La Sécurité des Crypto-Wallets

CryptoTerminal & BigBang

Séminaire Blockchain pour les Métiers

École Polytechnique, jeudi 21 mars 2019

Pascal.Urien@ethertrust.com
Pascal.Urien@Telecom-ParisTech.fr

TELECOM ParisTech

Bitcoin Transaction

```
01000000
01                Transaction Id
DE2D211EF429909B0AB8D2E7D25826A0EDD6281EC6DEDF2B822CE5014A349E72
01000000 Transaction Index
8A
47
                  r
30 44
02 20 0772ABD5D37D0CAAB881DBC8912628F93461839CC8D4BC007A355831A6061ED7
02 20 4CCCC34B34A9075FC09C9777EAB7A6F5612DA2130C1FF1C0E376AD9B2209D51D
01
                  s
41   Public Key
04 CFD7A542B8C823992AF51DA828E1B693CC5AB64F0CACF0F80C31A1ECA471786E
   285BDD3F1FE0A006BD70567885EF57EB149C8880CB9D5AF304182AC942E176CC
FFFFFFFF
01        Amount
D4180400000000000                 Payee hash160
19 76 A9 14 CB643DD608FB5C323A4A6342C1A6AC8048B409EB88 AC
00000000
01000000
```

Ethereum Transaction

```
F8 74 // RLP List, length= 116 bytes
   0C // nonce 1 byte =12 decimal
   85 06FC23AC00 // gasPrice = 30 GWei
   83 013880     // gasLimit = 80000 gas
   // recipient address 20 bytes
   94 6BAC1B75185D9051AF740AB909F81C71BBB221A6
   80 // Null Ether Value
   // Data 15 bytes "Temperature=25C"
   8F 54656D7065726174757265D323543
   1B // recovery parameter, 1 byte (27=+, 28=-)
   A0 // r, 32 bytes, ECDSA r parameter
   A9B58980F76EE6284800B82A2B5DF13E456887EC0CF426A5E5D6A738EB1784ED
   A0 // s, 32 bytes, ECDSA s parameter
   629633C6A3ED5FEE0FB40E2D1CF251345B885D372857B1A6C4762C9BE914281F
```

Public key is recovered from the signature two solutions + (27) and (-) (28)

https://etherscan.io/tx/0xd6904d832462ae17718c69e9caa0c3f3bed458382ac1f4e43b1aadd8e94744ad

SECP 256k1

$y^2 = x^3 + 7,\ x,y \in Z/pZ \quad p = 2^{256} + 2^{32} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 1$

G: GENERATOR

04 79BE667E F9DCBBAC 55A06295 CE870B07
029BFCDB 2DCE28D9 59F2815B 16F81798
483ADA77 26A3C465 5DA4FBFC 0E1108A8
FD17B448 A6855419 9C47D08F FB10D4B8

n: Curve Order

FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFE
BAAEDCE6 AF48A03B
BFD25E8C D0364141

ECDSA(r,s): → **Two integer values**

x private key ϵ [1,n-1]

P= xG = public key,

k ϵ [1,n-1] , kG= (x_R, y_R)

r = x_R mod n, e = H(M) 32 bytes LSB

s = k^{-1} (e + x r) mod n

VERIFY (r,s):

$u_1 = e\ s^{-1} \bmod n$

$u_2 = r\ s^{-1} \bmod n$

$(x_R, y_R) = u_1G + u_2P$

$v = x_R \bmod n$

Check v = r

RECOVER (r,s):

For R(x=r, y=y+) to R(x=r, y=y-)

$Q = r^{-1}(sR - eG)$

VERIFY(r,s) with Q as public Key

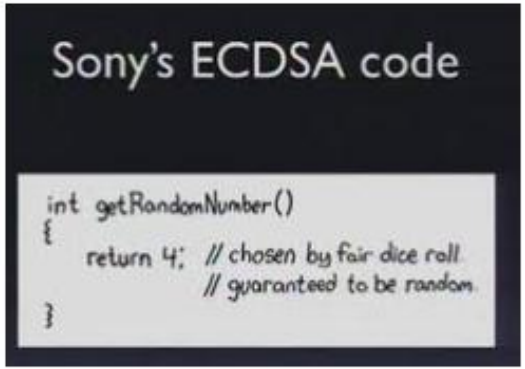
Console Hacking 2010, PS3 Epic Fail

27ième Chaos Communication Congress

Given a message signature (r,s), with $s = k^{-1} (e + x\ r) \bmod n$.

Given two signatures of two different messages, M1 and M2, with the same r (r,s1) and (r,s2), e1=h(M1), e2=h(M2).

The private key is computed as $x = (e1\ s2 - e2\ s1)\ r^{-1}\ (s1-s2)^{-1} \bmod n$

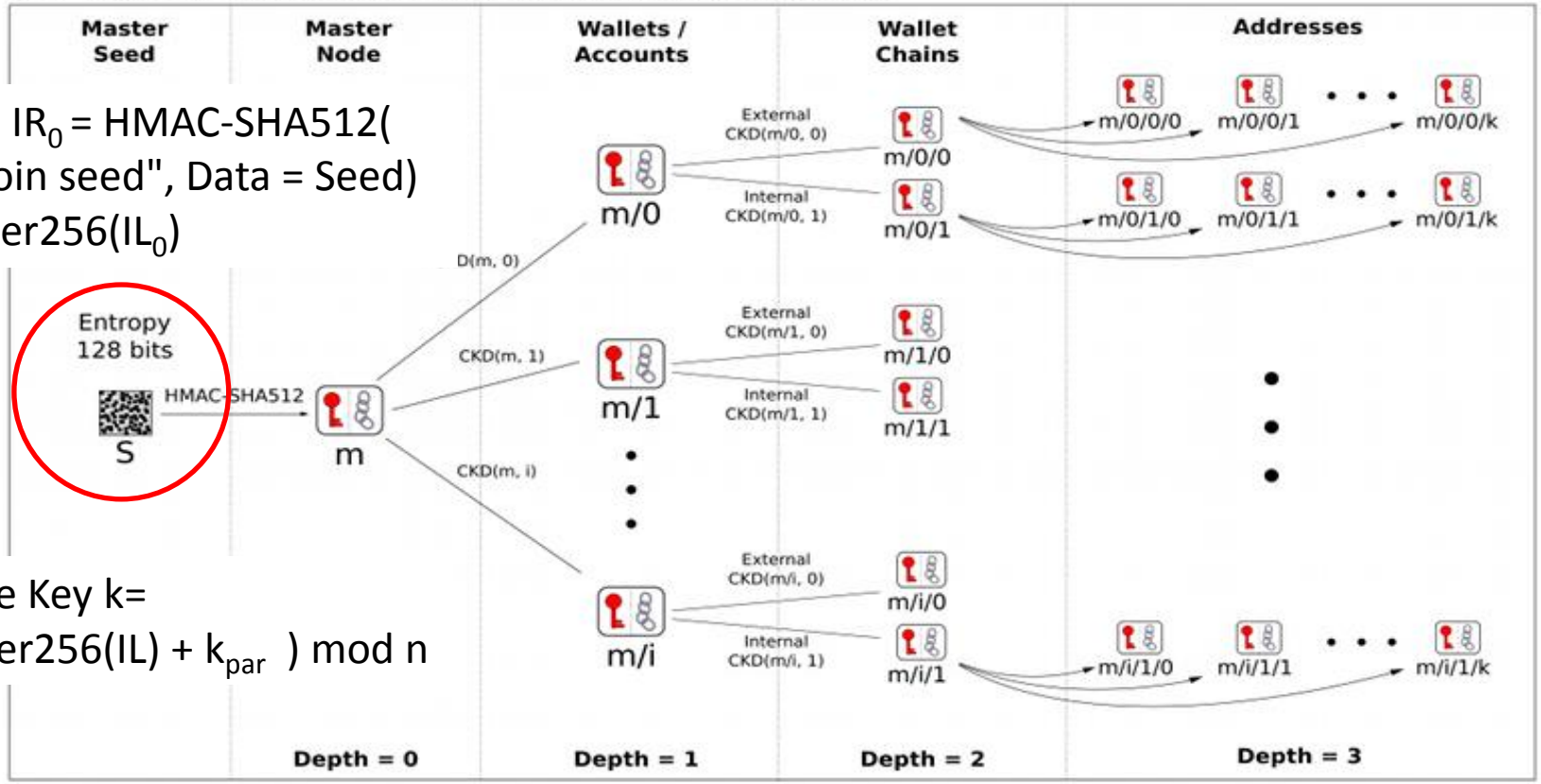


RFC 6979
Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)

BIP 32 - Hierarchical Deterministic Wallets

$IL_0 \,||\, IR_0 = \text{HMAC-SHA512}(\text{"Bitcoin seed"}, \text{Data} = \text{Seed})$

$k_0 = \text{ser256}(IL_0)$



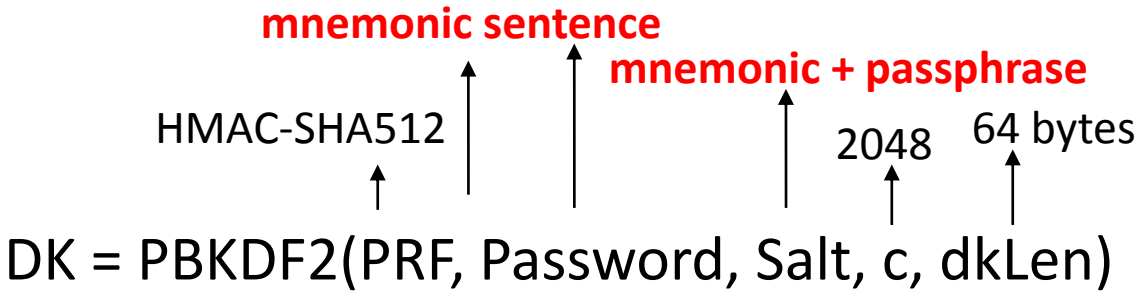
Private Key k=

$k = (\text{ser256}(IL) + k_{par}) \bmod n$

$\text{CKD}(m,i) = \text{HMAC-SHA512}(IR_{par}, \text{Data} = 0x00 \,||\, \text{ser256}(k_{par}) \,||\, \text{ser32}(i)) = IL \,||\, IR$

BIP 39

PBKDF-2: Password-Based Key Derivation Function (RFC 2898)

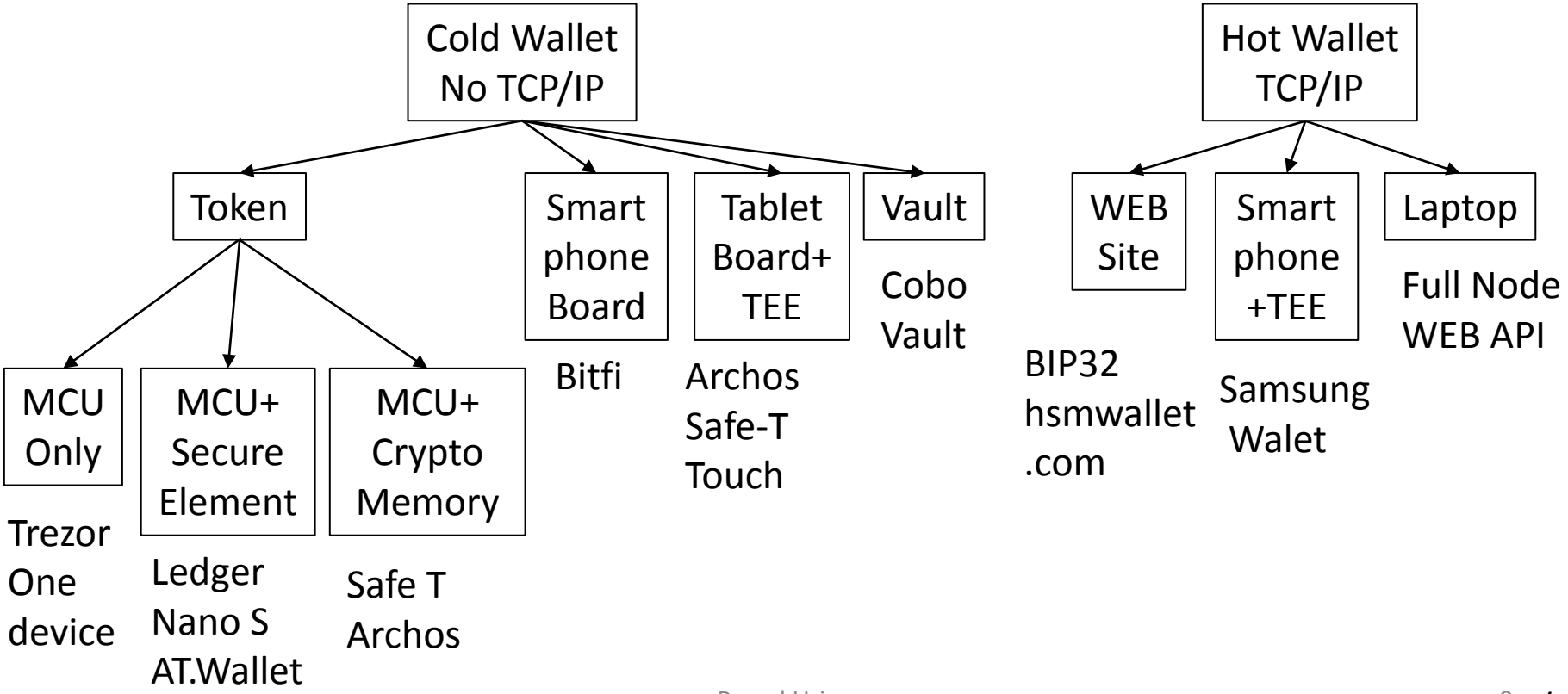


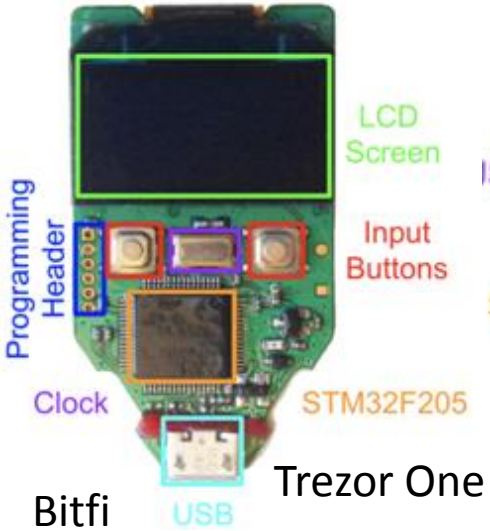
PASSWORD => SEED

What is a wallet ?

A safe place to generate blockchain transactions
A safe place to store private keys
A safe place to compute transaction signatures
A synchronization mechanism in needed in order to get the blockchain context

About Wallets



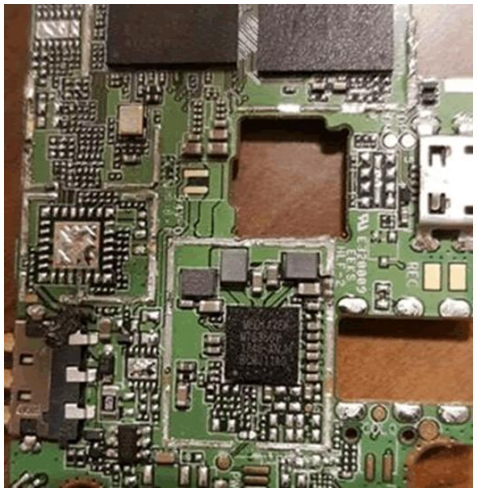


Trezor One

Ledger Nano S

Archos
Safe T

Bitfi



Cobo Vault

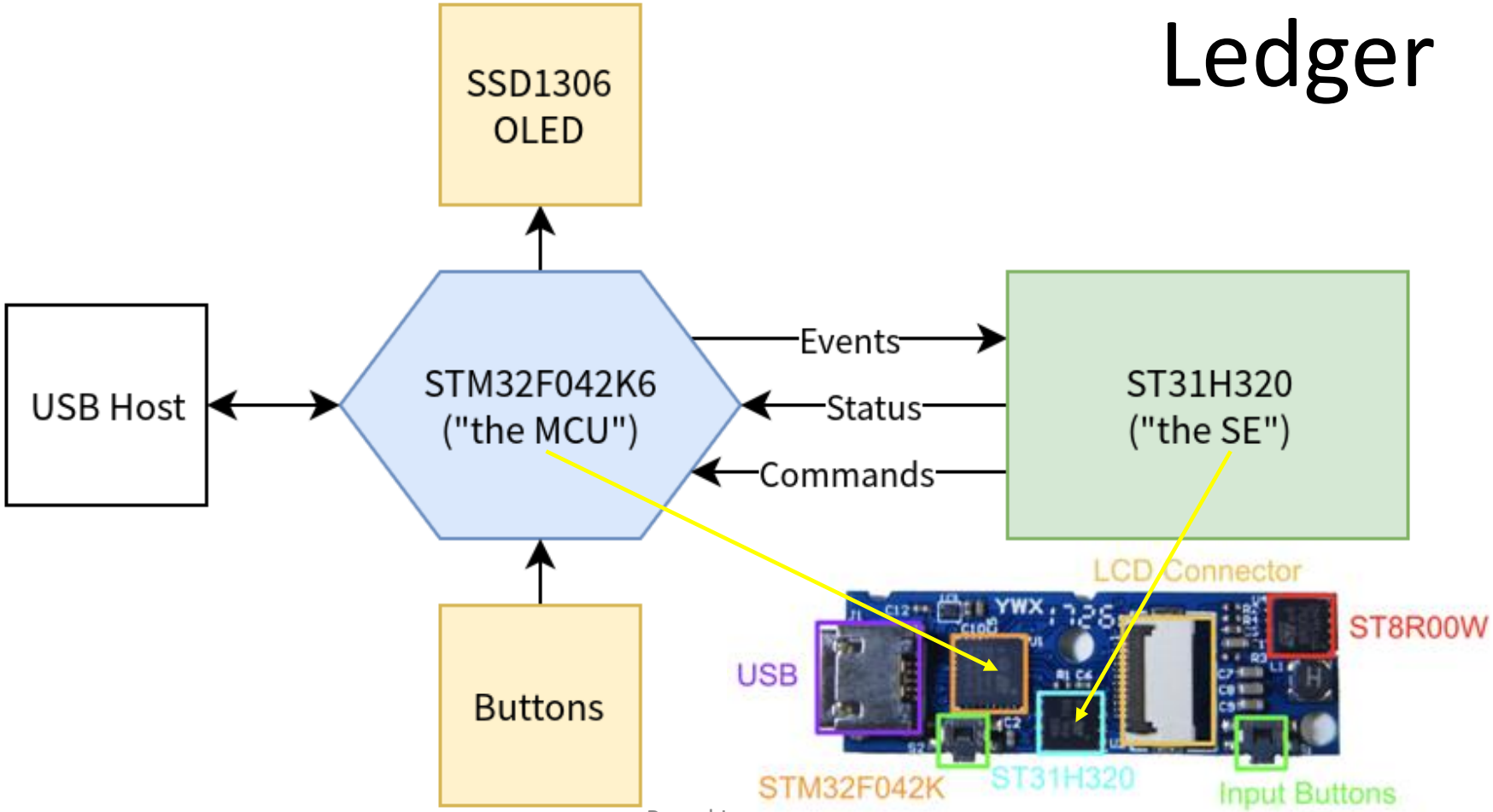
AT.Wallet

Samsung
Wallet

Archos
Safe-T
Touch

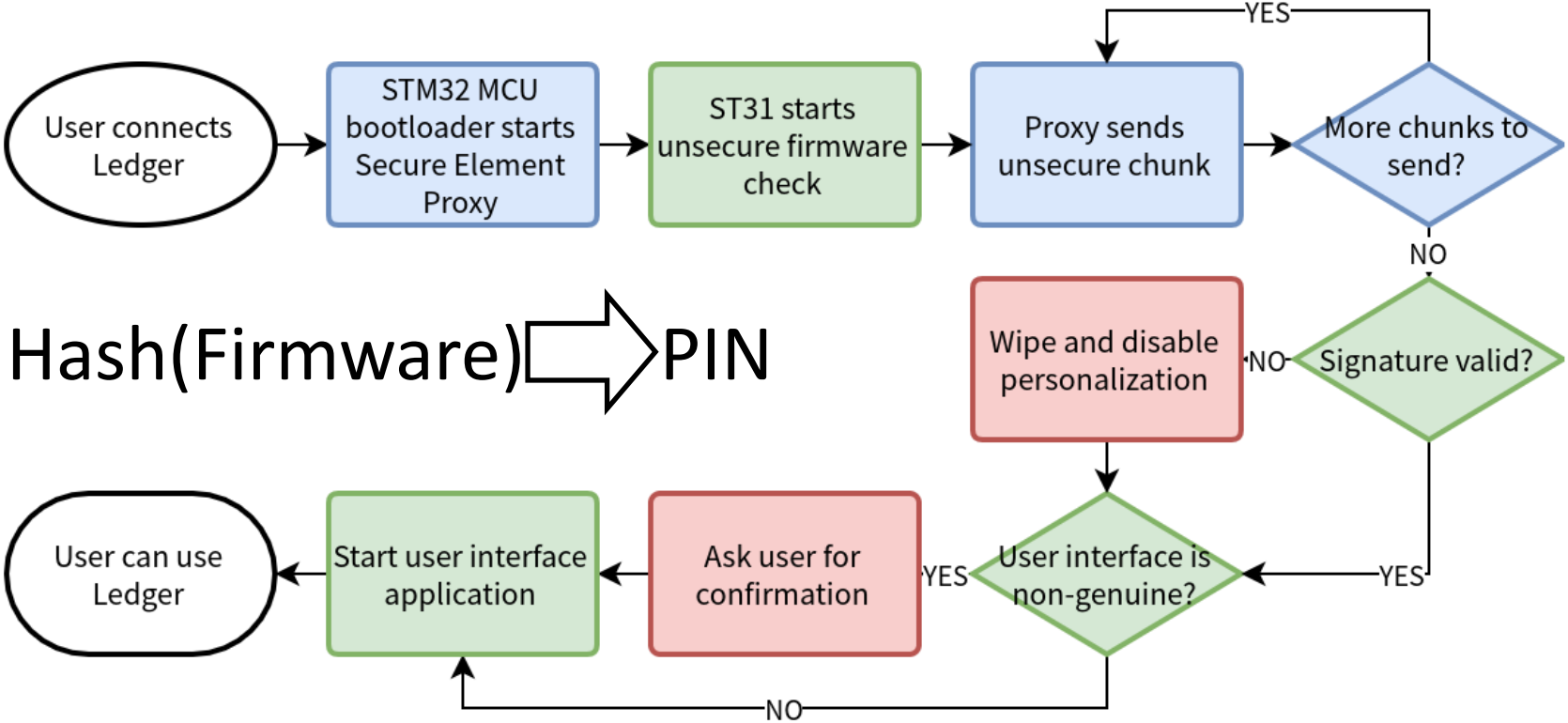
Threat\ Technology	PC Mobile	WEB Site	MCU Only	MCU+ Secure Element	MCU+ Crypto Memory	TEE	Vault	WEB Site + HSM	Ethertrust Crypto Terminal
Side Channel Attack									Secure Element
Malware APT									Big Bang
Firmware Software Integrity									Bare Metal
Injection									Firewall - keypad
PIN Code Password									Smartcard keypad - display
On-Line Attacks									Firewall Off Line

Ledger

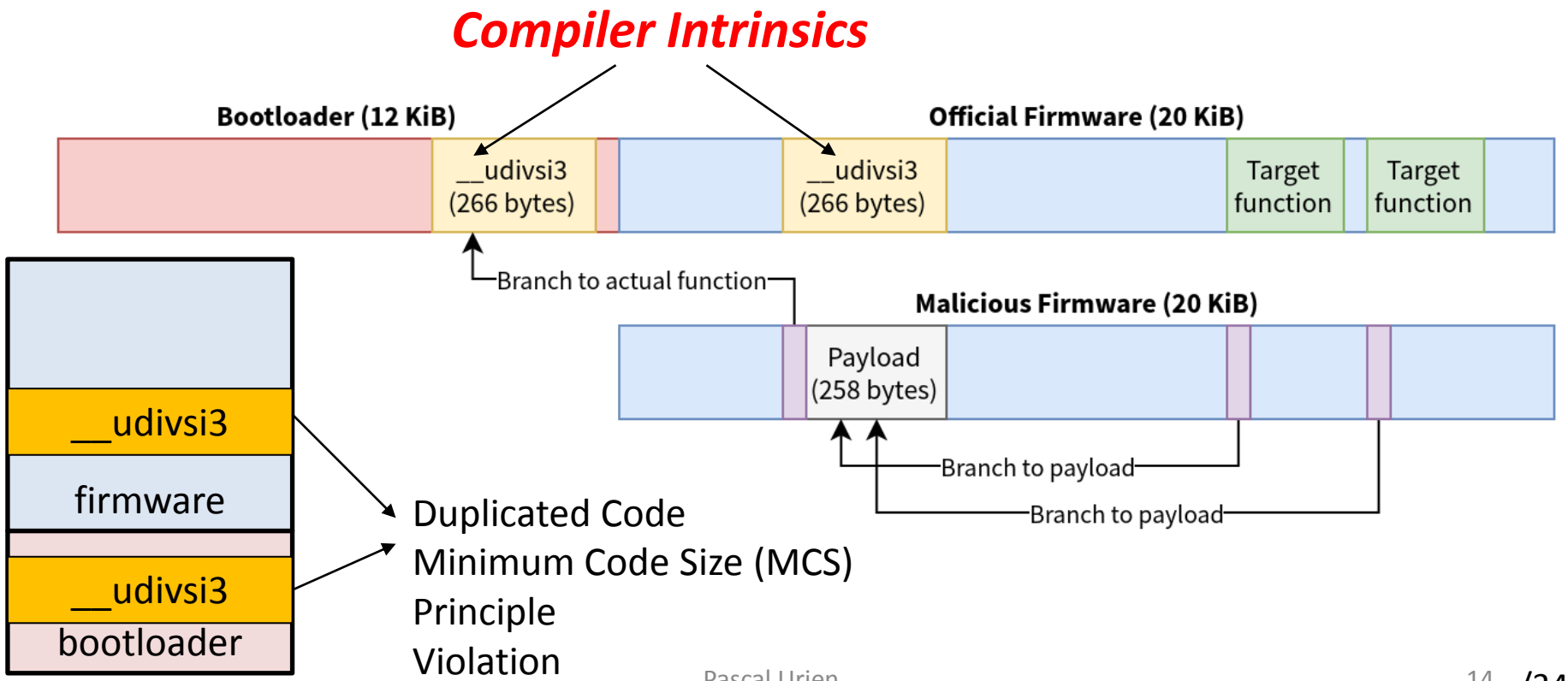

SSD1306
OLED
USB Host
STM32F042K6
("the MCU")
Events
Status
Commands
ST31H320
("the SE")
Buttons
LCD Connector
USB
ST8R00W
STM32F042K
ST31H320
Input Buttons

Ledger: Firmware Integrity Checking

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



Malware Installation



About the STM32F042K

- USB FS 2.0
- Frequency up to 48 MHz
- Memories
  - 16 to 32 Kbytes of Flash memory
  - 6 Kbytes of SRAM with HW parity

Main Processor

Secure Element

CPU

FLASH

RAM

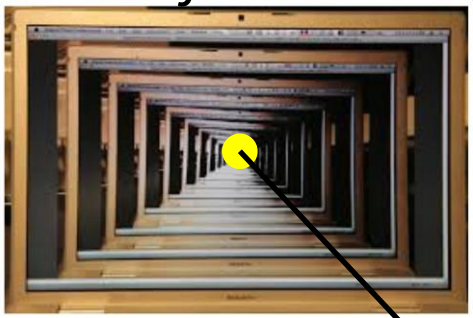
USB

SPI

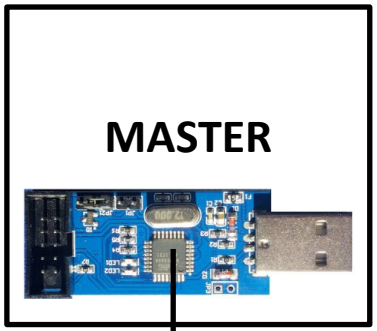
ROM ?

Serial Programming Interface (SPI)

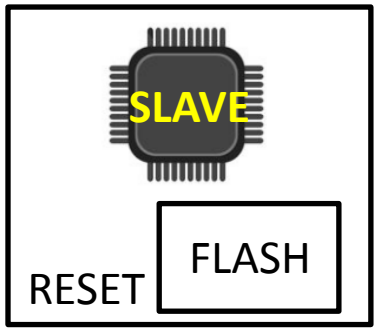
A flasher can be flashed



Serial Programmer

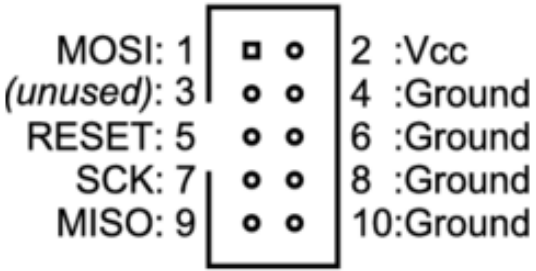


MCU



SCLK
MISO
MOSI
SS

SCLK (Serial Clock)
MOSI (Master Out Slave In)
MISO (Master In Slave Out)
SS (Slave Select)



AVR Programming Security Policy

All memories are erased

Serial Programming (SP) Requires RESET signal	Memories Access	Fuse: Disable RESET SP & PP
Parallel Programming (PP) Requires RESET signal	Fuses Access	Fuse: Disable SP from PP only
High Voltage Programming CLEAR under any conditions	Bootloader Configuration	Fuse: Others Voltage, Clock

EtherTrust Crypto Terminal (ECT)

Bringing Bank Card Security to Blockchain

Legacy Bank Card & Payment Terminal

*Terminaux de paiement Ingenico iWL220 et iWL250

Attention !

Chaque réparation ou ouverture d'un des composants par une personne non autorisée déclenche le module de sécurité et entraîne l'annulation de la garantie !

Élimination

Ce terminal de paiement est la propriété de Ingenico Payment Services et doit être restitué dans l'état initial ainsi que toute connectique y afférent à :

*https://payment-services.ingenico.com/~/media/documents/be/fr/ingenico-payment-services-terminals-mode-demploi-iwl250.ashx?la=fr

What is a Crypto Terminal ?

- A terminal
  - It communicates trough serial link
  - No TCP/IP stack
- Dedicated to security
  - Open Technologies
  - Removable smartcard EAL5+
  - Bare Metal
  - Integrity Probe (BigBang)
- Transaction Signature
- Transaction Generation (Ethereum)
- Not a Full Crypto Wallet
  - The Crypto Terminal is a firewall between smartcard and PC or Mobile

ECTv1

ECT In short

The Crypto Terminal, built with **OPEN technologies**, is equipped with an USB port supporting the ***USB Serial*** protocol. It includes a smartcard reader, and delivers the following services:

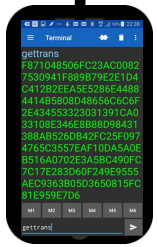
- Generation, calculation (BIP32), and secure storage of cryptographic keys;
- Generation of cryptographic signatures
  - in **synchronous mode** (driven from PC or mobile),
  - or in **asynchronous mode** (without connection to PC or mobile);
- Generation of signed transactions (Ethereum)
- All PINs are entered (like in the case of bank card terminal) on the Crypto Terminal;
- All signatures are acknowledged by the Crypto Terminal;
- Very High Security:
  - ***Removable Secure Element*** (smartcard) has EAL5+ certification.
  - ***Bare Metal***, all the Crypto Terminal software, including USB driver, can be securely flashed at any time.

SPI Programmer

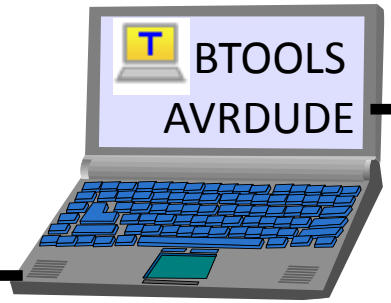
FIRMWARE

Crypto Terminal

OTG

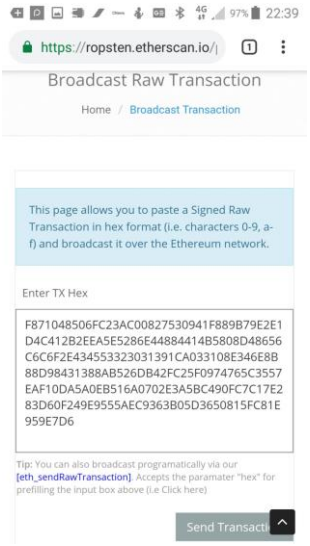


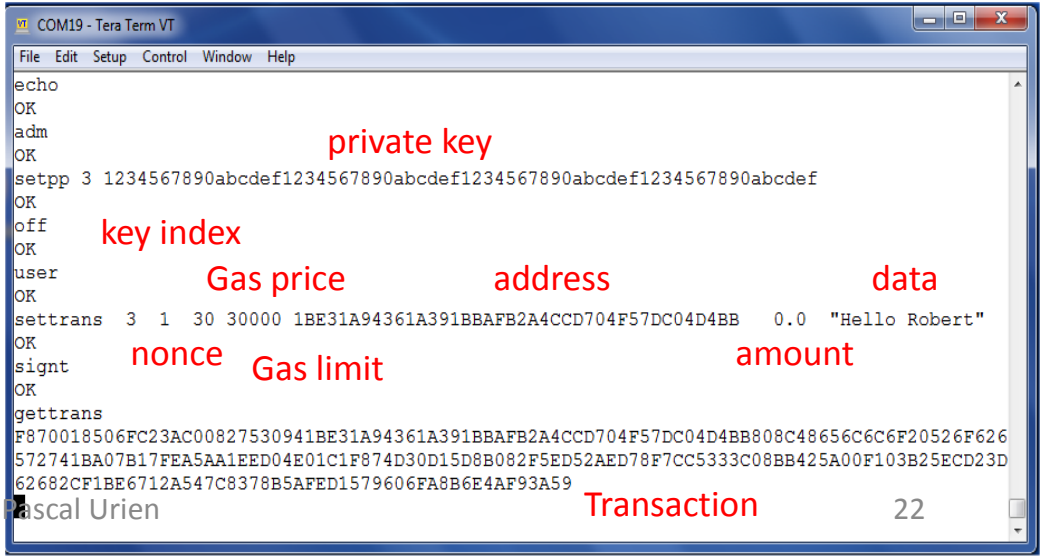
USB



https://ropsten.etherscan.io/pushTx

Blockchain Node





ECT commands

- Terminal Serial Commands
  - reboot, echo, disp, edit, enter, debug, nodebug, prompt, noprompt, on, off,status, changeadm, changeuser, changeuser2, adm, user, user2, computekey, genkey, setseed, getseed, getpub, btc, hash160, eth, setkey, setpp, getpriv, setlabel, getlabel, sign, recover, signr, signin, signout, signoutr, signdo, settrans, signt, gettrans, duplicate, read, write
- Crypto Currency (CC) Smartcard ISO7816 commands
  - Select, VerifyUserPin, VerifyUserPin2, VerifyAdminPin, ChangePin, GetStatus, Write, Read, ClearKeyPair & InitCurve, InitCurve & InitTree, GenerateKeyPair, DumpKeyPair, GetInfo, GetKeyParameter, SetKeyParameter, SignECDSA

Bare Metal ECT Hardware

SPI PROGRAMMER

USBasp FIRMWARE

USBasp Bootloader

ECT FIRMWARE

SECURE ELEMENT

CC FIRMWARE

JAVACARD OS

MAIN PROCESSOR ATMEGA2560

COMMUNICATION PROCESSOR ATMEGA16U2

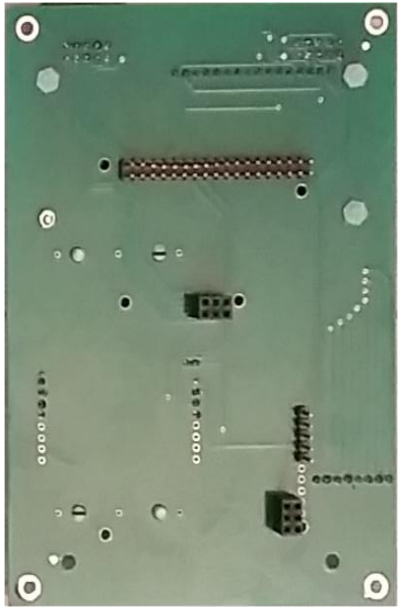
SMARTCARD SOCKET

DISPLAY

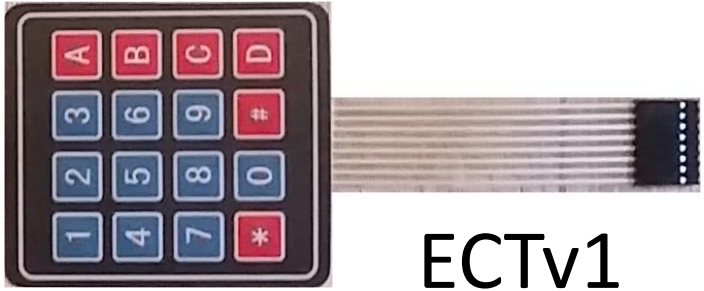
KEY PAD

Serial USB

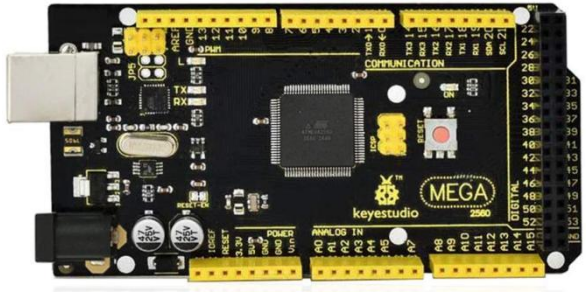
USB Serial FIRMWARE




MADE IN
FRANCE


A
B
C
D
3
6
9
#
2
5
8
0
1
4
7
*

ECTv1


COMMUNICATION
MEGA
keyestudio
POWER
ANALOG IN
DIGITAL

Some Crypto Terminal Figures

- Secure Element (javacard code)
  - javacard code, 2000 lines
    - Key Generation, Import, Calculation (BIP 32)
    - ECDSA Signature
- Crypto Terminal
  - Open Code (C), 10,000 lines
    - Security Policy
    - Command Interpreter
    - Signature Generation
    - Transaction Generation (Ethereum)
- PC/Mobile communication
  - Serial USB
    - Internet Connectivity
    - Blockchain Stack
    - Blockchain API

ISO7816 LIB

ECC LIB

CRYPTO LIB

BIG NUMBER LIB

KEYPAD LIB

LCD LIB

ETHER LIB

NO

KEY PRESSED

YES

MENU MANAGEMENT

KEY PAD

NO

CMD RECEIVED

YES

COMMAND PARSER

USB SERIAL

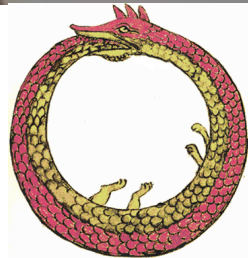
ECT FIRMWARE

CRYPTO CURRENCY FIRMWARE

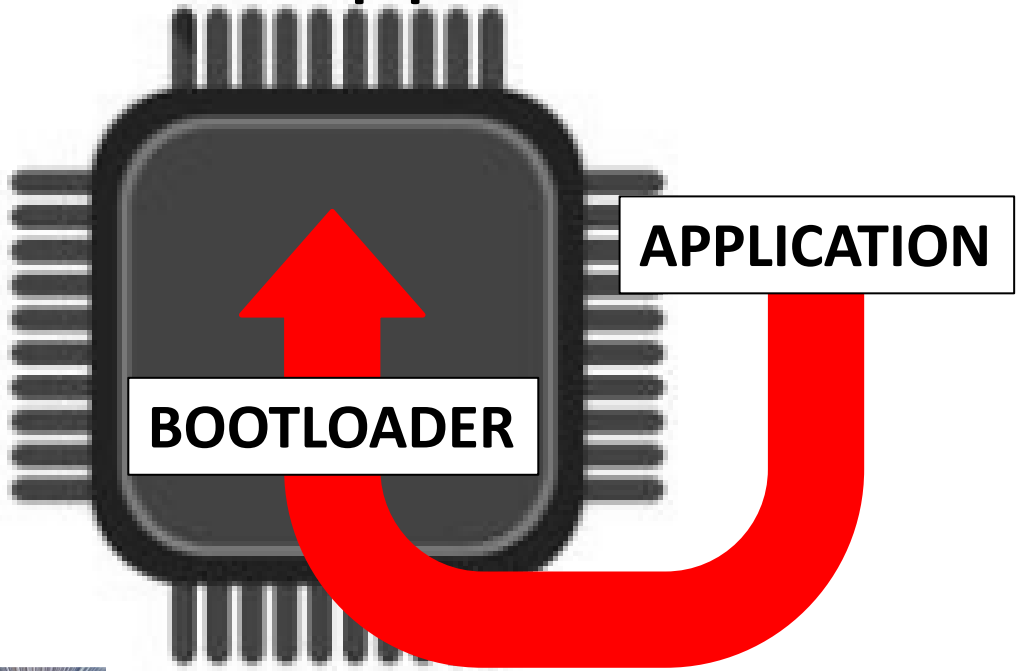
ECT Firmware

The Big Bang Paradigm

In classical systems
application trusts bootloader



In Classical System Downloaded Applications Trust Bootloader



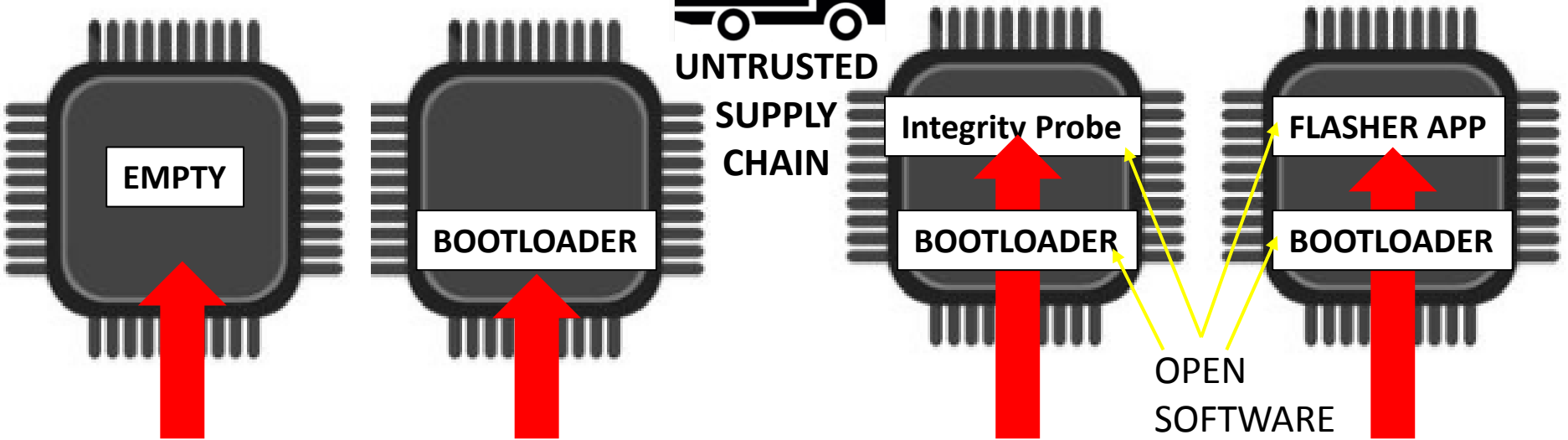
US warns of supply chain cyber-attacks

By Gordon Corera
Security correspondent

26 July 2018 Share

Bootloader may be modified by the supply chain

In the Big Bang Paradigm



Erase Chip

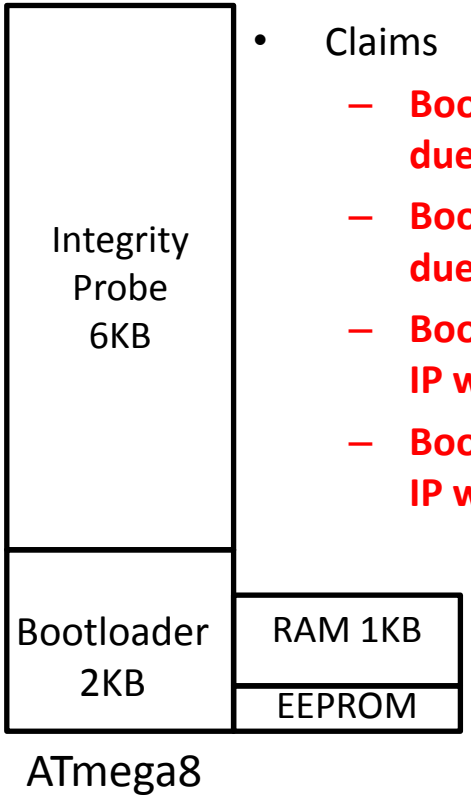
Bootloader is flashed in "*trusted*" environment

Integrity Probe Checks the Bootloader

FLASHER App

About the BigBang Integrity Probe

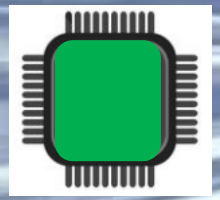
- Principles
  - All memories are hashed in a pseudo random order; this calculation generates an *Integrity Code* (IC).
  - The IC computing time (ICT) is included in the IC calculation
  - All unused (FLASH, RAM, EEPROM) memory are filled up with pseudo random values
  - All keys are obfuscated in the Integrity Probe code.
- Some figures
  - About 2^{54} permutation keys
  - Hash function, SHA3 Keccak256
  - ICT average time 18000ms
  - ICT maximum/minimum range 150ms
  - ICT precision 0,01 ms



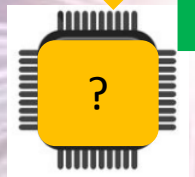
- Claims
  - **Bootloader is not able to compute IC due to hash function properties**
  - **Bootloader is not able to guess the ICT due to time dispersion (14 bits)**
  - **Bootloader is not able to monitor the IP without modifying the IC**
  - **Bootloader is not able to monitor the IP without modifying the ICT**

Integrity Code is displayed by led blinking (6 decimal digits) or thanks to a serial link

Erase Chip
Insert Bootloader

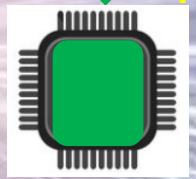


UNTRUSTED SUPLLY CHAIN



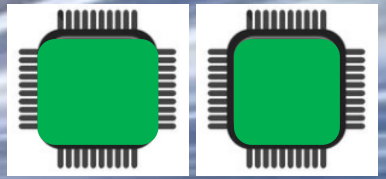
Load
Integrity
Probe

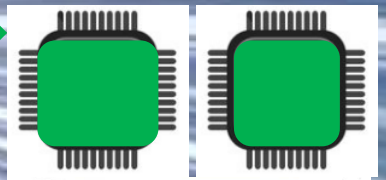
Bootloader
Verified



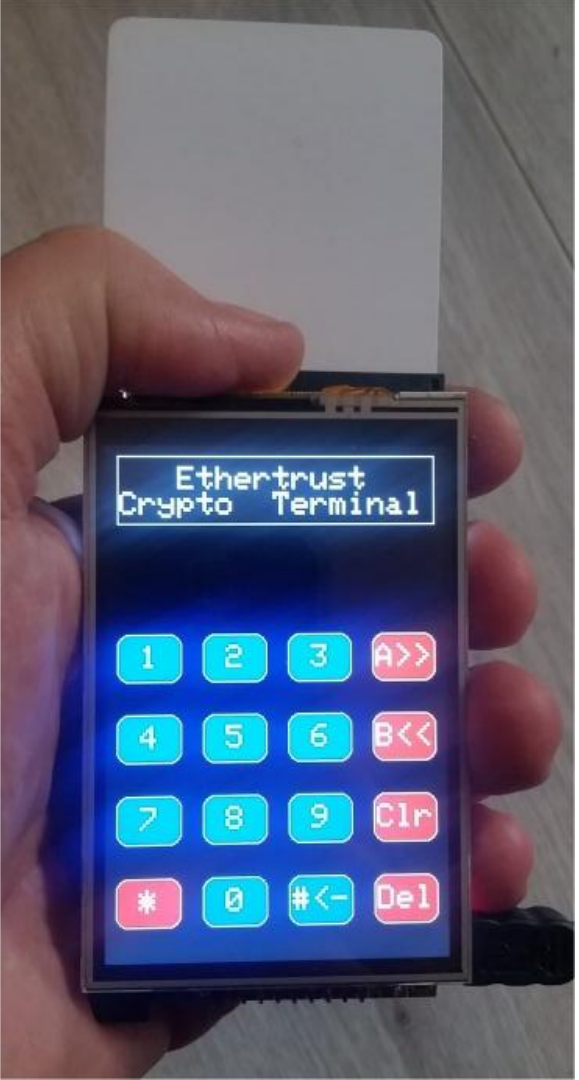
Load
Flasher
App

Create
Trusted
Flashers





Firmware



Questions ?

www.ethertrust.com

ECTv2
Touch Crypto Terminal