breaking bitcoin

Breaking Hardware Wallets

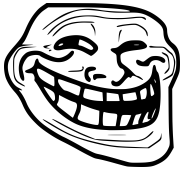
Breaking Bitcoin
September 2017

Nicolas Bacca
@btchip

Why Hardware Wallets ? - high level overview

breaking bitcoin

Public data



YES NO

Do you want to send
1.337 BTC to
1UnREADABLE

Operations on private data, with **user validation** and **proof of user presence**

Hardware Wallets - what is needed

breaking bitcoin

Protection against malware

- Protection of the private keys, the most critical asset
- Validation of the operation being performed, in a trusted environment

Protection against physical theft

Protection against bad cryptography

- Trustworthy RNG
- Side channel resistant implementations

How to break hardware ?

breaking bitcoin

Hack attack : software

Shack attack : low-budget hardware attack

Lab attack : “unlimited” time, resources

(From ARM Trustzone security guidelines : http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.prd29-genc-009492c/ch01s03s03.html)

Software attacks

breaking bitcoin

For generic programming error to buffer overflows, the usual things - nothing hardware specific

Repository of timely fixed TREZOR issues at
https://github.com/btchip/trezor-security-exploits

Shack attacks

breaking bitcoin

Obtaining information through observable leaks (timing / power) : SPA / DPA

Non invasive, non detectable

Chip can help to make things less observable, but implementation plays a major role (libsecp256k1, ctaes from Bitcoin Core help)

Fault injection

Invasive, hard to avoid, unless hardware helps - but not a “magic code change”

Clock/Vcc glitching

Bus/Memory modifications (more costly)

Lab attacks

breaking bitcoin

Chip decapping

Microscope analysis

Device cannot defend against such attackers supposing no constraints on time

#TLDR Protection Profile

breaking bitcoin

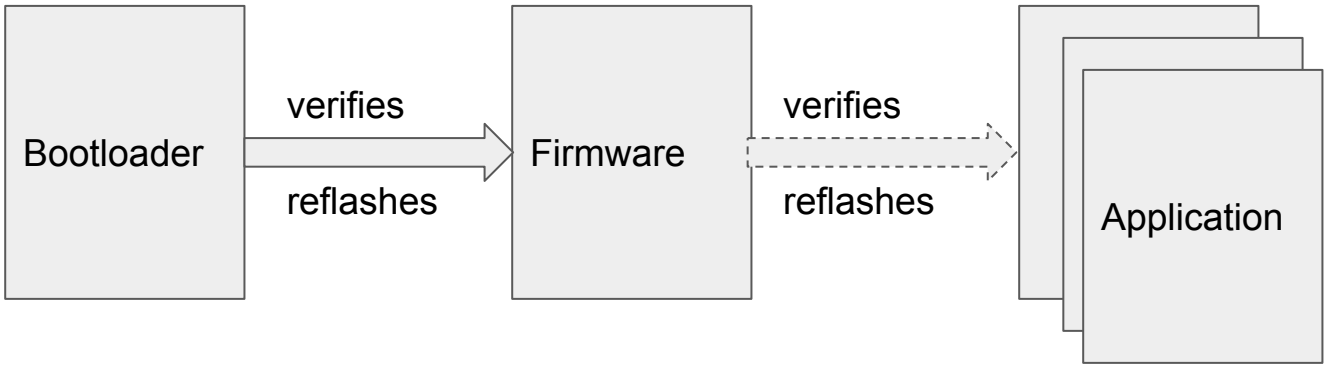
Hardware Wallet should not leak secrets on the go with a non too intrusive attack

An attacker that did her homework should not be able to run a SPA / glitching attack in a shop

Hardware Wallets should take some time to leak secrets when “borrowed”, preferably only using a highly intrusive method

Value of acceptable time may vary, at least 1 day ?

Hardware wallet chain of trust



Security vs convenience : keeping the user information while updating

Threat levels re. patchability

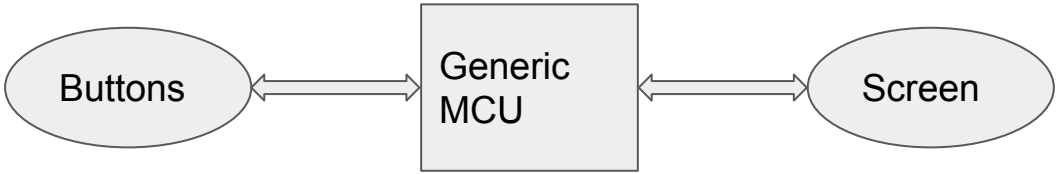
breaking bitcoin

DEFCON 3
Shack attack exploiting the chip

DEFCON 2
Shack attack exploiting the firmware

DEFCON 1
Software attack

Architecture : single generic MCU



Bitlox, KeepKey, TREZOR

Pros

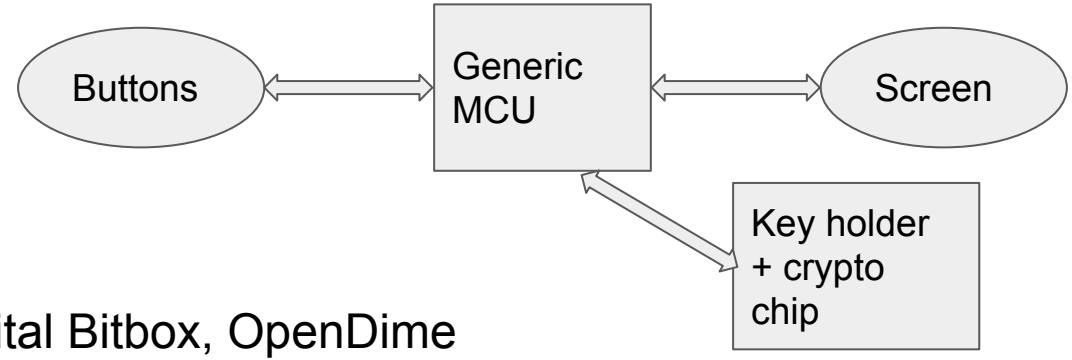
Auditability (up to the chip proprietary security mechanisms)

Cons

No proof of origin

Shack attacks : highly vulnerable

Architecture : generic MCU + dedicated crypto chip



Digital Bitbox, OpenDime

Pros

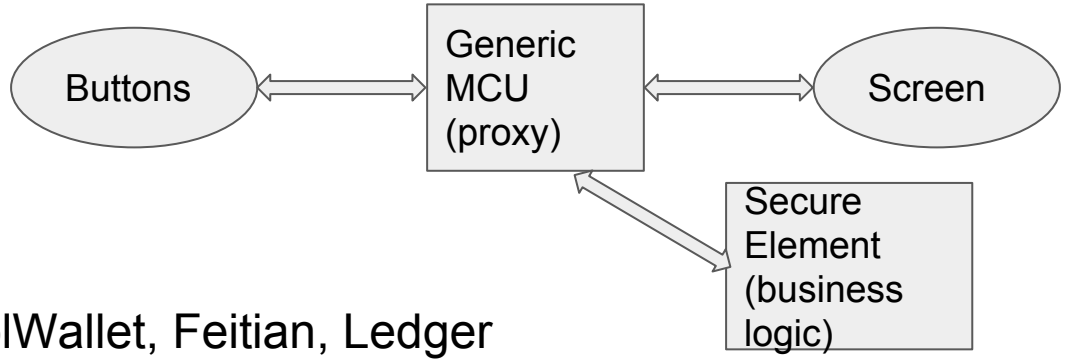
Better protection of assets than a Generic MCU

Cons

No proof of origin / Exotic architecture (business / secret split)

Shack attacks : not enough data to conclude

Architecture : Secure Element



CoolWallet, Feitian, Ledger

Pros

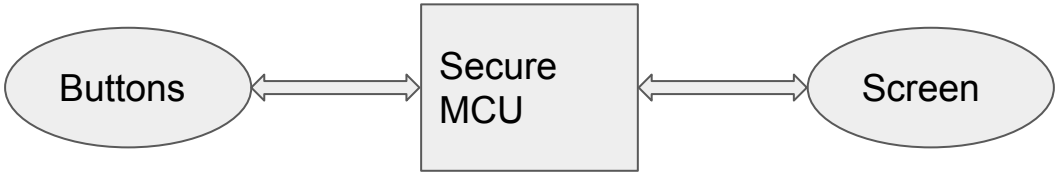
- Proof of origin

Cons

- Limited auditability

Shack attacks : strongly protected against

Architecture : secure MCU



Secalot

Pros

Auditability (up to the chip proprietary security mechanisms)

Cons

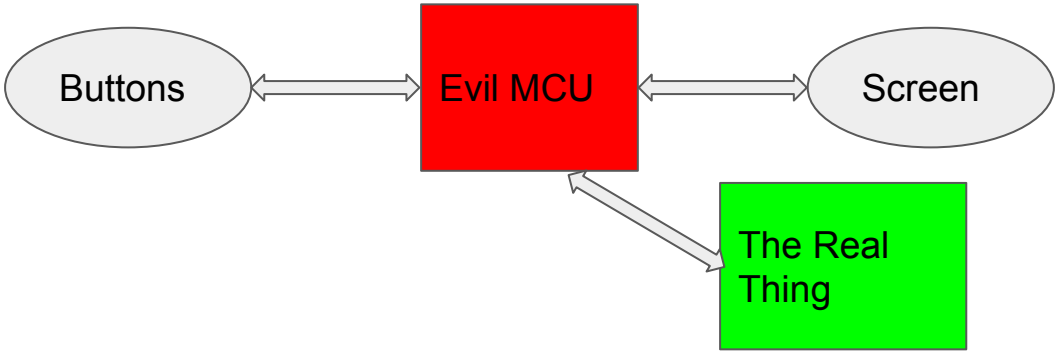
No proof of origin

Shack attacks : not enough data to conclude

Forgot anything ?

Impersonating the hardware is easy

Typical evil hardware wallet



Hard to protect against without visual inspection and/or building the device yourself

Traceability helps, to a given extent

Forgot anything ? take 2

Attacking from the UX angle is even easier

Payment Address SNAFU

breaking bitcoin

During a regular payment process, a newly generated address is used

If not checked using a second channel, no way to trust it - hardware wallets don’t help much in this situation

Payment requests (BIP 70) offering an end to end validation of the address are not popular

Confusing forks

breaking bitcoin

Latest example : Bitcoin Cash

- Same address format
- Anti-replay with a different signature algorithm, but too late if receiving

Malicious service risks when interacting with the device

- Obtain information about the other chain
- Sign on the other chain

Avoidable by extremely clear UX and limiting impact with tricks (such as locking to a specific HD derivation path)

Change account ransomware

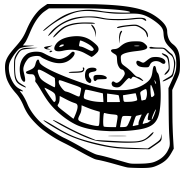
breaking bitcoin

Derive private key on path 44'/0'/0'/0 (BIP 32)

Send Change to 44'/0'/0'/1/entropy

Change looks fine. It really belongs to you, no problem here, I checked it

Ok, seems legit



Hey I got some entropy to sell you

Hardware wallets on existing hardware

breaking bitcoin

Attractive proposal : no additional hardware to buy

Achieved with modern CPUs featuring an isolation mode (Intel SGX, ARM TrustZone)

Same old issues issues

- Cryptographic algorithms can be vulnerable to passive attacks
- Little resistance against physical attacks (other than the complexity of the CPU)

And also new ones

Innovative virtualized hardware wallet issues

breaking bitcoin

Trusted display & I/O is often available as an optional feature

Different trust model, with two main options

Use attestation features constantly to “enhance” the security of the blockchain with trusted features (POET, Coco, …)

Use attestation features optionally to let the owner verify the integrity of the platform, then go back to a trustless model

Introducing BOLOS enclave on Intel SGX

breaking bitcoin

Open Source isolation model

- Moxie virtual CPU (well integrated with GNU toolchain)
- libsecp256k1 for ECC cryptography
- ctaes for AES encryption

Optional Intel attestation used to check the platform integrity

Platform code can be validated and recompiled by the user

Wallet code can be validated and modified by the user

Bounty at https://github.com/LedgerHQ/bolos-enclave-catchme (delayed a bit, because CVE-2017-5691 ...)

breaking bitcoin

Thank you, now go break some hardware
(hint : check your swag bag)
@btchip