RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

CONTRIBUTIONS À LA SÉCURITÉ DES LOGICIELS EMBARQUÉS DANS LA CHAÎNE DE CONFIANCE

Soutenance d'Habilitation à Diriger les Recherches

Guillaume BOUFFARD
Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)
https://cyber.gouv.fr

1. About Me

Who am I?

2014	Ph.D. thesis at University of Limoges *A Generic Approach for Protecting Java Card Smart Card Against Software Attacks*
Since 2014	Expert in embedded software security at ANSSI 2014–2022: ANSSI > Hardware Security Lab (LSC) Since 2023: ANSSI > Hardware and Software Architectures Lab (LAM)

Université de Limoges

What is ANSSI?

ANSSI: **French Cybersecurity Agency**.

- Is the national authority in charge of cybersecurity in France.
- Reports to the SGDSN (General Secretariat for Defence and National Security) which assists the Prime Minister.

Main missions:

(`https://cyber.gouv.fr/en/what-we-do`)

- **Defending** critical information systems and the victims of large-scale cyberattacks;
- **Knowing** the state of the art in cybersecurity and cyberspace threats;
- **Sharing** knowledge, recommendations, and expertise in digital safety;
- **Assisting** the national and international ecosystem;
- **Regulating** cybersecurity organisations, goods, and services.

ANSSI: Research Labs Activities

Expertise

Training

- Design and delivery of CFSSI training courses.
- Support for awareness and outreach events.

R&D

ANSSI: My Activities

Expertise

- Technical support to internal teams and certification bodies.
- Contributions to GlobalPlatform and JHAS workgroups.
- Collaboration with EU partners (e.g., BSI, ...).
- Support to national and European projects (e.g., France Identité, EU-Digital Identity).

Training

- Design and delivery of CFSSI training courses.
- Support for awareness and outreach events.
- **Design and delivery of university courses** (in my personal time).

R&D (≈33%)

- Focus on how to protect embedded software:
  - software and hardware attacks studies.
  - design of countermeasures.
- Supervision of Ph.D. students and interns.

2. Background

The Root of Trust (RoT)

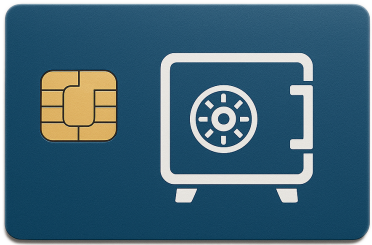
- Several functionalities must be executed in an environment that is capable of:
  - hosting sensitive apps:
    - where sensitive data is protected;
  - performing sensitive operations:
    - with no leakage.

The Root of Trust (RoT)

- Several functionalities must be executed in an environment that is capable of:
  - hosting sensitive apps:
    - where sensitive data is protected;
  - performing sensitive operations:
    - with no leakage.
- Root of Trust (RoT) is defined by GlobalPlatform [Glo18] as:
  - an element with a processing unit, code and data.
  - whose integrity **cannot be** verified.

[Glo18] GlobalPlatform. ***Root of Trust Definitions and Requirements.*** Version 1.1. June 2018 (https://globalplatform.org/wp-content/uploads/2018/07/GP_RoT_Definitions_and_Requirements_v1.1_PublicRelease-2018-06-28.pdf).

In the 2000s, Cybersecurity Relied on Smart Cards



- Tamper-resistant computing platform;
- Ubiquitous in daily life:
  - credit cards;
  - (U)SIM cards;
  - health cards (e.g., French *Carte Vitale*);
  - pay TV access cards;
  - …

The smart card is a **Secure Element** designed to act as a **hardware RoT**.

Secure Element: Minimalism for Maximum Security

Highly constrained architecture:

- Minimal hardware & software layout.
- Very limited embedded functionalities.
- Ultra-low power consumption.

Secure Element: Minimalism for Maximum Security

Highly constrained architecture:

- Minimal hardware & software layout.
- Very limited embedded functionalities.
- Ultra-low power consumption.

Security evaluations

- Resistance to **high attack potential** (Common Criteria **AVA_VAN.5** level).
- Long and rigorous process.
- Based on a few **targets of evaluation**:
  - threats and protections are clearly defined.
  - evaluations may rely on **Protection Profiles (PPs)** for common use cases.

Secure Element: Minimalism for Maximum Security

Highly constrained architecture:

- Minimal hardware & software layout.
- Very limited embedded functionalities.
- Ultra-low power consumption.

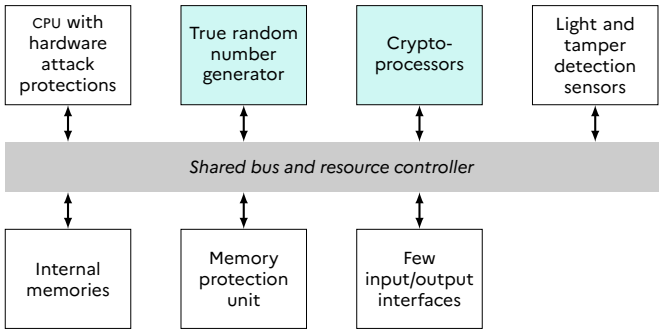
Security evaluations

- Resistance to **high attack potential** (Common Criteria **AVA_VAN.5** level).
- Long and rigorous process.
- Based on a few **targets of evaluation**:
  - threats and protections are clearly defined.
  - evaluations may rely on **Protection Profiles (PPs)** for common use cases.

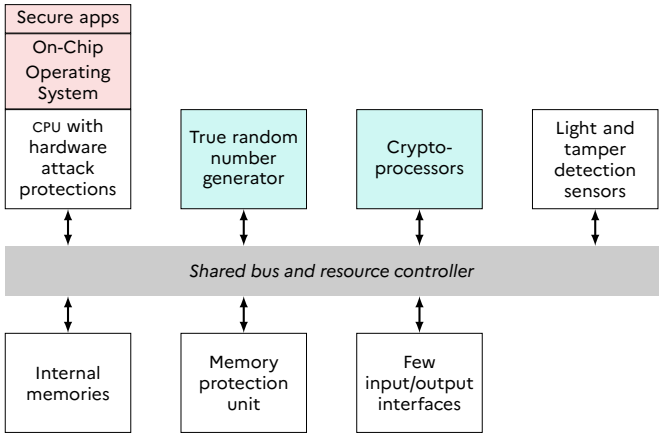
Between 2010 and 2018, more than 35 billion SEs were deployed [Glo19].

[Glo19] "6.2 Billion GlobalPlatform-Compliant Secure Elements Deployed in 2018". 2019 (https://globalplatform.org/latest-news/6-2-billion-globalplatform-compliant-secure-elements-deployed-in-2018/).

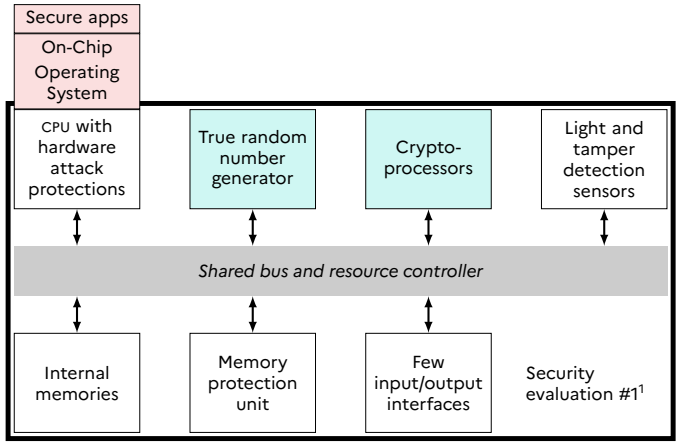
Typical Secure Element (SE) Architecture



Typical Secure Element (SE) Architecture

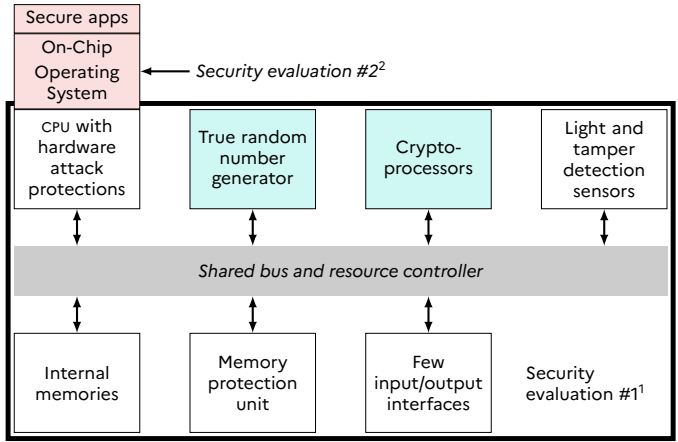


Typical Secure Element (SE) Architecture



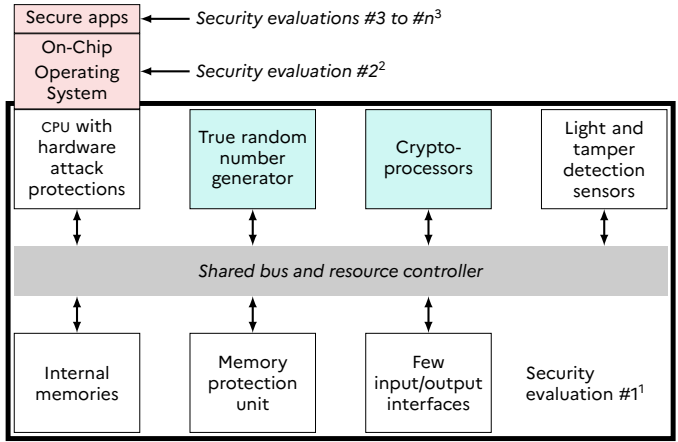
[1] **SE PP [Eur14] or embedded SE PP [Eur22].**

Typical Secure Element (SE) Architecture



[1] **SE PP [Eur14] or embedded SE PP [Eur22].**
[2] **Java Card PP [Ora21].**

Typical Secure Element (SE) Architecture



[1] **SE PP [Eur14] or embedded SE PP [Eur22].**
[2] **Java Card PP [Ora21].**
[3] **Secure apps' PP: (U)SIM [Rad10], identity [Sic12], payment [BS10], tachograph [Cen17], …**

Secure Element: Limitations

- SE are designed with a minimal attack surface:
  - only **one app** running at a time.
  - very **limited interfaces and resources**.
- Strong isolation

Secure Element: Limitations

- SE are designed with a minimal attack surface:
  - only **one app** running at a time.
  - very **limited interfaces and resources**.
- Strong isolation, but at the cost of:
  - **low performance**.
  - **limited extensibility**.
  - **restricted developer access**.

Secure Element: Limitations

- SE are designed with a minimal attack surface:
  - only **one app** running at a time.
  - very **limited interfaces and resources**.
- Strong isolation, but at the cost of:
  - **low performance**.
  - **limited extensibility**.
  - **restricted developer access**.

Not suitable for modern use cases requiring both **security** and **rich functionality**:

- Secure biometric authentication + encrypted storage + remote attestation.
- Running multiple secure services in parallel (e.g., payments + identity + DRM).

SEs need to be complemented with **more flexible secure environments**.

Trusted Execution Environment (TEE)

Initially designed as a performance-oriented emulation of a **hardware RoT**.

A Trusted Execution Environment (TEE) [Glo22] is an execution environment that:

- Mixes **security and performance** for sensitive applications;
- Runs only sensitive applications signed by a trusted entity;
- Ensures resistance to **software attacks** and certain **hardware attacks** [Glo20].

[Glo20] GlobalPlatform. ***TEE Protection Profile***. GPD_SPE_021. Version 1.3. July 2020 (https://globalplatform.org/specs-library/tee-protection-profile-v1-3/).
[Glo22] GlobalPlatform. ***TEE System Architecture***. Version 1.3. May 2022 (https://globalplatform.org/specs-library/tee-system-architecture/).

Rich Execution Environment (REE)

Designed for rich functionalities with direct access to system resources.

The Rich Execution Environment (REE) is an execution environment that:

- Runs a **standard Operating System (OS)** designed to support a wide range of devices;
  - primarily **functionality- and performance-oriented**;
- Hosts applications from **multiple sources**; (app stores, Internet, etc.)

Provides limited isolation guarantees, compared to a TEE.

Hardware
Root of Trust (RoT)

Hardware
Root of Trust (RoT)

Hardware
Root of Trust (RoT)

Trusted Execution
Environment (TEE)

Hardware Root of Trust (RoT)

Trusted Execution Environment (TEE)

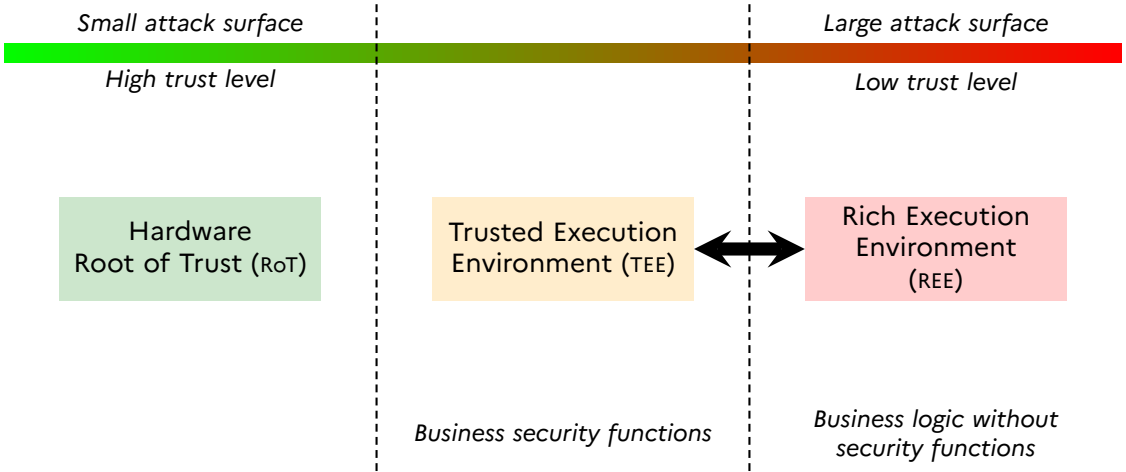
Rich Execution Environment (REE)

Hardware Root of Trust (RoT)

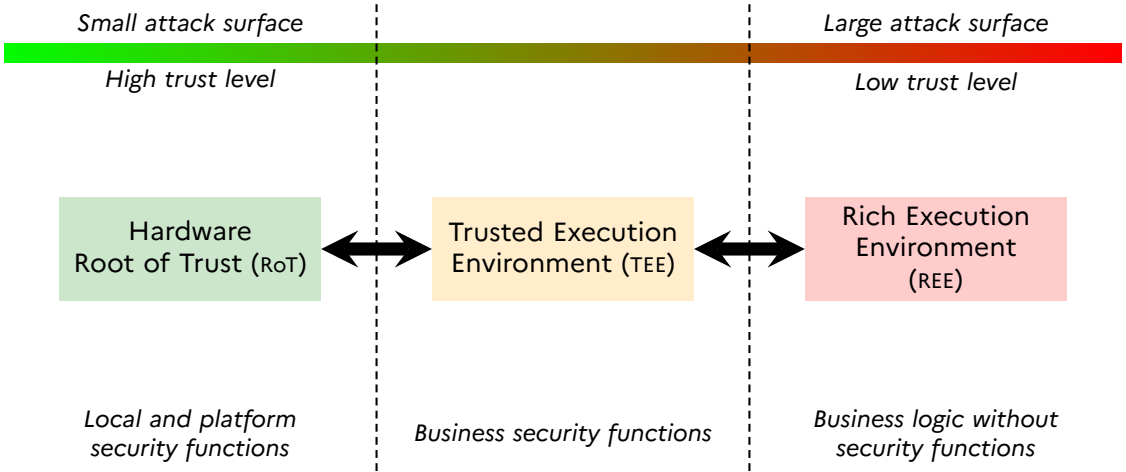
Trusted Execution Environment (TEE)

Rich Execution Environment (REE)

Business logic without security functions

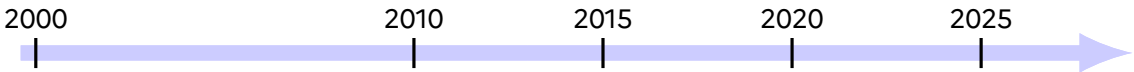

Small attack surface
Large attack surface
High trust level
Low trust level
Hardware
Root of Trust (RoT)
Trusted Execution
Environment (TEE)
Rich Execution
Environment
(REE)
Business security functions
Business logic without
security functions

The Chain of Trust (CoT)

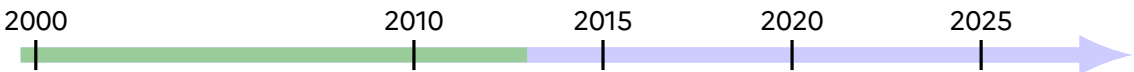


3. My Research Activities

Research Context

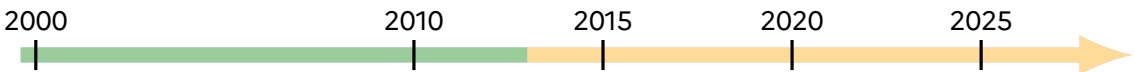


Research Context



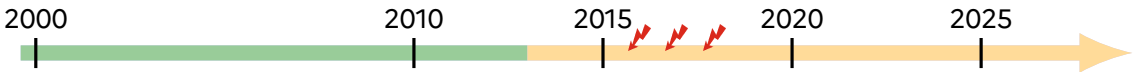
- Cybersecurity relies on SE:
  - SEs are designed for specific use-cases.

Research Context



- Cybersecurity relies on SE:
  - SEs are designed for specific use-cases.
- Sensitive operations increasingly moved to TEEs

Research Context

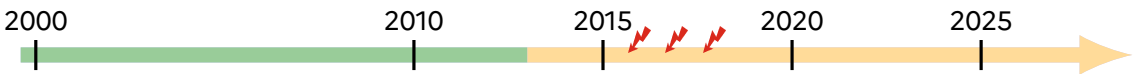


- Cybersecurity relies on SE:
  - SEs are designed for specific use-cases.
- Sensitive operations increasingly moved to TEEs

Transposition of hardware attacks from SE to TEEs [Vas+17; YSW18] and to REEs [Bos+16].

[Bos+16] "Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough", *CHES 2016*.
[Vas+17] "Laser-Induced Fault Injection on Smartphone Bypassing the Secure Boot", *FDTC 2017*.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

Research Context



- Cybersecurity relies on SE:
  - SEs are designed for specific use-cases.
- Sensitive operations increasingly moved to TEEs

Transposition of hardware attacks from SE to TEEs [Vas+17; YSW18] and to REEs [Bos+16].

Today

Digital services are ubiquitous, making **robust protection of the entire CoT** essential for trust and data security.

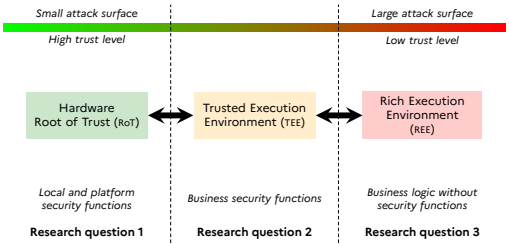
[Bos+16] "Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough", *CHES 2016*.
[Vas+17] "Laser-Induced Fault Injection on Smartphone Bypassing the Secure Boot", *FDTC 2017*.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

My Research Activities

Focusing on **embedded softwares security**

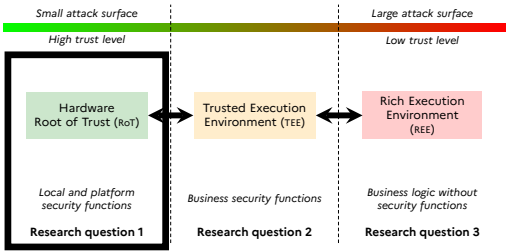
My Research Activities

Focusing on **embedded softwares security**, my research explores how security functions can be migrated from **hardware RoT** to more **powerful and versatile environments**.



My Research Activities

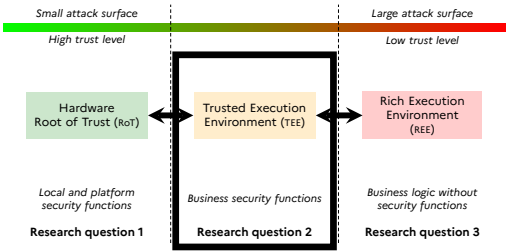
Focusing on **embedded softwares security**, my research explores how security functions can be migrated from **hardware RoT** to more **powerful and versatile environments**.



1 **How are *local and platform security functions* developed and used to enhance security?**

My Research Activities

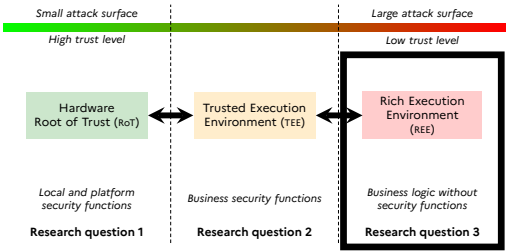
Focusing on **embedded softwares security**, my research explores how security functions can be migrated from **hardware RoT** to more **powerful and versatile environments**.



1. **How are *local and platform security functions* developed and used to enhance security?**
2. **How to achieve high security in TEEs for *business security functions*?**

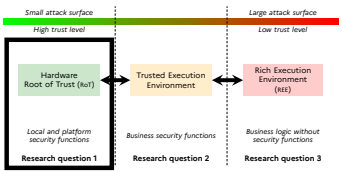
My Research Activities

Focusing on **embedded softwares security**, my research explores how security functions can be migrated from **hardware RoT** to more **powerful and versatile environments**.



1. **How are *local and platform security functions* developed and used to enhance security?**
2. **How to achieve high security in TEEs for *business security functions*?**
3. **How can sensitive apps run securely in the REE?**

3. My Research Activities



Research question 1:
How are *local and platform security functions*, provided by hardware RoT, developed and used to enhance security?

Secure Elements as Hardware RoTs

SE are the most widely deployed hardware RoTs worldwide.

- Resistance to **high attack potential**. (Common Criteria **AVA_VAN.5**)

My early research started on **closed SEs**:

- Both software and hardware implementations are proprietary and closed source.
- Focus on existing **software implementations** to understand design choices:
  - secure applications mostly studied by the community. (EMVCo [AM14; BST21], (U)SIM [Sec25])

[AM14] "EMV: why payment systems fail", *Communication of ACM 2014*.
[BST21] "The EMV Standard: Break, Fix, Verify", *S&P 2021*.
[Sec25] "eSIM security". 2025 (`https://security-explorations.com/esim-security.html`).

Secure Elements as Hardware RoTs

SE are the most widely deployed hardware RoTs worldwide.

- Resistance to **high attack potential**. (Common Criteria **AVA_VAN.5**)

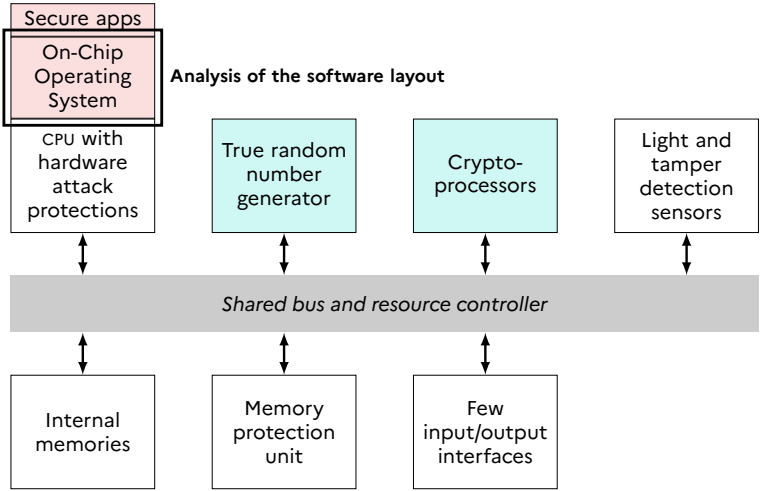
My early research started on **closed SEs**:

- Both software and hardware implementations are proprietary and closed source.
- Focus on existing **software implementations** to understand design choices:
  - secure applications mostly studied by the community. (EMVCo [AM14; BST21], (U)SIM [Sec25])

Focus on the **software stack beneath secure applications**.

[AM14] "EMV: why payment systems fail", *Communication of ACM 2014*.
[BST21] "The EMV Standard: Break, Fix, Verify", *S&P 2021*.
[Sec25] "eSIM security". 2025 (`https://security-explorations.com/esim-security.html`).

Contributions to Hardware RoT Security



The On-Chip Operating System

Most of the on-chip OS embedded in SE includes:

- A minimal and hardened OS;
- A Java Card Virtual Machine (JCVM). (≈ 150 Common Criteria certified products per year)
  - 6 billion devices embed a JCVM are deployed annually [Pas22].

[Pas22] "Oracle Celebrates the Java Card Forum's 25th Anniversary". 2022 (`https://blogs.oracle.com/java/post/java-card-forum-25-years-anniversary`).

The On-Chip Operating System

Most of the on-chip OS embedded in SE includes:

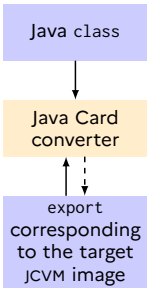
- A minimal and hardened OS;
- A Java Card Virtual Machine (JCVM). (≈ 150 Common Criteria certified products per year)
  - 6 billion devices embed a JCVM are deployed annually [Pas22].

The Java Card technology provides:

- A development environment to build secure applications;
- A platform-independent runtime environment;
- A multiple-applicative environmnent;
- A strong application isolation.

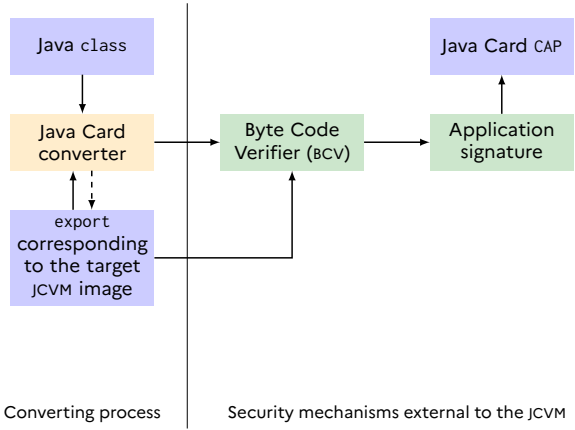
[Pas22] "Oracle Celebrates the Java Card Forum's 25th Anniversary". 2022 (`https://blogs.oracle.com/java/post/java-card-forum-25-years-anniversary`).

Analysis of the Java Card Platform Security

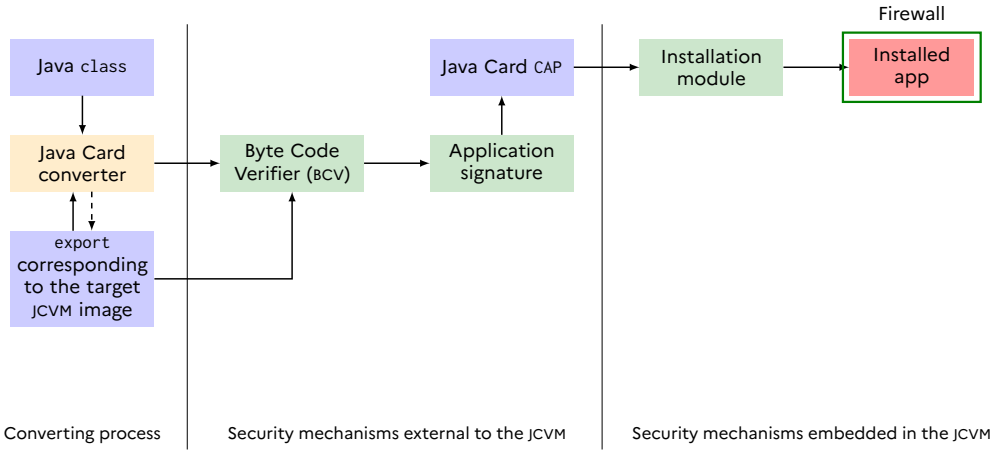


Converting process

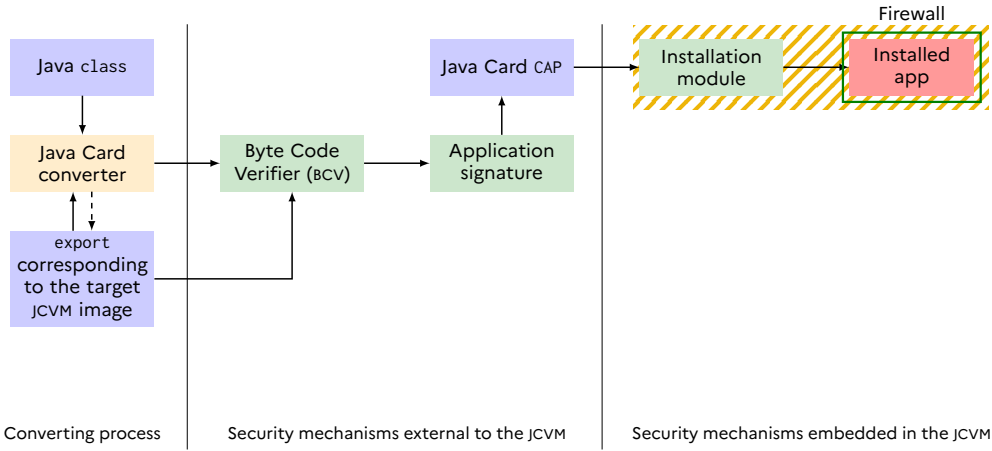
Analysis of the Java Card Platform Security



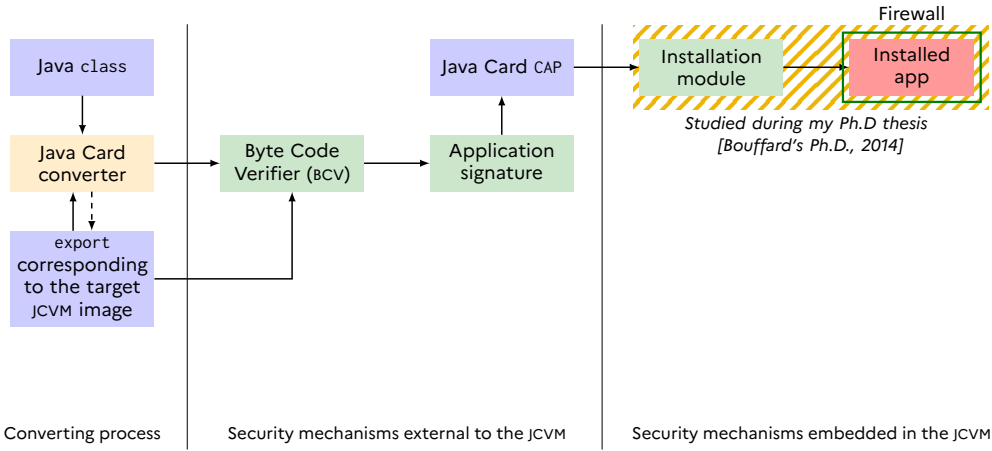
Analysis of the Java Card Platform Security



Analysis of the Java Card Platform Security

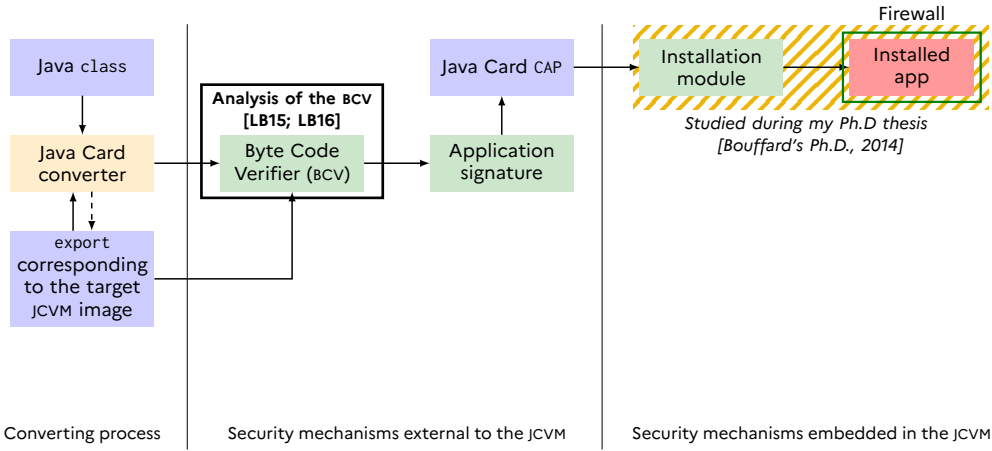


Analysis of the Java Card Platform Security



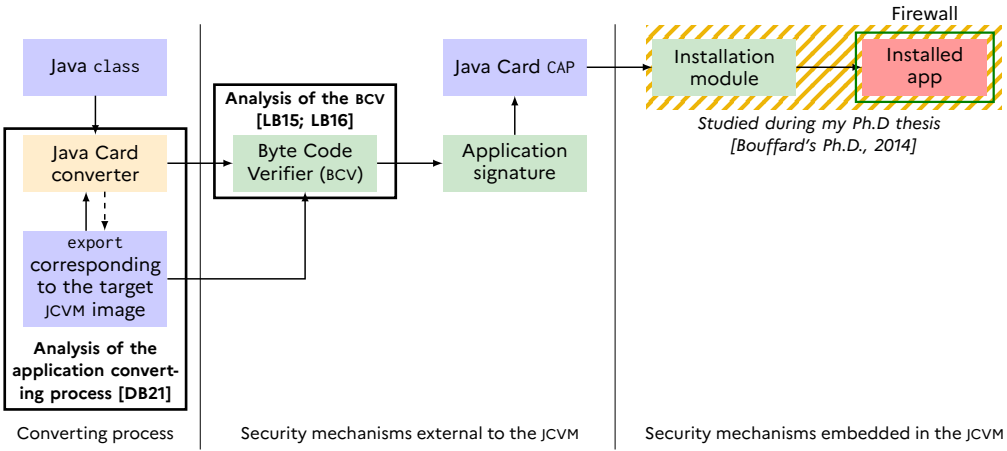
[Bouffard's Ph.D., 2014] "A Generic Approach for Protecting Java Card Smart Card Against Software Attacks", Université de Limoges.

Analysis of the Java Card Platform Security



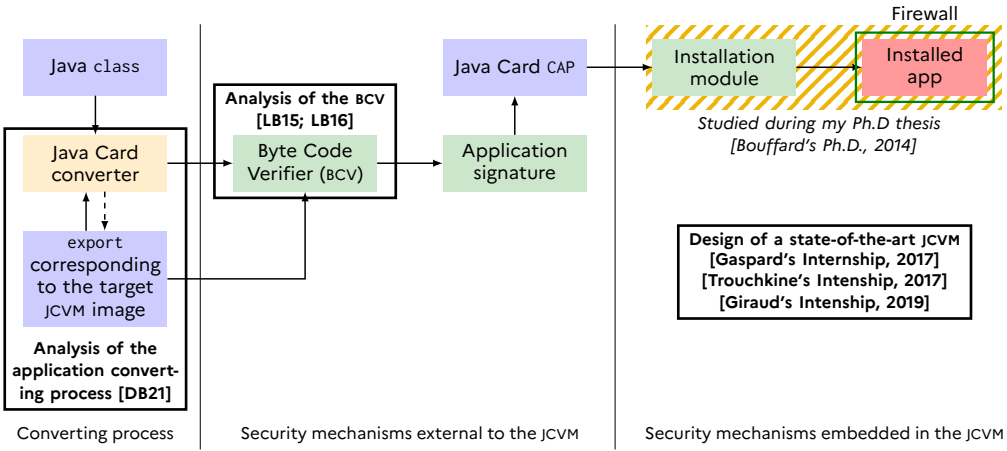
[LB15] "Java Card Virtual Machine Compromising from a Bytecode Verified Applet", *CARDIS 2015*.
[LB16] "Fuzzing and Overflows in Java Card Smart Cards", *SSTIC 2016*.

Analysis of the Java Card Platform Security



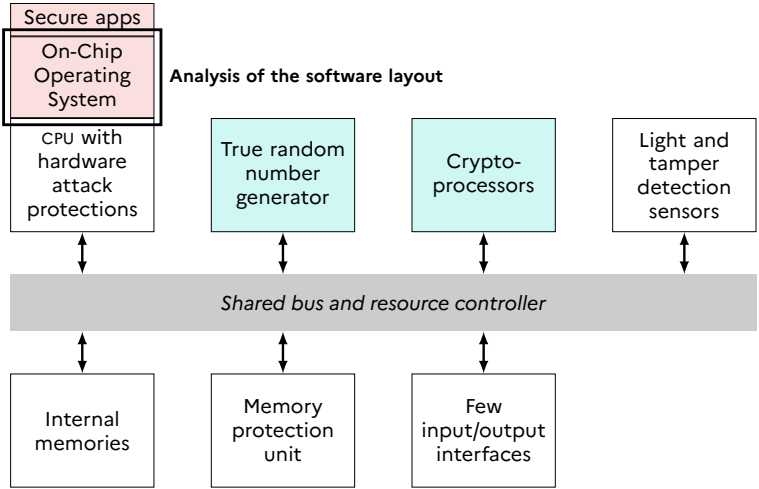
[DB21] "PhiAttack - Rewriting the Java Card Class Hierarchy", *CARDIS 2021*.
[LB15] "Java Card Virtual Machine Compromising from a Bytecode Verified Applet", *CARDIS 2015*.
[LB16] "Fuzzing and Overflows in Java Card Smart Cards", *SSTIC 2016*.

Analysis of the Java Card Platform Security

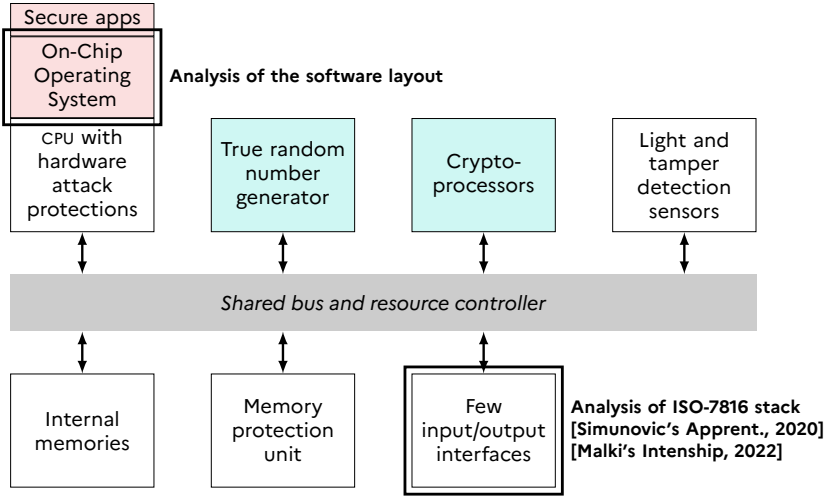


[Gaspard's Intenship, 2017] "Implementation of a Secure Operating System for Java Card-based Secure Element", École Polytechnique.
[Giraud's Intenship, 2019] "Secure Implementation of GlobalPlatform for Java Card Platform", INSA.
[Trouchkine's Intenship, 2017] "Hardware Implementation of a Java Card Virtual Machine", École des Mines de Saint-Étienne.

Contributions to Hardware RoT Security (cont.)

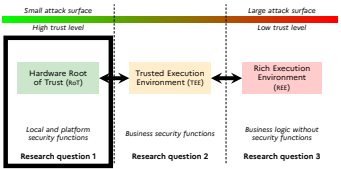


Contributions to Hardware RoT Security (cont.)



[Malki's Intership, 2022] "Fingerprinting of Embedded Software Implementation", École 42.
[Simunovic's Apprent., 2020] "Security Analysis of the ISO-7816 Stack", ESISAR.

Research Question 1: Summary of Contributions



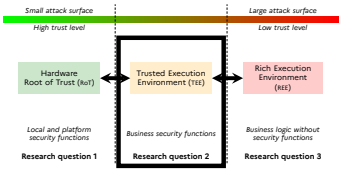
Research question 1:
How are *local and platform security functions*, provided by hardware RoT, developed and used to enhance security?

SEs = strongest Hardware RoTs but hardware & software are closed and proprietary.

Research focus: **embedded software security** without access to target internals.

1. Anticipated risks from **misused tools and environments**
   - by studying deployed JCVM implementation and toolchains.
2. Designed a **state-of-the-art JCVM**
   - minimizing reliance on external elements to strengthen implementation security.
3. Evaluated the security of **communication interfaces**
   - by uncovering leakage and fingerprinting opportunities in deployed implementations.

3. My Research Activities



Research question 2:
How to achieve high security in TEEs for *business security functions*?

TEE Security: Requirements and Challenges

The TEE must be both high-performance and secure area

- Deployed in application SoCs (> 2015)
- Security evaluations:
  - PP for TEE [Glo20]
  - Resistance to **Basic** or **Enhanced-Basic** **attack potential** (Common Criteria **AVA_VAN.2** or **3** level)

[Glo20] GlobalPlatform. ***TEE Protection Profile***. GPD_SPE_021. Version 1.3. July 2020 (https://globalplatform.org/specs-library/tee-protection-profile-v1-3/).

TEE Security: Requirements and Challenges

The TEE must be both high-performance and secure area

- Deployed in application SoCs (> 2015)
- Security evaluations:
  - PP for TEE [Glo20]
  - Resistance to **Basic** or **Enhanced-Basic** **attack potential** (Common Criteria **AVA_VAN.2** or **3** level)

The TEE PP requires:

- Resistance to software attacks
- Resistance to hardware attacks

[Glo20] GlobalPlatform. ***TEE Protection Profile***. GPD_SPE_021. Version 1.3. July 2020 (https://globalplatform.org/specs-library/tee-protection-profile-v1-3/).

TEE Security: Requirements and Challenges

The TEE must be both high-performance and secure area

- Deployed in application SoCs (> 2015)
- Security evaluations:
  - PP for TEE [Glo20]
  - Resistance to **Basic** or **Enhanced-Basic** **attack potential** (Common Criteria **AVA_VAN.2** or **3** level)

The TEE PP requires:

- Resistance to software attacks => ✓ (covered in evaluations)
- Resistance to hardware attacks

[Glo20] GlobalPlatform. ***TEE Protection Profile***. GPD_SPE_021. Version 1.3. July 2020 (https://globalplatform.org/specs-library/tee-protection-profile-v1-3/).

TEE Security: Requirements and Challenges

The TEE must be both high-performance and secure area

- Deployed in application SoCs (> 2015)
- Security evaluations:
  - PP for TEE [Glo20]
  - Resistance to **Basic** or **Enhanced-Basic** **attack potential** (Common Criteria **AVA_VAN.2** or **3** level)

The TEE PP requires:

- Resistance to software attacks **=>** ✓ (covered in evaluations)
- Resistance to hardware attacks **=> ?**

[Glo20] GlobalPlatform. ***TEE Protection Profile***. GPD_SPE_021. Version 1.3. July 2020 (https://globalplatform.org/specs-library/tee-protection-profile-v1-3/).

TEE Security: Requirements and Challenges

The TEE must be both high-performance and secure area

- Deployed in application SoCs (> 2015)
- Security evaluations:
  - PP for TEE [Glo20]
  - Resistance to **Basic** or **Enhanced-Basic** **attack potential** (Common Criteria **AVA_VAN.2** or **3** level)

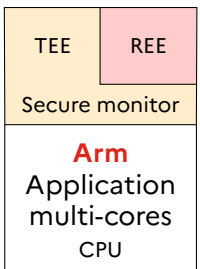
The TEE PP requires:

- Resistance to software attacks **=>** ✓ (covered in evaluations)
- Resistance to hardware attacks **=> ?**

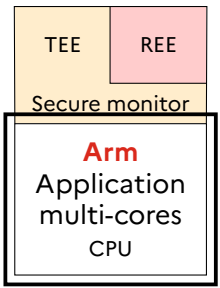
In this work, I focus on Arm TrustZone. (99% of the deployed mobile CPUs [Kin24])

[Glo20] GlobalPlatform. ***TEE Protection Profile.*** GPD_SPE_021. Version 1.3. July 2020 (`https://globalplatform.org/specs-library/tee-protection-profile-v1-3/`).
[Kin24] ***Arm Stock: AI Chip Favorite Is Overpriced, Forbes 2024.***

TEE Architecture on Arm Application CPUs



TEE Architecture on Arm Application CPUs



Impact analysis of hardware attacks on **application CPU**:

- Prior work confirmed side-channel threats on application CPUs [Bal+15; Lon+15].
- Mid-2010s, the exploitability of **fault injection** was **still debated**.
  - Unlike SEs, application CPU complexity hinders analysis.

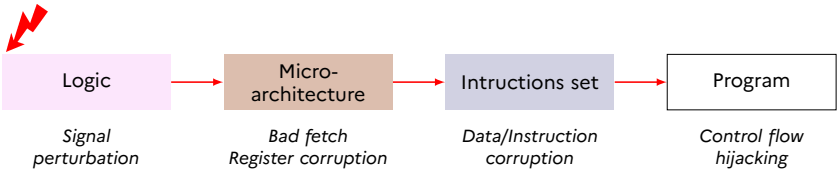
[Bal+15] "DPA, Bitslicing and Masking at 1 GHz", *CHES 2015*.
[Lon+15] "SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip", *CHES 2015*.

Fault Effects Characterization

inspired from [YSW18]

[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

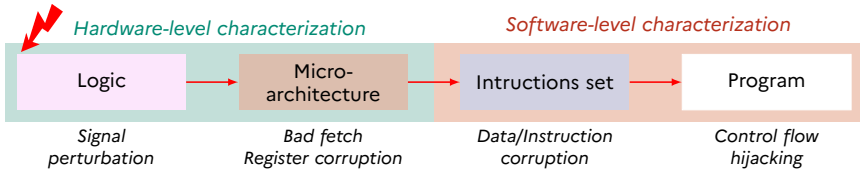
Fault Effects Characterization



inspired from [YSW18]

[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

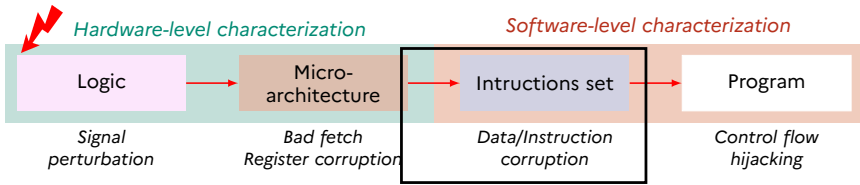
Fault Effects Characterization



inspired from [YSW18]

[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

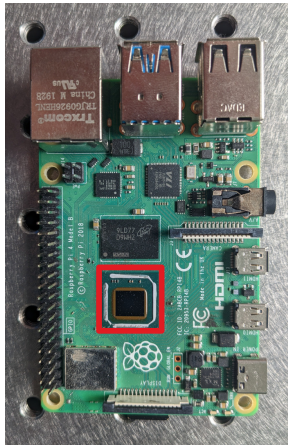
Fault Effects Characterization



inspired from [YSW18]

[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

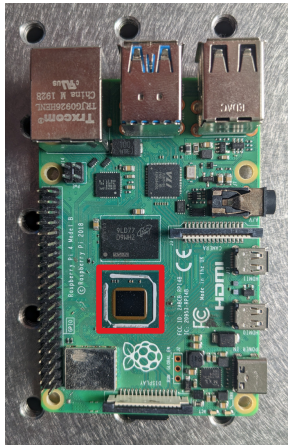
Fault Effects Analysis from ISA Point of View

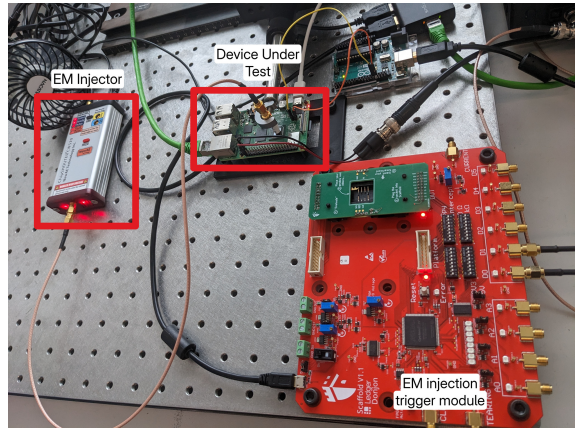


[ITB23] "Pew Pew, I'm root! De la caractérisation à l'exploitation: un voyage plein d'embûches", ***JAIF 2023.***

Fault Effects Analysis from ISA Point of View

Trouchkine's Ph.D., 2021

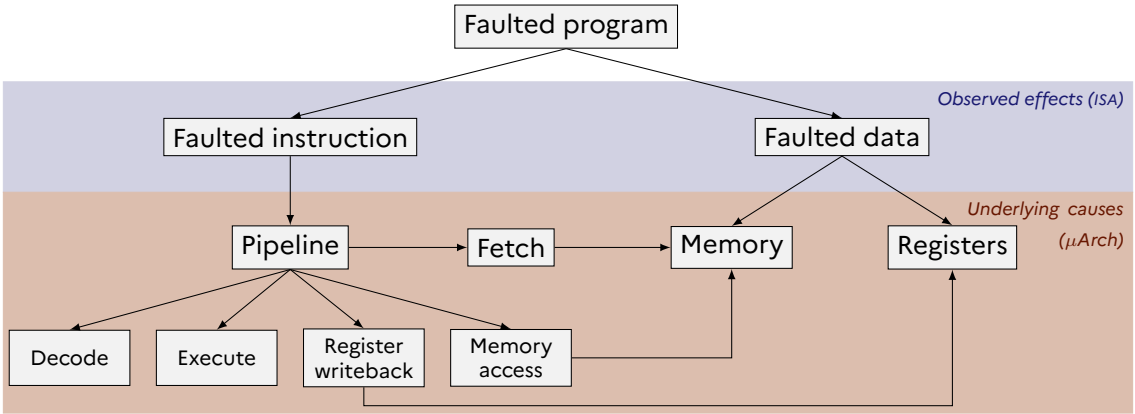




[ITB23] "Pew Pew, I'm root! De la caractérisation à l'exploitation: un voyage plein d'embûches", *JAIF 2023*.

Fault Effects Analysis from ISA Point of View

Trouchkine's Ph.D., 2021

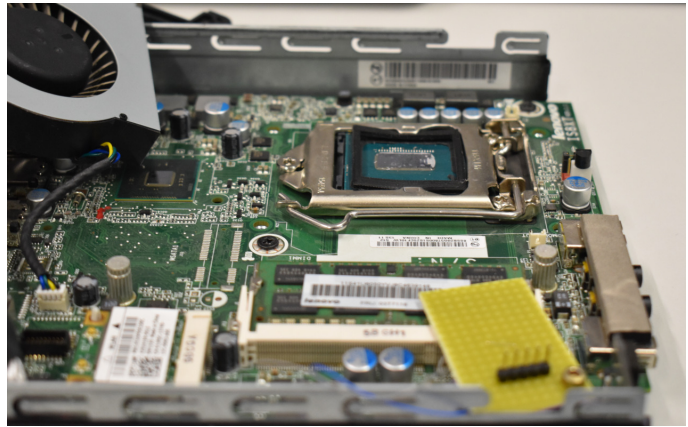


[TBC19] "Fault Injection Characterization on Modern CPUs", *WISTP 2019.*

[Tro+21] "Electromagnetic fault injection against a complex CPU, toward new micro-architectural fault models", *JCEN 2021.*

Fault Effects Analysis from ISA Point of View

Architecture-agnostic approach [TBC21]



[TBC21] "EM Fault Model Characterization on SoCs: From Different Architectures to the Same Fault Model", *FDTC 2021*.

Synthesis of Trouchkine's Ph.D. thesis

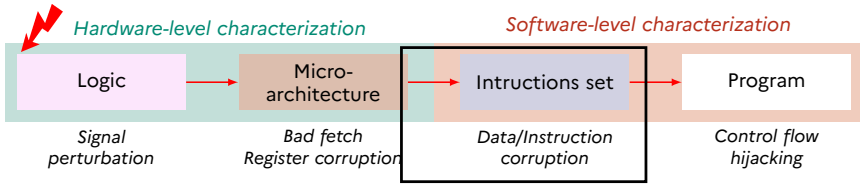
Key findings

1. Defined an architecture-agnostic approach to measure fault effects [TBC19] on **microarchitecture blocks** [Tro+21] from the ISA level;
   - analysis of faults disturbing the **MMU** and **cache management** [Tro+21].
2. Applied the methodology on **Arm** and **Intel** CPUs embedding TEE [TBC21].

Demonstrated that faults directly affect execution in **simple software contexts**.

[TBC19] "Fault Injection Characterization on Modern CPUs", *WISTP 2019.*
[Tro+21] "Electromagnetic fault injection against a complex CPU, toward new micro-architectural fault models", *JCEN 2021.*
[TBC21] "EM Fault Model Characterization on SoCs: From Different Architectures to the Same Fault Model", *FDTC 2021.*

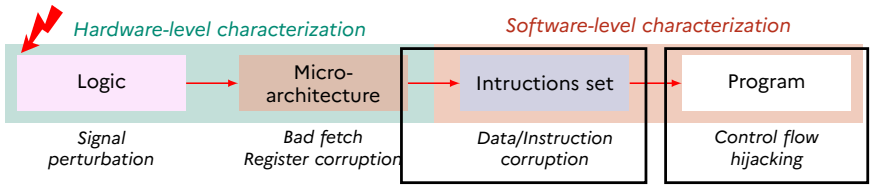
Fault Effects Characterization (cont.)



inspired from [YSW18]

[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

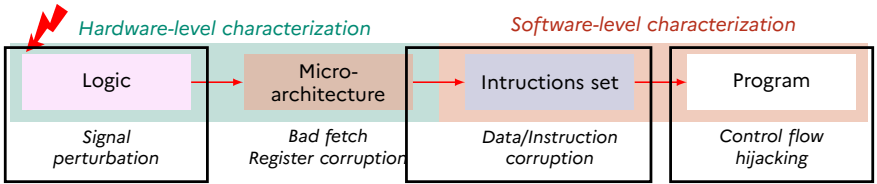
Fault Effects Characterization (cont.)



inspired from [YSW18]

[ITB23] "Pew Pew, I'm root! De la caractérisation à l'exploitation: un voyage plein d'embûches", *JAIF 2023*.
[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

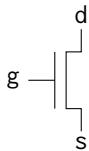
Fault Effects Characterization (cont.)



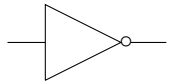
inspired from [YSW18]

[Marotta's Ph.D., 2025] "Effects of synchronous clock glitch on the security of integrated circuits", Université de Rennes.
[ITB23] "Pew Pew, I'm root! De la caractérisation à l'exploitation: un voyage plein d'embûches", *JAIF 2023*.
[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.
[YSW18] "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation", *JHSS 2018*.

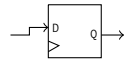
Fault Effects Analysis on Electronic Logic



Transistors



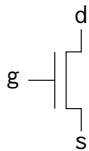
Logic gates



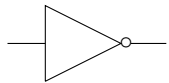
Latches

(Flip-flops)

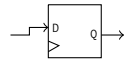
Fault Effects Analysis on Electronic Logic



Transistors

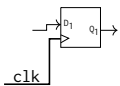


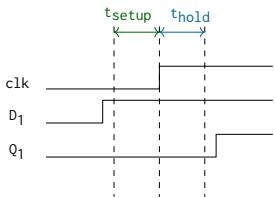
Logic gates

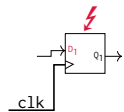


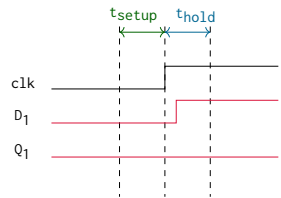
Latches

(Flip-flops)









Electromagnetic Fault Injection Impact on Logic

Electromagnetic Fault Injection (EMFI) attack: [CB19; Yua+12]

- Perturbs the clock distribution ⇒ Phase-Locked Loop (PLL);
- Results in unintended glitches on the clock signal.

[CB19] "Microcontroller Sensitivity to Fault-Injection Induced by Near-Field Electromagnetic Interference", ***EMC 2019.***
[Yua+12] "Electromagnetic interference analysis using an embedded phase-lock loop", ***APEMC 2012.***

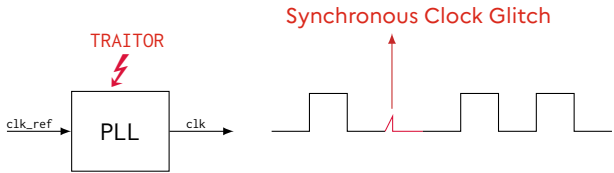
Electromagnetic Fault Injection Impact on Logic

Electromagnetic Fault Injection (EMFI) attack: [CB19; Yua+12]

- Perturbs the clock distribution ⇒ Phase-Locked Loop (PLL);
- Results in unintended glitches on the clock signal.

Observation:

- Fault effects comparable to controlled clock glitches; (injectable by TRAITOR [Cla+21])
- The **observed fault model** does not match with the litterature [Deh+12; DLM21; Nab+23].



[Cla+21] "TRAITOR: A Low-Cost Evaluation Platform for Multifault Injection", *AsiaCCS 2021*.
[Deh+12] "Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES", *FDTC 2012*.
[DLM21] "Modeling and Simulating Electromagnetic Fault Injection", *TCAD 2021*.
[Nab+23] "A Tale of Two Models: Discussing the Timing and Sampling EM Fault Injection Models", *FDTC 2023*.

Fault Effects at Logic

Method:

- Target: LFSR implemented with flip-flops (FF) embedded in an FPGA;
- FPGA experiments with controlled clock glitches using TRAITOR [Cla+21];
- Complemented with transistor-level simulations.

Findings: [Mar+24]

- Introduced the Energy-Threshold Fault Model;
  - fault sensitivity depends on **intrinsic** (manufacturing variability, routing) and **extrinsic factors** (cross-talk, neighboring activity);
- Voltage amplitude **is more significant than** glitch width in determining correct sampling.

[Cla+21] "TRAITOR: A Low-Cost Evaluation Platform for Multifault Injection", *AsiaCCS 2021*.
[Mar+24] "Characterizing and Modeling Synchronous Clock-Glitch Fault Injection", *COSADE 2024*.

Synthesis of Marotta's Ph.D. thesis

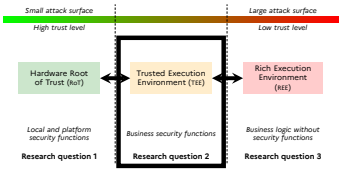
Key findings

1. Proposed an approach to **simulate EMFI at the logic level** [Mar+24]; (based on TRAITOR)
   - discovered a new Energy-Threshold Fault Model.
2. Transposed this model to study and explain **fault effects on microcontrollers** [Mar25].

Showed that the study was limited to a single latch type,
not **fully reflecting complex designs**.

[Mar+24] "Characterizing and Modeling Synchronous Clock-Glitch Fault Injection", *COSADE 2024*.
[Marotta's Ph.D., 2025] "Effects of synchronous clock glitch on the security of integrated circuits", Université de Rennes.

Research Question 2: Summary of Contributions



Research question 2:
How to achieve high security in TEEs for *business security functions*?

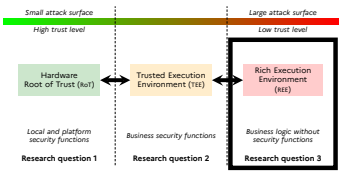
TEEs = secure environments within application CPUs, but exposed to hardware attacks.

Research focus: **fault injection characterization** at multiple abstraction levels.

1. From ISA-level analysis of closed CPUs implementation; (Arm TrustZone)
2. To logic-level analysis of known implementations. (flip-flops on FPGA/ASIC)

Identified the need for **dedicated countermeasures** for application CPUs

3. My Research Activities



Research question 3:
How can sensitive apps run securely in the REE?

REE: Requirements and Challenges

The REE is a high-performance, feature-rich environment

REE: Requirements and Challenges

The REE is a high-performance, feature-rich environment where apps' security relies:

- On **general-purpose** mitigations; (MMU, ASLR, CFI, sandboxing)
- On services offloaded to TEE/hardware RoT.

REE reality:

- **Untrusted by design**; (user-controlled, multi-users, third-party apps)
- No formal **Common Criteria** evaluation of the full stack;
- Diverse threats:
  - kernel/driver bugs, supply-chain & update issues, malware/rooting, ….

REE: Requirements and Challenges

The REE is a high-performance, feature-rich environment where apps' security relies:

- On **general-purpose** mitigations; (MMU, ASLR, CFI, sandboxing)
- On services offloaded to TEE/hardware RoT.

REE reality:

- **Untrusted by design**; (user-controlled, multi-users, third-party apps)
- No formal **Common Criteria** evaluation of the full stack;
- Diverse threats:
  - kernel/driver bugs, supply-chain & update issues, malware/rooting, ….

In this work, I study how sensitive applications can run in the REE when access to TEE/RoT is **limited or unavailable**. (agreements required with each smartphone vendor)

How to Protect Sensitive Applications in the REE?

Adversary with **full control of the execution**.

How to Protect Sensitive Applications in the REE?

Adversary with **full control of the execution**.

Method: ⇒ Defense in depth

- Use obfuscated apps to store and manipulate senstive assets.

New threat model: ⇒ Hardware attacks transposed into software

- Fault injection via binary instrumentation [Bos+16]. (inspired by methods originally targeting SEs)

[Bos+16] "Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough", ***CHES 2016.***

How to Protect Sensitive Applications in the REE?

Adversary with **full control of the execution**.

Method: ⇒ Defense in depth

- Use obfuscated apps to store and manipulate senstive assets.

New threat model: ⇒ Hardware attacks transposed into software

- Fault injection via binary instrumentation [Bos+16]. (inspired by methods originally targeting SEs)

Current focus:

- Protect implementations of symmetric algorithms in obfuscated apps against software-level fault injection;
- **[Giraud's Ph.D., 2024]**: extend this protection to implementations of asymmetric algorithms.

[Bos+16] "Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough", *CHES 2016*.
[Giraud's Ph.D., 2024] "Application security on uncontrolled systems", École Normale Supérieure.

Case Study: McEliece in White-Box Context

Security Analysis [GB23]

- Target: McEliece cryptosystem on Arm platforms [Pet+15];

[GB23] "Faulting original McEliece's implementations is possible", *SILM@EuroS&PW 2023*.
[Pet+15] "Countermeasure against the SPA attack on an embedded McEliece cryptosystem", *MAREW 2015*.

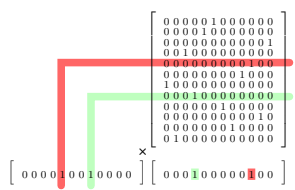
Case Study: McEliece in White-Box Context

Security Analysis [GB23]

- Target: McEliece cryptosystem on Arm platforms [Pet+15];
- Apply software-level fault injection on decryption:
  - replace `EOR` with `RSB` instruction; (1-bit instruction modification)
  - 40–70% entropy reduction of secret key.
- Mitigation: design a variant of McEliece **immune** to this attack.

```
1  uint32_t accu[1024/32] = {0};
2  for(int i = 0; i < 1024; i++) {
3    if(((vector[i/32] >> (31-(i%32))) & 0x01) != 0) {
4      for(int j = 0; j < (1024/32); j++) {
5        accu[j] =
6          accu[j] ^ matrix[i*(1024/32)+j];
7  }}}
```

```
10684   e51b300c    ldr r3, [fp, #-12]
10688   e0822003    add r2, r2, r3
1068c   e59f30d8    ldr r3, [pc, #216]
10690   e08f3003    add r3, pc, r3
10694   e7933102    ldr r3, [r3, r2, lsl #2]
10698   e0212003    eor r2, r1, r3
1069c   e51b300c    ldr r3, [fp, #-12]
106a0   e1a03103    lsl r3, r3, #2
106a4   e24b1004    sub r1, fp, #4
106a8   e0813003    add r3, r1, r3
106ac   e5032024    str r2, [r3, #-36]
```



[GB23] "Faulting original McEliece's implementations is possible", *SILM@EuroS&PW 2023*.
[Pet+15] "Countermeasure against the SPA attack on an embedded McEliece cryptosystem", *MAREW 2015*.

Synthesis of Giraud's Ph.D. thesis

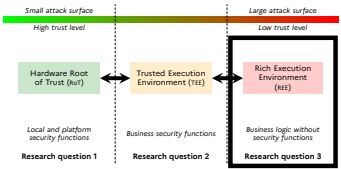
Key findings

1. Studied obfuscated applications in untrusted environments;
   - focusing on resilience against **binary instrumentation attacks**;
2. Applied the approach to a **post-quantum asymmetric algorithm** [GB23]. (McEliece)

Demonstrated that trust cannot be directly extended to REE.

[GB23] "Faulting original McEliece's implementations is possible", *SILM@EuroS&PW 2023*.

Research Question 3: Summary of Contributions



Research question 3:
How can sensitive apps run securely in the REE?

REE = open and potentially untrusted environment
Sensitive applications must be secured without strong isolation.

Research focus: protecting **sensitive assets** in the REE.

1. Investigated **white-box security models**;
2. Applied **software-level fault injection**;
3. Proposed **obfuscation and modified designs**.

Summary of my Contributions

Analyzed the security of embedded software:

1. In SEs, focusing on risks from misused environments;
2. Characterized the impact of **fault injection attacks** across system levels
   - to better understand their consequences;
3. Work mainly based on closed implementations
   - limiting internal visibility but reflecting real-world constraints;
4. Measured the limits of existing solutions
   - highlighting the need for **dedicated countermeasures** in performance-oriented implementations.

Summary of my Contributions

Analyzed the security of embedded software:

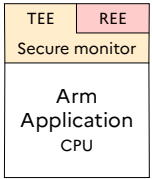
1. In SEs, focusing on risks from misused environments;
2. Characterized the impact of **fault injection attacks** across system levels
   - to better understand their consequences;
3. Work mainly based on closed implementations
   - limiting internal visibility but reflecting real-world constraints;
4. Measured the limits of existing solutions
   - highlighting the need for **dedicated countermeasures** in performance-oriented implementations.

What comes next

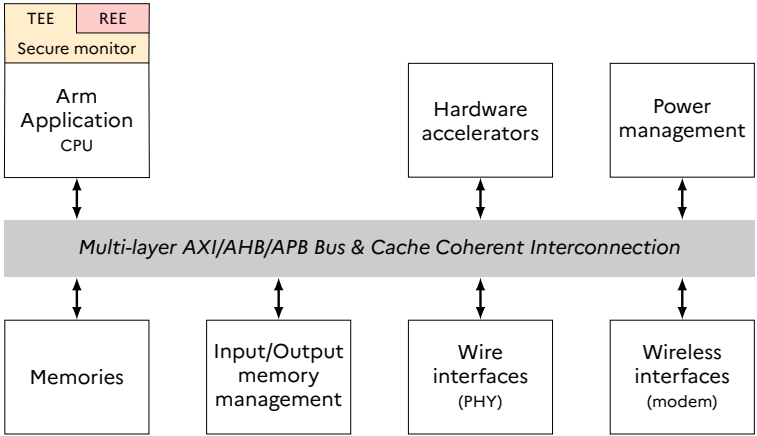
- Contribute to the **design of secure and performance-oriented implementations**;
- Propose **hardware attack-resistant solutions** for future TEE platforms.

4. Perspectives

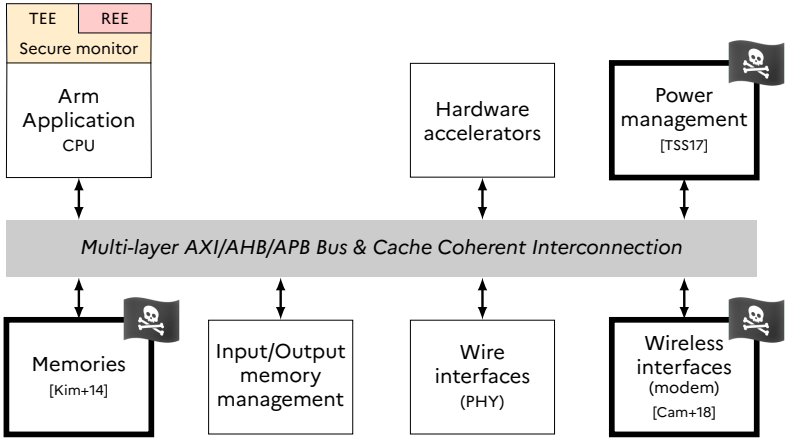
Application System on Chip (SoC) Hardware Architecture



Application System on Chip (SoC) Hardware Architecture

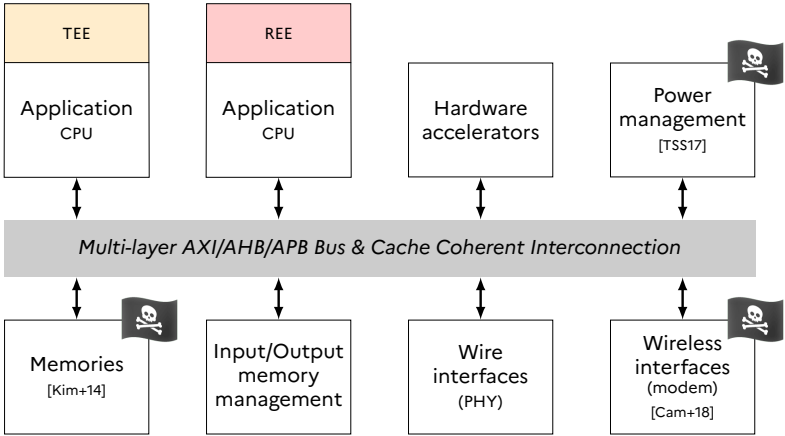


Application System on Chip (SoC) Hardware Architecture



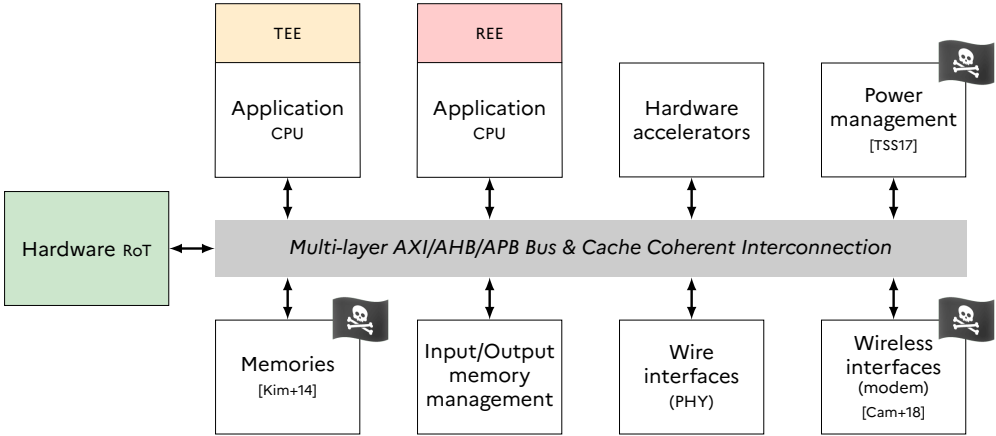
[Kim+14] "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors", *ISCA 2014*.
[Cam+18] "Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers", *ACM CCS 2018*.
[TSS17] "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management", *USENIX Security 2017*.

Application System on Chip (SoC) Hardware Architecture



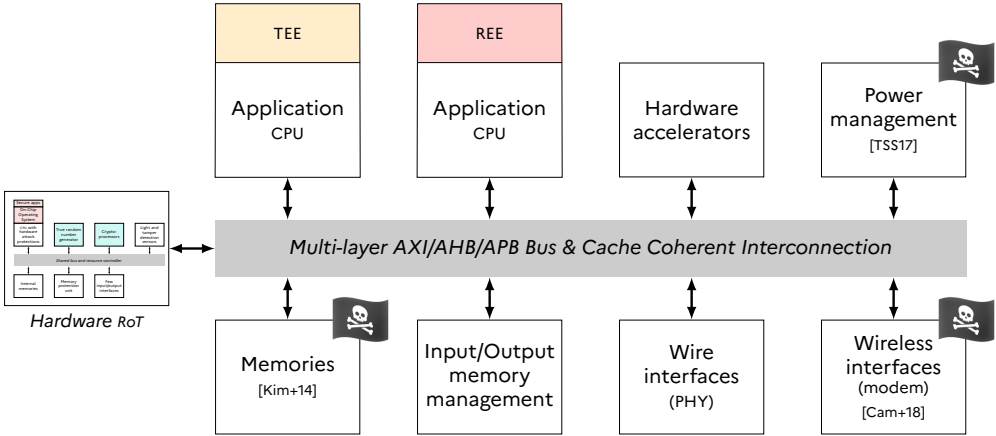
[Kim+14] "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors", *ISCA 2014*.
[Cam+18] "Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers", *ACM CCS 2018*.
[TSS17] "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management", *USENIX Security 2017*.

Application System on Chip (SoC) Hardware Architecture



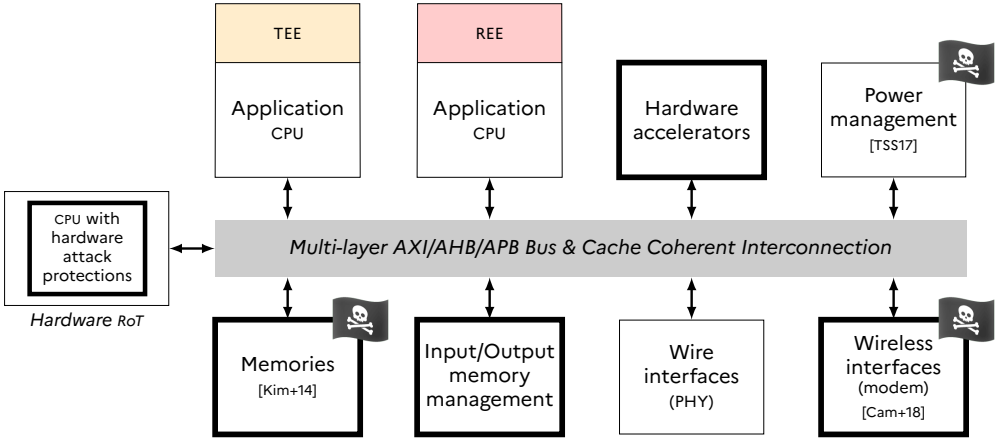
[Kim+14] "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors", *ISCA 2014*.
[Cam+18] "Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers", *ACM CCS 2018*.
[TSS17] "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management", *USENIX Security 2017*.

Application System on Chip (SoC) Hardware Architecture



[Eur14] Eurosmart. ***Smartcard IC Platform Protection Profile with Augmentation Packages.*** BSI-CC-PP-0084. Version 1.0. Jan. 2014 (`https://www.commoncriteriaportal.org/files/ppfiles/pp0084b_pdf.pdf`).

Application System on Chip (SoC) Hardware Architecture



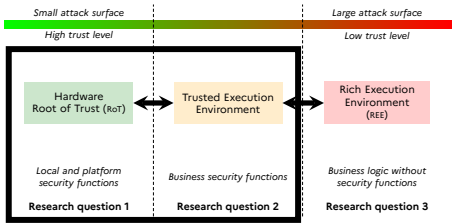
[Eur22] **Eurosmart.** ***Secure Sub-System in System-on-Chip Protection Profile.*** **BSI-CC-PP-0117. Version 1.5. Mar. 2022 (`https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0117a_pdf`).**

Toward Hardware-Resilient TEEs on Application SoCs

How can the TEE be secured **against hardware attacks** in application SoCs?

Research directions:

1. Understand security design in modern SE hardware;
2. Analyze the specificities of application SoCs;
   (shared modules, application central processing units (CPUs), complex interconnects)
3. Scale SEs protections to secure TEEs against **hardware attacks**.



Step 1: Understand Security Design in Modern SE Hardware

Deployed SEs are based on a **closed architecture**. (Arm SecurCore SC300)

Step 1: Understand Security Design in Modern SE Hardware

Deployed SEs are based on a **closed architecture**. (Arm SecurCore SC300)

RISC-V as an opportunity!

Step 1: Understand Security Design in Modern SE Hardware

Deployed SEs are based on a **closed architecture**. (Arm SecurCore SC300)

RISC-V as an opportunity!

Analysis of secure-oriented open-source implementations:

- **OpenTitan** (open-source SE) driven by LowRisc;
  - early work conducted within the PEPR Arsene funding project [Bikou's Internship, 2024].
- **CV32E40S** (open-source CPU) supported by OpenHW;
  - early work carried out in joint collaboration with CEA/List [Silva Araújo's Internship, 2024].

[Bikou's Internship, 2024] "Analysis of an Open-Source Secure Component Architecture", Sorbonne Université.
[Silva Araújo's Internship, 2024] "Security analysis of open-source RISC-V processors", École des Mines de Saint-Étienne.

Step 1: Understand Security Design in Modern SE Hardware

Deployed SEs are based on a **closed architecture**. (Arm SecurCore SC300)

RISC-V as an opportunity!

Analysis of secure-oriented open-source implementations:

- **OpenTitan** (open-source SE) driven by LowRisc;
  - early work conducted within the PEPR Arsene funding project [Bikou's Internship, 2024].
- **CV32E40S** (open-source CPU) supported by OpenHW;
  - early work carried out in joint collaboration with CEA/List [Silva Araújo's Internship, 2024].

An increasing number of **ASICs** are based on **open-source implementations** [SO25].

[SO25] "Fabrication begins for production OpenTitan silicon". 2025 (`https://opensource.googleblog.com/2025/02/fabrication-begins-for-production-opentitan-silicon.html`).

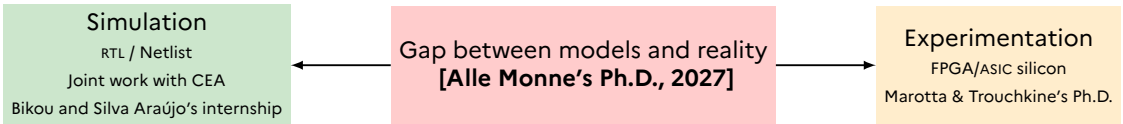
Step 1: Security Analysis at Every Stage of Hardware RoT Design

Simulation
RTL / Netlist
Joint work with CEA
Bikou and Silva Araújo's internship

Experimentation
FPGA/ASIC silicon
Marotta & Trouchkine's Ph.D.

[Bikou's Internship, 2024] "Analysis of an Open-Source Secure Component Architecture", Sorbonne Université.
[Marotta's Ph.D., 2025] "Effects of synchronous clock glitch on the security of integrated circuits", Université de Rennes.
[Silva Araújo's Internship, 2024] "Security analysis of open-source RISC-V processors", École des Mines de Saint-Étienne.
[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.

Step 1: Security Analysis at Every Stage of Hardware RoT Design

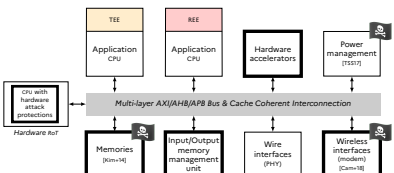


[Bikou's Internship, 2024] "Analysis of an Open-Source Secure Component Architecture", Sorbonne Université.
[Alle Monne's Ph.D., 2027] "Formalization and Analysis of Countermeasures Against Fault Injection Attacks on Open-Source Processors", Université Grenoble Alpes.
[Marotta's Ph.D., 2025] "Effects of synchronous clock glitch on the security of integrated circuits", Université de Rennes.
[Silva Araújo's Internship, 2024] "Security analysis of open-source RISC-V processors", École des Mines de Saint-Étienne.
[Trouchkine's Ph.D., 2021] "System-on-Chip Physical Security Evaluation", Université Grenoble Alpes.

Step 2: Addressing Application SoC-Specific Threats

Challenge: TEE runs on an application CPU **without dedicated protections**.

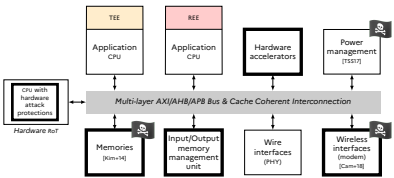
Secure and performance-oriented apps must consider hardware-level vulnerabilities.



Step 2: Addressing Application SoC-Specific Threats

Challenge: TEE runs on an application CPU **without dedicated protections**.

Secure and performance-oriented apps must consider hardware-level vulnerabilities.



[Gonidec's Ph.D., 2026]:

- Power management units as attack vectors;
- Survey of **TEE threats induced by power management units** [Gon+25].

[Gonidec's Ph.D., 2026] "Securing RISC-V System-on-Chip against Energy-based Attacks", Université Bretagne Sud.
[Gon+25] "Do Not Trust Power Management: A Survey on Internal Energy-based Attacks Circumventing Trusted Execution Environments Security Properties", *TECS 2025*.

Step 3: Secure TEEs against Hardware Attacks

- Builds on **Step 1**: understand countermeasures in SE;
- Builds on **Step 2**: analyze threats specific to application SoCs.

Transpose and adapt SE protections to secure **dedicated CPUs for TEE**.

This work is at an early stage:

1. Identify which SE protections can scale to TEE context;
   - how to adapt the *Secure Sub-System in SoC PP* [Eur22] to application processors?
2. Propose dedicated countermeasures against **hardware attacks**:
   - PTCC FORWARD project: countermeasures for application CPU against hardware attacks.

[Eur22] Eurosmart. ***Secure Sub-System in System-on-Chip Protection Profile.*** BSI-CC-PP-0117. Version 1.5. Mar. 2022 (`https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Reporte/ReportePP/pp0117a_pdf`).

From Security-oriented CoT …

The **CoT** is not only a foundation for **securing sensitive apps** …

...to Safety-Critical CoT

... it can also be transposed to safety-critical systems, where both **safety** and **security** must coexist.

...to Safety-Critical CoT

... it can also be transposed to safety-critical systems, where both **safety** and **security** must coexist.

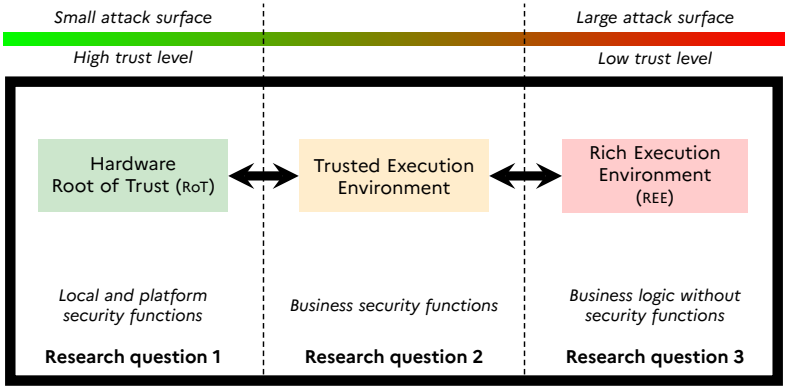
Safety-Critical Systems = systems whose failure may cause harm from people, assets, or the environment.

- Deployed in **medical**, **industrial**, and **transportation** sectors;
- Growing connectivity ⇒ stronger security requirements.

Safety constraints: functions cannot be disabled, even under attack.

New Research Question

How can the **CoT** model be adapted to strengthen **safety**?



Case Study: Connected Vehicles

- **High connectivity** (Bluetooth, Wi-Fi, cellular) + **sensors** (cameras, LiDAR, radars);

[TB25] "300 secondes chrono: prise de contrôle d'un infodivertissement automobile à distance", *SSTIC 2025*.

Case Study: Connected Vehicles

- **High connectivity** (Bluetooth, Wi-Fi, cellular) + **sensors** (cameras, LiDAR, radars);
- Internal ANSSI project on a representative 2020 vehicle [TB25];
- **Bluetooth** chosen as focus: always active, even without user connection.
  - the Bluetooth stack is implemented within the REE.

[TB25] "300 secondes chrono: prise de contrôle d'un infodivertissement automobile à distance", *SSTIC 2025*.

Case Study: Connected Vehicles

- **High connectivity** (Bluetooth, Wi-Fi, cellular) + **sensors** (cameras, LiDAR, radars);
- Internal ANSSI project on a representative 2020 vehicle [TB25];
- **Bluetooth** chosen as focus: always active, even without user connection.
  - the Bluetooth stack is implemented within the REE.

```
struct sdpServInfo[0] {
 /* 0x0000 */  void    * next;
 /* 0x0004 */  void    * prev;
 /* 0x0008 */  uint8_t * ptr_pkt_data;
    // ...
 /* 0x0088 */  uint8_t   pkt_header [5];
 /* 0x008D */  uint8_t   pkt_data   [507];
};
struct sdpServInfo[1] {
 /* 0x0288 */  void    * next;
 /* 0x028C */  void    * prev;
 /* 0x0290 */  uint8_t * ptr_pkt_data;
    // ...
 /* 0x0310 */  uint8_t   pkt_header [5];
 /* 0x0314 */  uint8_t   pkt_data   [507];
}; // size = 648 (0x288) bytes
// ...
```

Discovery of a **0-click unauthenticated RCE** vulnerability.

[TB25] "300 secondes chrono: prise de contrôle d'un infodivertissement automobile à distance", *SSTIC 2025*.

Case Study: Connected Vehicles

- **High connectivity** (Bluetooth, Wi-Fi, cellular) + **sensors** (cameras, LiDAR, radars);
- Internal ANSSI project on a representative 2020 vehicle [TB25];
- **Bluetooth** chosen as focus: always active, even without user connection.
  - the Bluetooth stack is implemented within the REE.

```
struct sdpServInfo[0] {
  /* 0x0000 */  void    * next;
  /* 0x0004 */  void    * prev;
  /* 0x0008 */  uint8_t * ptr_pkt_data;
     // ...
  /* 0x0088 */  uint8_t   pkt_header [5];
  /* 0x008D */  uint8_t   pkt_data   [507];
};
struct sdpServInfo[1] {
  /* 0x0288 */  void    * next;
  /* 0x028C */  void    * prev;
  /* 0x0290 */  uint8_t * ptr_pkt_data;
     // ...
  /* 0x0310 */  uint8_t   pkt_header [5];
  /* 0x0314 */  uint8_t   pkt_data   [507];
}; // size = 648 (0x288) bytes
// ...
```

Discovery of a **0-click unauthenticated RCE** vulnerability.

Security Implications

- **Compromise of the REE** ⇒ send unauthorized CAN messages. (existing hardware RoT filters some critical CAN messages)
- Highlights the need for a more complete CoT [Glo23].

[TB25] "300 secondes chrono: prise de contrôle d'un infodivertissement automobile à distance", *SSTIC 2025*.
[Glo23] GlobalPlatform. *Trust & Security in Automotive Systems*. Tech. rep. Oct. 2023 (https://globalplatform.org/wp-content/uploads/2023/10/GP-Trust-for-Secure-AutoServices-White-Paper_Web_Spreads.pdf).

Towards a Unified Perspective on Safety and Security

Objective: Improved **security** with **functional safety** in connected and critical systems.

Towards a Unified Perspective on Safety and Security

Objective: Improved **security** with **functional safety** in connected and critical systems.

- Builds on expertise from hardware RoTs and TEEs security;
- Focus on **connected and autonomous vehicles**:
  - resistance to **hardware attacks** [Küh+25; Mel24; OFl20; Wer+23]; (fault injection, side-channel)
  - integration with **new infrastructures** [Dud19]. (in-motion charging, connected roads)
- Broader scope: **medical and industrial sectors**: (studied by several ANSSI teams)
  - where security and safety must coexist;

[Dud19] "V2G Injector: Whispering to cars and charging units through the Power-Line", ***SSTIC 2019*****.**
[Küh+25] "Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV", ***Asia CCS 2025*****.**
[Mel24] "Bypassing the Renesas RH850/P1M-E read protection using fault injection". 2024 (`https://icanhack.nl/blog/rh850-glitch/`).
[OFl20] "BAM BAM!! On Reliability of EMFI for in-situ Automotive ECU Attacks", ***ESCAR EU 2020*****.**
[Wer+23] "Back in the Driver's Seat: Recovering Critical Data from Tesla Autopilot Using Voltage Glitching", ***CCC 2023*****.**

Towards a Unified Perspective on Safety and Security

Objective: Improved **security** with **functional safety** in connected and critical systems.

- Builds on expertise from hardware RoTs and TEEs security;
- Focus on **connected and autonomous vehicles**:
  - resistance to **hardware attacks** [Küh+25; Mel24; OFl20; Wer+23]; (fault injection, side-channel)
  - integration with **new infrastructures** [Dud19]. (in-motion charging, connected roads)
- Broader scope: **medical and industrial sectors**: (studied by several ANSSI teams)
  - where security and safety must coexist;
- Advocacy for rigorous **evaluation and certification** of vehicle CoT [CAR21].

[CAR21] CAR 2 CAR Communication Consortium. ***Protection Profile V2X Hardware Security Module.*** BSI-CC-PP-0114. Version 1.0.1. Dec. 2021 (https://commoncriteriaportal.org/nfs/ccpfiles/files/ppfiles/pp0114a_pdf.pdf).

Towards a Unified Perspective on Safety and Security

Objective: Improved **security** with **functional safety** in connected and critical systems.

- Builds on expertise from hardware RoTs and TEEs security;
- Focus on **connected and autonomous vehicles**:
  - resistance to **hardware attacks** [Küh+25; Mel24; OFl20; Wer+23]; (fault injection, side-channel)
  - integration with **new infrastructures** [Dud19]. (in-motion charging, connected roads)
- Broader scope: **medical and industrial sectors**: (studied by several ANSSI teams)
  - where security and safety must coexist;
- Advocacy for rigorous **evaluation and certification** of vehicle CoT [CAR21].

Propose solutions where **security reinforces safety**, ensuring resilience in increasingly interconnected environments.

[CAR21] CAR 2 CAR Communication Consortium. ***Protection Profile V2X Hardware Security Module.*** BSI-CC-PP-0114. Version 1.0.1. Dec. 2021 (https://commoncriteriaportal.org/nfs/ccpfiles/files/ppfiles/pp0114a_pdf.pdf).

5. Conclusion

Conclusion

Research within ANSSI:

- Only **33%** of my time dedicated to research;
- Strong link with national security missions and evaluations.

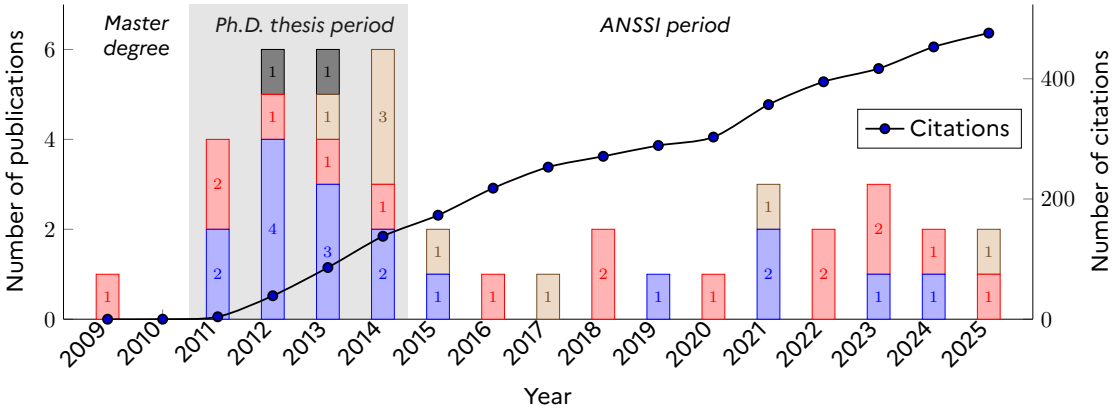
Research areas:

- From hardware RoTs to TEEs and safety-critical systems;
- Contributions at both **academic** and **operational** levels;
  - built on **strong collaborations** with CEA/Leti, CEA/List, DGA-MI, IETR, INRIA, UBS/Lab-STICC, and several ANSSI teams;
- Supervision of **5 Ph.D. thesis** (3 defended / 2 ongoing), **9 internships** and **1 apprenticeship**.

Summary and Future Directions:

- Understanding and securing the full Chain of Trust;
- Towards bridging **security** and **safety**.

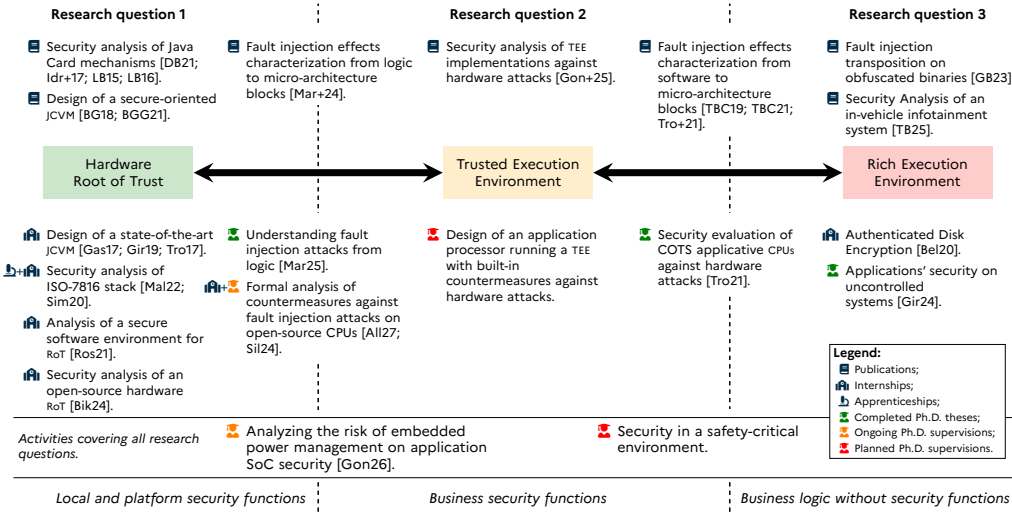
My Publications through the Years



International conferences (6 / 11) National conferences (10 / 5)
Journal articles (4 / 4) Book chapters (- / 2)

- Citations: 476
- *h*-index: 13

Thank you for your Attention 🙏 🙂



Bibliography

[All27] Jonah Alle Monne. “Formalization and Analysis of Countermeasures Against Fault Injection Attacks on Open-Source Processors”. PhD thesis. Grenoble, France: Université Grenoble Alpes, 2027. In preparation.

[AM14] Ross J. Anderson and Steven J. Murdoch. “EMV: why payment systems fail”. In: *Communications of the ACM* 57.6 (2014), pp. 24–28. DOI: 10.1145/2602321.

[Bal+15] Josep Balasch, Benedikt Gierlichs, Oscar Reparaz, and Ingrid Verbauwhede. “DPA, Bitslicing and Masking at 1 GHz”. In: *Proceedings of the 17th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Tim Güneysu and Helena Handschuh. Vol. 9293. Lecture Notes in Computer Science. Saint-Malo, France: Springer, Sept. 2015, pp. 599–619. DOI: 10.1007/978-3-662-48324-4_30.

[BST21] David A. Basin, Ralf Sasse, and Jorge Toro-Pozo. “The EMV Standard: Break, Fix, Verify”. In: *Proceedings of the 42nd IEEE Symposium on Security and Privacy S&P*. San Francisco, CA, USA: IEEE, May 2021, pp. 1766–1781. DOI: 10.1109/SP40001.2021.00037.

Bibliography (cont.)

[Bel20] Yanis Belkheyar. “Authenticated Disk Encryption”. Master’s thesis. Paris, France: Université Paris 7, Sept. 2020.

[Bik24] Angie-Sofia Bikou. “Analysis of an Open-Source Secure Component Architecture”. Master’s thesis. Paris, France: Sorbonne Université, Sept. 2024.

[BS10] BMS and SFPMEI. *PP Secure Access Module for Electronic Money system Protection Profile*. Feb. 2010.

[Bos+16] Joppe W. Bos, Charles Hubain, Wil Michiels, and Philippe Teuwen. “Differential Computation Analysis: Hiding Your White-Box Designs is Not Enough”. In: *Proceedings of the 18th International Conference on Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Vol. 9813. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Aug. 2016, pp. 215–236. DOI: 10.1007/978-3-662-53140-2_11.

Bibliography (cont.)

[Bou14] Guillaume Bouffard. “A Generic Approach for Protecting Java Card Smart Card Against Software Attacks”. PhD thesis. Limoges, France: Université de Limoges, Oct. 2014.

[BG18] Guillaume Bouffard and Léo Gaspard. “Hardening a Java Card Virtual Machine Implementation with the MPU”. In: *Symposium sur la sécurité des technologies de l’information et des communications (SSTIC)*. Rennes, France, June 2018.

[BGG21] Guillaume Bouffard, Vincent Giraud, and Léo Gaspard. “Java Card Virtual Machine Memory Organization: a Design Proposal”. In: *CoRR* (2021). arXiv: 2110.10037.

[Cam+18] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. “Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers”. In: *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. Ed. by David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang. Toronto, ON, Canada: ACM, Oct. 2018, pp. 163–177. DOI: 10.1145/3243734.3243802.

Bibliography (cont.)

[CAR21] CAR 2 CAR Communication Consortium. *Protection Profile V2X Hardware Security Module*. BSI-CC-PP-0114. Version 1.0.1. Dec. 2021.

[Cen17] European Commission — Joint Research Centre. *Digital Tachograph - Tachograph Card (TC PP)*. BSI-CC-PP-0091-2017. Version 1.0. May 2017.

[CB19] Ludovic Claudepierre and Philippe Besnier. "Microcontroller Sensitivity to Fault-Injection Induced by Near-Field Electromagnetic Interference". In: *Proceedings of the International Symposium on Electromagnetic Compatibility (EMC)*. Sapporo, Japan, June 2019, pp. 673–676. DOI: 10.23919/EMCTokyo.2019.8893701.

[Cla+21] Ludovic Claudepierre, Pierre-Yves Péneau, Damien Hardy, and Erven Rohou. "TRAITOR: A Low-Cost Evaluation Platform for Multifault Injection". In: *Proceedings of the 21rd International Symposium on Advanced Security on Software and Systems (ASSS)*. Ed. by Weizhi Meng and Li Li. Virtual Event, Hong Kong: ACM, June 2021, pp. 51–56. DOI: 10.1145/3457340.3458303.

Bibliography (cont.)

[Deh+12] Amine Dehbaoui, Jean-Max Dutertre, Bruno Robisson, and Assia Tria. "Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES". In: *Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. Ed. by Guido Bertoni and Benedikt Gierlichs. Leuven, Belgium: IEEE Computer Society, Sept. 2012, pp. 7–15. DOI: 10.1109/FDTC.2012.15.

[DB21] Jean Dubreuil and Guillaume Bouffard. "PhiAttack - Rewriting the Java Card Class Hierarchy". In: *Proceedings of the 20th International Conference on Smart Card Research and Advanced Applications (CARDIS)*. Ed. by Vincent Grosso and Thomas Pöppelmann. Vol. 13173. Lecture Notes in Computer Science. Lübeck, Germany: Springer, Nov. 2021, pp. 275–288. DOI: 10.1007/978-3-030-97348-3_15.

[Dud19] Sébastien Dudek. "V2G Injector: Whispering to cars and charging units through the Power-Line". In: *Symposium sur la sécurité des technologies de l'information et des communications (SSTIC)*. Rennes, France, June 7, 2019.

Bibliography (cont.)

[DLM21] Mathieu Dumont, Mathieu Lisart, and Philippe Maurine. “Modeling and Simulating Electromagnetic Fault Injection”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 40.4 (2021), pp. 680–693. DOI: 10.1109/TCAD.2020.3003287.

[Eur14] Eurosmart. *Smartcard IC Platform Protection Profile with Augmentation Packages*. BSI-CC-PP-0084. Version 1.0. Jan. 2014.

[Eur22] Eurosmart. *Secure Sub-System in System-on-Chip Protection Profile*. BSI-CC-PP-0117. Version 1.5. Mar. 2022.

[Gas17] Léo Gaspard. “Implementation of a Secure Operating System for Java Card-based Secure Element”. Master’s thesis. Palaiseau, France: École Polytechnique, Sept. 2017.

[Gir19] Vincent Giraud. “Secure Implementation of GlobalPlatform for Java Card Platform”. Master’s thesis. Rennes, France: INSA, Sept. 2019.

Bibliography (cont.)

[Gir24] Vincent Giraud. “Application security on uncontrolled systems. Study of the risks, protections, stakes and interests around trust in off-the-shelf computer products”. PhD thesis. Paris, France: École Normale Supérieure, Sept. 2024.

[GB23] Vincent Giraud and Guillaume Bouffard. “Faulting original McEliece’s implementations is possible. How to mitigate this risk?” In: *IEEE European Workshops on Symposium on Security and Privacy (EuroS&PW)*. Delft, Netherlands: IEEE, July 2023, pp. 311–319. DOI: 10.1109/EuroSPW59978.2023.00039.

[Glo19] Global Platform. *6.2 Billion GlobalPlatform-Compliant Secure Elements Deployed in 2018*. May 2019. URL: https://globalplatform.org/latest-news/6-2-billion-globalplatform-compliant-secure-elements-deployed-in-2018/.

[Glo18] GlobalPlatform. *Root of Trust Definitions and Requirements*. Version 1.1. June 2018.

[Glo20] GlobalPlatform. *TEE Protection Profile*. GPD_SPE_021. Version 1.3. July 2020.

[Glo22] GlobalPlatform. *TEE System Architecture*. Version 1.3. May 2022.

Bibliography (cont.)

[Glo23] GlobalPlatform. *Trust & Security in Automotive Systems*. Tech. rep. Oct. 2023.

[Gon26] Gwenn Le Gonidec. "Securing RISC-V System-on-Chip against Energy-based Attacks". PhD thesis. Rennes, France: Université Bretagne Sud, 2026. In preparation.

[Gon+25] Gwenn Le Gonidec, Guillaume Bouffard, Jean-Christophe Prévotet, and Maria Méndez Real. "Do Not Trust Power Management: A Survey on Internal Energy-based Attacks Circumventing Trusted Execution Environments Security Properties". In: *ACM Transactions on Embedded Computing Systems* 24.4 (July 2025). ISSN: 1539-9087. DOI: 10.1145/3735556.

[Idr+17] Noreddine El Janati El Idrissi, Guillaume Bouffard, Jean-Louis Lanet, and Said El Hajji. "Trust can be misplaced". In: *Journal of Cryptographic Engineering* 7.1 (2017), pp. 21–34. DOI: 10.1007/s13389-016-0142-5.

Bibliography (cont.)

[ITB23] Alexandre Iooss, Thomas Trouchkine, and Guillaume Bouffard. “Pew Pew, I’m root! De la caractérisation à l’exploitation: un voyage plein d’embûches”. fr. In: *Journée thématique sur les attaques par injection de fautes (JAIF)* (Sept. 2023).

[Kim+14] Yoongu Kim, Ross Daly, Jeremie S. Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. “Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors”. In: *Proceedings of 41st International Symposium on Computer Architecture (ISCA)*. Minneapolis, MN, USA: IEEE Computer Society, June 2014, pp. 361–372. DOI: 10.1109/ISCA.2014.6853210.

[Kin24] Beth Kindig. *Arm Stock: AI Chip Favorite Is Overpriced*. Mar. 2024. URL: https://www.forbes.com/sites/bethkindig/2024/03/21/arm-stock-ai-chip-favorite-is-overpriced/ (visited on 07/07/2025).

Bibliography (cont.)

[Küh+25] Niclas Kühnapfel, Christian Werling, Hans Niklas Jacob, and Jean-Pierre Seifert. "Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV". In: *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security (Asia CCS)*. ASIA CCS '25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 1235–1249. ISBN: 9798400714108. DOI: 10.1145/3708821.3710820.

[LB15] Julien Lancia and Guillaume Bouffard. "Java Card Virtual Machine Compromising from a Bytecode Verified Applet". In: *Proceedings of the 14th International Conference Smart Card Research and Advanced Applications (CARDIS)*. Vol. 9514. Lecture Notes in Computer Science. Bochum, Germany: Springer, Nov. 2015, pp. 75–88. DOI: 10.1007/978-3-319-31271-2_5.

[LB16] Julien Lancia and Guillaume Bouffard. "Fuzzing and Overflows in Java Card Smart Cards". In: *Symposium sur la sécurité des technologies de l'information et des communications (SSTIC)*. Rennes, France, June 2016.

Bibliography (cont.)

[Lon+15] Jake Longo, Elke De Mulder, Dan Page, and Michael Tunstall. “SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip”. In: *Proceedings of the 17th International Workshop Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Tim Güneysu and Helena Handschuh. Vol. 9293. Lecture Notes in Computer Science. Saint-Malo, France: Springer, Sept. 2015, pp. 620–640. DOI: 10.1007/978-3-662-48324-4_31.

[Mal22] Louisa Malki. “Fingerprinting of Embedded Software Implementation”. Master’s thesis. Paris, France: École 42, Sept. 2022.

[Mar25] Amélie Marotta. “Effects of synchronous clock glitch on the security of integrated circuits”. PhD thesis. Rennes, France: Université de Rennes, June 2025.

Bibliography (cont.)

[Mar+24] Amélie Marotta, Ronan Lashermes, Guillaume Bouffard, Olivier Sentieys, and Rachid Dafali. "Characterizing and Modeling Synchronous Clock-Glitch Fault Injection". In: *Proceedings of the 15th International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*. Ed. by Romain Wacquez and Naofumi Homma. Vol. 14595. Lecture Notes in Computer Science. Gardanne, France: Springer, Apr. 2024, pp. 3–21. DOI: 10.1007/978-3-031-57543-3_1.

[Mel24] Willem Melching. *Bypassing the Renesas RH850/P1M-E read protection using fault injection*. Nov. 8, 2024. URL: https://icanhack.nl/blog/rh850-glitch/ (visited on 07/07/2025).

[Nab+23] Roukoz Nabhan, Jean-Max Dutertre, Jean-Baptiste Rigaud, Jean-Luc Danger, and Laurent Sauvage. "A Tale of Two Models: Discussing the Timing and Sampling EM Fault Injection Models". In: *Proceedings of the Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*. Prague, Czech Republic: IEEE, Sept. 2023, pp. 1–12. DOI: 10.1109/FDTC60478.2023.00010.

Bibliography (cont.)

[OFl20] Colin O'Flynn. "BAM BAM!! On Reliability of EMFI for in-situ Automotive ECU Attacks". In: *IACR Cryptology ePrint Archive* (2020), p. 937. eprint: 2020/937.

[Ora21] Oracle. *Java Card Protection Profile – Open Configuration*. Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, May 2021.

[Pas22] Calinel Pasteanu. *Oracle Celebrates the Java Card Forum's 25th Anniversary*. Oct. 2022. URL: https://blogs.oracle.com/java/post/java-card-forum-25-years-anniversary (visited on 07/07/2025).

[Pet+15] Martin Petrvalsky, Tania Richmond, Milos Drutarovsky, Pierre-Louis Cayrel, and Viktor Fischer. "Countermeasure against the SPA attack on an embedded McEliece cryptosystem". In: *Proceedings of 25th IEEE International Conference Radioelektronika (MAREW)*. Pardubice, Czech Republic, Apr. 2015, pp. 462–466. DOI: 10.1109/RADIOELEK.2015.7129055.

[Rad10] Société Française du Radiotéléphone (SFR). *(U)SIM Java Card Platform Protection Profile Basic and SCWS Configurations*. Version 2.0.2. June 2010.

Bibliography (cont.)

[Ros21] Ever Atilano Rosales. “Security of the Secure Boot against Fault Attacks”. Master’s thesis. Paris, France: Université Paris 6, Sept. 2021.

[Sec25] Security Explorations. *eSIM security*. Aug. 2025. URL: https://security-explorations.com/esim-security.html (visited on 08/28/2025).

[Sic12] Bundesamt für Sicherheit in der Informationstechnik (BSI). *Common Criteria Protection Profile — Machine Readable Travel Document with “ICAO Application”, Extended Access Control with PACE (EAC PP)*. BSI-CC-PP-0056-V2-2012. Version 1.3.2. Dec. 2012.

[Sil24] Mário da Silva Araújo. “Security analysis of open-source RISC-V processors”. Master’s thesis. Gardanne, France: École des Mines de Saint-Étienne, Sept. 2024.

[Sim20] Boris Simunovic. “Security Analysis of the ISO-7816 Stack”. Master’s thesis. Valence, France: ESISAR, Sept. 2020.

Bibliography (cont.)

[SO25] Cyrus Stoller and Miguel Osorio. *Fabrication begins for production OpenTitan silicon*. Feb. 2025. URL: https://opensource.googleblog.com/2025/02/fabrication-begins-for-production-opentitan-silicon.html.

[TSS17] Adrian Tang, Simha Sethumadhavan, and Salvatore J. Stolfo. "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management". In: *Proceedings of the 26th USENIX Security Symposium*. Ed. by Engin Kirda and Thomas Ristenpart. Vancouver, BC, Canada: USENIX Association, Aug. 2017, pp. 1057–1074.

[TB25] Philippe Trébuchet and Guillaume Bouffard. "300 secondes chrono: prise de contrôle d'un infodivertissement automobile à distance". In: *Symposium sur la sécurité des technologies de l'information et des communications (SSTIC)*. June 2025.

Bibliography (cont.)

[Tro17] Thomas Trouchkine. “Hardware Implementation of a Java Card Virtual Machine”. Master’s thesis. Gardanne, France: École des Mines de Saint-Étienne, Sept. 2017.

[Tro21] Thomas Trouchkine. “System-on-Chip Physical Security Evaluation”. PhD thesis. Grenoble, France: Université Grenoble Alpes, Mar. 2021.

[TBC19] Thomas Trouchkine, Guillaume Bouffard, and Jessy Clédière. “Fault Injection Characterization on Modern CPUs”. In: *Proceedings of the 13th International Conference Information Security Theory and Practice (WISTP)*. Ed. by Maryline Laurent and Thanassis Giannetsos. Vol. 12024. Lecture Notes in Computer Science. Paris, France: Springer, Dec. 2019, pp. 123–138. DOI: 10.1007/978-3-030-41702-4_8.

[TBC21] Thomas Trouchkine, Guillaume Bouffard, and Jessy Clédière. “EM Fault Model Characterization on SoCs: From Different Architectures to the Same Fault Model”. In: *Proceedings of the 18th Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*. Milan, Italy: IEEE, Sept. 2021, pp. 31–38. DOI: 10.1109/FDTC53659.2021.00014.

Bibliography (cont.)

[Tro+21] Thomas Trouchkine, Sébanjila Kevin Bukasa, Mathieu Escouteloup, Ronan Lashermes, and Guillaume Bouffard. “Electromagnetic fault injection against a complex CPU, toward new micro-architectural fault models”. In: *Journal of Cryptographic Engineering (JCEN)* (Mar. 2021). DOI: 10.1007/s13389-021-00259-6.

[Vas+17] Aurélien Vasselle, Hugues Thiebeauld, Quentin Maouhoub, Adèle Morisset, and Sébastien Ermeneux. “Laser-Induced Fault Injection on Smartphone Bypassing the Secure Boot”. In: *Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. aipei, Taiwan: IEEE Computer Society, Sept. 2017, pp. 41–48. DOI: 10.1109/FDTC.2017.18.

[Wer+23] Christian Werling, Niclas Kühnapfel, Hans Niklas Jacob, and Oleg Drokin. “Back in the Driver’s Seat: Recovering Critical Data from Tesla Autopilot Using Voltage Glitching”. In: *Proceedings of the 37th Chaos Communication Congress* (Dec. 2023).

Bibliography (cont.)

[Yua+12] Shih-Yi Yuan, Yu-Lun Wu, Richard Perdriau, Shry-Sann Liao, and Hao-Ping Ho. "Electromagnetic interference analysis using an embedded phase-lock loop". In: *Proceedings of Asia-Pacific Symposium on Electromagnetic Compatibility (APEMC)*. Singapore, May 2012. DOI: 10.1109/APEMC.2012.6237910.

[YSW18] Bilgiday Yuce, Patrick Schaumont, and Marc Witteman. "Fault Attacks on Secure Embedded Software: Threats, Design, and Evaluation". In: *Journal of Hardware and Systems Security* 2.2 (2018), pp. 111–130. DOI: 10.1007/s41635-018-0038-1.