Trojan-tolerant Hardware

+ Supply Chain Security in Practice

Vasilios Mavroudis
Doctoral Researcher, UCL

Dan Cvrcek
CEO, Enigma Bridge
Highlights

- The private life of keys
- Weak links of the supply chain
- Lessons learned from airplanes
- Demo of our crypto hardware
- Protocols, Maths & Magic
- Politics, Distrust & Hardware Security
The Private Life of Keys

1. Someone designs an integrated circuit (IC)
2. IC is fabricated
3. IC is delivered to hardware vendor
4. Vendor loads firmware & assembles device
5. Device is sent to customer
6. Customer generates and stores key on the device
The Private Life of Keys

1. Someone designs the processor chip
2. Foundry fabricates the chip
3. Haulage transports chips
4. System vendor programs firmware
5. Distributors deliver a device to you
6. You create and use your key on the device

We can’t protect all the steps … but we can duplicate them

https://enigmabridge.com/mpc
Hardware Security Modules

Physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocesssing.

Features:
- Cryptographic key generation, storage, management
- Tamper-evidence, Tamper-resistance, Tamper-response
- Security Validation & Certification

Crypto Operations are carried out in the device
No need to output the private keys!
Hardware Security Modules

**Common Applications**
- Public Key Infrastructures
- Payment Processing Systems
- SSL Connections
- DNSSEC
- Transparent Data Encryption

**Cost**
- Hardware ($>10k)
- Integration Cost
- Operational/Support

https://enigmabridge.com/mpc
HSM Guarantees

1. Someone designs an integrated circuit (IC)
2. IC is fabricated
3. IC is delivered to hardware vendor
4. Vendor loads firmware & assembles device
5. Device is sent to customer
6. Customer generates and stores key on the device
What could go wrong?

- **Bugs**
  - CVE-2015-5464
    The HSM allows remote authenticated users to bypass intended key-export restrictions...

- **Backdoors/HT?**
  - NSA’s Own Hardware Backdoors May Still Be a “Problem from Hell”
  - Expert Says NSA Have Backdoors Built Into Intel And AMD Processors
  - Snowden: The NSA planted backdoors in Cisco products

https://enigmabridge.com/mpc
Proposed Solutions

- Trusted Foundries
  - Very expensive
  - Prone to errors/bugs

- Split-Manufacturing
  - Still Expensive
  - Prone to errors/bugs

- Post-fabrication Inspection
  - Expensive (+ re-tooling)
  - A huge pain, doesn’t scale

Arms Race
  - Adversaries always one step forward
  - Can never be 100% certain

It’s safe to assume we will never win
Solution from the sky and space

**Lockstep systems** are fault-tolerant computer systems that run the same set of operations at the same time in parallel.

- **Dual redundancy** allows error detection and error correction
- **Triple redundancy** automatic error correction, via majority vote
  → Triple Redundant 777 Primary Flight Computer

https://enigmabridge.com/mpc
Safety != Security

Fault-tolerant systems are built for safety and the computations are simply replicated

Not enough for security!

Keys would have to be copied across all processors

Security of our keys would depend on the weakest link
Our Solution

1. Someone designs an integrated circuit (IC)
2. IC is fabricated
3. IC is delivered to hardware vendor
4. Vendor loads firmware & assembles device
5. Device is sent to customer
6. Customer generates and stores key on the device
Who we are

Vasilios Mavroudis
Doctoral Researcher, UCL

Dan Cvrcek
CEO, Enigma Bridge

George Danezis
Professor, UCL

Petr Svenda
Assistant Professor, MUni
CTO, Enigma Bridge
Ingredients of the Solution

1. Hardware Components (IC)
   - Independent Fabrication
   - Non-overlapping Supply Chains
   - Programmable
   - Affordable
   - Bonus if COTS

2. Cryptographic Protocols
   - No single trusted party
   - Full Distribution of Secrets
   - Distributed Processing
   - Provably Secure (i.e., Math)
Smart Cards

Many Independent Manufacturers
- Private Fabrication Facilities
- Disjoint Supply Chains (location, factories, design)

Programmable Secure Execution Environment
- NIST FIPS140-2 standard, Level 4
- Common Criteria EAL4+/5+

Off-the-shelf Cost $5-$40
Multiparty Computation Protocols

Distributed Operations
- Random number Generation
- Key Pair Generation
- Decryption
- Signing

Provably Protect against
- $N-1$ Malicious & Colluding parties
- $N$ Malicious & non-colluding parties

https://enigmabridge.com/mpc
THE prototype
Many Smart Cards

Components
- Multiples of 120 smartcards
- TCP/UDP access to smartcards
- FPGA manages the communication bus
  - 1Gbit/s bandwidth for requests
SIMONA boards with 120 JCs
Gigabit link to untrusted Linux server
FPGA
- JC /ISO7816->TCP
JavaCards
- FIPS140-2 Level 3
- CC EAL5+
Plugging it into a cloud service

- Registration proxy
- Wrapper (e.g. PKCS11)
- Language binding
- External API (JSON)
- Security domain Manager (virtualisation)

Controllers:
- FIPS140-2 L3 hardware
- FIPS140-2 L3 hardware
- FIPS140-2 L3 hardware

Monitoring dashboards

https://enigmabridge.com/mpc
Giving smart-cards an infrastructure

mpc.enigmabridge.com

MPC RESTful server

SIMONA board (192.168.42.10)
LAS VEGAS

SIMONA board (84.92.209.143)
CAMBRIDGE UK

MacBook-2
Giving smart-cards an infrastructure

mpc.enigmabridge.com

MPC RESTful server

SIMONA board (192.168.42.10)
LAS VEGAS

https://enigmabridge.com/mpc

SIMONA board (84.92.209.143)
CAMBRIDGE UK

- ARM
- Intel
- SPARC
Giving smart-cards an infrastructure

mpc.enigmabridge.com

MPC RESTful server

SIMONA board (192.168.42.10)
LAS VEGAS

https://enigmabridge.com/mpc

MacBook-2
Demo 2
 MPC RESTful server

- RESTful requests (switch evil)
- MPC key generation
- web-socket servers

ICs with Hardware Trojans

MacBook-2

mpc.enigambridge.com

https://enigambridge.com/mpc
Demo 3
PERFORMANCE
Tolerance vs Runtime

![Graph showing Tolerance vs Runtime with lines for Key Generation, Signing, and Decryption.](https://enigmabridge.com/mpc)
Scalability

Linear, no bottleneck
Protocols
Key Points

- No single IC is trusted with a secret (e.g., private key)
- Misbehaving ICs can be detected by honest ones
- If one IC is excluded from any protocol, user can tell

Bonus: Minimize interaction between ICs for performance
Sharing a Secret

- Split a secret in *shares*
- The secret can be *reconstructed* later
- Without *sufficient* shares not a single bit is leaked
- Splitting Parameters:
  - How many shares the secret is split into (*n*)
  - How many shares you need to reconstruct the secret (*t*)

*In our case: Each 3 ICs hold shares for a secret*

https://enigmabridge.com/mpc
Classic Key Generation

Single IC System
1. Bob asks for new key pair
2. Backdoored IC generates compromised key
3. Private Key is “securely” stored
4. Weak Public key is returned

Problems
- Malicious IC has full access to the private key
- Bob can’t tell if he got a “bad” key

https://enigmabridge.com/mpc
Distributed Key Generation

*THE* Public Key

Public Keys

ICs holding key shares

https://enigmabridge.com/mpc
Distributed Key Generation

*THE* Public Key

Public Keys

ICs holding key shares

https://enigmabridge.com/mpc
Distributed Key Generation

*THE* Public Key

Public Keys

ICs holding key shares

https://enigmabridge.com/mpc
Distributed Key Generation

1. ICs holding key shares
2. Public Keys
3. Public Keys
4. *THE* Public Key
Distributed Key Generation

Key Points
- No single IC is trusted with a secret (e.g., private key) ✔
- Misbehaving ICs can be detected by honest ones ✔
- If one IC is excluded from any protocol, user can tell ✔

Bonus: Minimize interaction between ICs for performance ✗
Classic Decryption

Single IC System
1. Bob asks for ciphertext decryption
2. Backdoored IC decrypts ciphertext
3. Bob retrieves plaintext

The IC needs full access to the private key to be able to decrypt ciphertexts.

https://enigambridge.com/mpc
Distributed Decryption

Help me decrypt this email

Decryption Shares
ICs holding key shares

https://enigmabridge.com/mpc
Distributed Decryption

Decryption Shares
ICs holding key shares

Help me decrypt this email
Distributed Decryption

Help me decrypt this email

Decryption Shares
ICs holding key shares

https://enigmabridge.com/mpc
Distributed Decryption

Key Points
- No single IC is trusted with a secret (e.g., private key) ✔
- Misbehaving ICs can be detected by honest ones -
- If one IC is excluded from any protocol, user can tell ✔

Bonus: Minimize interaction between ICs for performance ✔
Classic Signing

Single IC System

1. Bob asks for document signing
2. Backdoored IC signs the plaintext
3. Bob retrieves signature

The IC needs full access to the private key to be able to sign plaintexts.

https://enigmabridge.com/mpc
Distributed Signing

ICs holding key shares

Signature Shares

Help me sign this document

https://enigmabridge.com/mpc
Distributed Signing I

Caching
1. Bob sends a caching request
2. The ICs verify Bob’s authorization
3. Generate a random group element based on j
4. Bob sums the random elements

Properties
- Caching for thousands of rounds (j)
- Bob stores $R_j$

https://enigmabridge.com/mpc
Distributed Signing II

**Signing**

1. Bob asks for document signing & sends $R_j$, $j$, and the hash of $m$
2. ICs verify his authorization
3. ICs check if $j$ has been used again
4. ICs compute their signature share
5. Bob sums all signature shares

**Properties**

- All ICs must participate
- Significant speed up with caching

https://enigmabridge.com/mpc
Distributed Signing

Key Points
- No single IC is trusted with a secret (e.g., private key) ✔
- Misbehaving ICs can be detected by honest ones ✔
- If one IC is excluded from any protocol, user can tell ✔

Bonus: Minimize interaction between ICs for performance ✔
How we made it scale

https://enigmabridge.com/mpc
How we made it scale

https://enigmabridge.com/mpc
How we made it scale
How we made it scale
How we made it scale

https://enigmabridge.com/mpc
How we made it scale

But how can all these groups have shares for the same key?
Key Replication

1. Group A generates a public key
2. A1, A2, A3 send their shares to B1, B2, B3
3. Each IC in B receives shares from A1, A2, A3
4. Each IC in B combines the 3 shares and retrieves its private key
Key Replication

1. Group A generates a public key
2. A1, A2, A3 send their shares to B1, B2, B3
3. Each IC in B receives shares from A1, A2, A3
4. Each IC in B combines the 3 shares and retrieves its private key
5. A1, A3 and B2 collude

The adversary retrieves the secret!
Key Replication

1. Group A generates a public key
2. Then each IC in A splits its private key in three shares and sends them to B1, B2, B3
3. Each IC in B receives shares from A1, A2, A3
4. Each IC in B combines the 3 shares and retrieves its private key share

The full public keys of A and B are the same!
geopolitics
“We can guarantee security if there is at least one honest IC that is not backdoored or faulty.”
“We can guarantee security if there is at least one honest IC that is not backdoored or faulty.”

What if all ICs are malicious?
Government-level adversaries
- Deep access to fabrication facilities
- Very sophisticated techniques
- Very hard to detect their Backdoors/Trojans
- Very secretive; highly classified
- Won’t share their backdoor details
Government-level adversaries

- Deep access to fabrication facilities
- Very sophisticated techniques
- Very hard to detect their Backdoors/Trojans
- Very secretive; highly classified
- Won’t share their backdoor details
- Unlikely to collude with anyone

https://enigmabridge.com/mpc
“We can guarantee security even when all ICs are malicious, if at least one does not collude.”
Conclusions & Future

New crypto hardware architecture

- For the first time, tolerates faulty & malicious hw
- Decent Performance
- Scales nicely; just keep adding ICs
- Suitable for commercial-off-the-shelf components
- Existing malicious insertion countermeasures are very welcome!
Trojan-tolerant Hardware
+ Supply Chain Security in Practice

find more at
https://enigmabridge.com/mpc

Vasilios Mavroudis
Doctoral Researcher, UCL

Dan Cvrcek
CEO, Enigma Bridge