HWallet
The simplest Bitcoin hardware wallet

FOSDEM '19

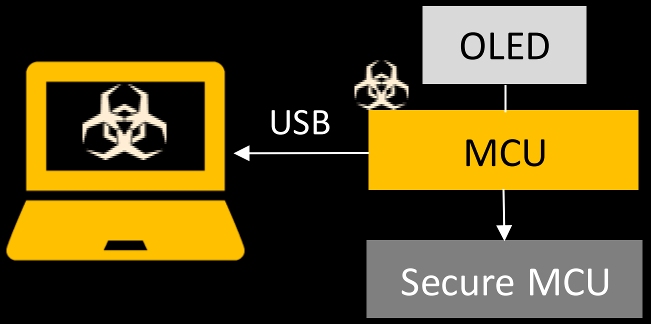
Nemanja Nikodijević
<nemanja@hacke.rs>

Vulnerabilities in hardware wallets

Ledger

https://saleemrashid.com/2018/03/20/breaking-ledger-security-model/

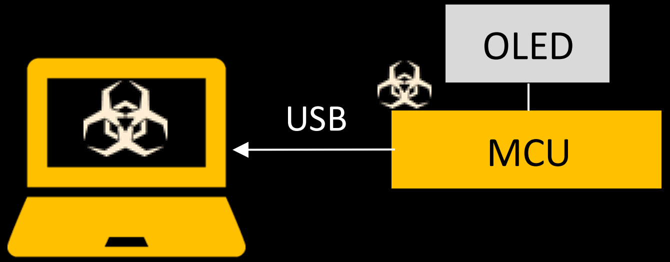
While the software on the SE can be attested to, the MCU is a non-secure chip and its firmware can be replaced by an attacker



TREZOR

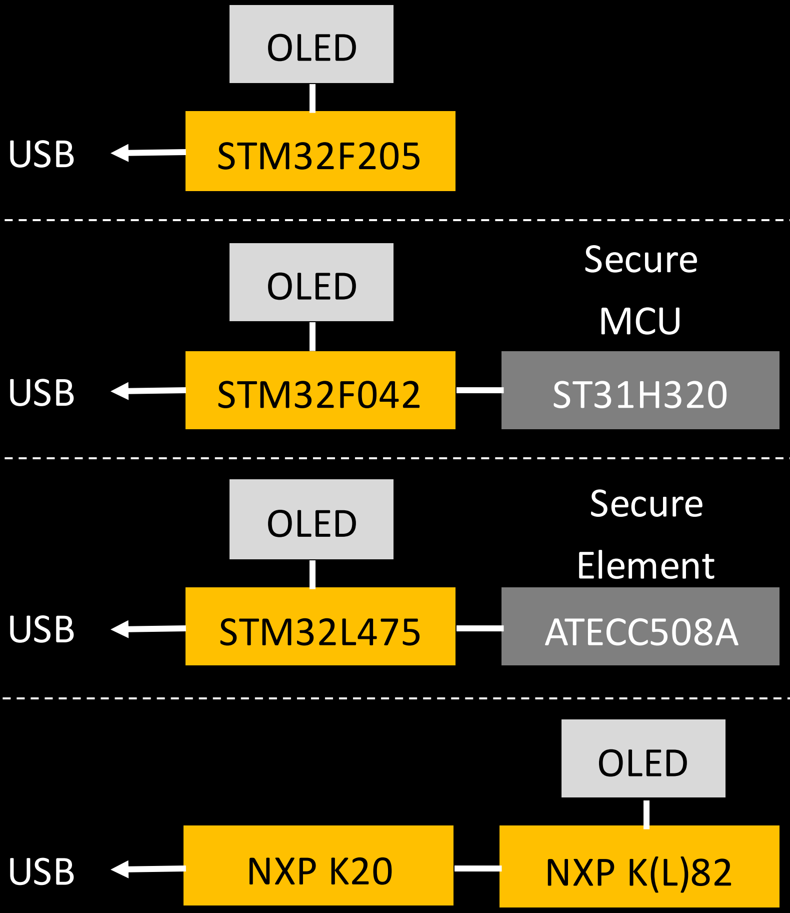
https://blog.trezor.io/fixing-physical-memory-access-issue-in-trezor-2b9b46bb4522

...an attacker with physical access to a TREZOR device could have created a custom firmware which extracts the seed from the RAM of the device.



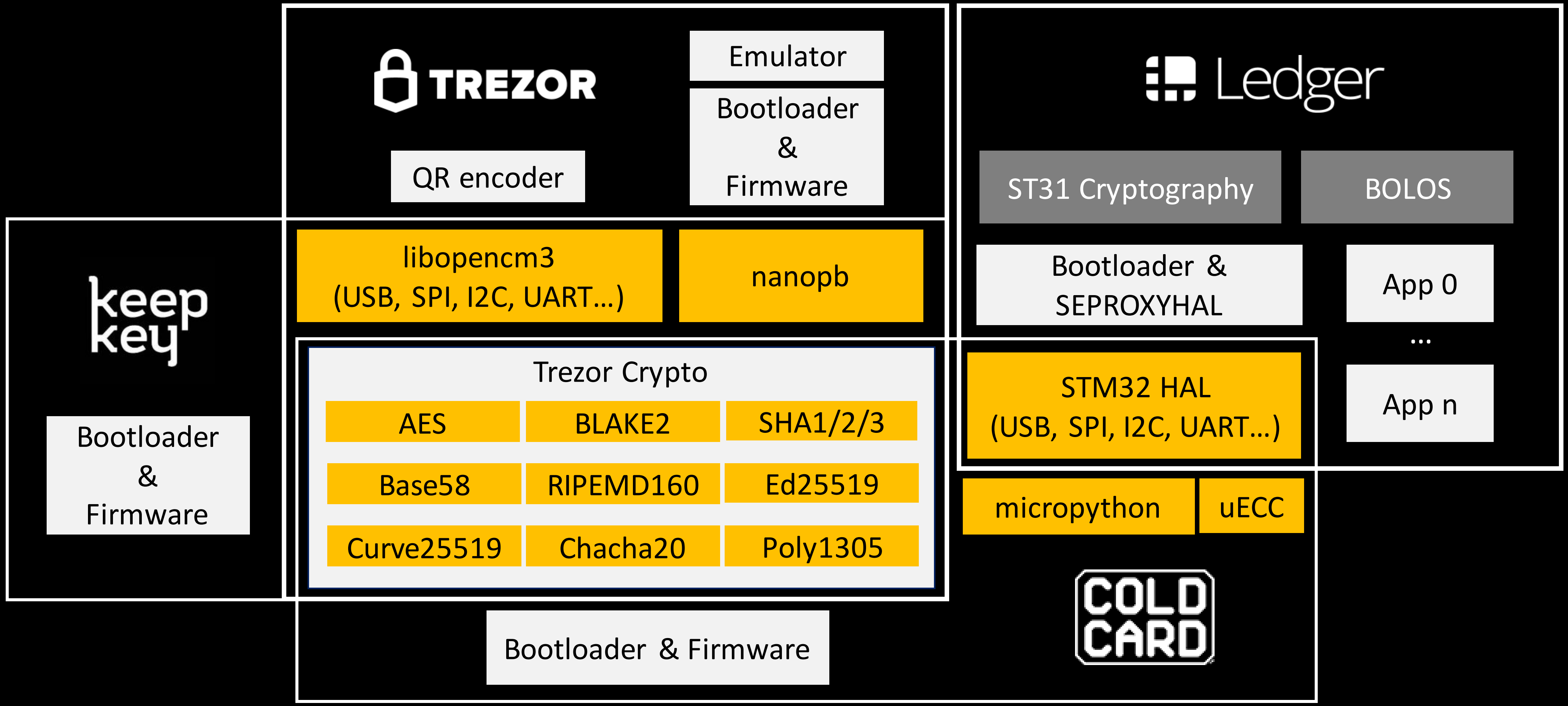
nemanja@hacke.rs

Hardware wallets



	Hardware Acceleration			Open
	TRNG	SHA256	secp256k1	Source
TREZOR keepkey	X	X	X	✓
Ledger	✓	?	✓	X
COLDCARD	X	✓	X	✓
HWallet	✓	✓	✓	✓

Library dependencies



TREZOR

Emulator

Bootloader & Firmware

QR encoder

libopencm3 (USB, SPI, I2C, UART…)

nanopb

Trezor Crypto

AES

BLAKE2

SHA1/2/3

Base58

RIPEMD160

Ed25519

Curve25519

Chacha20

Poly1305

keep key

Bootloader & Firmware

Ledger

ST31 Cryptography

BOLOS

Bootloader & SEPROXYHAL

App 0

…

App n

STM32 HAL (USB, SPI, I2C, UART…)

micropython

uECC

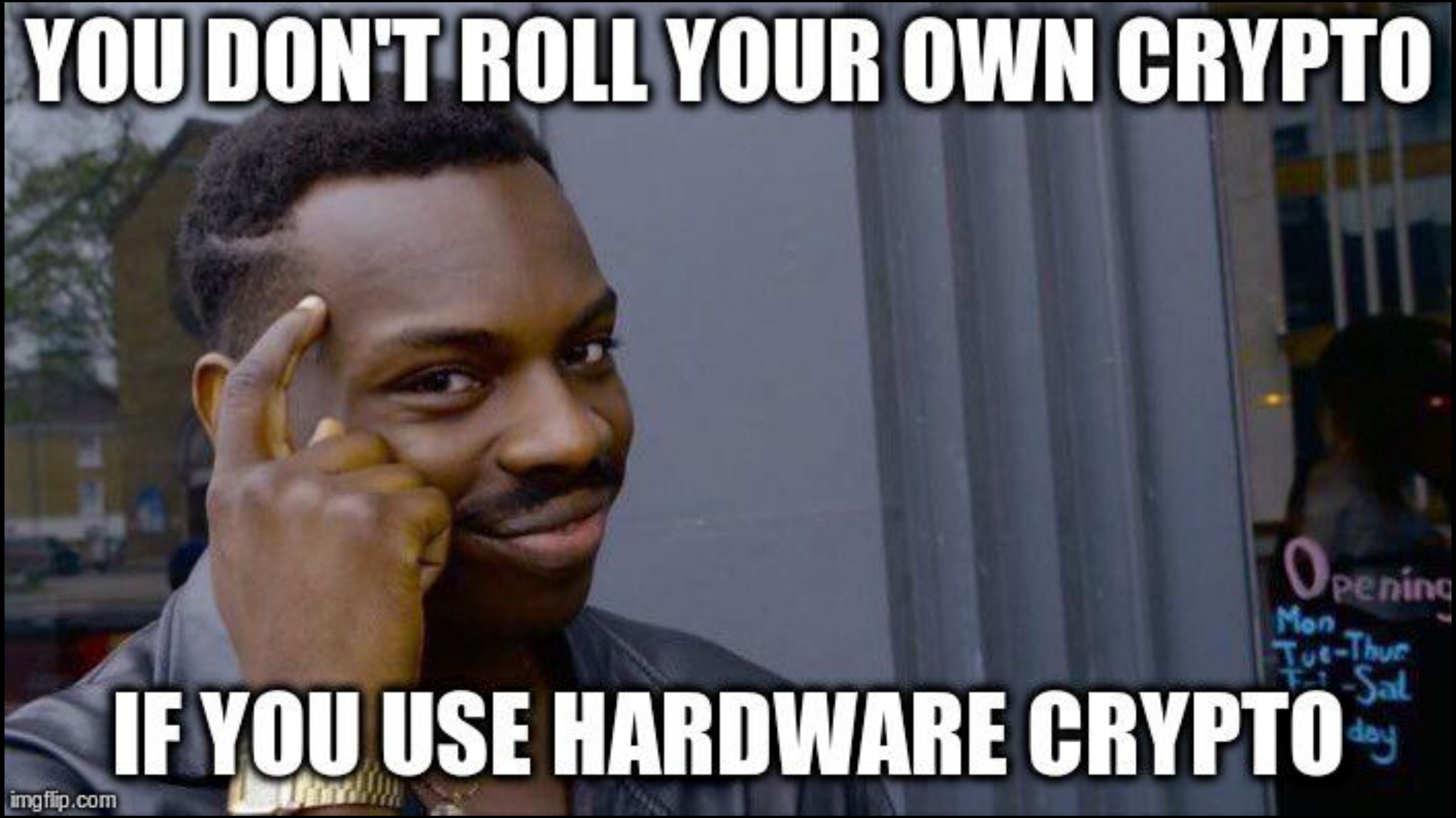
COLD CARD

Bootloader & Firmware

● third party libs ○ open source ● closed source

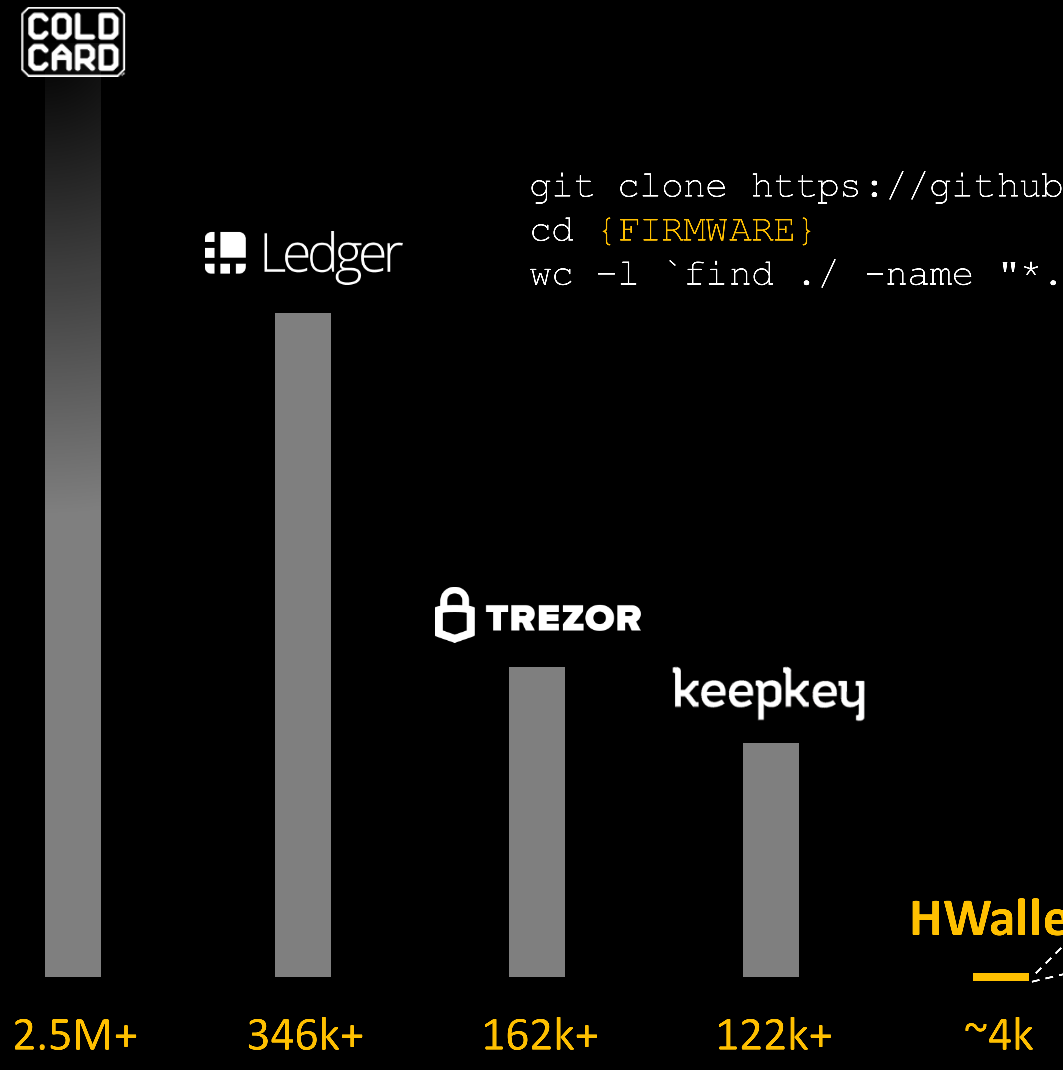
nemanja@hacke.rs

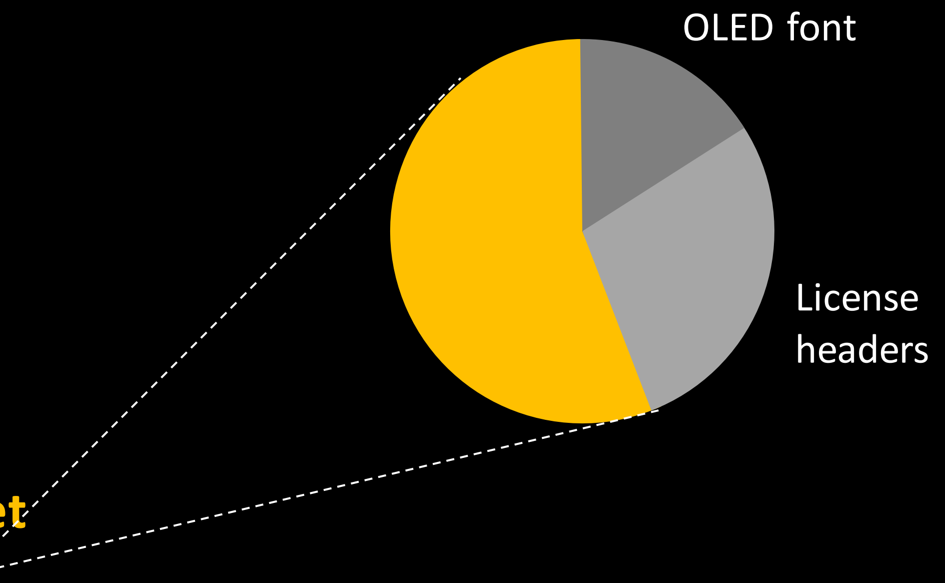
Don't roll your own crypto!



Code size comparison

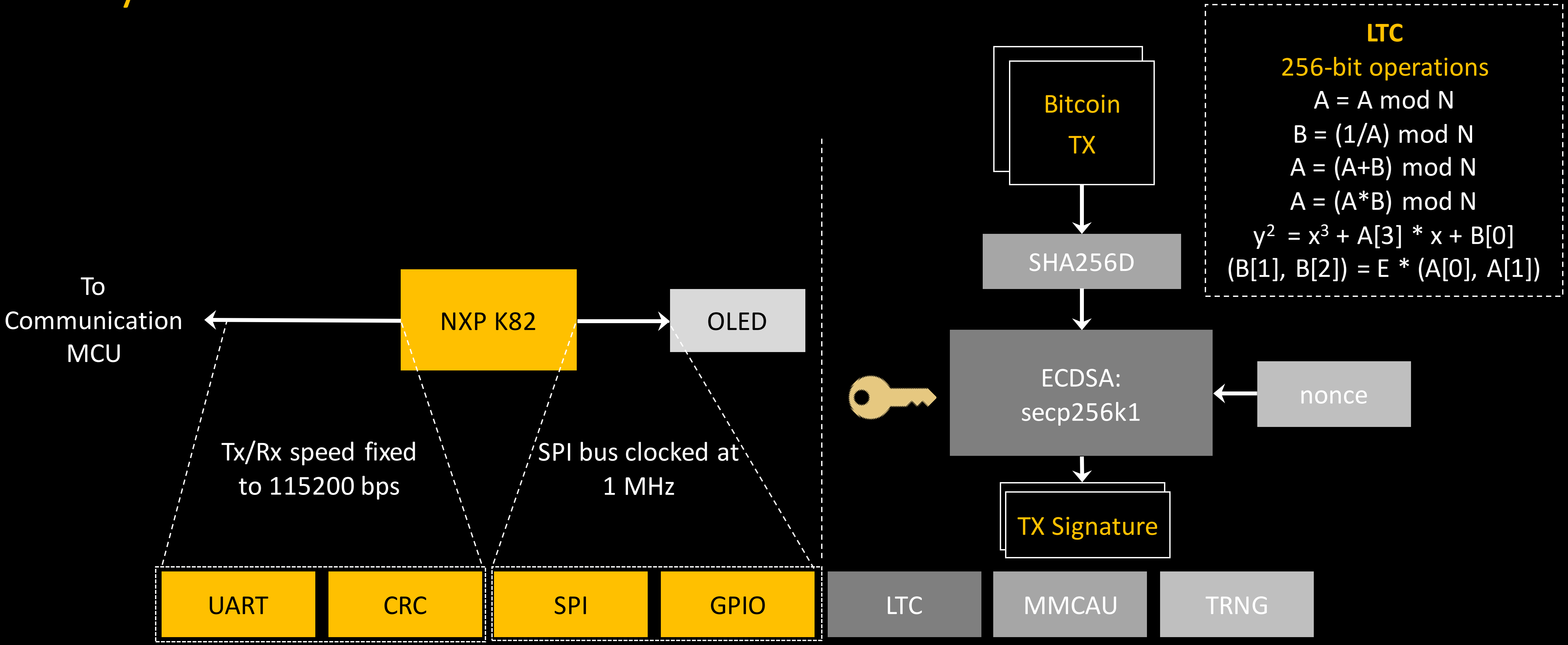
```
git clone https://github.com/{PRODUCT}/{FIRMWARE} --recurse-submodules
cd {FIRMWARE}
wc -l `find ./ -name "*.c" -o -name "*.h"`
```





nemanja@hacke.rs

Code layers



To Communication MCU

NXP K82

OLED

Tx/Rx speed fixed to 115200 bps

SPI bus clocked at 1 MHz

UART

CRC

SPI

GPIO

LTC

MMCAU

TRNG

Bitcoin TX

SHA256D

ECDSA: secp256k1

nonce

TX Signature

LTC

256-bit operations

A = A mod N

B = (1/A) mod N

A = (A+B) mod N

A = (A*B) mod N

$y^2 = x^3 + A[3] * x + B[0]$

(B[1], B[2]) = E * (A[0], A[1])

https://gitlab.com/nemanjan/hwallet

nemanja@hacke.rs

Code layers

```
typedef struct {
    uint16_t type;
    uint16_t length;
    uint8_t data[32];
    uint32_t crc;
} Packet;

PACKET_Send();
PACKET_Receive();
```

```
typedef struct {
    SPIx* spi;
    GPIOx* dcGpio;
    GPIOx* rstGpio;
    uint8_t dcPin;
    uint8_t rstPin;
    uint8_t buffer[ ];
} OLED;

OLED_WriteRow();
OLED_Clear();
```

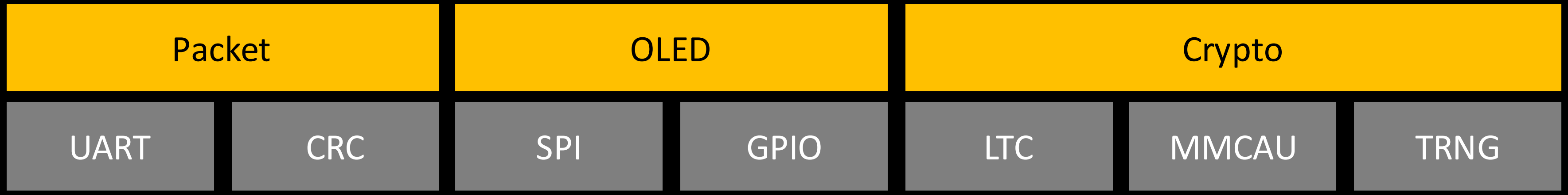
```
CRYPTO_Random();
CRYPTO_SHA256();
CRYPTO_ECDSA_Sign();
CRYPTO_ECDSA_GetPublicKey();
typedef struct {
    uint8_t num[32];
    uint8_t len;
} Bignum;
CRYPTO_Bignum_Init();
CRYPTO_Bignum_Mod();
CRYPTO_Bignum_Div();
CRYPTO_Bignum_Sub();
CRYPTO_Bignum_IsNull();
```

B' = (1/B) mod N

A' = A – A mod B

(A/B) mod N = (A'B') mod N

N - a large prime, larger than any A or B, e.g. **p** from secp256k1



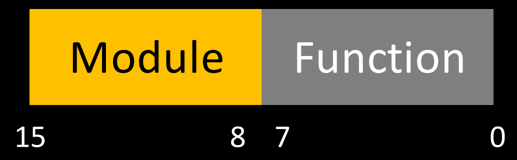
https://gitlab.com/nemanjan/hwallet

nemanja@hacke.rs

Code layers

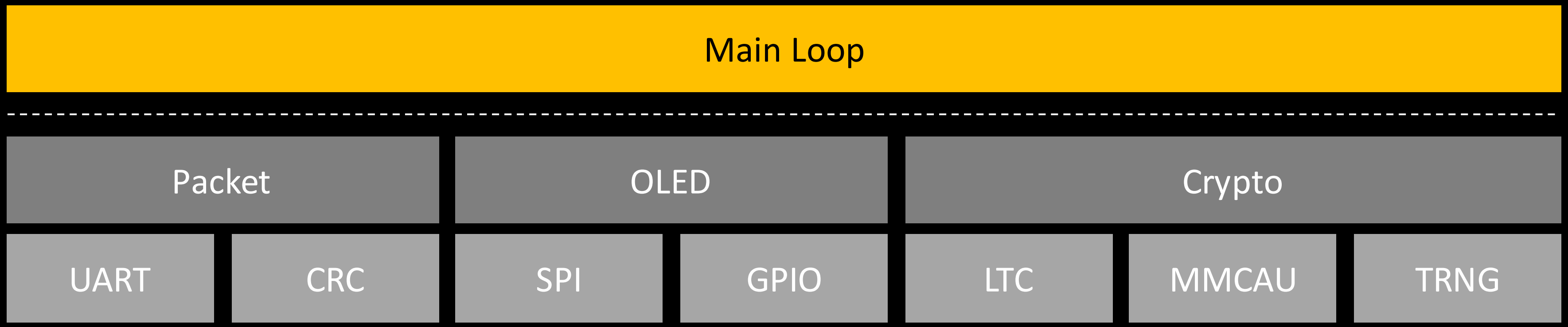
```
while(1) {
    Packet msg;
    PACKET_Receive(&msg);
    switch(PACKET_MODULE(msg.type)) {
        case PACKET_BITCOIN:
            Bitcoin_Process(&msg);
        ...
    };
}
```

Packet type



Module | Function

15 8 7 0



Main Loop

Packet | OLED | Crypto

UART | CRC | SPI | GPIO | LTC | MMCAU | TRNG

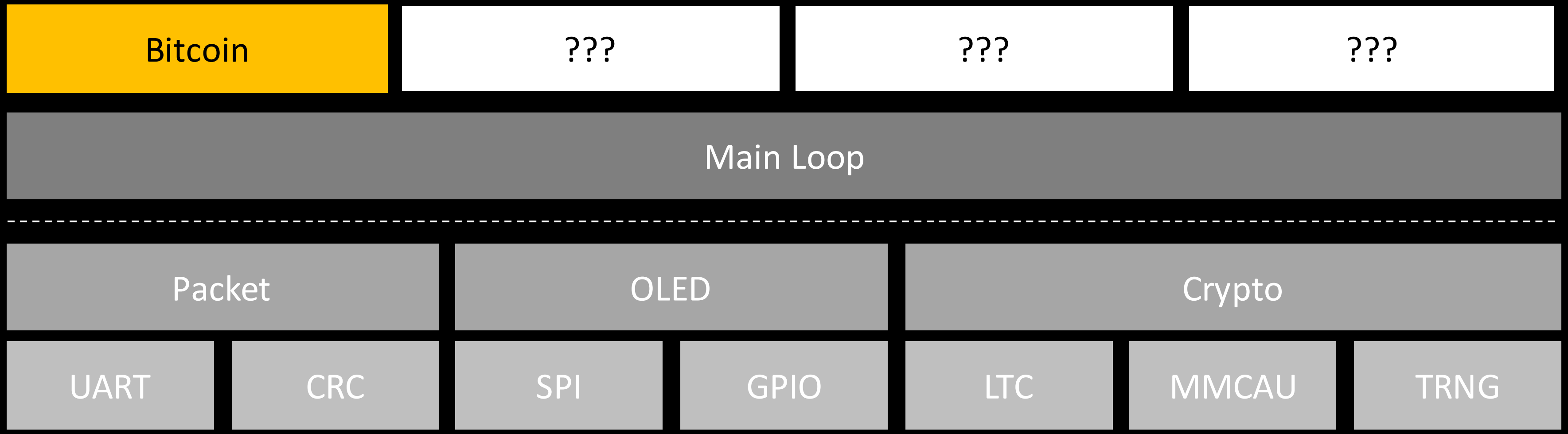
https://gitlab.com/nemanjan/hwallet

nemanja@hacke.rs

Code layers

```
void Bitcoin_Process(Packet* msg) {
    switch(PACKET_FUNC(msg->type)) {
        case BITCOIN_FUNC_INIT_TX:
            Bitcoin_Tx_Init();
        ...
    };
}
```

.00100000 BTC
to 15BoW83tLDb6Znx
qWbzRkERebWK8JJAr6g



Bitcoin | ??? | ??? | ???

Main Loop

Packet | OLED | Crypto

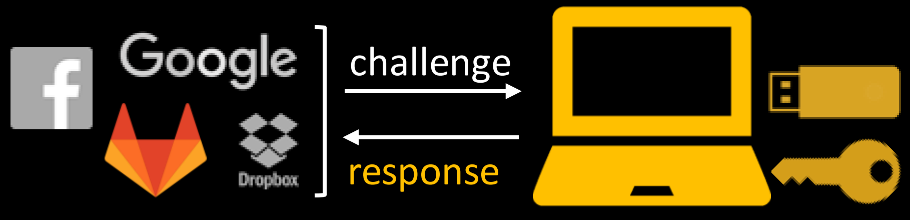
UART | CRC | SPI | GPIO | LTC | MMCAU | TRNG

https://gitlab.com/nemanjan/hwallet

nemanja@hacke.rs

What's next?

FIDO U2F

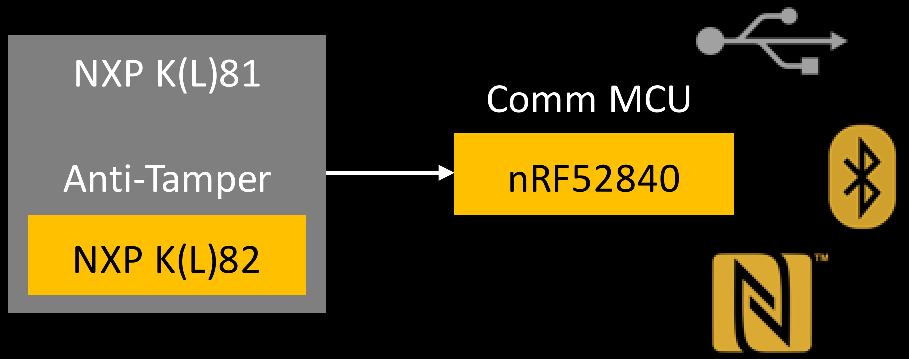


challenge

response

WebAuthn

CTAP



NXP K(L)81

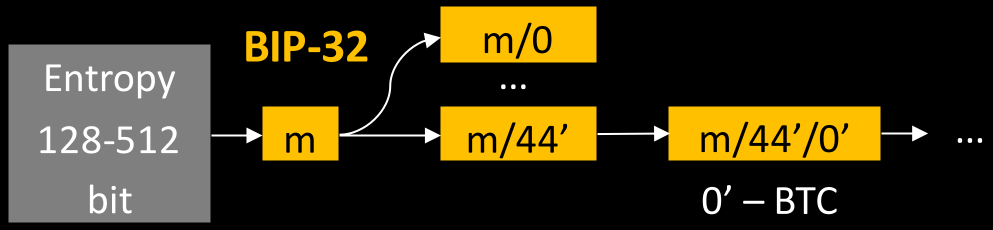
Anti-Tamper

NXP K(L)82

Comm MCU

nRF52840

Recovery seed



Entropy 128-512 bit

BIP-32

m

m/0

…

m/44’

m/44’/0’

…

0’ – BTC

60’ – ETH

144’ – XRP

BIP-44

… witch collapse practice feed shame …

BIP-39

More cryptocurrencies

nemanja@hacke.rs

Questions?