HWallet
The simple cryptocurrency hardware wallet
What is a hardware wallet?

A **hardware wallet** is a special type of bitcoin wallet which stores the user’s private keys in a secure hardware device. They have major advantages over standard software wallets:

- private keys are often stored in a protected area of a microcontroller, and cannot be transferred out of the device in plaintext
- immune to computer viruses that steal from software wallets
- can be used securely and interactively, private keys never need to touch potentially vulnerable software
- much of the time, the software is open source, allowing a user to validate the entire operation of the device

https://en.bitcoin.it/wiki/Hardware_wallet

---

**Trezor**


...the buffer overflows, allowing the attacker to write up to 60 bytes of data into a protected part of the memory. Depending on the memory layout the flaw can be escalated to arbitrary code execution...
What is a hardware wallet?

A **hardware wallet** is a special type of bitcoin wallet which stores the user’s private keys in a secure hardware device. They have major advantages over standard software wallets:

- private keys are often stored in a protected area of a microcontroller, and cannot be transferred out of the device in plaintext
- immune to computer viruses that steal from software wallets
- can be used securely and interactively, private keys never need to touch potentially-vulnerable software
- much of the time, the software is open source, allowing a user to validate the entire operation of the device

https://en.bitcoin.it/wiki/Hardware_wallet

Ledger

# Hardware wallets

<table>
<thead>
<tr>
<th>Wallet</th>
<th>Hardware</th>
<th>TRNG</th>
<th>SHA256</th>
<th>secp256k1</th>
<th>Open Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>TREZOR</td>
<td>STM32F205</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ledger</td>
<td>STM32F042</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ColdCard</td>
<td>STM32L475</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HWallet</td>
<td>NXP K20</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Features:**
- **USB**
- **OLED**
- **Secure MCU**
- **Secure Element**
- **TRNG**
- **SHA256**
- **secp256k1**

**Open Source:**
- ✓: Yes
- □: No
- ?: Unknown
Don't roll your own crypto!

YOU DON'T ROLL YOUR OWN CRYPTO

IF YOU USE HARDWARE CRYPTO
Code size comparison

git clone https://github.com/{PRODUCT}/{FIRMWARE} --recurse-submodules

cd {FIRMWARE}

wc -l `find ./ -name "*.c" -o -name "*.h"`

<table>
<thead>
<tr>
<th>Code Size</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HWallet</td>
<td>2.5M+</td>
</tr>
<tr>
<td>keepkey</td>
<td>346k+</td>
</tr>
<tr>
<td>TREZOR</td>
<td>162k+</td>
</tr>
<tr>
<td>Ledger</td>
<td>122k+</td>
</tr>
<tr>
<td>Coldcard</td>
<td>~4k</td>
</tr>
</tbody>
</table>

OLED font
License headers

nemanja@hacke.rs
Code layers

To Communication MCU

TX/Rx speed fixed to 115200 bps

SPI bus clocked at 1 MHz

NXP K82

OLED

UART

CRC

SPI

GPIO

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])

SHA256D

ECDSA: secp256k1

TX Signature

BTC TX

nonce

https://gitlab.com/nemanjan/hwallet

nemanja@hacke.rs
typedef struct {
    uint16_t type;
    uint16_t length;
    uint8_t data[32];
    uint32_t crc;
} Packet;

PACKET_Send();
PACKET_Receive();

typedef struct {
    SPI* spi;
    GPIOMUX* dcGpio;
    GPIOMUX* rstGpio;
    uint8_t dcPin;
    uint8_t rstPin;
    uint8_t buffer[ ];
} OLED;

OLED_WriteRow();
OLED_Clear();

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
while(1) {
    Packet msg;
    PACKET_Receive(&msg);
    switch(PACKET_MODULE(msg.type)) {
        case PACKET_BITCOIN:
            Bitcoin_Process(&msg);
            ...
    }
}
void Bitcoin_Process(Packet* msg) {
    switch(PACKET_FUNC(msg->type)) {
    case BITCOIN_FUNC_INIT_TX:
        Bitcoin_Tx_Init();
        ...
    }
}

https://gitlab.com/nemanjan/hwallet
POC | GTFO
What's next?

FIDO U2F

Google

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/44'

BIP-44
m/44'/0'
0' – BTC
60' – ETH
144' – XRP

More cryptocurrencies

BIP-39

… with collapse practice feed shame …
Questions?