Fault Attacks:
How Practical are Fault Injection Attacks, Really?

Hamed Ramezanipour, Ali Nouri

Research Center for Developing Advanced Technologies (RCDAT)

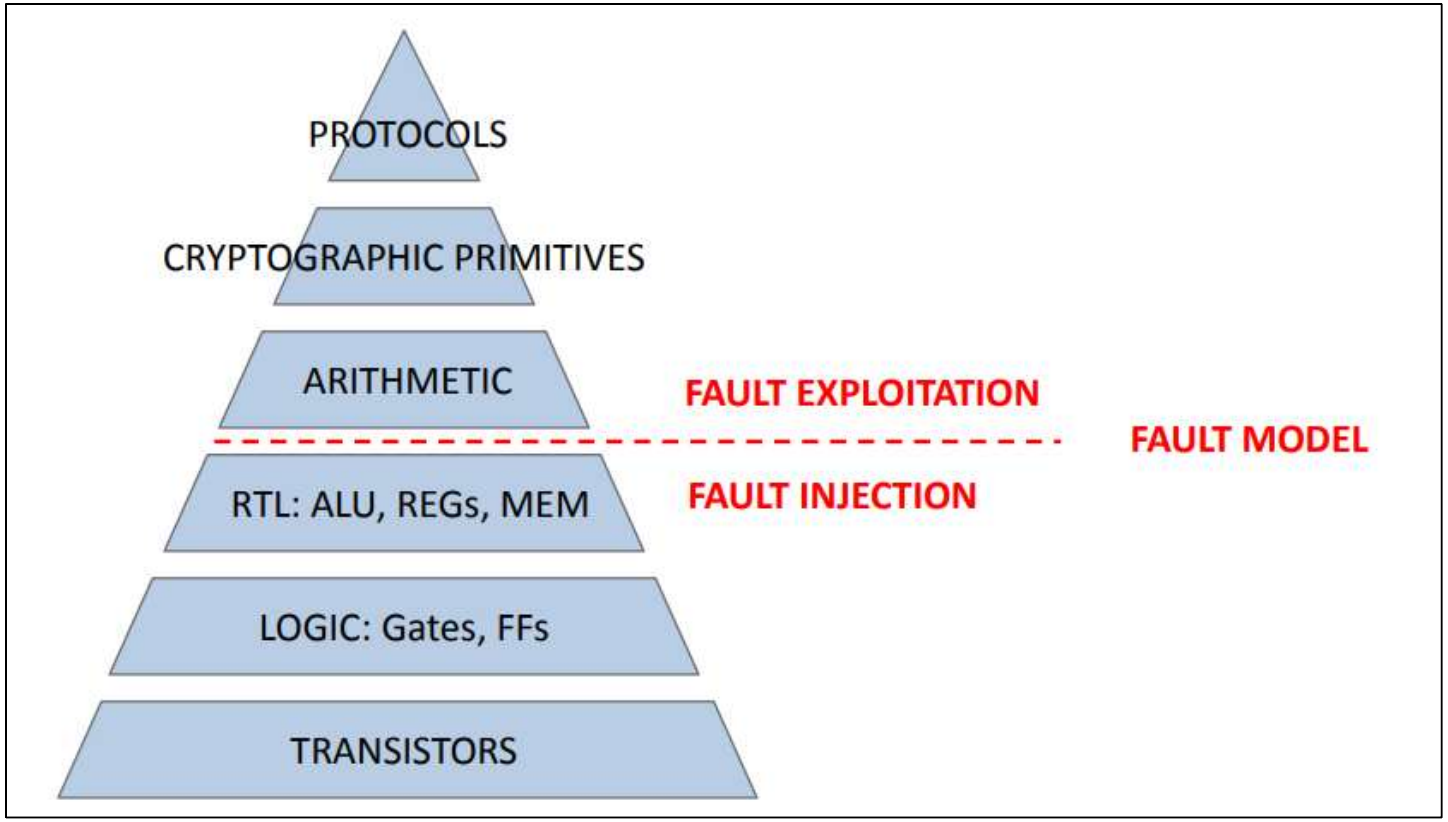
October 2025

Agenda

- ❑ Introduction
- ❑ Fault Attacks
- ❑ Fault Analysis
- ❑ Fault Countermeasures
- ❑ Case studies
- ❑ Further Resources
- ❑ Conclusions
- ❑ References

Introduction

Introduction



The embedded design space

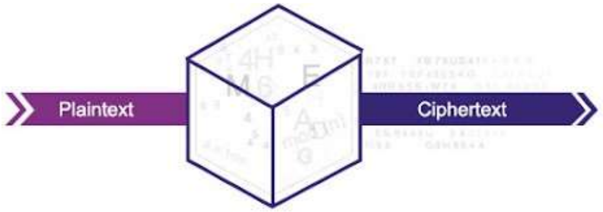
Introduction

- Classification of Attacks

Gray-Box Model

The attacker has access (sometimes physical) to the device.

White-Box Model

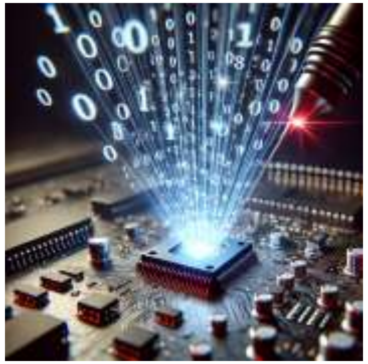


The attacker is inside the system and has access to intermediate values and...!

Black-Box Model

The attacker has access to a set of plaintexts and their corresponding ciphertexts.

- A primitive may be secure in black box model but not secure in grey or white box models!
- This talk is about one of the **Grey-box** model.



Fault Attack

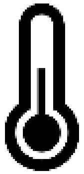
Introduction

- ❑ Implementation

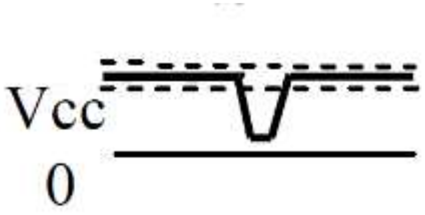
Ways to generate a fault [Lomne, 2015]

There are different ways to generate a fault:
- Electrical glitch on pins (VCC, CLK, I/O,)
- Electrical glitch on the die (FBBI)
- Light injection
- ElectroMagnetic (EM) field injection

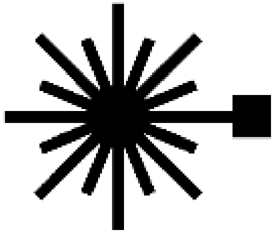
Temperature

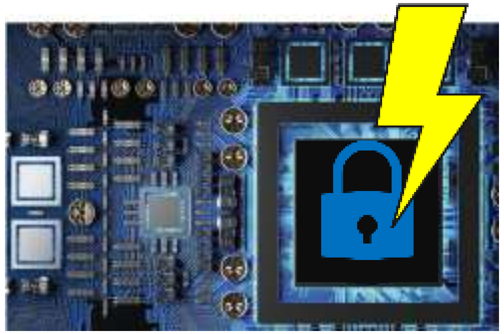


Voltage Spikes

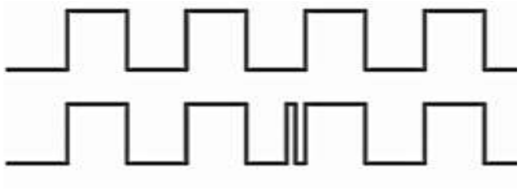


Laser Beam

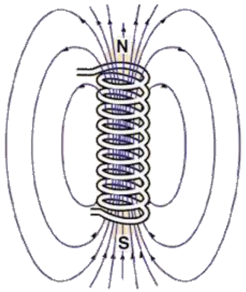




Device Under Attack



Clock Glitches



Electromagnetic Pulses

Introduction

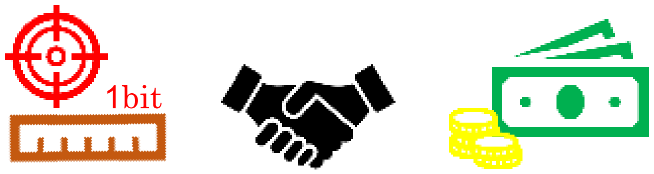
- ❑ Comparison

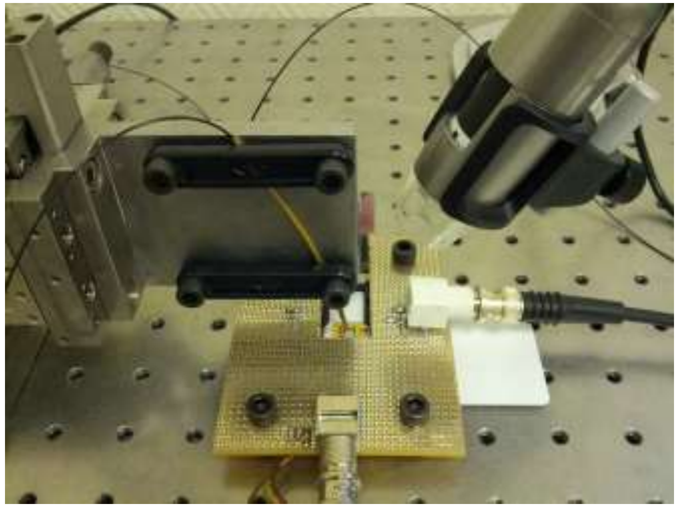
Global/Low-Cost/Low-Precision

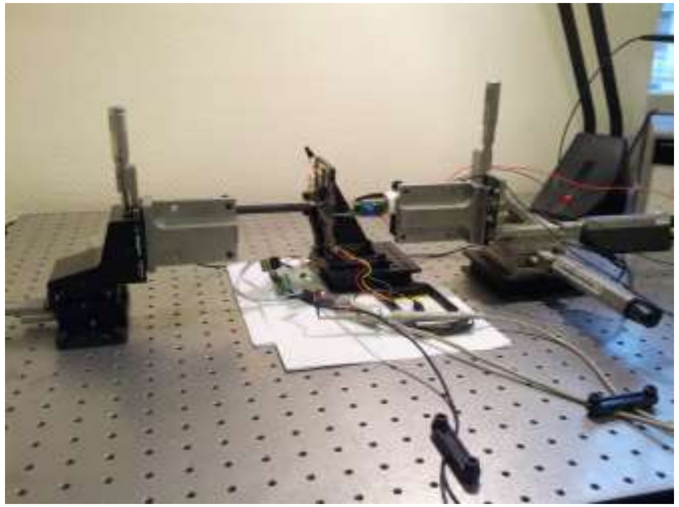
Clock/Voltage glitch, temperature

Local/High-Cost/High-Precision

Laser, Electromagnetic, Ion Beam

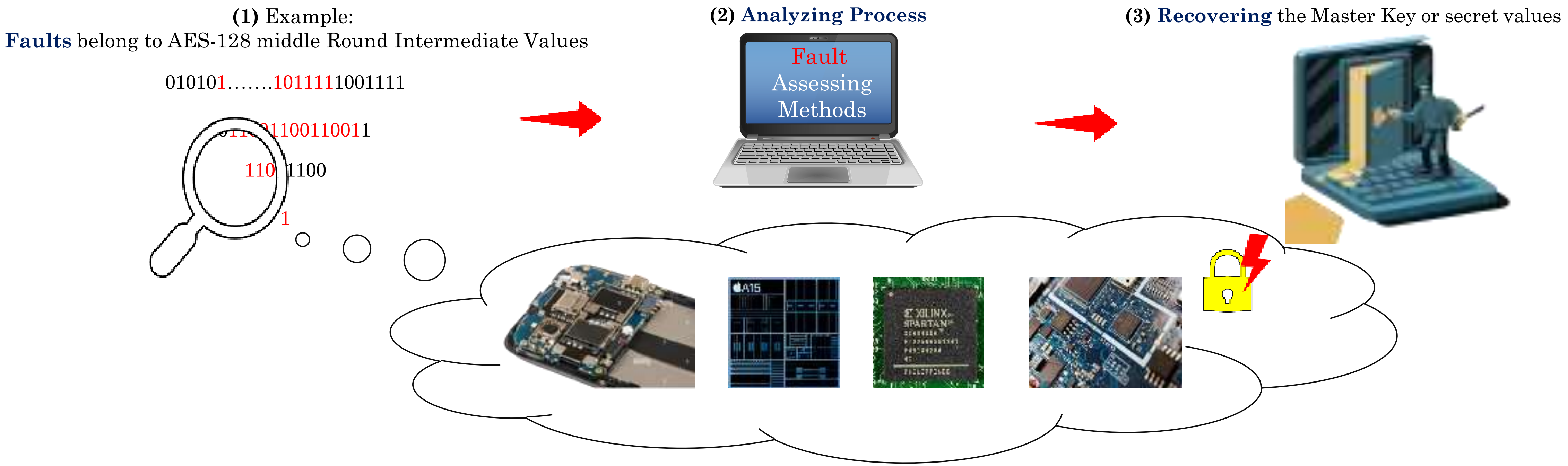






Introduction

- ❑ Fault Evaluation Route



Introduction

- Recent Practical Scenarios in Real World

Voltage Glitch for Recovering Critical Data From Tesla Autopilot [1]

Voltage Glitch on Secure Boot of SpaceX Starlink [2]

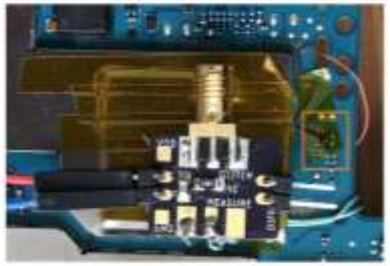
EMFI Against Secure Automotive Bootloaders [3]

EMFI Against lattice-based signature [4]

Clock Glitch Against Glitch Detector Structure in 12th Generation Intel Core Processor [5]

EMFI on Apple A4 [6]

Glitching the keepkey hardware wallet [7]



Voltage fault injection on PlayStation Vita [8]

[1]Niclas et al, 2023 [2]Lennert, 2022 [3]Enrico et al, 2023 [4]Prasanna et al, 2019 [5]Amund et al, 2024 [6]Carlton Shepherd et al, 2022 [7]Riscure team, 2022 [8] Yifan Lu, 2019.

Introduction

- Why Fault attacks are important?

•**Reveal vulnerabilities** not found in normal testing

•**Real-world consequences**: bypass checks, leak secrets, disrupt safety

•**Low-cost for attackers** yet effective against complex systems

•**Applicable across domains**: smartcards, IoT, automotive, defense

Fault Attacks

❑ Fault attacks categories

- **Non-Invasive**:
  No physical modification; e.g., EMFI.

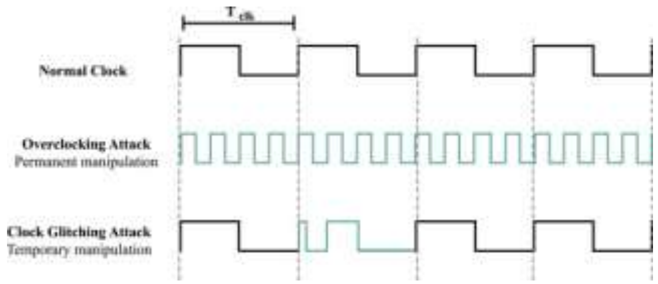
- **Semi-Invasive**:
  Limited physical access without full chip removal; e.g., clock/voltage glitching.

- **Invasive**:
  Direct modification of the chip with physical access; e.g., laser fault injection after decapsulation.

- The focus on this presentation:
  - Clock glitching.
  - Voltage glitching.
  - EMFI.

- Clock Glitching → manipulate the system clock to skip or corrupt instructions.

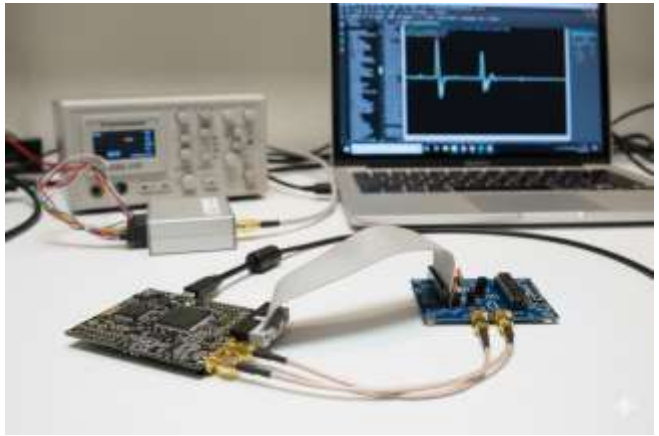
- How to perform clock glitching (tools & approaches)

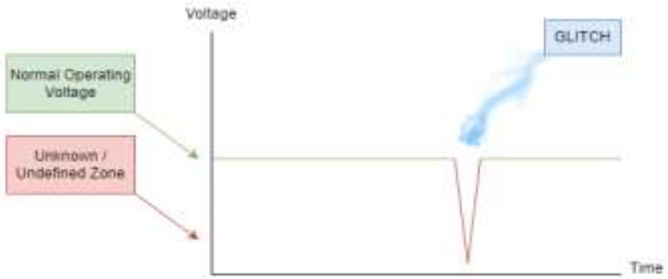


Devices: ChipWhisperer, clock generator, FPGA, oscilloscope & logic analyzer.

Methods: Inject short pulses or jitter on the clock line, replace crystal with controlled source, gate/skip clock cycles with transistor/FPGA.

Monitor: oscilloscope (clock waveform), logic analyzer (trigger timing), UART/LEDs (DUT behavior).





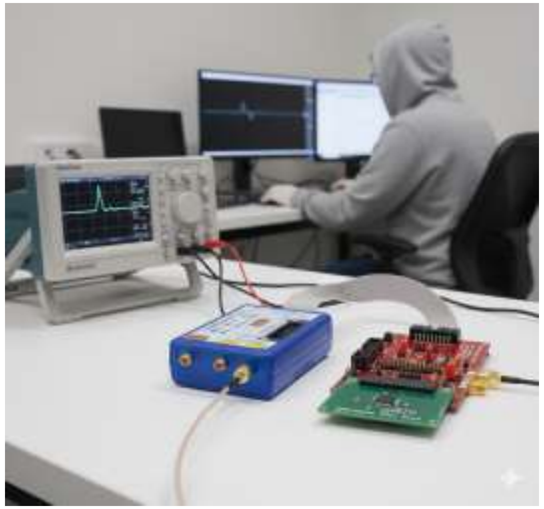
- Voltage Glitching → inject short voltage drops/spikes to disrupt logic.

- How to perform voltage glitching (tools & approaches)

Devices: ChipWhisperer, Custom Glitching Hardware, Crowbar circuit.

Methods: Short V_{DD} drop, spikes, capacitor discharge, ground transient injection, crowbar pulses, timed supply gating.

Monitor: High-speed Oscilloscopes.



- Electromagnetic Fault Injection (EMFI) → use EM pulses to induce transient faults in circuits.

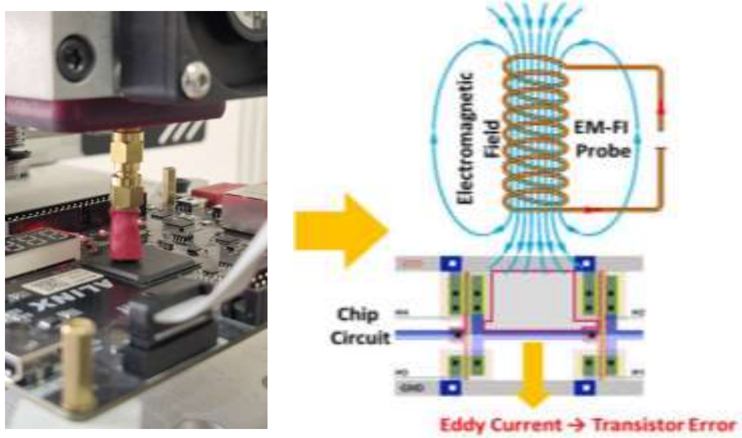
- How to perform EMFI (tools & approaches)

  Devices: ChipShouter, EM pulse guns, custom EM coils, high-voltage pulse generators, Langer near-field probes, Riscure EMFI.

  Methods: Near-field EM pulses injected into silicon, spatial scanning for sensitive spots, timed pulses during execution.

  Positioning: motorized or manual XYZ stages let you do systematic spatial scans.

  Monitor: Python scripts to log, timestamp, and classify outputs.



Key parameters in Fault injection attacks [1]

Parameters	Description
Number of Fault Injection Events (n)	The number of physical fault injections performed in a specified time window
Fault Location (l)	**Precise:** Specific gate(s) **Loose:** Specific cluster of gates, no/partial control on which gates are affected **No control:** Random location
Fault Timing (t)	**Precise:** Specific clock cycle/operation **Loose:** Set of clock cycles/operations **No control:** Random timing
Number of Affected Bits (b)	The number of affected bits by the fault injection
Duration of the Injected Fault (d)	**Transient:** Limited, self-recoverable **Persistent:** Limited, needs to be explicitly overwritten **Destructive:** Irreversible
Targeted Gate Type (g)	**Combinational** gates only **Memory** gates only **Both**
Fault Type (p)	**Set:** Faulting to 1 **Reset:** Faulting to 0 **Random:** Random outcome **Flip:** Flipping the value **Custom:** Attacker specified gate modification

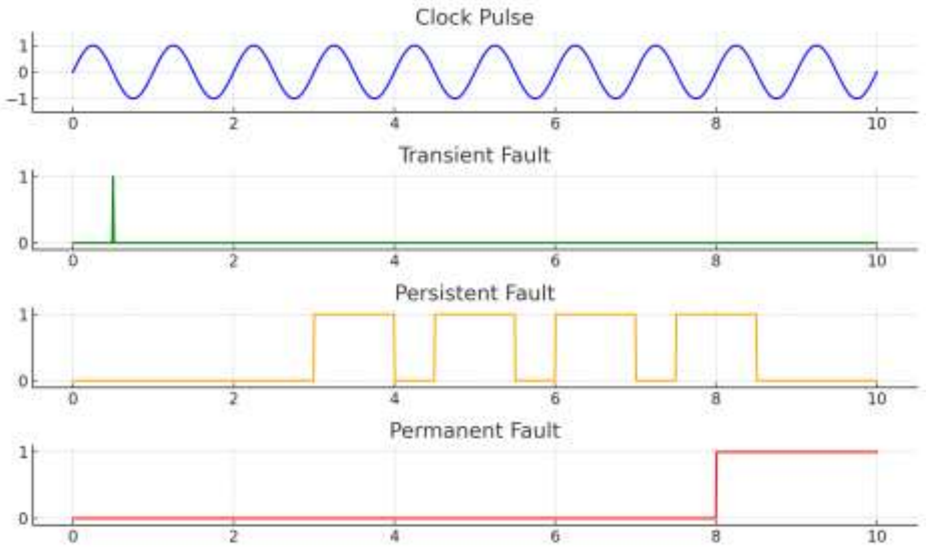
Fault Analysis

Fault Analysis Methods

- Classifications of Fault model: Duration and impact

- How long it will stay:
  - Transient Fault: Affect one encryption
  - Permanent Fault: Always present
  - Persistent Fault: Hybrid model between transient and permanent. Persist over several encryptions but disappears on reboot. Typically targets stored constants (ex. S-box in memory)

- What it will affect:
  - Modification of operation flow
  - Changing a specific bit/ byte/ word or random
  - Which round, a specific round or random



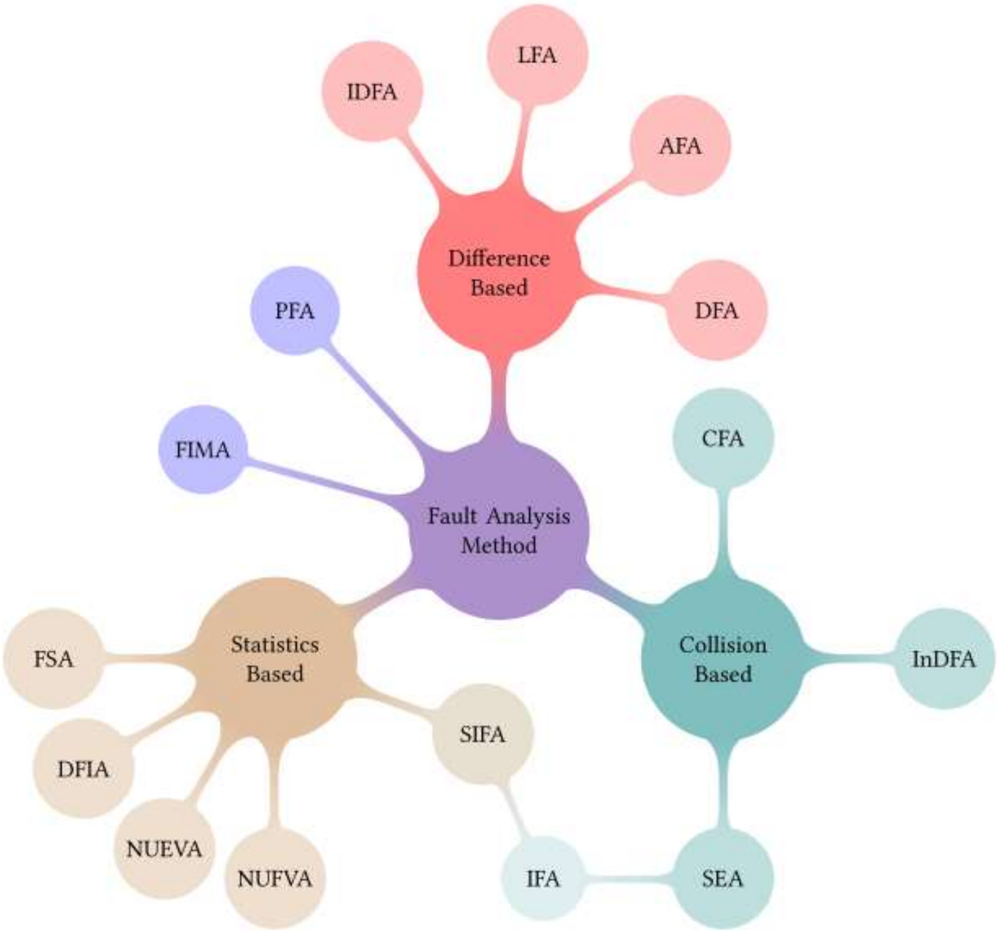
Fault Analysis Methods

❑ Cryptanalysis approaches

Different models of FA from Cryptanalysis point of view

- Differential fault analysis (DFA)
- Statistical fault analysis (SFA, SIFA, SEFA, …)
- Persistent fault analysis (PFA, MPFA, SIPFA)
- …

Fault Analysis Methods



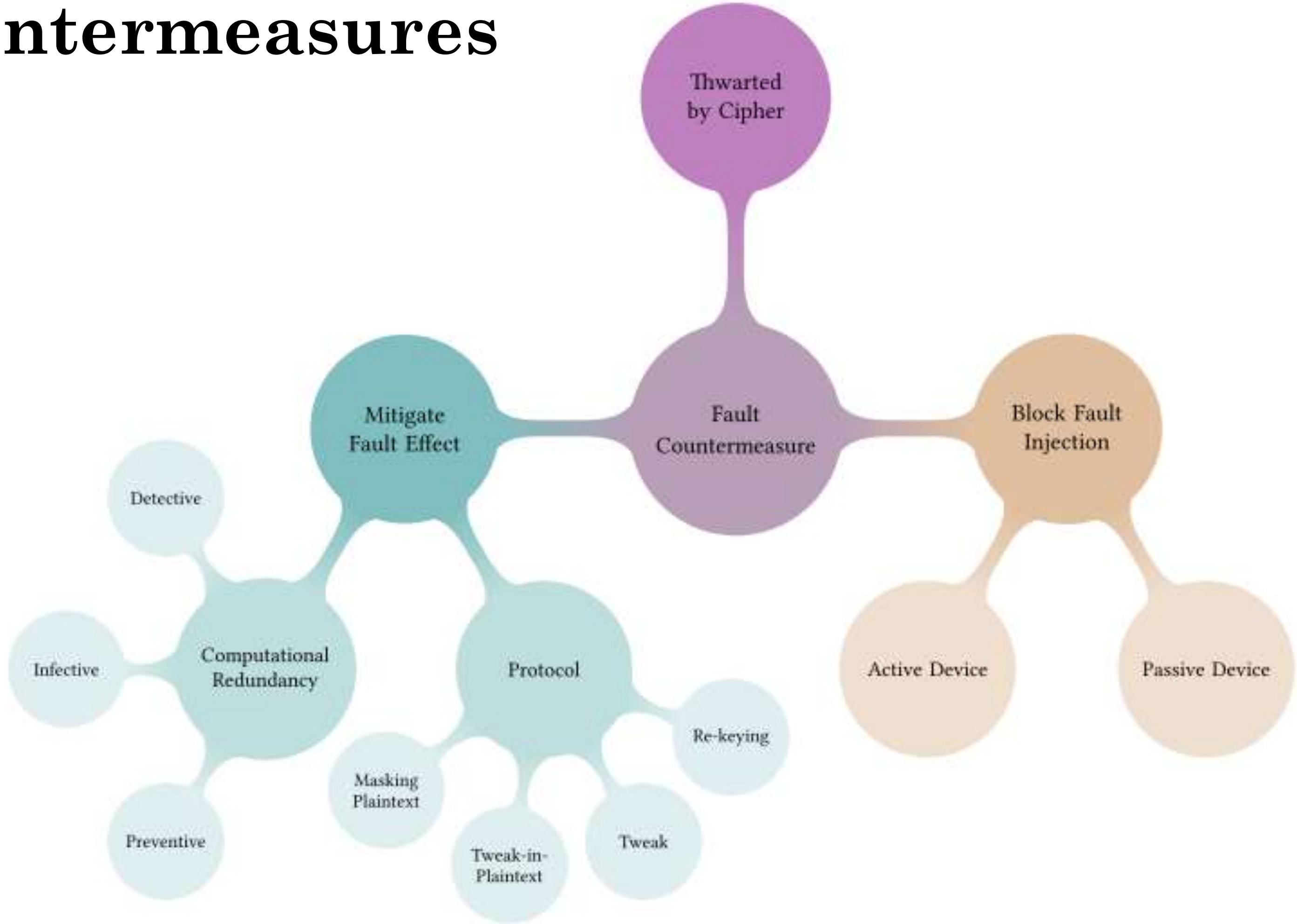
Classification of fault analysis methods. [2]

Fault Countermeasures

Fault Countermeasures

- Redundancy based implementation
  - Detection
  - Correction
  - Infection
  - Cipher-based
- **Physical Circuit**
  - **Shield**
  - **Physical Detector**
  - **Tamper Detector**
  - **Package**
- **Protocol**
  - **Re-keying**
  - **Masking plaintext**
  - **Tweak**
  - **Padding**

Fault Countermeasures

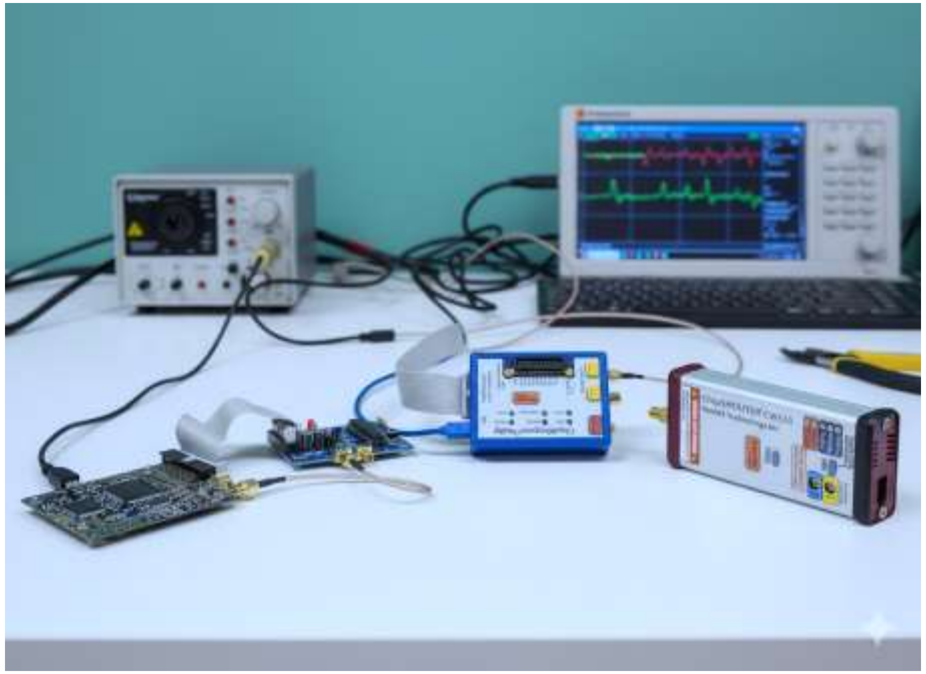


Generic overview of fault attack countermeasures. [2]

Case studies

❑ NewAE experimental setup:

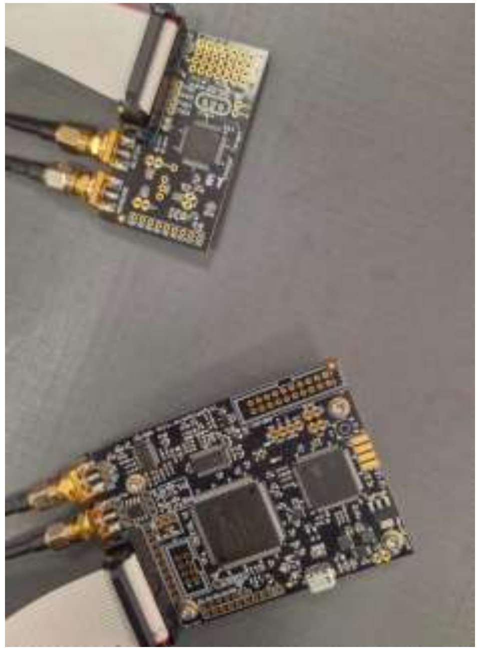
- Chipshouter
- Chipwhisperer lite
- Chipwhisperer Husky
- CW305 Target board



❑ Our attack scenarios:

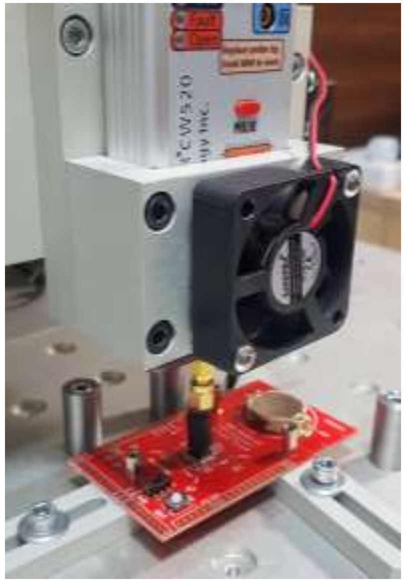
- Using Chipwhisperer lite to make Voltage glitch on STM32 target.
- Using Chipwhisperer Husky to make Clock glitch on cw305 Target board.
- Using Chipshouter with its original Target

- ❑ **Using Chipwhisperer lite to make Voltage glitch on STM32 target.**



- Using Chipwhisperer Husky to make Clock glitch on cw305 Target board.

- **Using Chipshouter with its [original Target]**



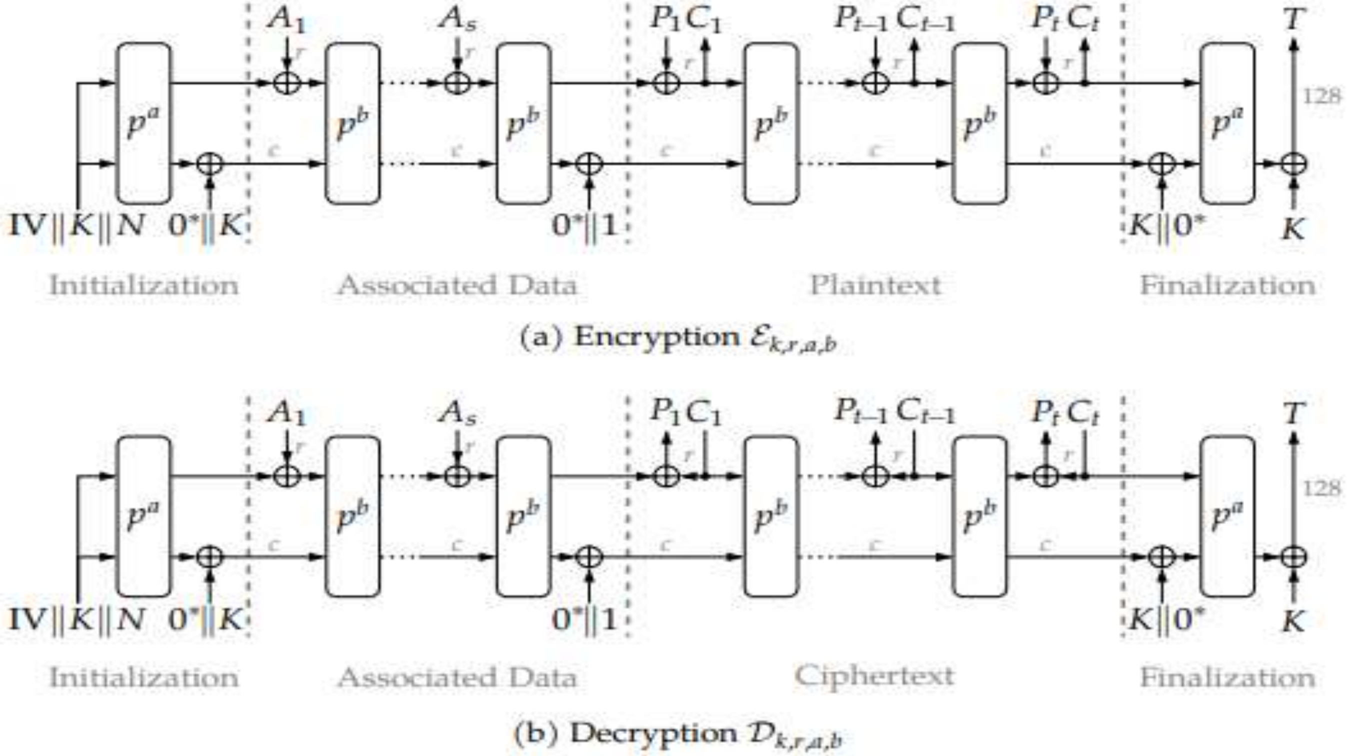
❑ Our experimental setup:

- Chipshouter
- Artix7 Xilinx board.
- STM32f405 development board.

❑ Our attack scenarios:

- Using Chipshouter to make ASCON algorithm output faulty on:
  1. Artix7 FPGA
  2. STM32f405 ARM

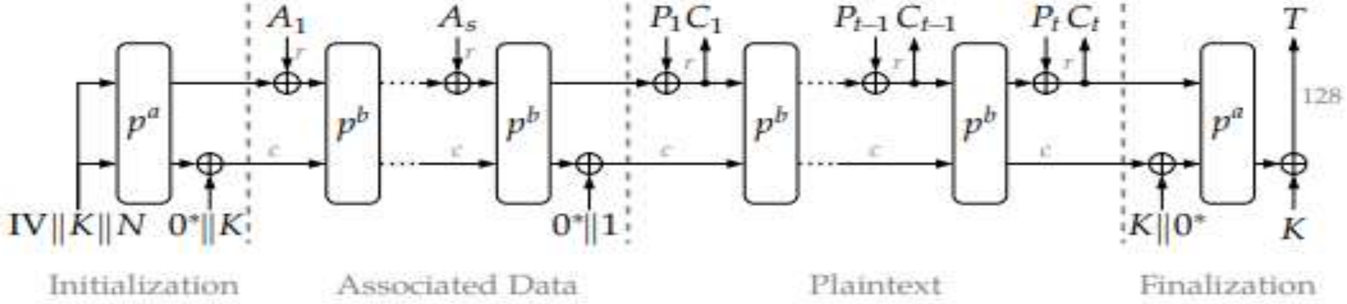
- ❑ **Using Chipshouter to make ASCON algorithm output faulty on ARTIX7.**



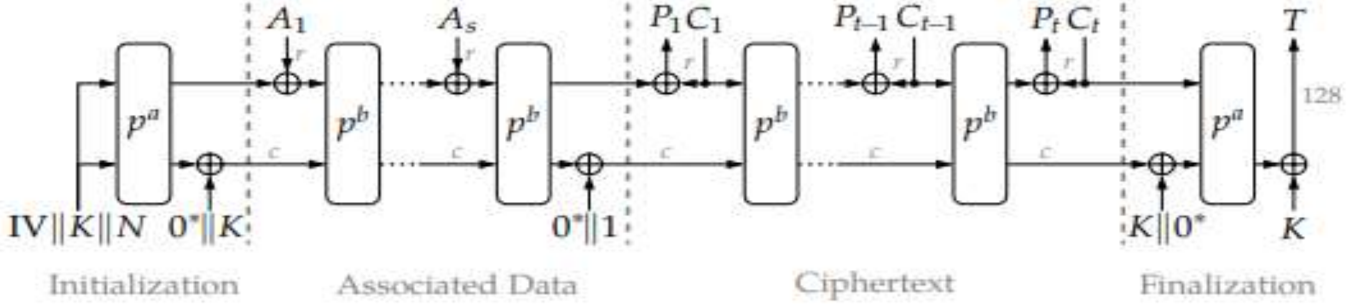
(a) Encryption $\mathcal{E}_{k,r,a,b}$

(b) Decryption $\mathcal{D}_{k,r,a,b}$

Name	Algorithm	Bit size of				Rounds	
		key	nonce	tag	data block	p^a	p^b
Ascon-128	$\text{Ascon}_{12,6}$-128-64	128	128	128	64	12	6



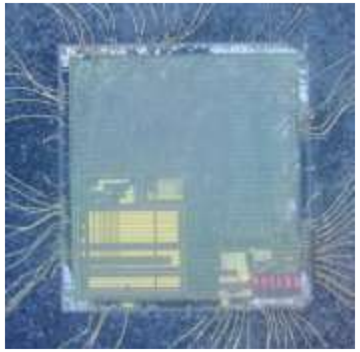
(a) Encryption $\mathcal{E}_{k,r,a,b}$



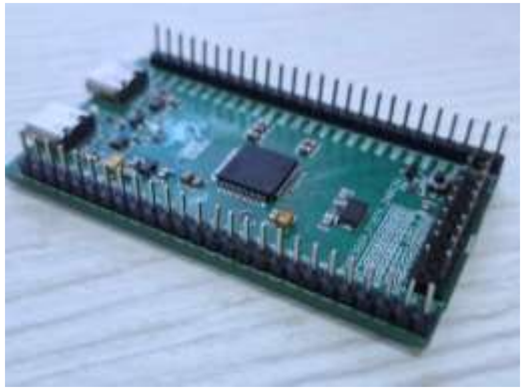
(b) Decryption $\mathcal{D}_{k,r,a,b}$

Name	Algorithm	Bit size of				Rounds	
		key	nonce	tag	data block	p^a	p^b
Ascon-128	Ascon$_{12,6}$-128-64	128	128	128	64	12	6

- ❑ **Using Chipshouter to make ASCON algorithm output faulty on ARTIX7.**



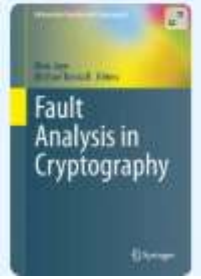
- ❑ **Using Chipshouter to make ASCON algorithm output faulty on STM32F405.**

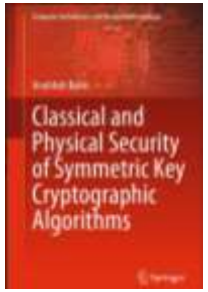


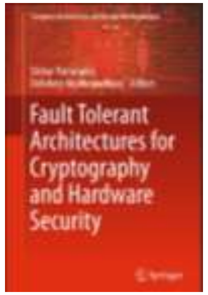
Further Resources

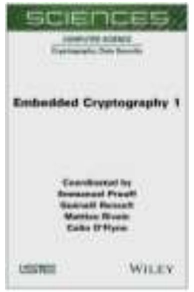
❑ Useful Books:

- **The Hardware Hacking Handbook: Breaking Embedded Security with Hardware Attacks**, by Colin O'Flynn and Jasper van Woudenberg.
- **Fault Analysis in Cryptography**, by Michael Tunstall, Marc Joye.
- **Hardware Security Training, Hands-on!** by Farimah Farahmandi, Mark Tehranipoor, and N. Nalla Anandakumar.
- **Hardware Security Primitives,** by Mark Tehranipoor , Nitin Pundir , Nidish Vashistha , Farimah Farahmandi.
- **Introduction to Hardware Security and Trust,** by Mark Tehranipoor
- **Classical and Physical Security of Symmetric Key Cryptographic Algorithms,** by Anubhab Baksi.
- **Fault Tolerant Architectures for Cryptography and Hardware Security,** by Sikhar Patranabis, Debdeep Mukhopadhyay
- **Embedded Cryptography 1,** Emmanuel Prouff, Guénaël Renault, Mattieu Rivain, and Colin O'Flynn.









❑ Major Academic Platforms:

➢ Where to Publish: top journals and conferences

- IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)
- Journal of Cryptographic Engineering (Springer).
- Journal of Hardware and Systems Security (Springer).
- IEEE Transactions on Information Forensics and Security (TIFS).
- IEEE Transactions on Dependable and Secure Computing (TDSC).
- IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD).

- CHES — Cryptographic Hardware and Embedded Systems (IACR).
- IEEE HOST — Hardware Oriented Security and Trust.
- COSADE — Constructive Side-Channel Analysis and Secure Design.
- FDTC — Fault Diagnosis and Tolerance in Cryptography.
- CARDIS — Smart Card Research & Advanced Applications.
- DATE — Design, Automation & Test in Europe.
- DAC — Design Automation Conference.

➢ Who's leading the research: Top labs around world

- IAIK, Graz University of Technology, Austria.
- The Labs of "EnICS Labs", Israel.
- Ruhr-Universitat Bochum, Germany.
- Cryptographic Engineering Research Group (CERG), George Mason University, USA.
- Hardware & Embedded Systems Lab (HESL), Nanyang Technological University of Singapore.

- Computer Security and Cryptography Group, ETH Zurich
- Hardware Security and Cryptographic Processor Lab, Tsinghua University, Beijing, China
- Keysight Technologies Netherlands Riscure BV.
- Secure-IC, France.
- Fraunhofer Institute for Applied and Integrated Security.

Conclusion

- Fault injection reveals **hardware vulnerabilities** missed by normal testing.
- Fault analysis methods help make meaningful sense of complex faulty outputs.
- After identifying weaknesses, applying suitable fault countermeasures is the best way to strengthen the system.
- Practical fault injection via Clock glitch, voltage glitch, and EMFI.
- Practical tools like **Chipshouter** and **Chipwhisperer** make these attack accessible for research.

• Key Inspired References

[1] Dilara Toprakhisar , Svetla Nikova , and Ventzislav Nikov, “SoK: Parameterization of Fault Adversary Models Connecting Theory and Practice”, Springer Nature Switzerland AG 2024 E. Oswald (Ed.): CT-RSA 2024, LNCS 14643, pp. 433–459, 2024.

[2] Anubhab Baksi and Shivam Bhasin, et al., “A Survey on Fault Attacks on Symmetric Key Cryptosystems”, *ACM Computing Surveys*, Vol. 55, No. 4, Article 86. Publication date: November 2022.

[3] Niclas Kuhnapfel, Robert Buhren. et al., “EM-Fault It Yourself: Building a Replicable EMFI Setup for Desktop and Server Hardware”, *IEEE Physical Assurance and Inspection of Electronics (PAINE).* 2022.

[4] Aghaie, A., Moradi, A., et al., “Impeccable circuits”, *IEEE Trans. Comput.* 69(3), 361–376 (2020).

[5] Carlton Shepherda, Konstantinos Markantonakisa, et al., “Physical Fault Injection and Side-Channel Attacks on Mobile Devices: A Comprehensive Analysis”, arXiv:2105.04454v6 [cs.CR] 22 Mar 2022.

[6] Jean-Max Dutertrea, Alexandre Menua, et al., “Experimental Analysis of the Electromagnetic Instruction Skip Fault Model and Consequences for Software Countermeasures”.

[7] Colin O’Flynn, “EMFI for Safety-Critical Testing of Automotive Systems”.

[8] Hao Guo, Sayandeep Saha, et al., “Vulnerability Assessment of Ciphers To Fault Attacks Using Reinforcement Learning”.

Any questions?

Thank you!