals from tags
- A “tag” (standards: “PICC”)
  - couples to the reader and modulates its field-power consumption
  - alternating between absorption and reflection with a signal ("backscatter")

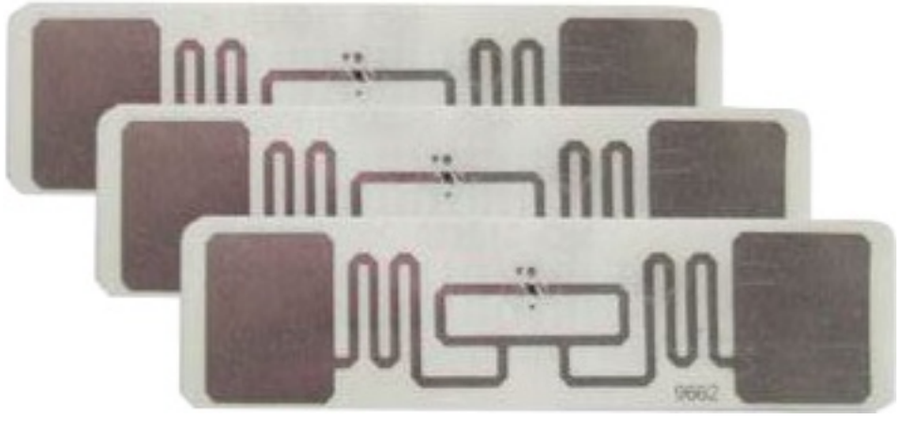


Interface Types

- Low Frequency (LF) -- 125-135 kHz
  - LC coil; long and thin wire coil



- High Frequency (HF) -- 13.56 MHz
  - LC coil; less-long, thin wire coil
- Ultra High Frequency (UHF) -- 850-950 MHz
  - Dipole Antenna

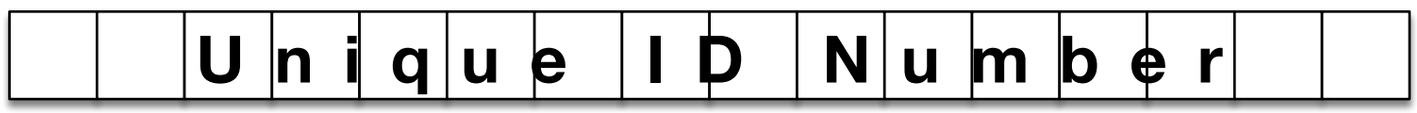


NFC

- More applications, more smartphones -- more standards
- Less distinction between reader and tag; multiple modes: classic reader/tag, NFC card emulation, peer-to-peer communications
- Industry standards group
- NDEF records and placement

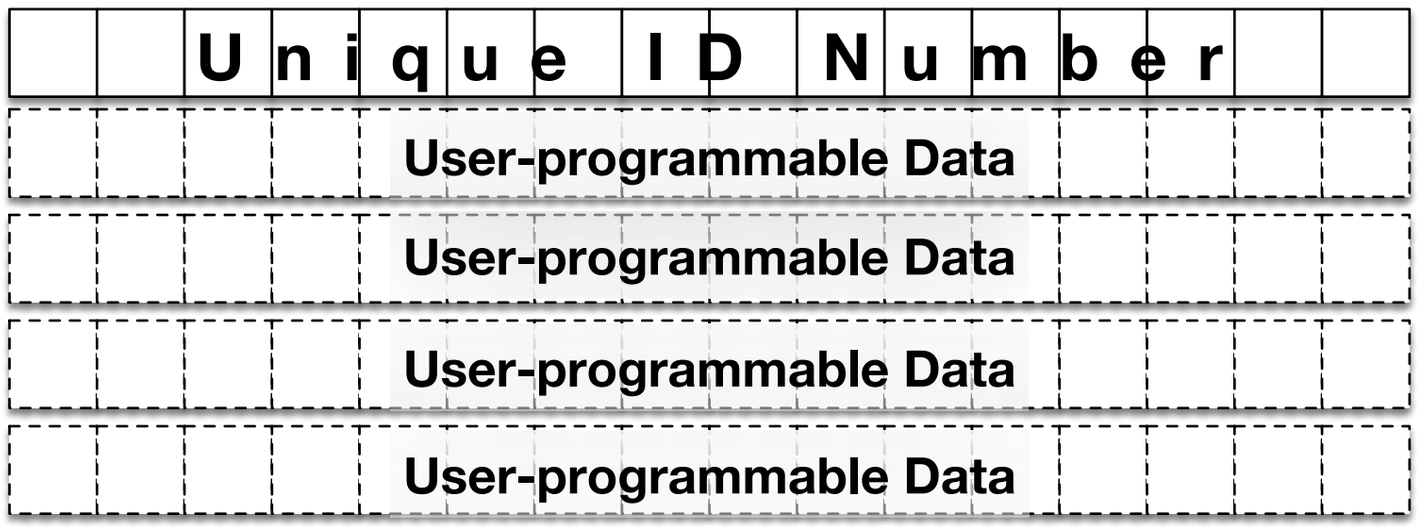
Memory Tags

- Small flash, an RF frontend, and a microcontroller
- Generally, each with a unique ID set at the factory
- Optionally, some user-programmable data



Memory Tags

- Small flash, an RF frontend, and a microcontroller
- Generally, each with a unique ID set at the factory
- Optionally, some user-programmable data
- Usually based on ISO 14443A ("type A")



"Unique" IDs

- Same number, every time (usually)
- What the vendor datasheet *says*
  - "the block is programmed and write-protected in production testing”
- What the developers hear
  - “This unique ID can be used to check for the presence of a known, previously-registered tag”
- What the world does
  - Third-party IC designers have created “magic” cards, whose UID can be changed arbitrarily
  - Uses additional command opcodes not used in the original protocol

"Good Enough" Implementations

- User's can't critique what they can't see
- Locks keep honest people honest
- Inexpensive and fast-to-implement UID-only systems are pervasive
- Weak cryptography prevents casual abuse
    - The door is shut, but it is not locked either

Reader Interactions

- Simplest: read just the “unique” ID and compare against a database
- More complex: reader authenticates to card, card authenticates to reader; protected memory contains a certificate or token
- Keys
  - Simple: same key for every tag
  - More-complex: diversified keys, based on the tag UID

Mifare Classic

- Cheap and widely deployed
- 1kB or 4kB of memory
  - 1k: Laid out into 16 Sectors of 4 Blocks, each 16 bytes long
    - 64 blocks * 16 bytes == 1024 Bytes
    - 3 blocks of user-writable data per-sector
      - except Block 0 (contains UID and card data)

Key A Access Bits Key B

Sector	Block	Data	Access	A | Acc. | B r w | r w | r w [info] r | w | i | d/t/r
0	0	1a1f60ed880804000293fd89ed20901d	000	-
	1	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	2	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	3	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]
1	4	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	5	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	6	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	7	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]
2	8	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	9	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	10	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	11	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]
3	12	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	13	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	14	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	15	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]
4	16	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	17	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	18	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	19	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]
5	20	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	21	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	22	00000000000000000000000000000000	000	A/B | A/B | A/B | A/B [transport]
	23	ffffffffffffff078069ffffffffffff	001	- A | A A | A A [transport]

Tools

- Card emulation
  - Magic Cards
  - Chameleon (Mini, Ultra)
- Proxmark3
- Low-level reader control (PN532 boards)

Proxmark3

- SDR is to Radio what Proxmark3 is to LF/HF RFID
- Software-defined interfaces with an FPGA enables supporting a huge variety of types
- Swappable antennas

PN532 Boards

- An NXP chip, mounted into a reference design board
- Integral PCB Antenna
- Supports multiple interfaces: UART, SPI, I2C

UARTs

- Awesome for connecting separate systems together: asynchronous, serial/ordered, flexible, scalable, no separate clock

- Transfers some atomic unit of bits in one direction: these days, 8 bits/1 byte at a time

- Held high, pulled low

- Ubiquitous in embedded electronics

- USB Adapters are useful for exploring new systems; frequently used for a text console to embedded OSes

Kits For Today

- PN532 Kit (Qty. ~40)
  - PN532 board, USB-Serial adapter, hookup wires, a Magic 4-byte Mifare Classic card
  - €5-€10
- Proxmark3 Easy (Qty. 2)
  - Proxmark3 Easy with modern firmware, USB cable
  - €30-€50

Today's Reader

- Diversified keys based on UID
- Default access conditions
- A key can read B key

Today's Reader

- Key Algorithm uses SHA256 for diffusion, XOR for a secret
  - Sector Pre-key: SHA256(UID + Sector Number + ("a"|"b"))[0 .. 5]
  - Sector Key: XOR(Secret, Sector Pre-key)

Today's Reader

- Example:
    - UID: 1A 1F 60 ED, Sector: 1, Key: A
      Example secret: 112233445566
    - Sector Pre-key:
        - SHA256(1A1F60ED0161)
        - b8d1d830f22a7d5e7f144fa49027baec816841c621fe79974d4ec1c8e1f2fe70
        - b8d1d830f22a
        - XOR(112233445566, b8d1d830f22a)
        - Sector 1 Key A: a9f3eb74a74c
- Cyberchef Link

Software

- VM Images
  - Based on Kali Linux. Hopefully with all software you'll want today.
    - https://github.com/nfc-tools/mfoc-hardnested
      - User/Password: kali/kali
- Compile-it-yourself
  - Based on libnfc; configure it for our reader:
    - Edit **/etc/nfc/libnfc.conf** and add:
      - device.connstring = "pn532_uart:/dev/ttyUSB0"
- Proxmark3: https://github.com/RfidResearchGroup/proxmark3/

Difficulty Levels

- Easy mode: one sector secured, stores a flag value and a difficulty level
    - Faster time-to-satisfaction (PN532: 5-30 mins, Proxmark3: 1 minute)
- Hard mode: sectors 1-15 secured, Sector 0 with default keys
    - Takes longer (PN532: best case ~1 hour, Proxmark3: ~3-5 minutes)

How To Play

- Come up to me to get a blue tag programmed
  - Easy Mode or Hard Mode
- Solder, assemble, and connect your reader
- Place the blank card on the reader and dump it with the default keys:
  - mfoc-hardnested -P 250 -T 50 -O blank.mfd -F
- Place the blue tag on the reader and run
  - mfoc-hardnested -P 250 -T 50 -O camp.mfd -F
- Success! Once you have both dump files, you can clone the card contents.

How To Play

- Examine your fresh dump file:
  - python3 ~/src/mfdread/mfdread.py ~/camp.mfd

How To Play

- Write the blue tag contents onto the blank card, using the blank card keys
    - This process changes the keys
    - nfc-mfclassic w a u camp.mfd blank.mfd
- http://151.216.195.155:8000/