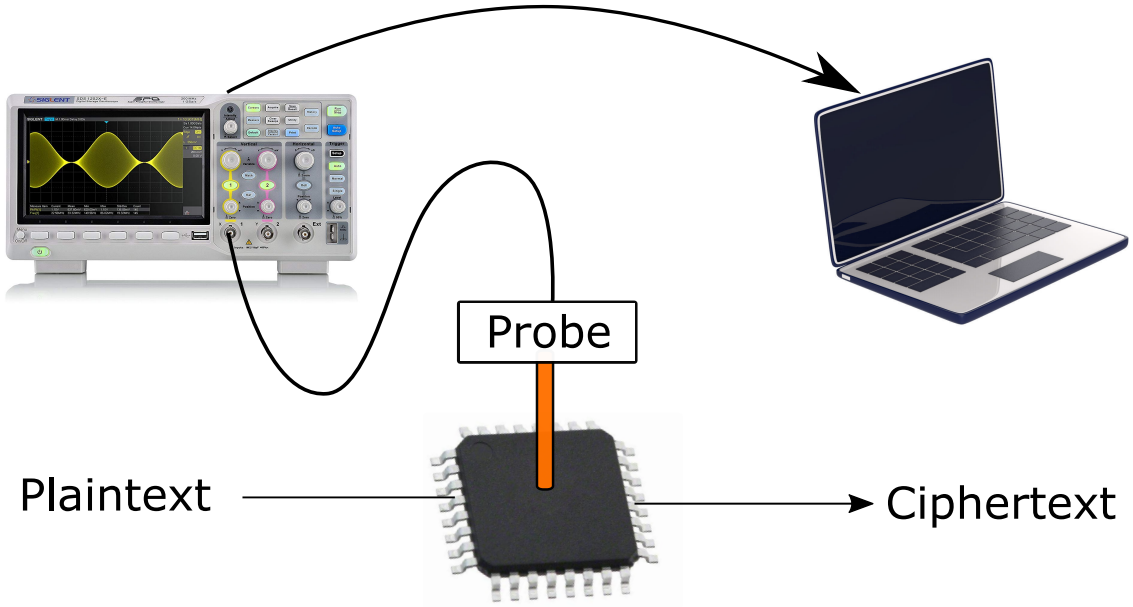
Remote Side-Channel Attacks on Heterogeneous SoC

Joseph GRAVELLIER (EMSE)
Jean-Max DUTERTRE (EMSE)
Yannick TEGLIA (THALES)
Philippe LOUBET-MOUNDI (THALES)
Francis OLIVIER (THALES)

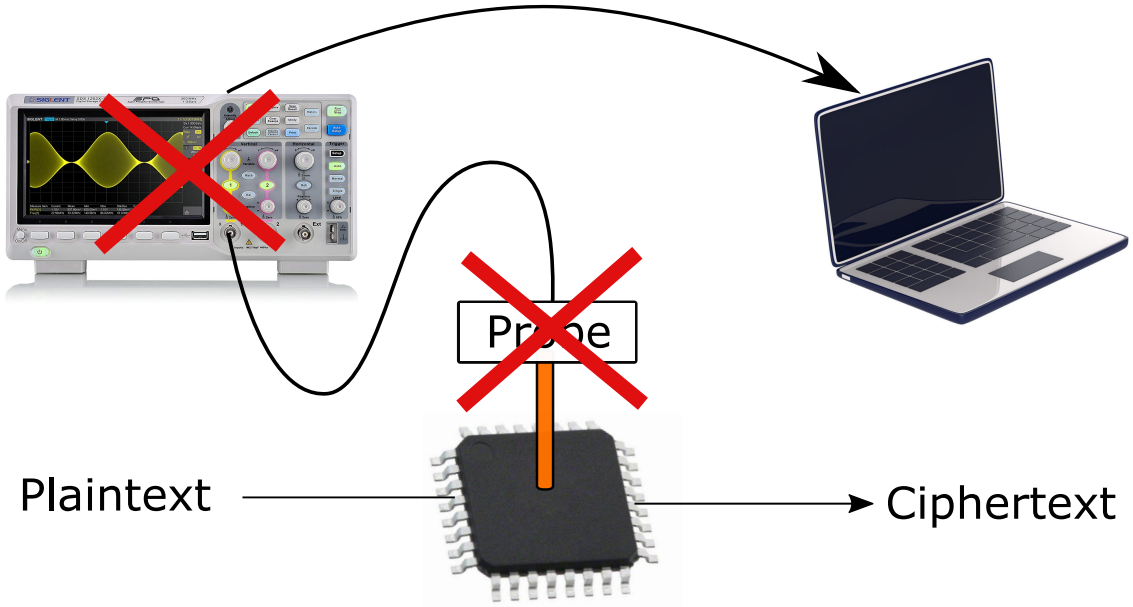
Laboratoire de Sécurité des Architectures et des Systèmes,
F-13541 Gardanne France
Thales - 13600 La Ciotat, France

November 2019

- Type: fault injection attack (FIA) & side-channel attack (SCA).
- Target: smart cards, microcontrollers, system on chip...
- Means: oscilloscope, laser, EM probe...
- Range: **local**, direct physical access required.



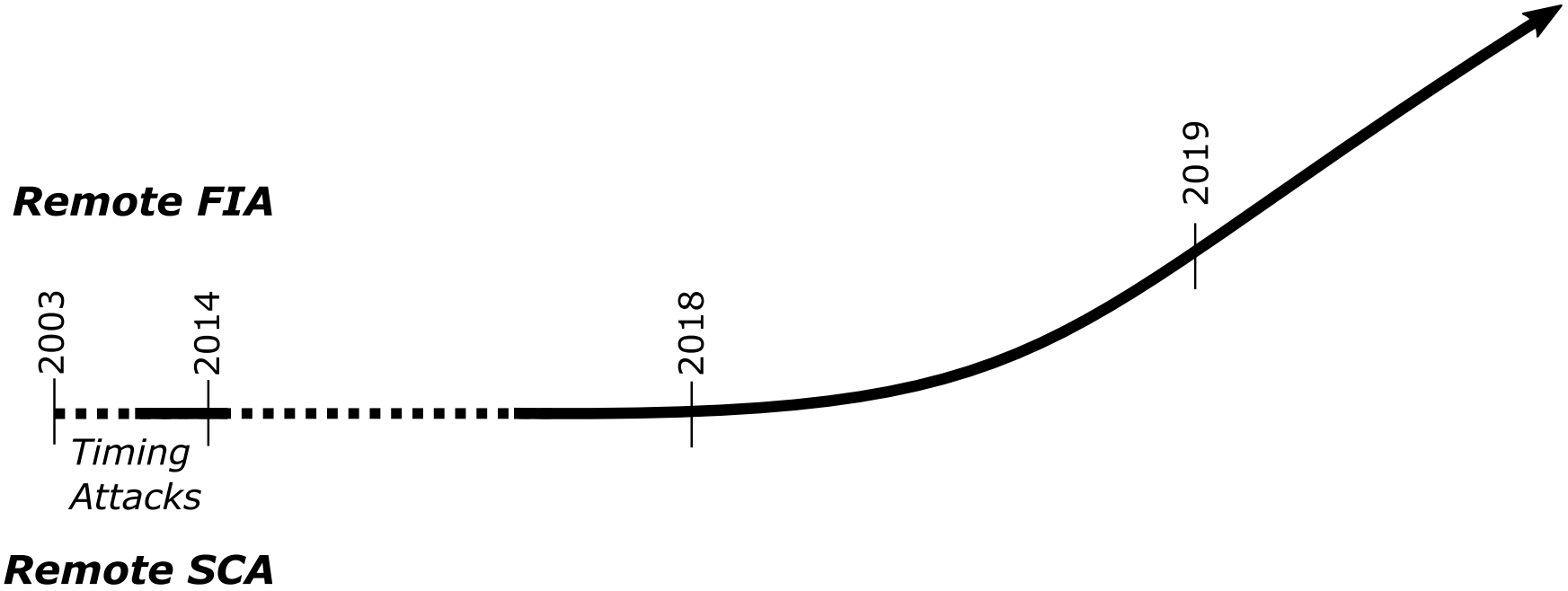
- Type: fault injection attack (FIA) & side-channel attack (SCA).
- Range: **remote, access to a network required.**
- Target: connected devices (IoT), data centers...
- Means: **resources available within the target**.



About remote hardware attacks

A temporal overview

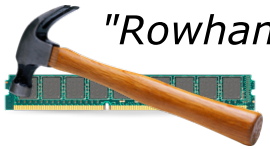
- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



About remote hardware attacks

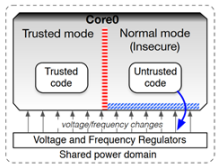
A temporal overview

- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



"Rowhammer"

Remote FIA



"CLKscrew"

2003 2014 2018 2019

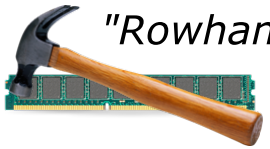
Timing Attacks

Remote SCA

About remote hardware attacks

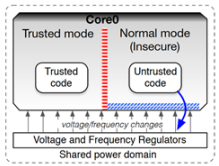
A temporal overview

- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



"Rowhammer"

Remote FIA



"CLKscrew"

2003 — 2014 — 2018 — 2019

Timing Attacks

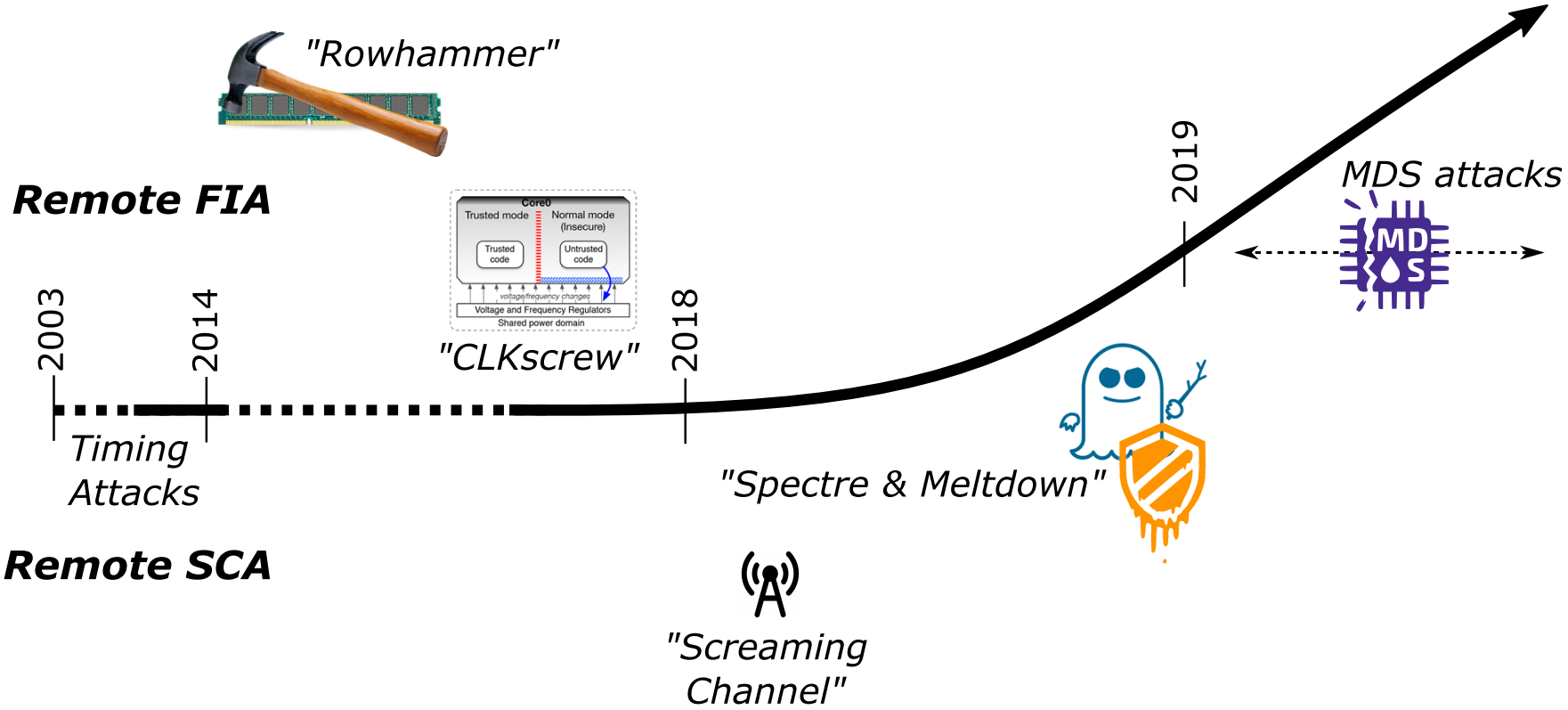
Remote SCA

"Screaming Channel"

About remote hardware attacks

A temporal overview

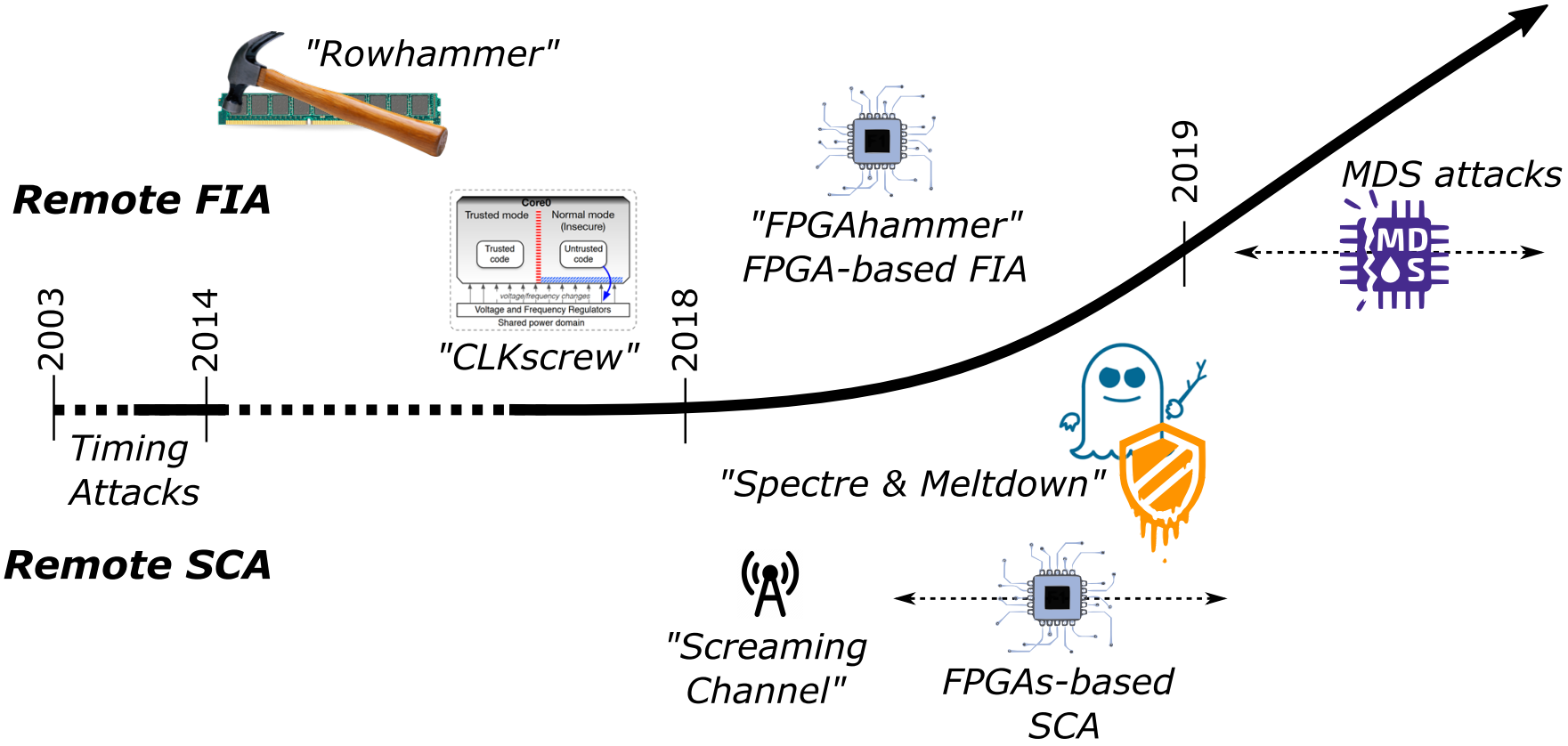
- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



About remote hardware attacks

A temporal overview

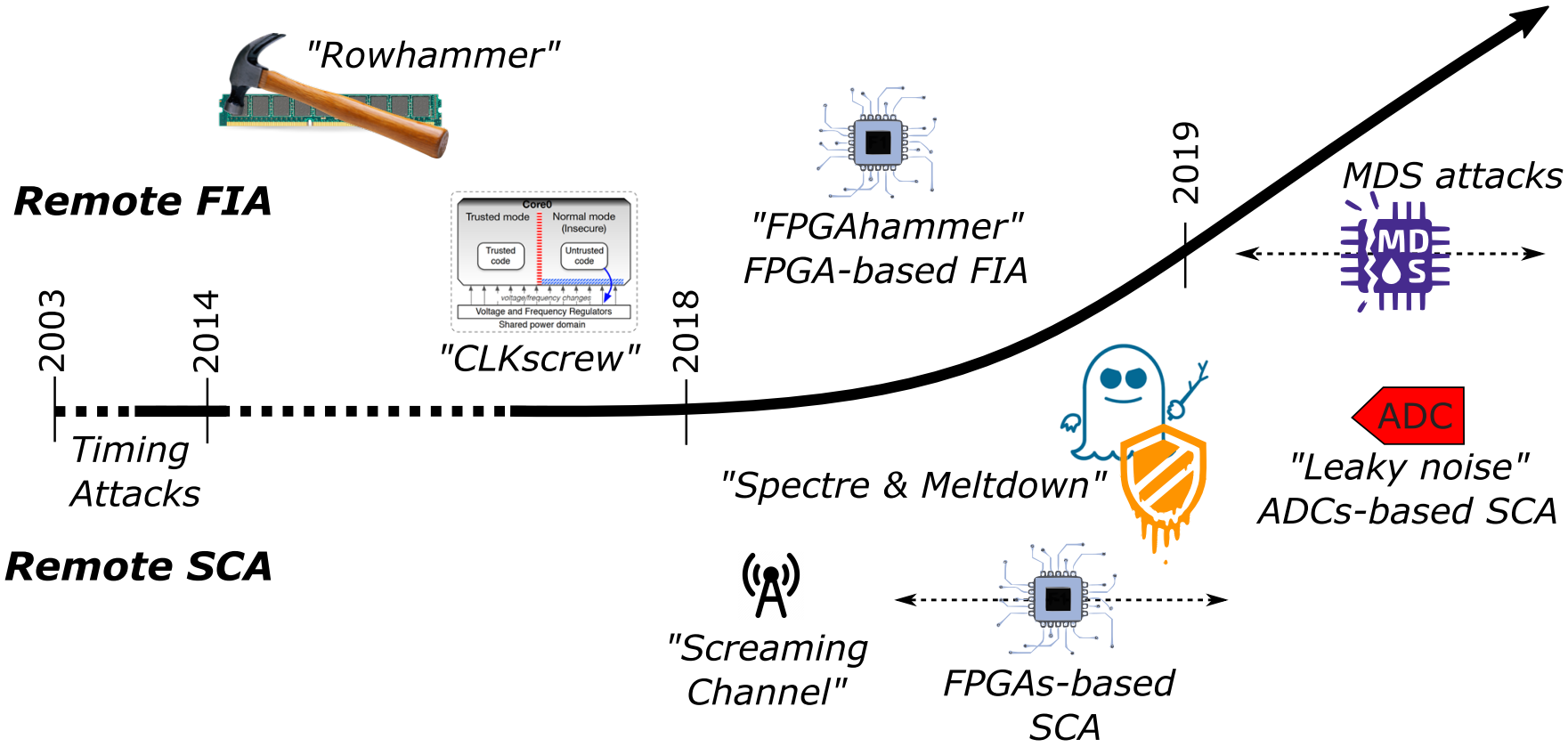
- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



About remote hardware attacks

A temporal overview

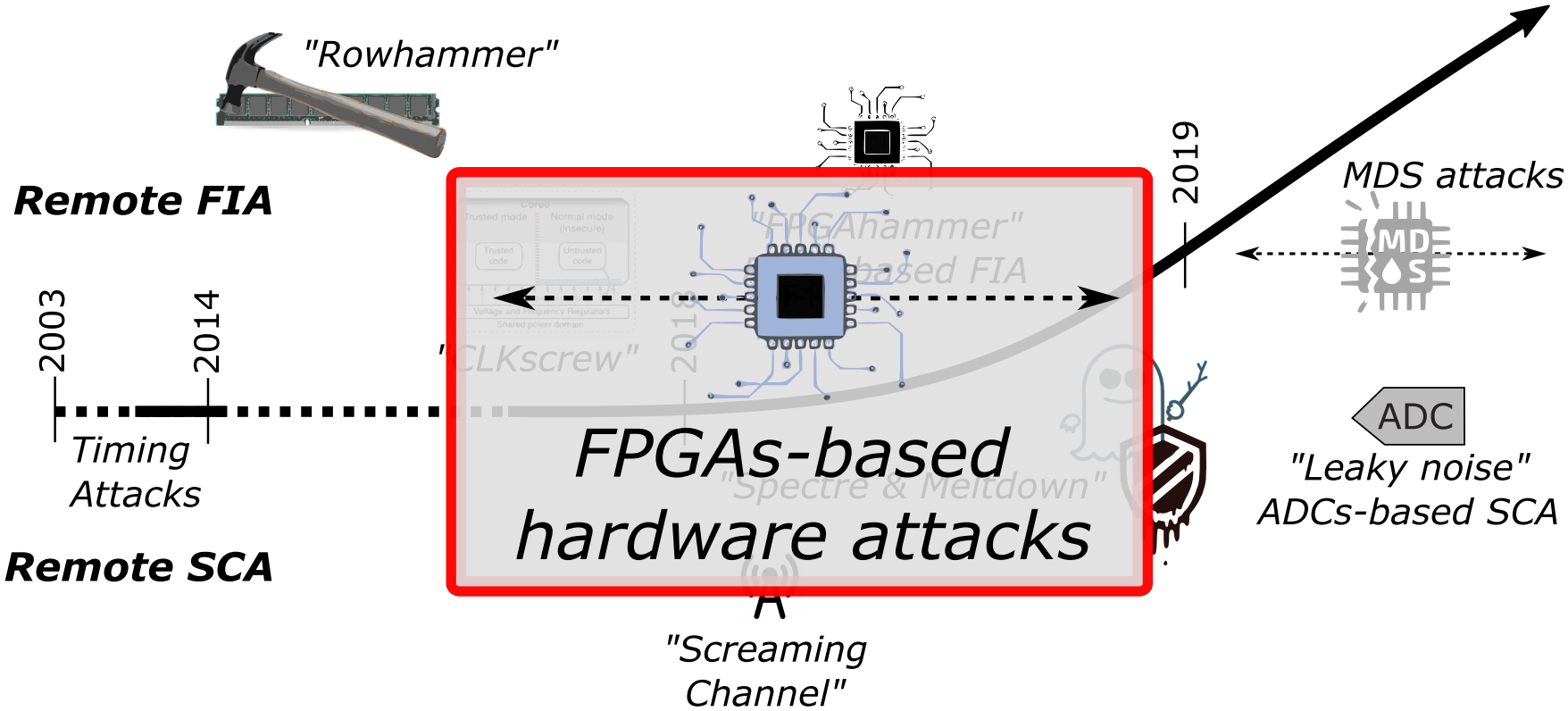
- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



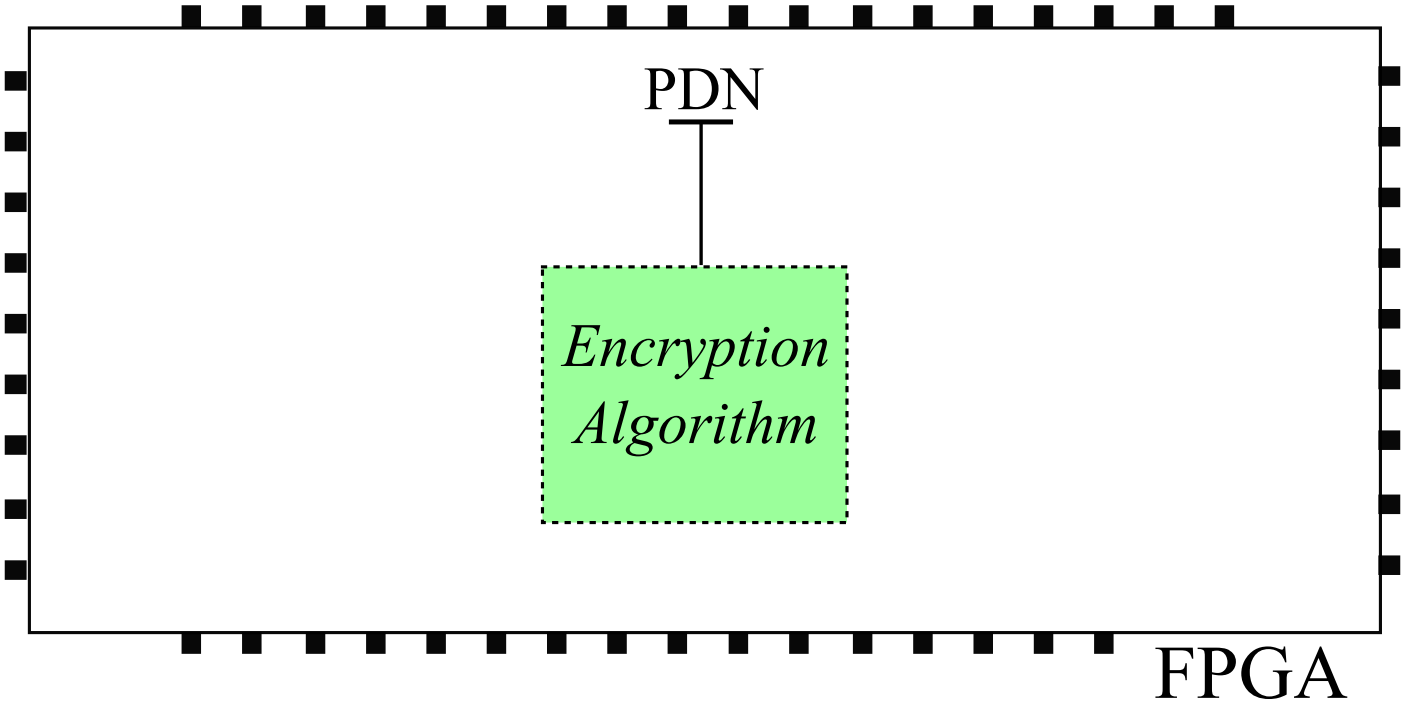
About remote hardware attacks

A temporal overview

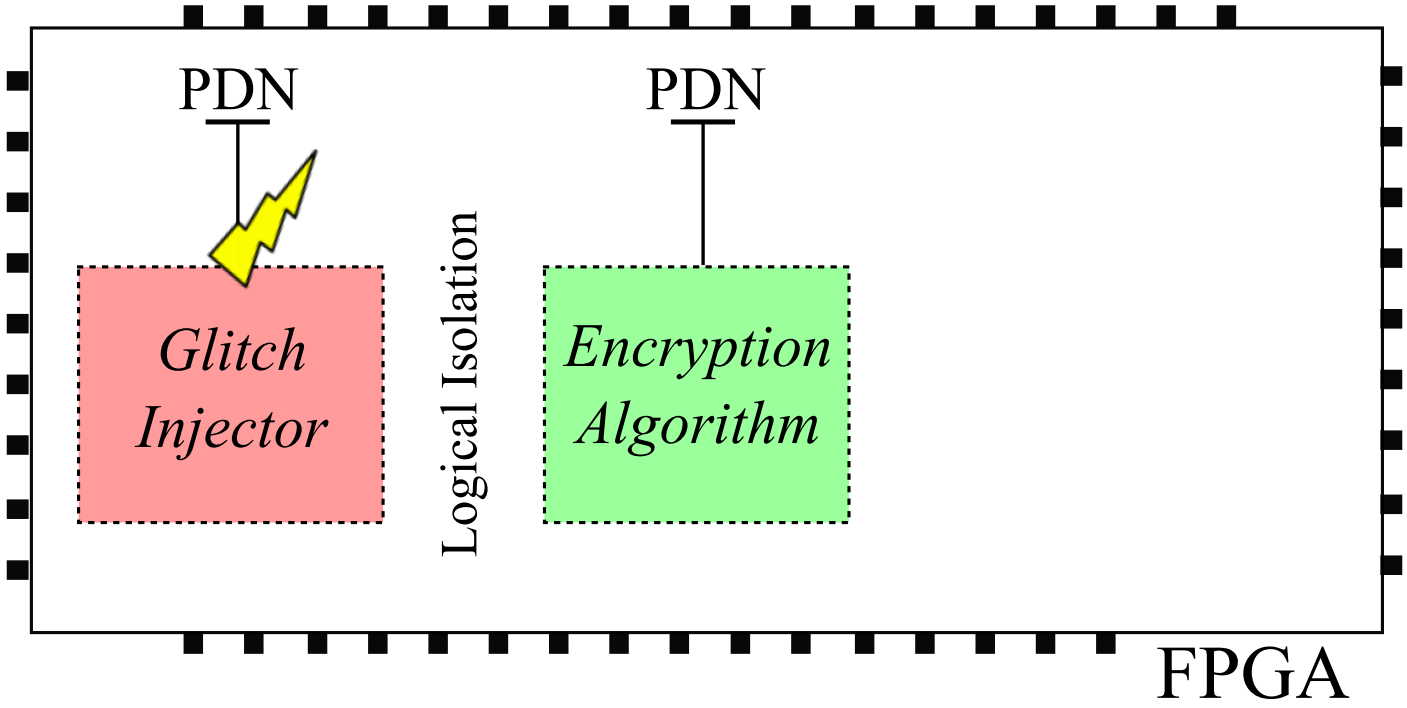
- Remote hardware attack topic keeps on gaining in popularity:
  - Emergence of cloud services, IoT, decentralized computing



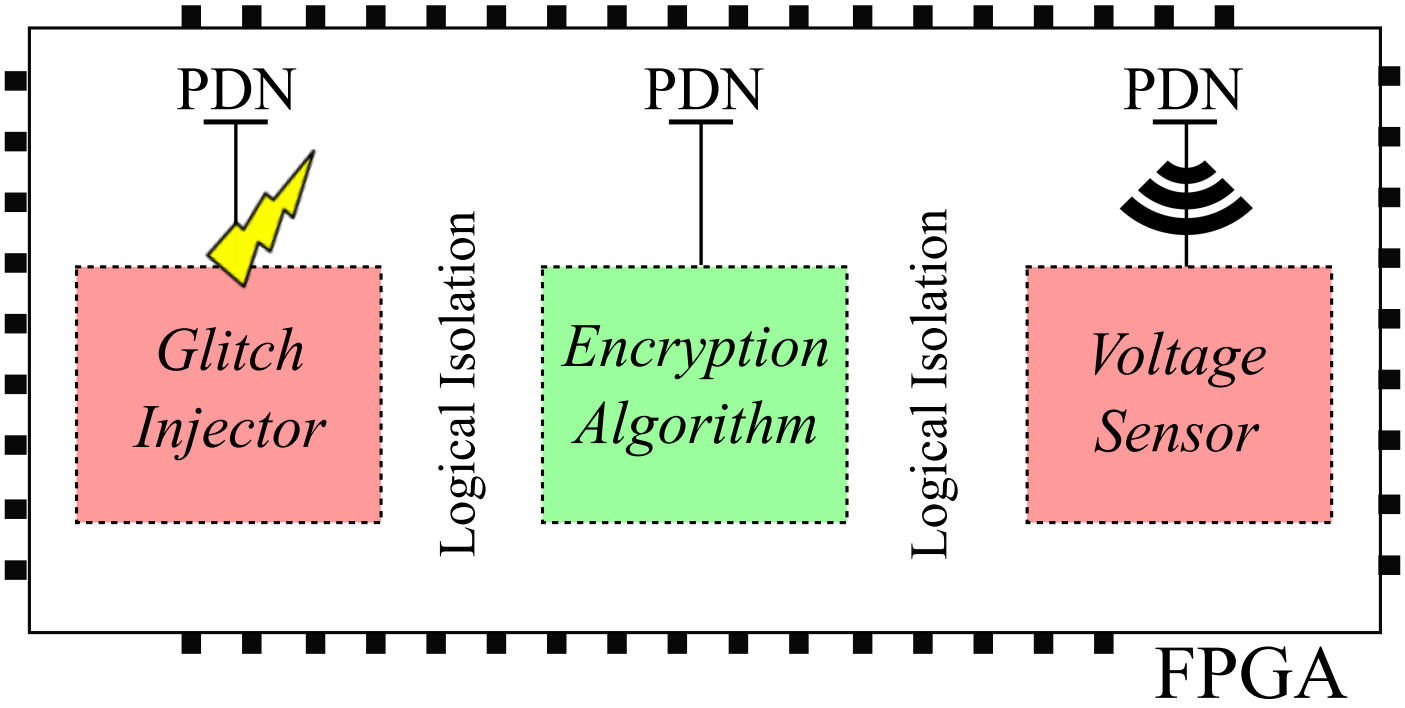
- Usual hardware attacks can be entirely reproduced within FPGA logic:
  - Encryption **algorithm** implementation.
  - Voltage glitch **injector** implementation (Krautter et al).
  - Voltage **sensor** implementation (Schellenberg et al).



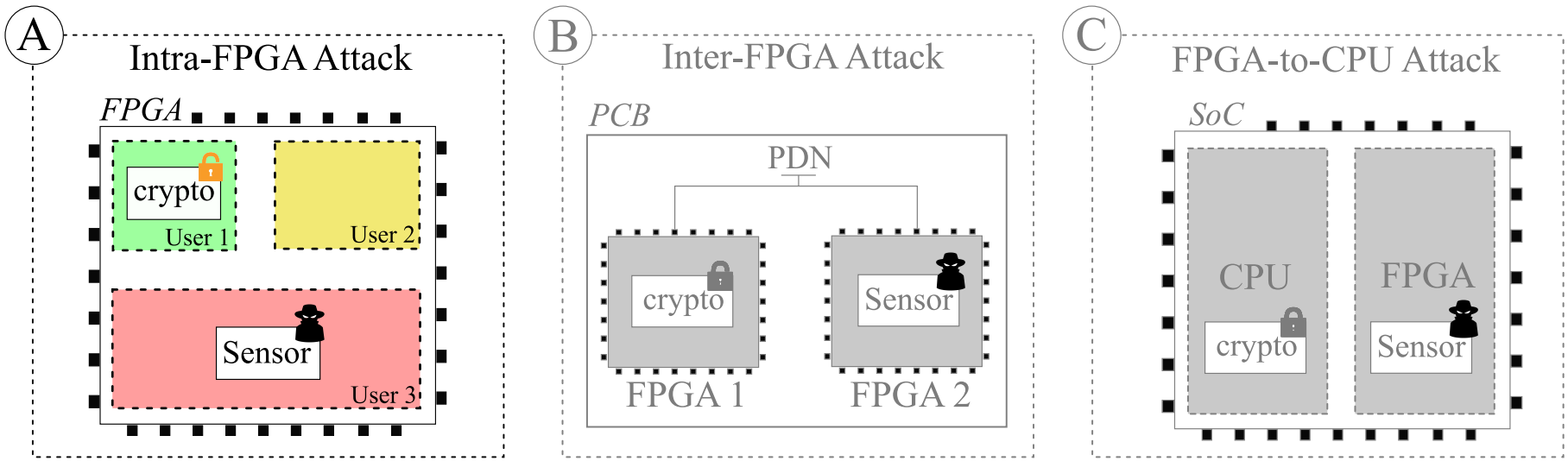
- Usual hardware attacks can be entirely reproduced within FPGA logic:
  - Encryption **algorithm** implementation.
  - Voltage glitch **injector** implementation (Krautter et al).
  - Voltage **sensor** implementation (Schellenberg et al).



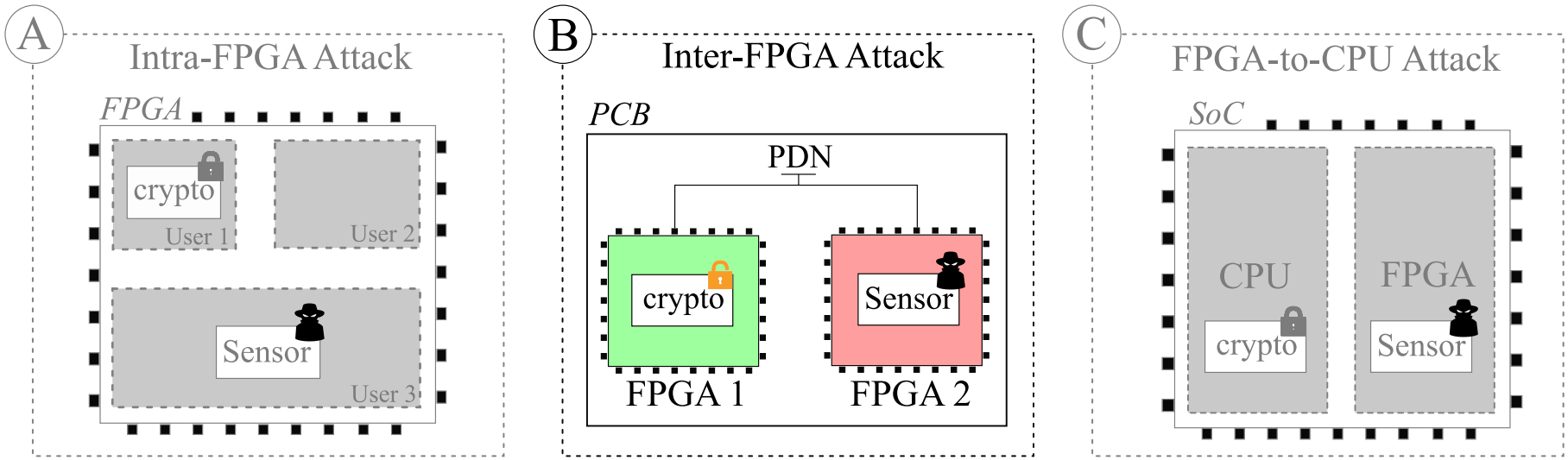
- Usual hardware attacks can be entirely reproduced within FPGA logic:
  - Encryption **algorithm** implementation.
  - Voltage glitch **injector** implementation (Krautter et al).
  - Voltage **sensor** implementation (Schellenberg et al).



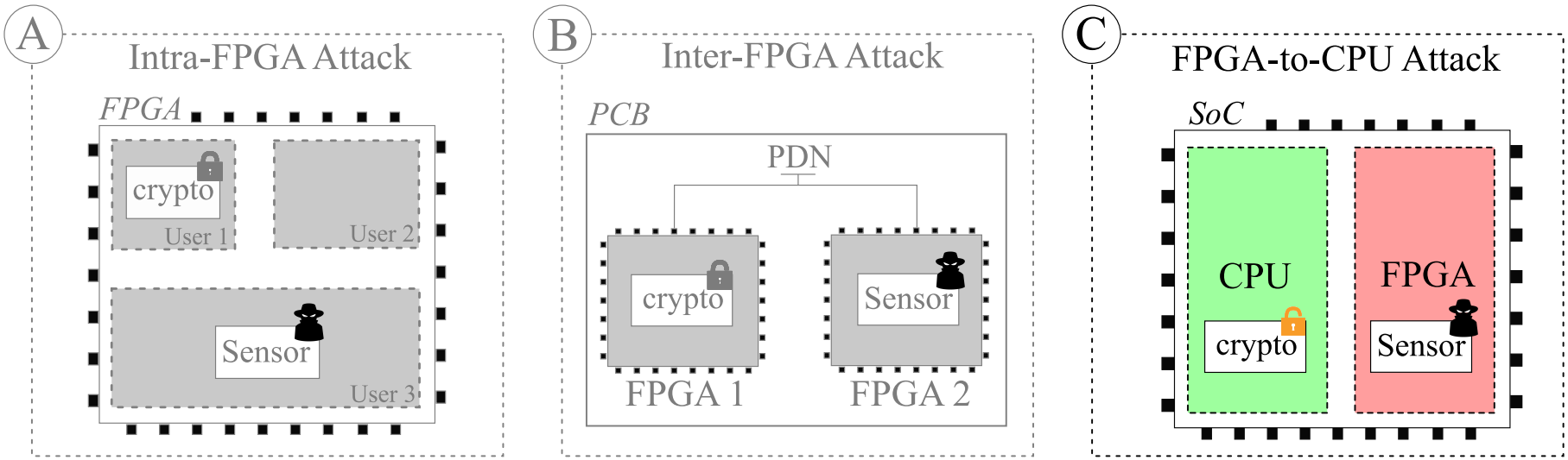
- Target: connected devices that embeds FPGAs.
  - (A) Multi-user FPGAs in **cloud datacenters** (Schellenberg et al).
  - (B) Printed circuit boards **PCB** (Schellenberg et al).
  - (C) **Heterogeneous** connected **SoCs** (Zhao et al).



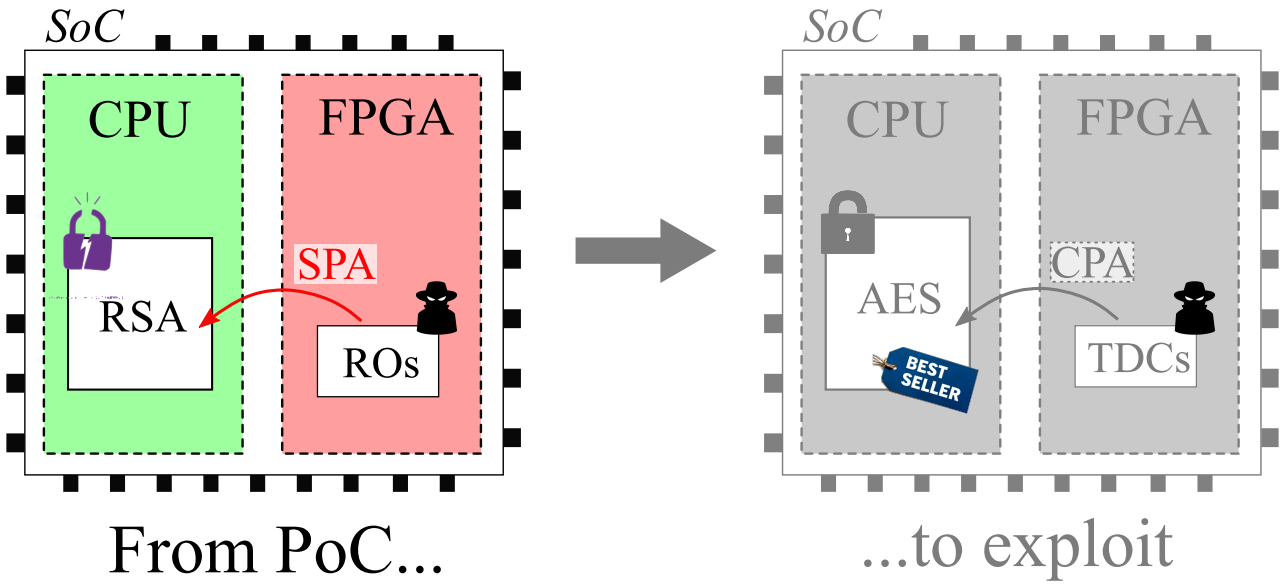
- Target: connected devices that embeds FPGAs.
  - (A) Multi-user FPGAs in **cloud datacenters** (Schellenberg et al).
  - (B) Printed circuit boards **PCB** (Schellenberg et al).
  - (C) **Heterogeneous** connected **SoCs** (Zhao et al).



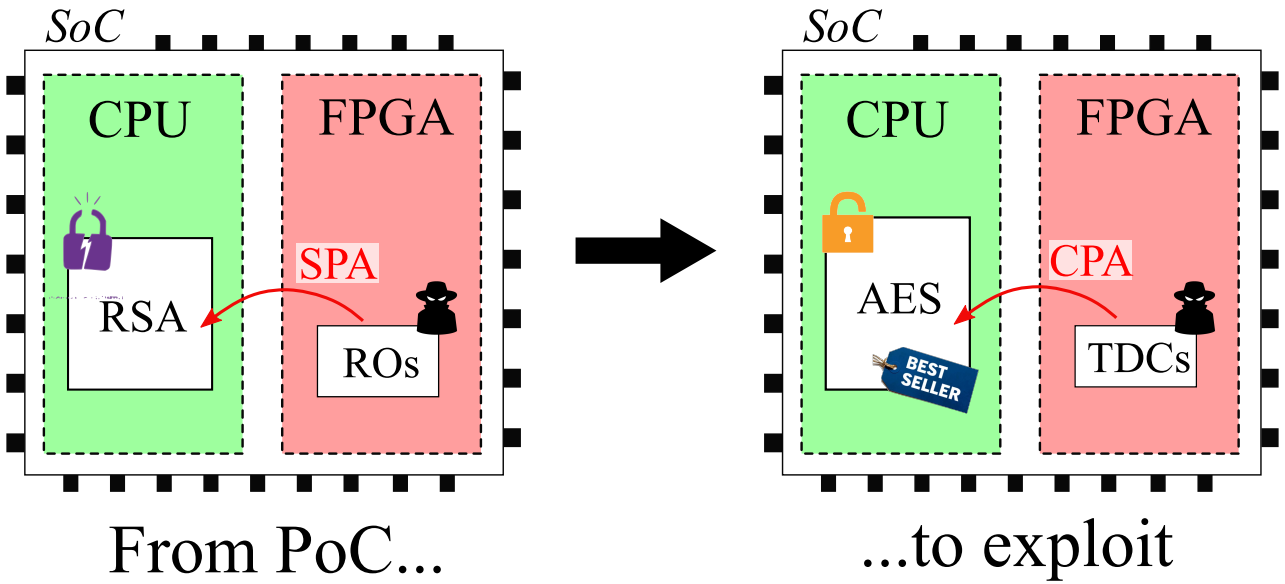
- Target: connected devices that embeds FPGAs.
  - (A) Multi-user FPGAs in **cloud datacenters** (Schellenberg et al).
  - (B) Printed circuit boards **PCB** (Schellenberg et al).
  - (C) **Heterogeneous** connected **SoCs** (Zhao et al).



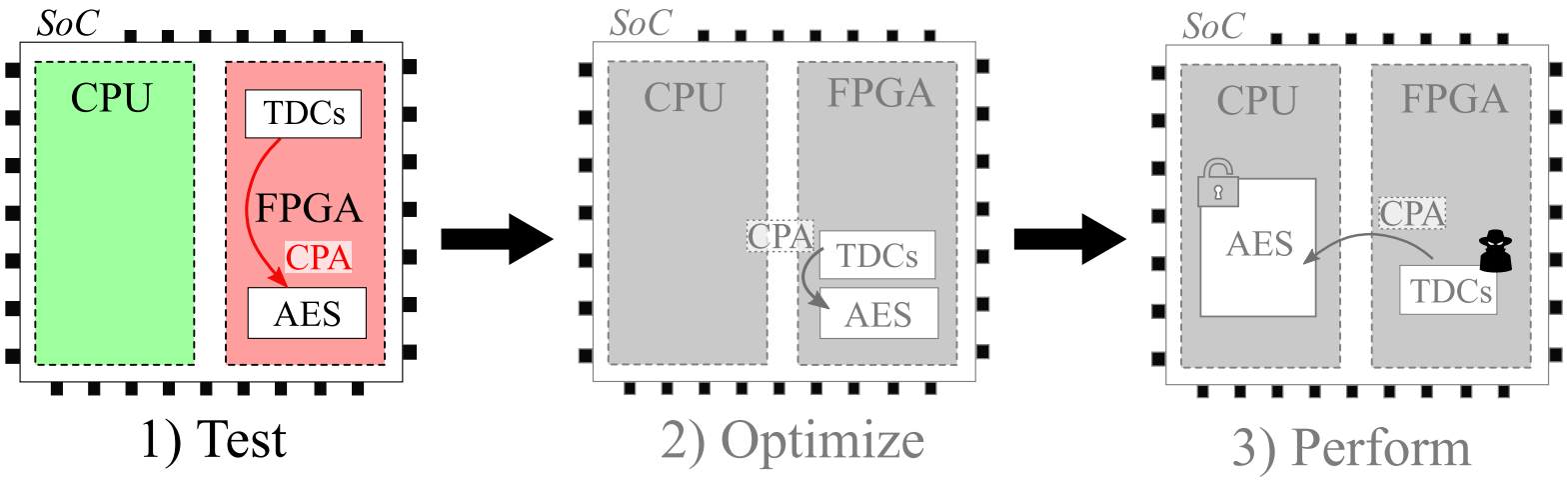
- Already proved:
  - CPU computations can be eavesdropped by FPGA-based sensors.
  - **SPA** attack on **self-written** software RSA using **ROs**.
- Our Goal:
  - Perform FPGA-based **CPA** attacks against **open-source** and **deployed** software AES implementations.



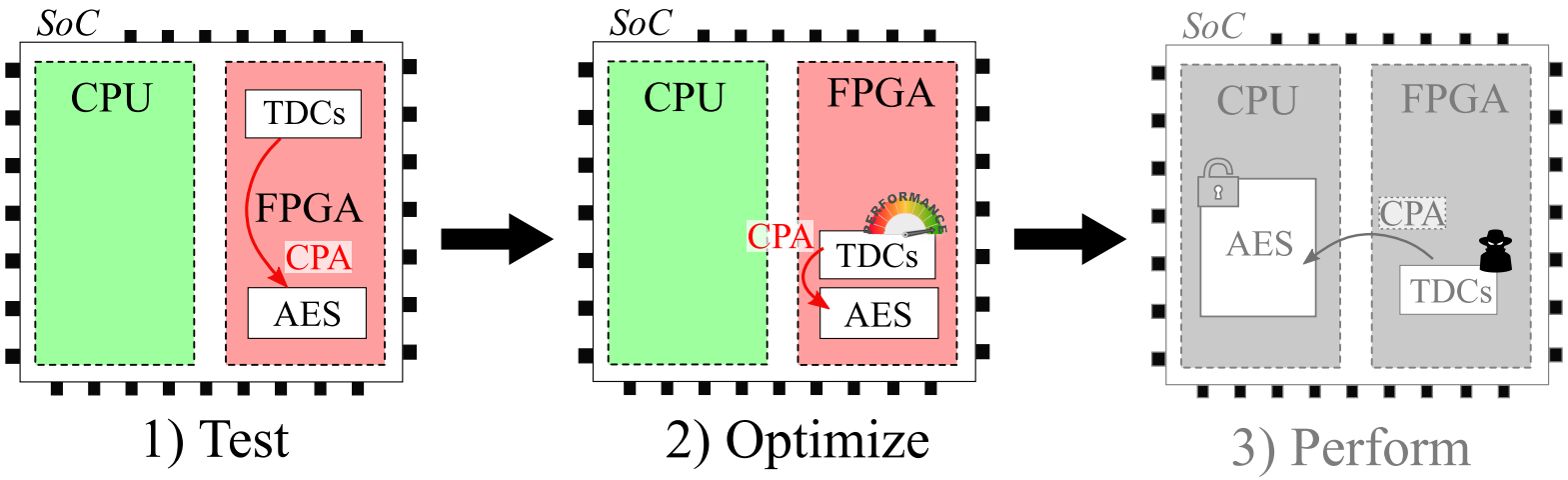
- Already proved:
  - CPU computations can be eavesdropped by FPGA-based sensors.
  - **SPA** attack on **self-written** software RSA using **ROs**.
- Our Goal:
  - Perform FPGA-based **CPA** attacks against **open-source** and **deployed** software AES implementations.



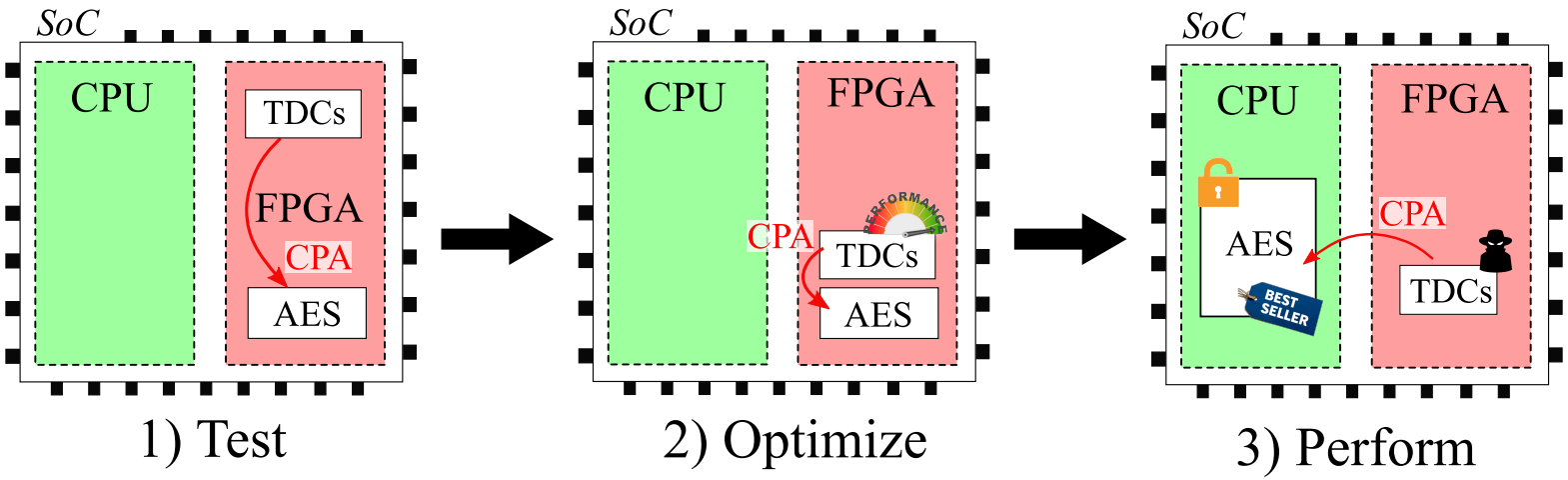
- Iterative implementation:
  1. **Test** SCA on **hardware** AES implementation.
  2. **Optimize** setup toward SCA on software AES.
  3. **Perform** SCA on **software** AES implementations.



- Iterative implementation:
  1. **Test** SCA on **hardware** AES implementation.
  2. **Optimize** setup toward SCA on software AES.
  3. **Perform** SCA on **software** AES implementations.

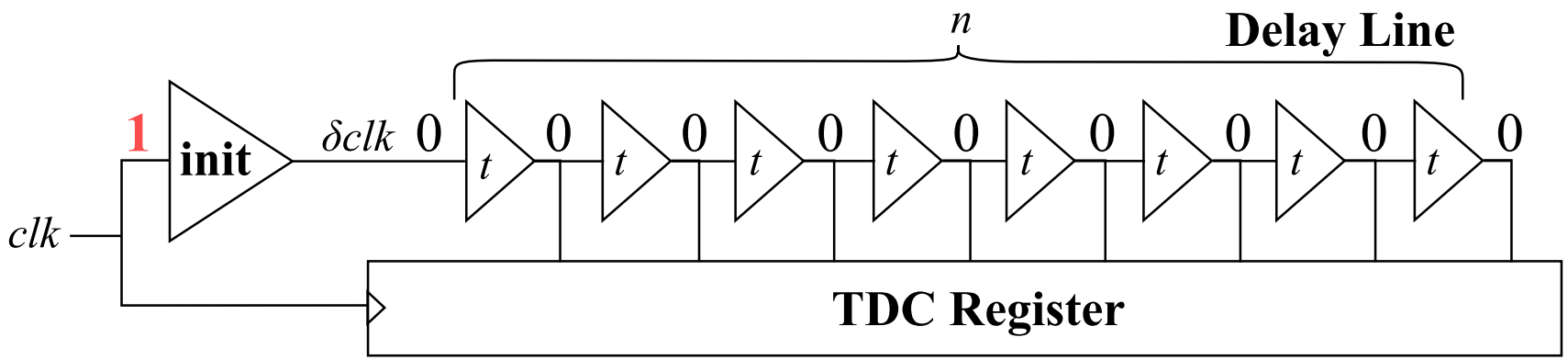


- Iterative implementation:
  1. **Test** SCA on **hardware** AES implementation.
  2. **Optimize** setup toward SCA on software AES.
  3. **Perform** SCA on **software** AES implementations.

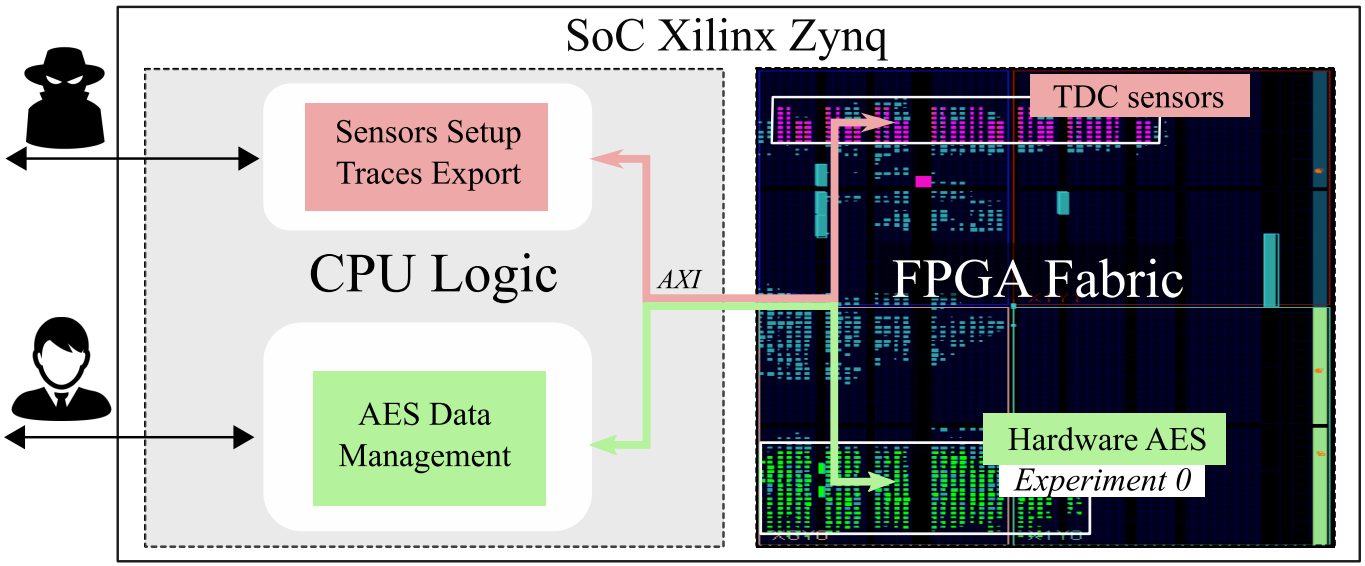


1. **Hardware** AES encryption key retrieval.
2. FPGA-based SCA **Optimization**.
3. **Software** AES encryption key retrieval.

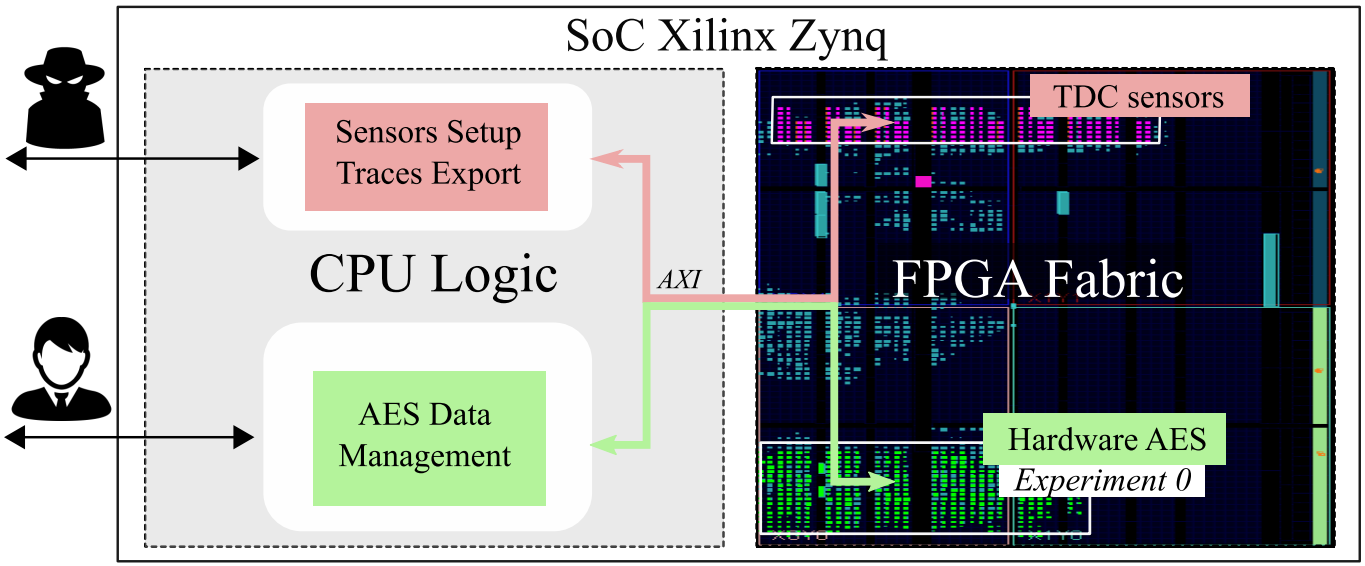
- Power supply fluctuations ⇒ Propagation delay variations.
- Time-To-Digital converter basics:
  - A *clk* **signal** propagates through a **delay line**.
  - A **register** periodically captures the **delay line** state.



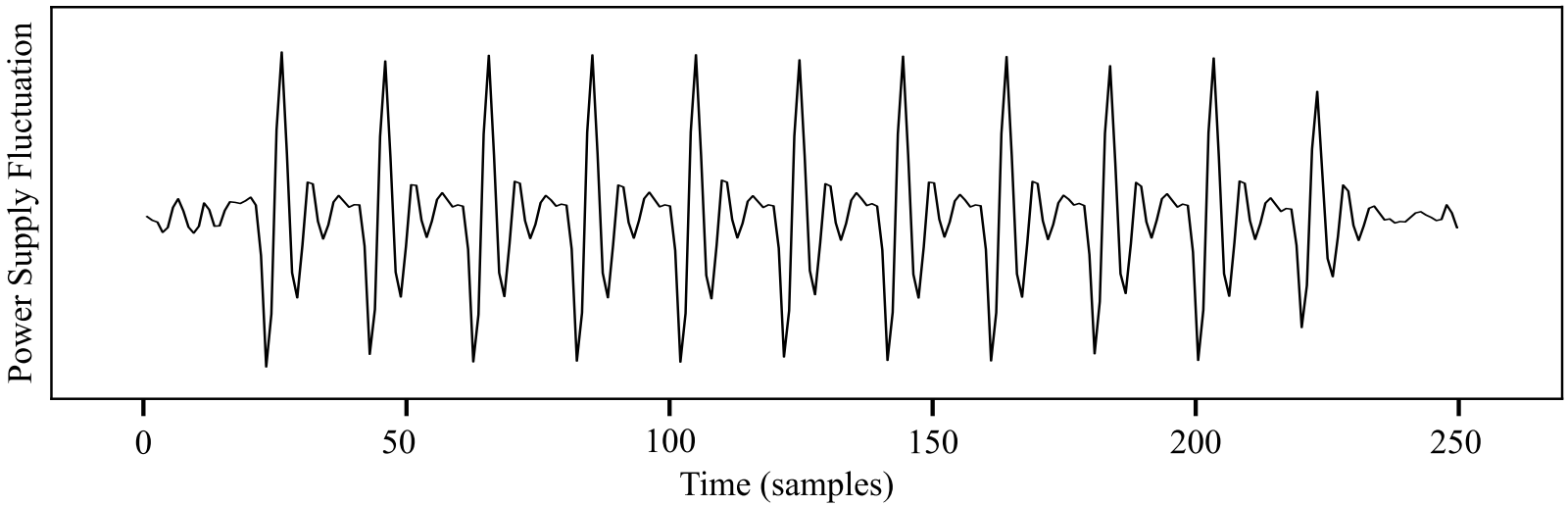
- Target: Xilinx Zynq 7000 heterogeneous SoC
  - FPGA (Xilinx Artix-7) - TDC sensors and AES algorithm
  - CPU (ARM Cortex-A9) - Traces export and AES management



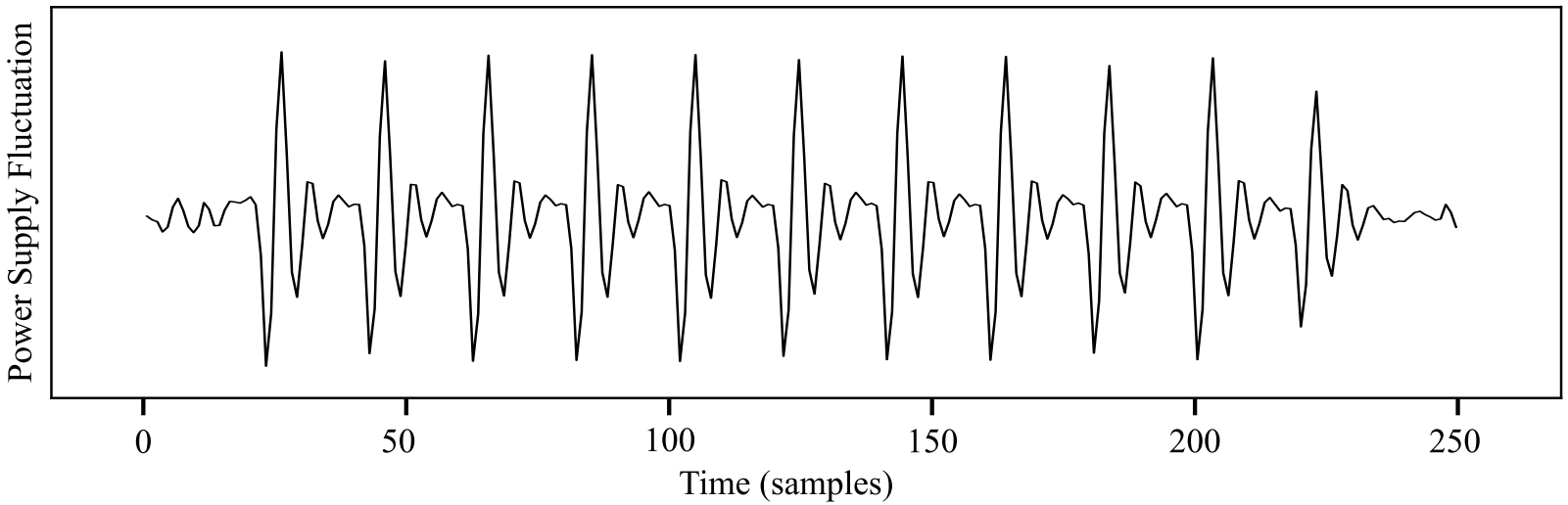
- Target: Xilinx Zynq 7000 heterogeneous SoC
  - FPGA (Xilinx Artix-7) - TDC sensors and AES algorithm
  - CPU (ARM Cortex-A9) - Traces export and AES management
- Experimental setup:
  - TDCs placed horizontally far away from AES => worst case scenario



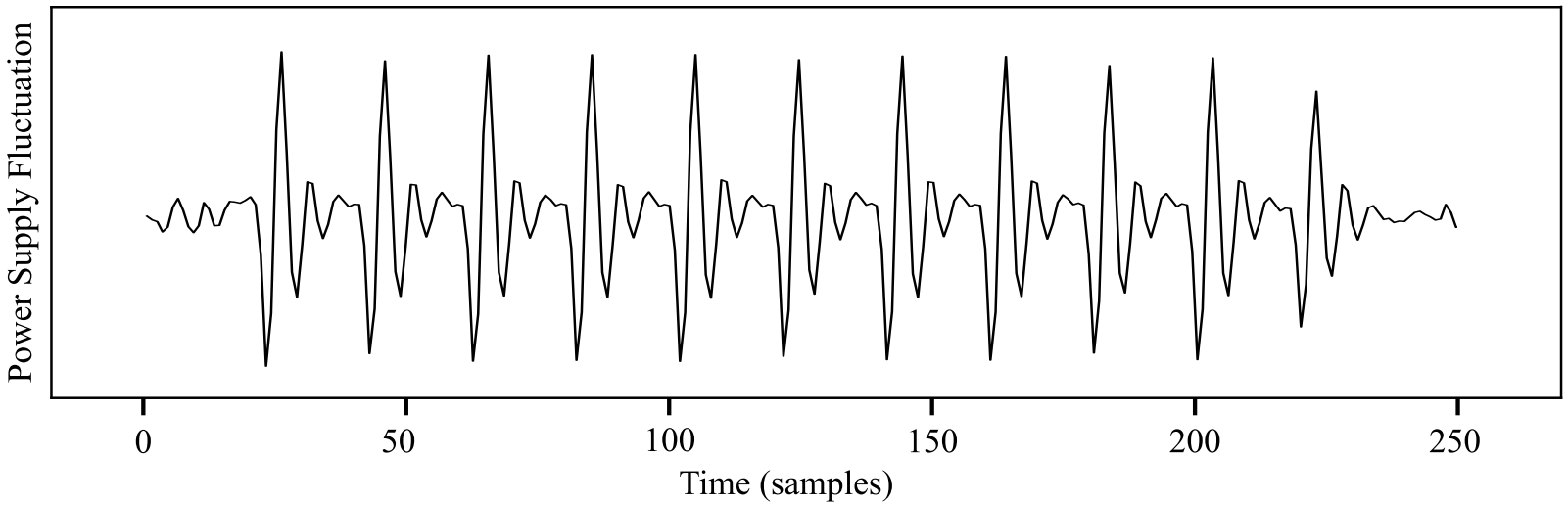
- Custom VHDL AES designed for the attack.
  - Key size **128 bit**, Datapath **128 bit**.



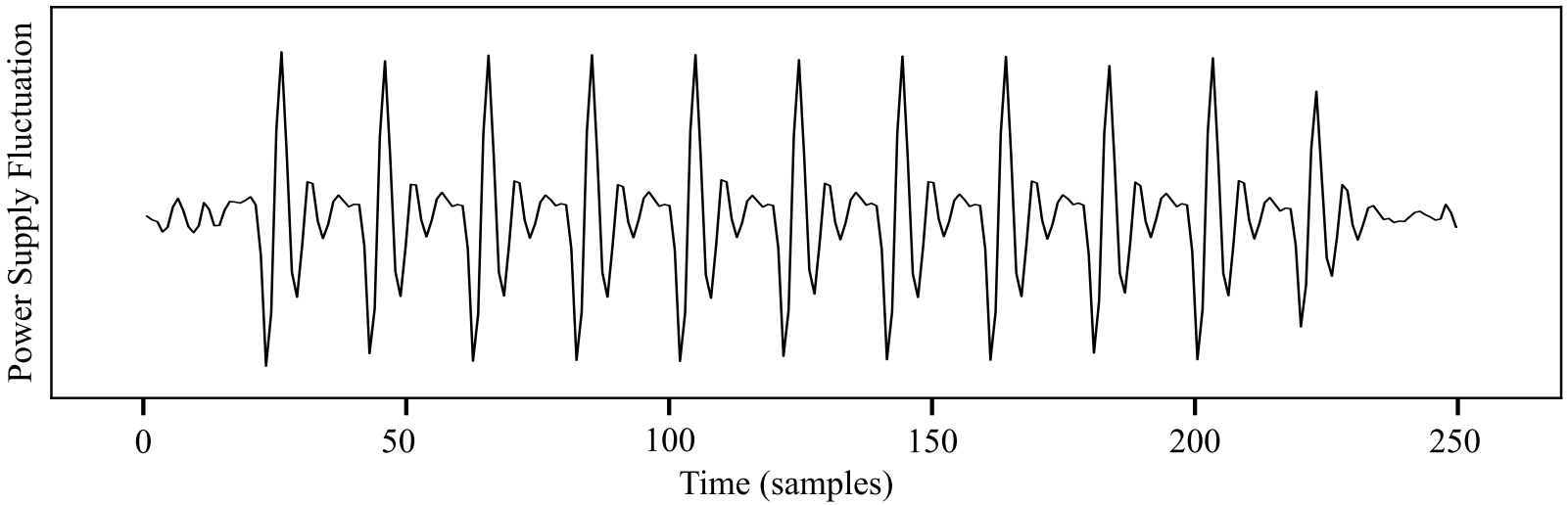
- Custom VHDL AES designed for the attack.
  - Key size **128 bit**, Datapath **128 bit**.
  - AES encryption time @10MHz $\Rightarrow$ **1.1**μs



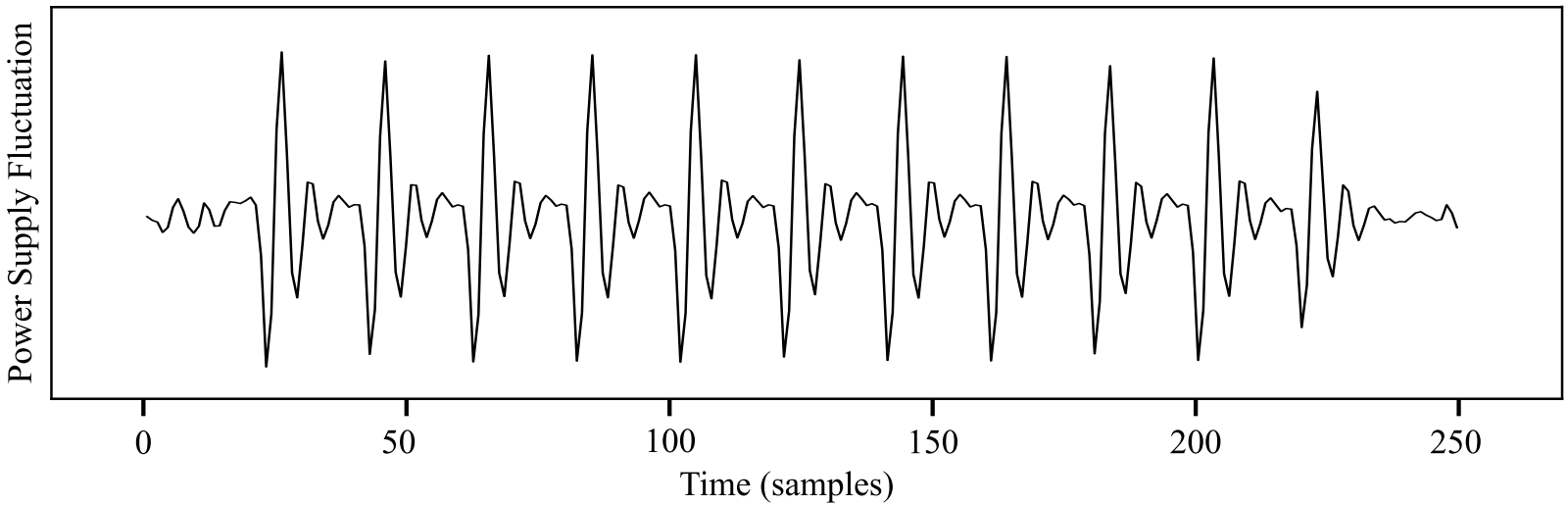
- Custom VHDL AES designed for the attack.
  - Key size **128 bit**, Datapath **128 bit**.
  - AES encryption time @10MHz $\Rightarrow$ **1.1**μs
  - Synchronisation $\Rightarrow$ Encryption and measurement launched **simultaneously**.



- Custom VHDL AES designed for the attack.
  - Key size **128 bit**, Datapath **128 bit**.
  - AES encryption time @10MHz $\Rightarrow$ **1.1**μs
  - Synchronisation $\Rightarrow$ Encryption and measurement launched **simultaneously**.
  - CPA model $\Rightarrow$ AES Last round $HW[ARK_9 \oplus ARK_{10}]$



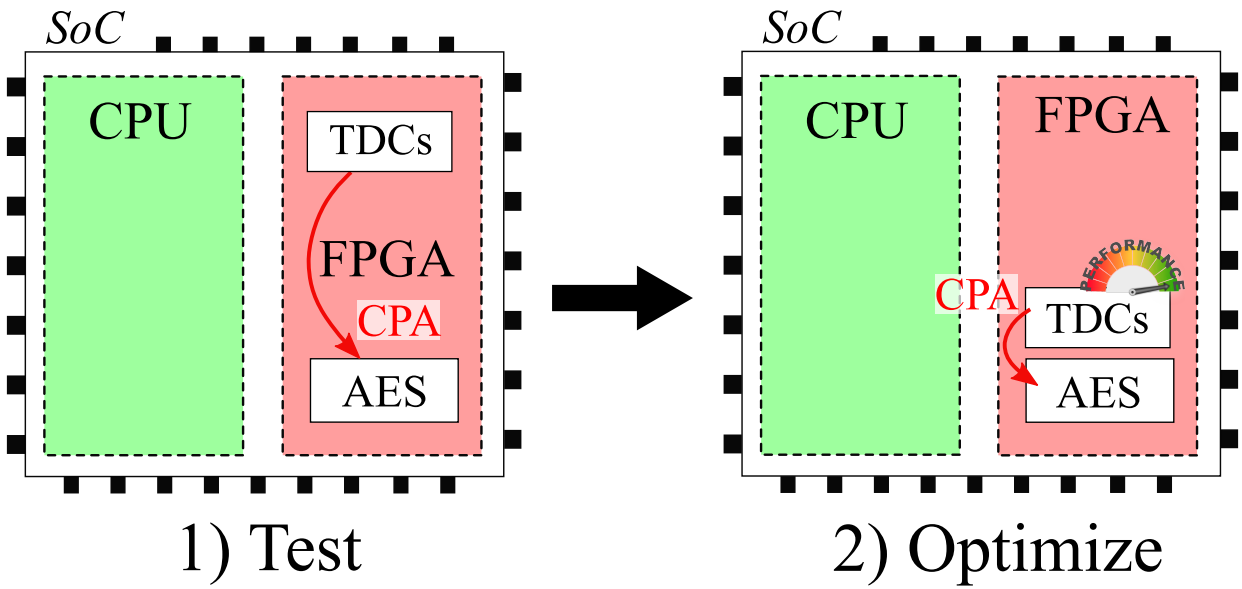
- Custom VHDL AES designed for the attack.
  - Key size **128 bit**, Datapath **128 bit**.
  - AES encryption time @10MHz $\Rightarrow$ **1.1**μs
  - Synchronisation $\Rightarrow$ Encryption and measurement launched **simultaneously**.
  - CPA model $\Rightarrow$ AES Last round $HW[ARK_9 \oplus ARK_{10}]$



- Results: number of traces required to infer an AES key byte: **4,483**.

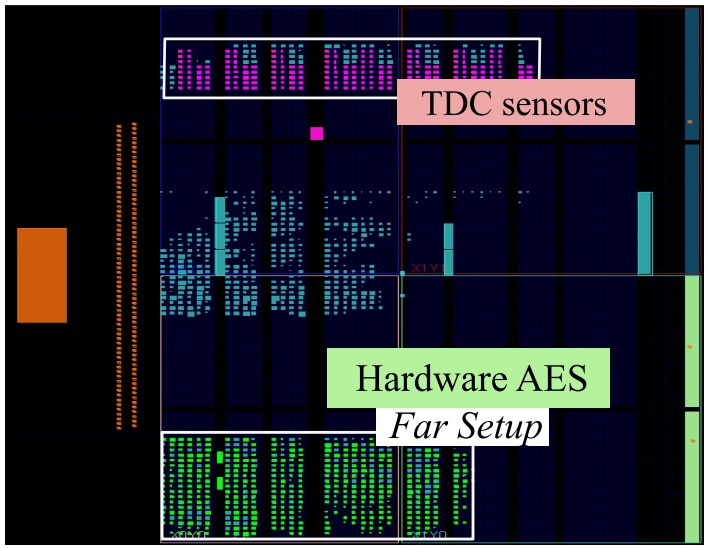
1. **Hardware** AES encryption key retrieval.
2. FPGA-based SCA **Optimization**.
3. **Software** AES encryption key retrieval.

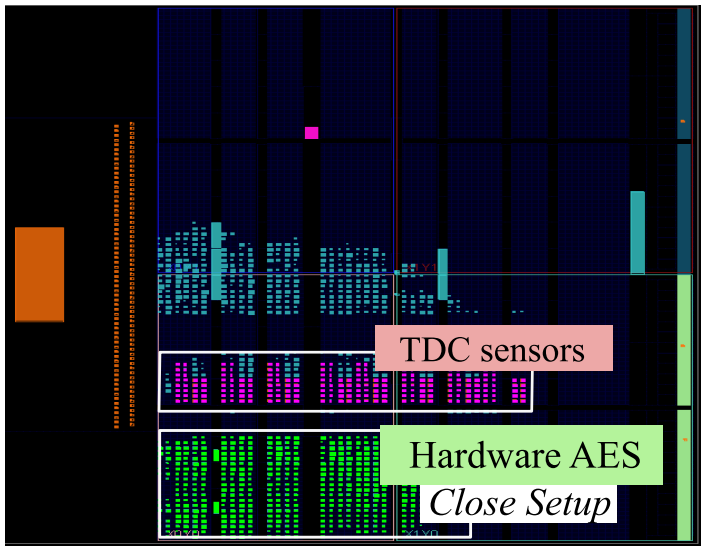
- Several levers:
  - Placement: TDCs **proximity** to the target.
  - Performance: TDCs **structure** modifications.



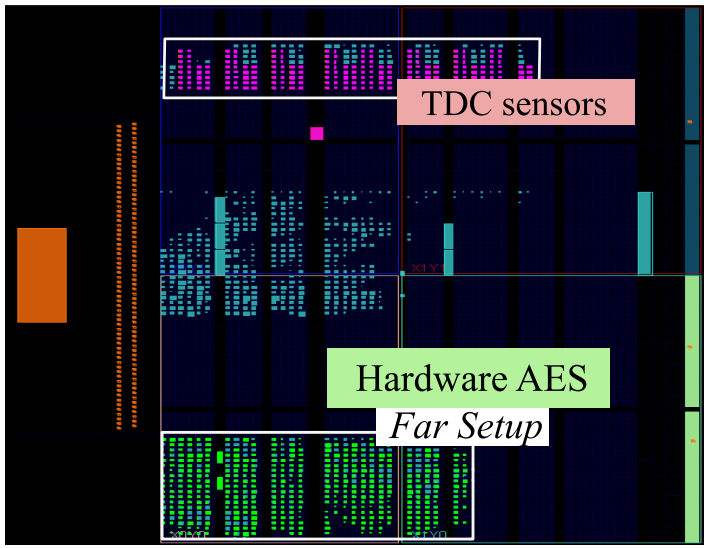
- Assumption:
  - Sensor **proximity** to the target should improve CPA results.
  - Less distance means less acquired noise.

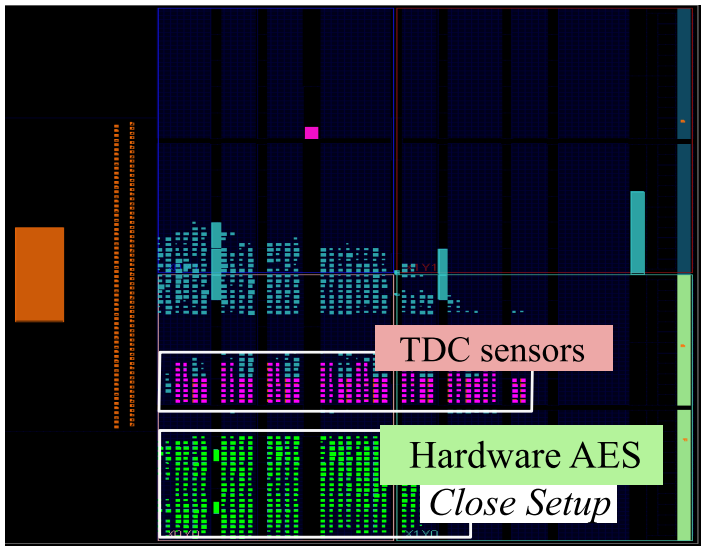
- Assumption:
  - Sensor **proximity** to the target should improve CPA results.
  - Less distance means less acquired noise.
- Experimental Setup:
  - **Far setup:** 80 slices between AES & TDCs.
  - **Close setup:** 6 slices between AES & TDCs.





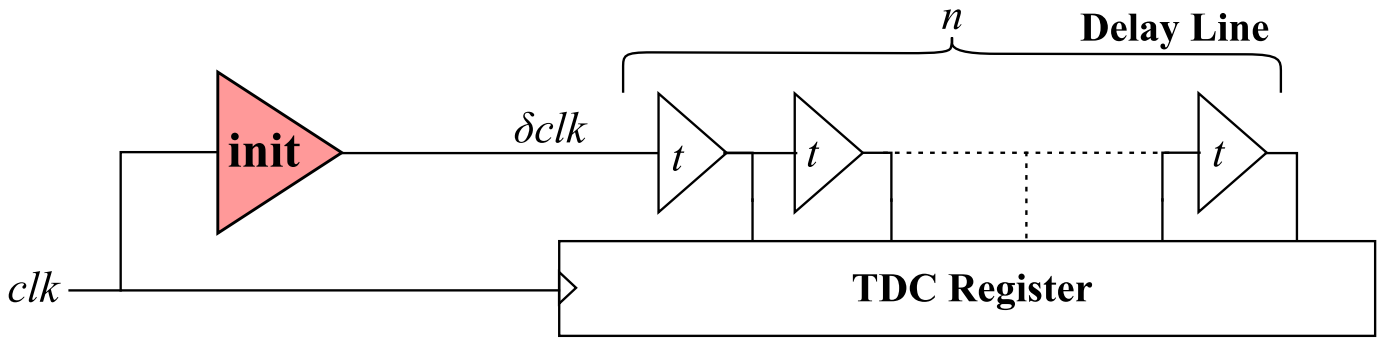
- Assumption:
  - Sensor **proximity** to the target should improve CPA results.
  - Less distance means less acquired noise.
- Experimental Setup:
  - **Far setup:** 80 slices between AES & TDCs.
  - **Close setup:** 6 slices between AES & TDCs.





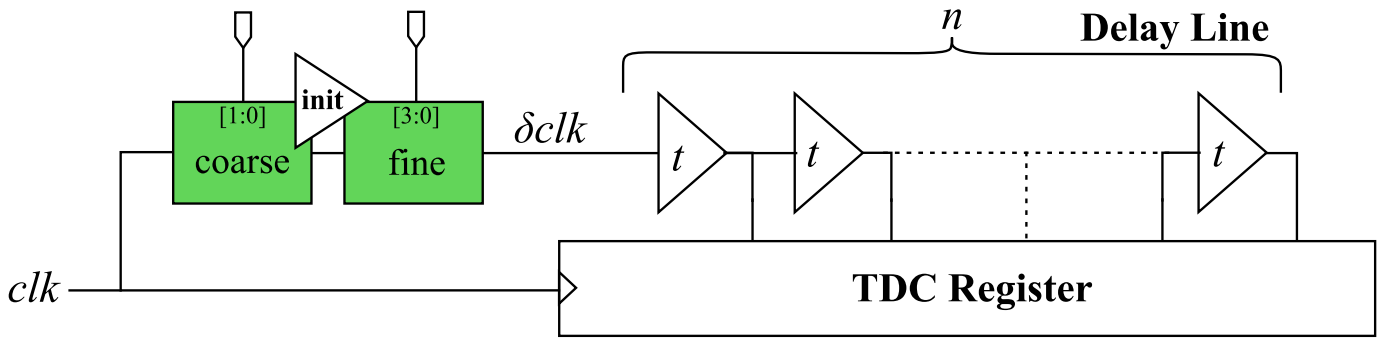
- Results: CPA traces required drops from 4,483 to **3,440**.

- Fixed (classic) **init** delay:
  - Add 180° phase shift to form δclk signal.
  - Integrates voltage fluctuations during a **half** *clk* period.
- Reconfigurable (new) **init** delay:
  - Add $n * 180°$ phase shift to form δclk signal.
  - Integrates voltage fluctuations during $n *$ **half** *clk* period.



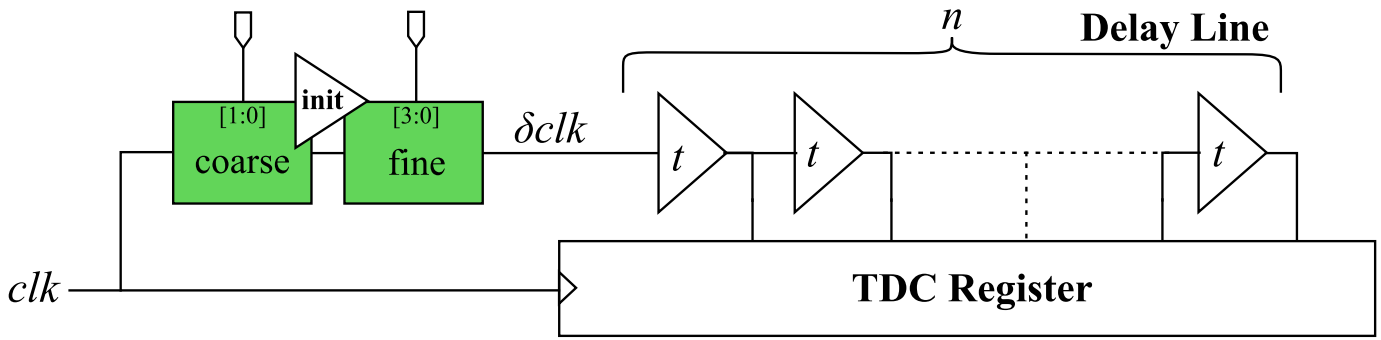
Fixed *init* Delay

- Fixed (classic) **init** delay:
  - Add 180° phase shift to form δclk signal.
  - Integrates voltage fluctuations during a **half** *clk* period.
- Reconfigurable (new) **init** delay:
  - Add $n * 180°$ phase shift to form δclk signal.
  - Integrates voltage fluctuations during $n *$ **half** *clk* period.



Reconfigurable *init* Delay

- Fixed (classic) **init** delay:
  - Add 180° phase shift to form δclk signal.
  - Integrates voltage fluctuations during a **half** *clk* period.
- Reconfigurable (new) **init** delay:
  - Add $n * 180°$ phase shift to form δclk signal.
  - Integrates voltage fluctuations during $n *$ **half** *clk* period.



Reconfigurable *init* Delay

Results: CPA traces required **drops** from 3,440 to **1,381**.

- Results:

TDC Calibration	Average number of Traces	Optimization Factor
No	4,483	/
Placement	3,440	1,30
Init + Placement	1,381	3,25

- Results:

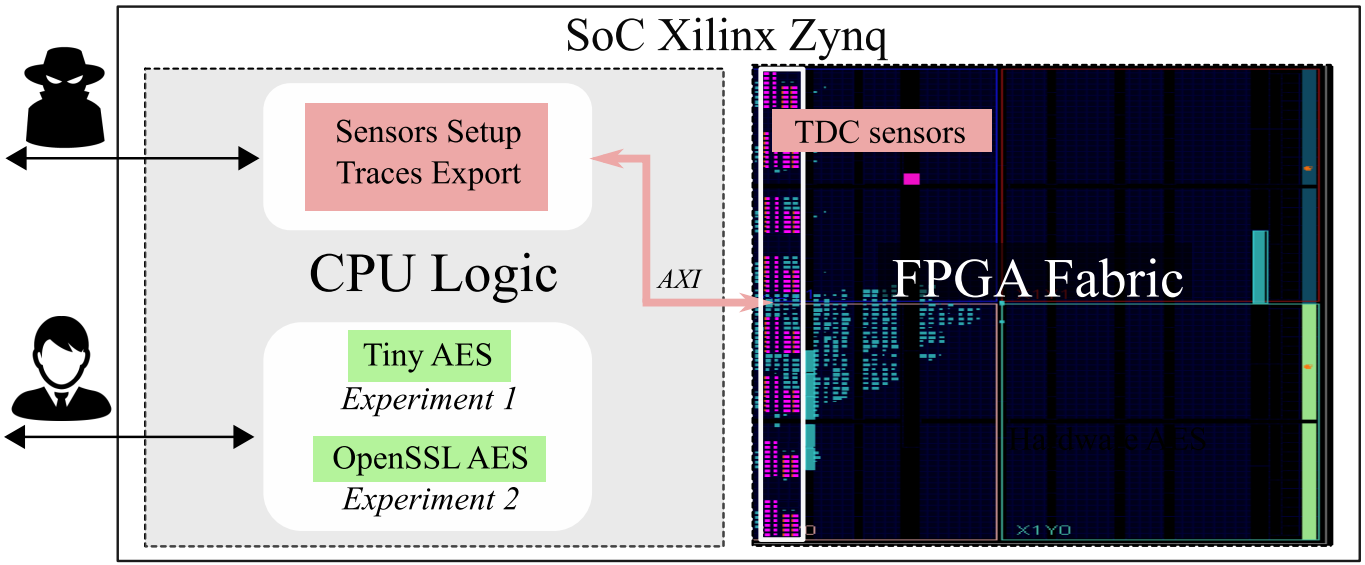
TDC Calibration	Average number of Traces	Optimization Factor
No	4,483	/
Placement	3,440	1,30
Init + Placement	1,381	3,25

- TDCs calibration is **substantial** for the following CPU attacks.
  - Low CPU-to-FPGA side-channel leakage.
  - CPU frequency @666MHz >> TDC frequency @200MHz.

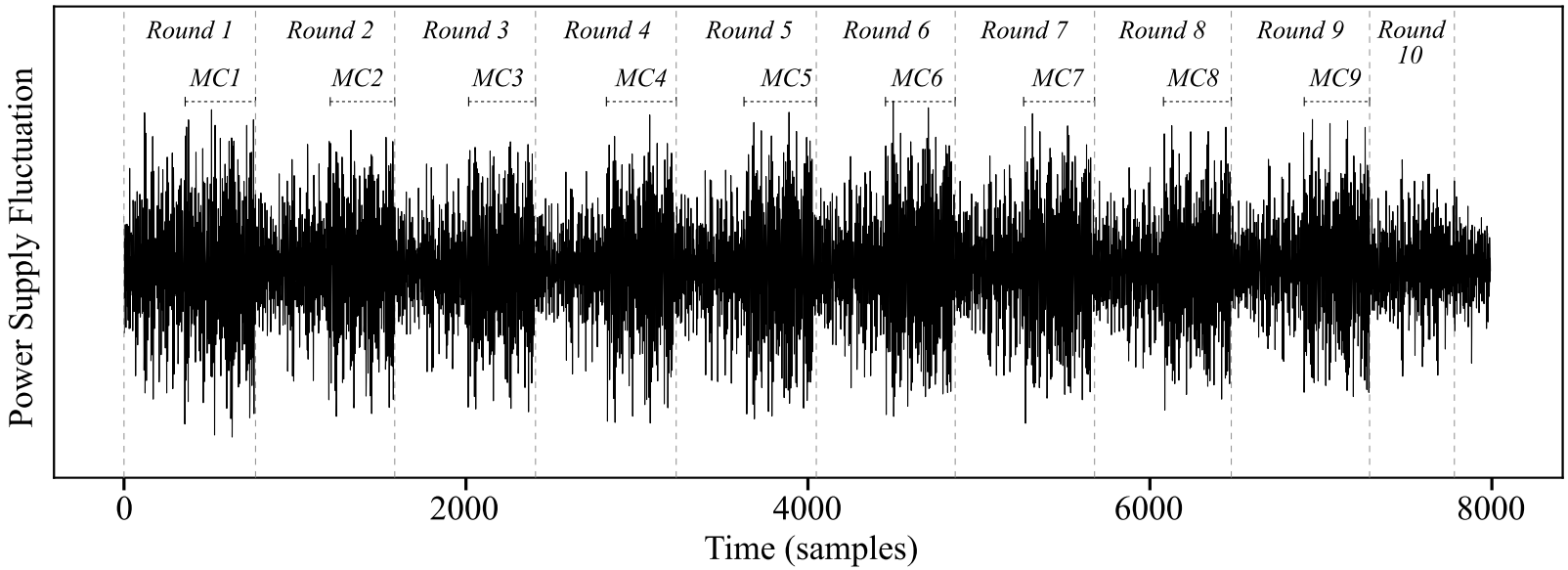
1. **Hardware** AES encryption key retrieval.
2. FPGA-based SCA **Optimization**.
3. **Software** AES encryption key retrieval.

- Two freely-available software AES studied (Bare-metal programming):
  - Tiny AES 128 - **8 bit** data-path.
  - OpenSSL AES 128 - **32 bit** data-path (T-Table)

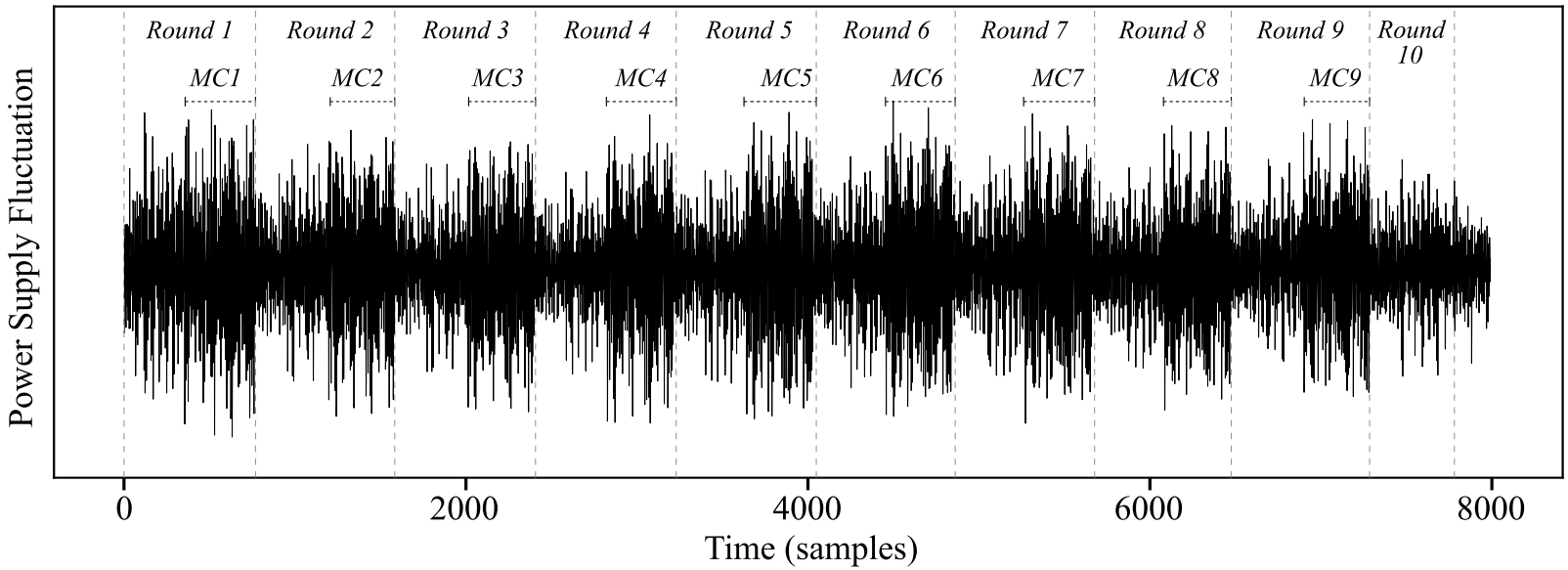
- Two freely-available software AES studied (Bare-metal programming):
  - Tiny AES 128 - **8 bit** data-path.
  - OpenSSL AES 128 - **32 bit** data-path (T-Table)
- Experimental setup:
  - 8 TDCs **placed vertically** on FPGA left part => make sense according to the implemented view.



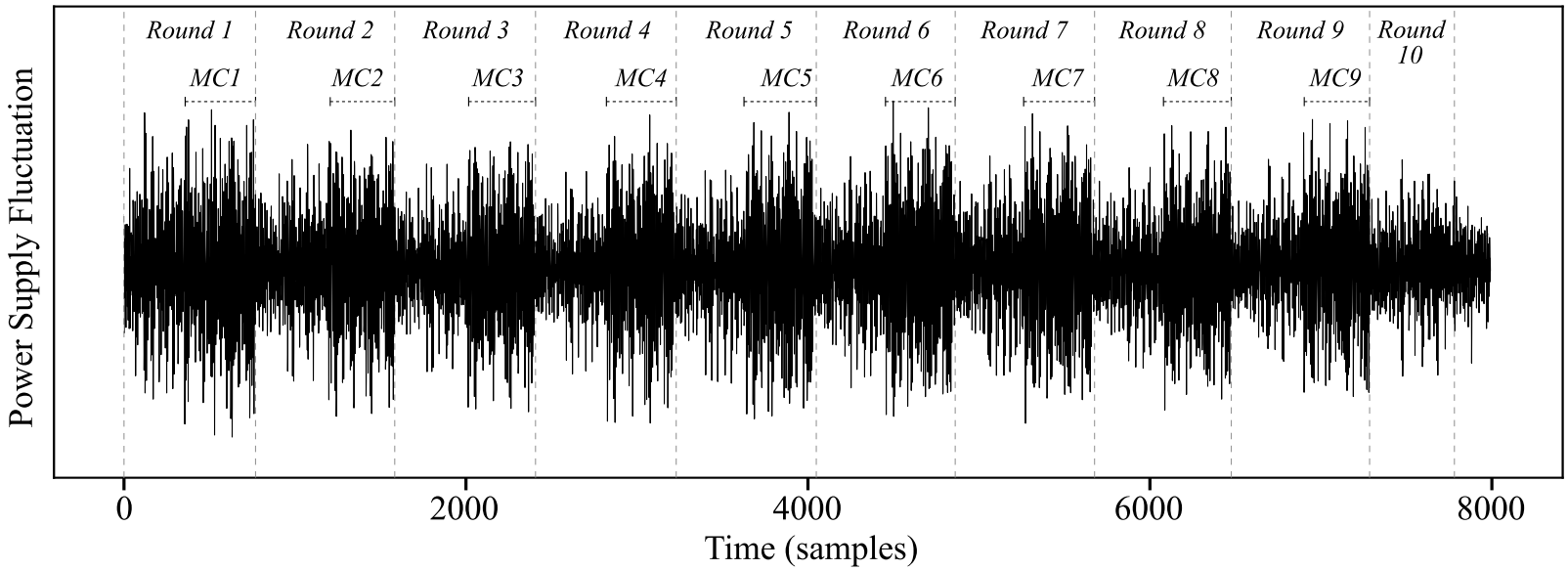
- Small and portable implementation of the AES written in C.
  - Encryption time @666MHz $\Rightarrow$ **40**μs.



- Small and portable implementation of the AES written in C.
  - Encryption time @666MHz $\Rightarrow$ **40**μs.
  - CPA model $\Rightarrow$ AES First round Sbox: $HW[Sbox(k \oplus m)]$.

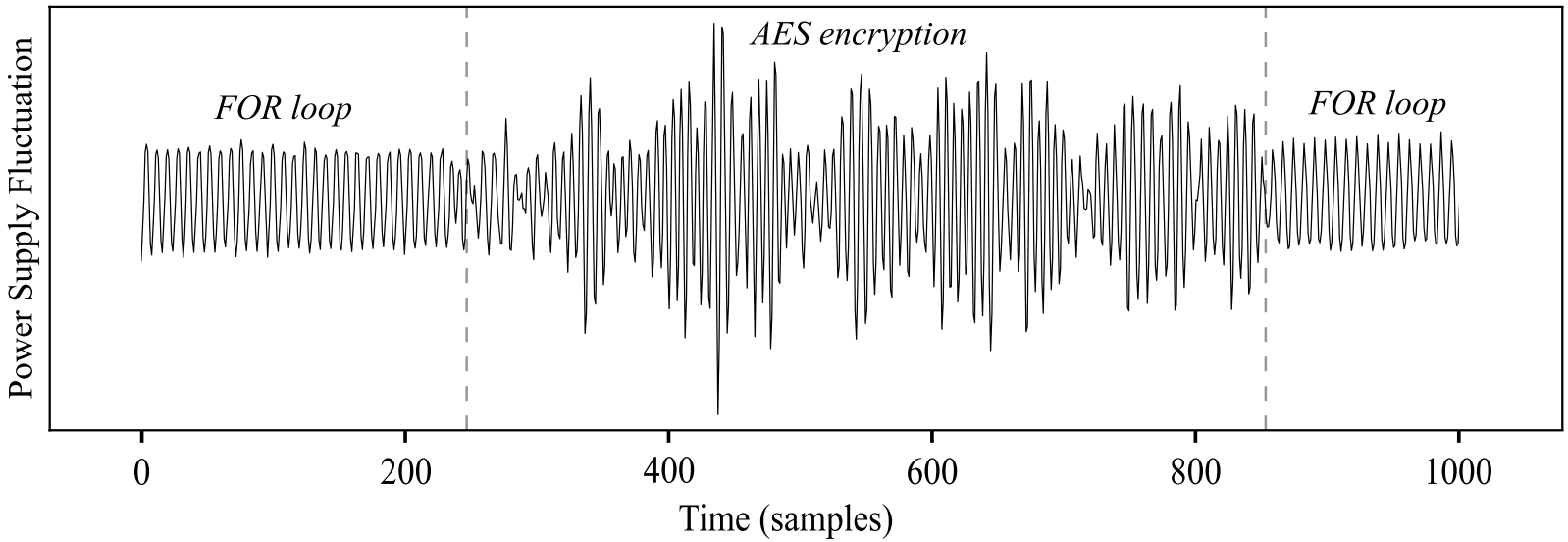


- Small and portable implementation of the AES written in C.
  - Encryption time @666MHz $\Rightarrow$ **40**μs.
  - CPA model $\Rightarrow$ AES First round Sbox: $HW[Sbox(k \oplus m)]$.

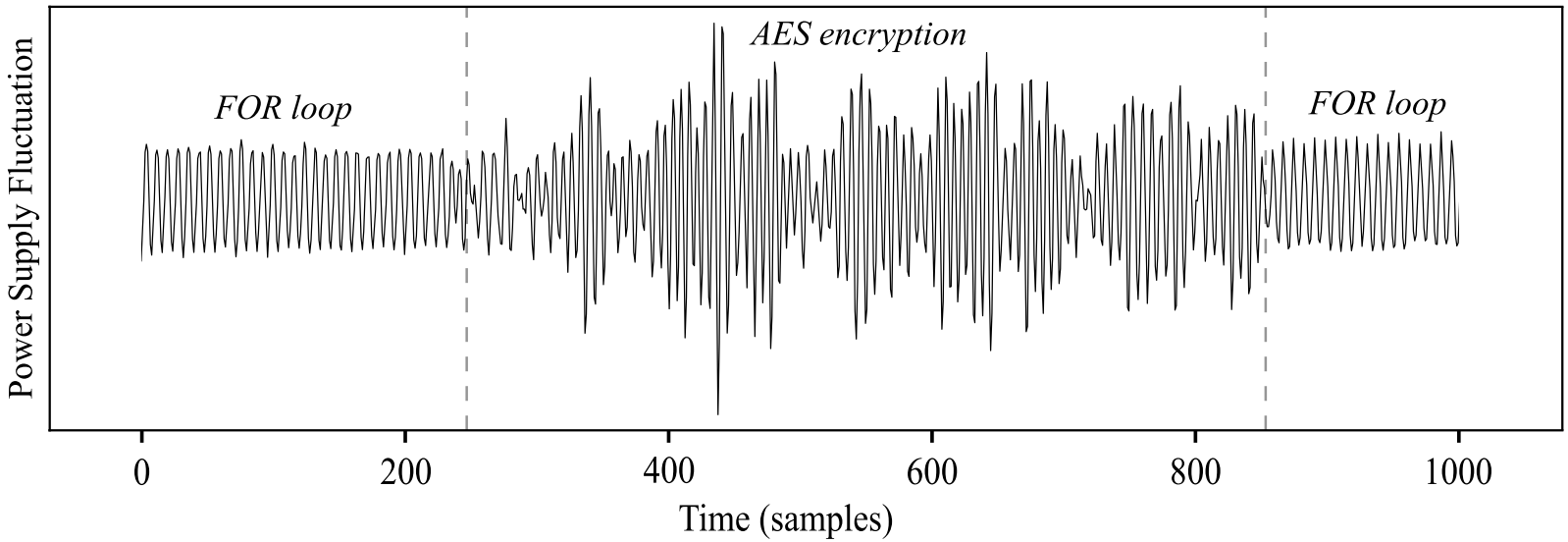


- **Number** of traces required to infer an AES key byte: **111,758**.

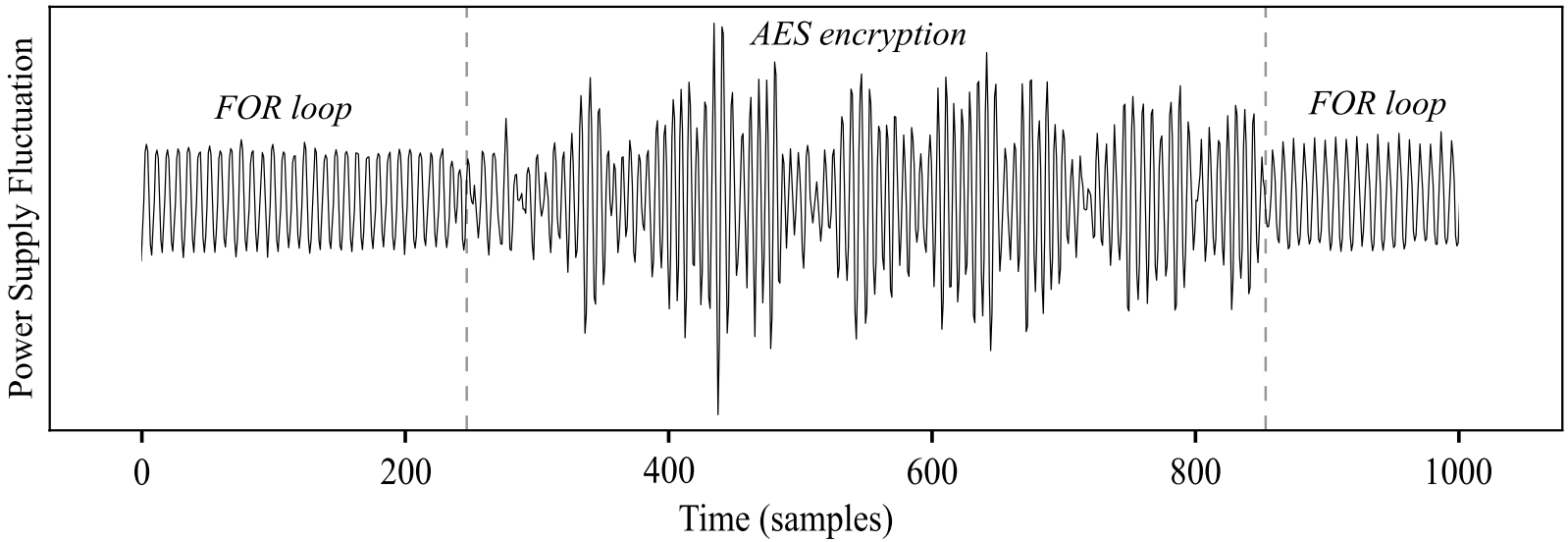
- Crypto library used for secure channels over computer networks.
  - Datapath **32 bit** (T-table).



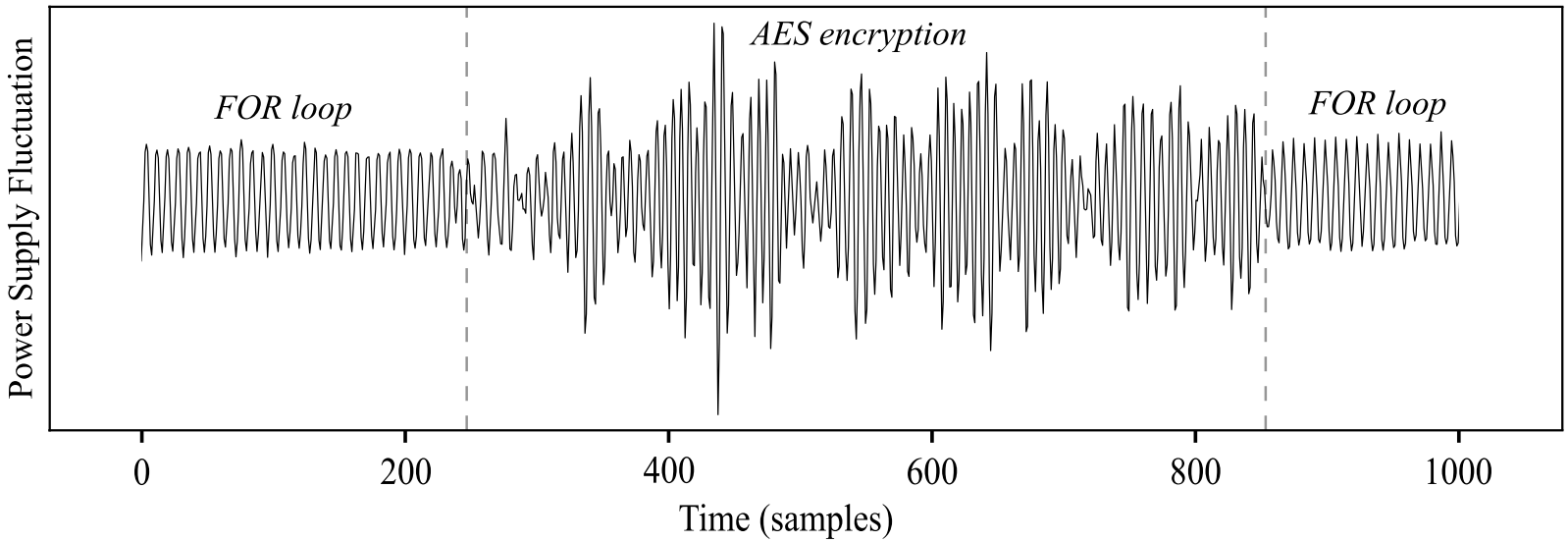
- Crypto library used for secure channels over computer networks.
  - Datapath **32 bit** (T-table).
  - AES encryption time @666MHz $\Rightarrow$ **2.90**μs



- Crypto library used for secure channels over computer networks.
  - Datapath **32 bit** (T-table).
  - AES encryption time @666MHz $\Rightarrow$ **2.90**μs
  - CPA model $\Rightarrow$ AES First round *Sbox*: $HW[Sbox(k \oplus m)]$

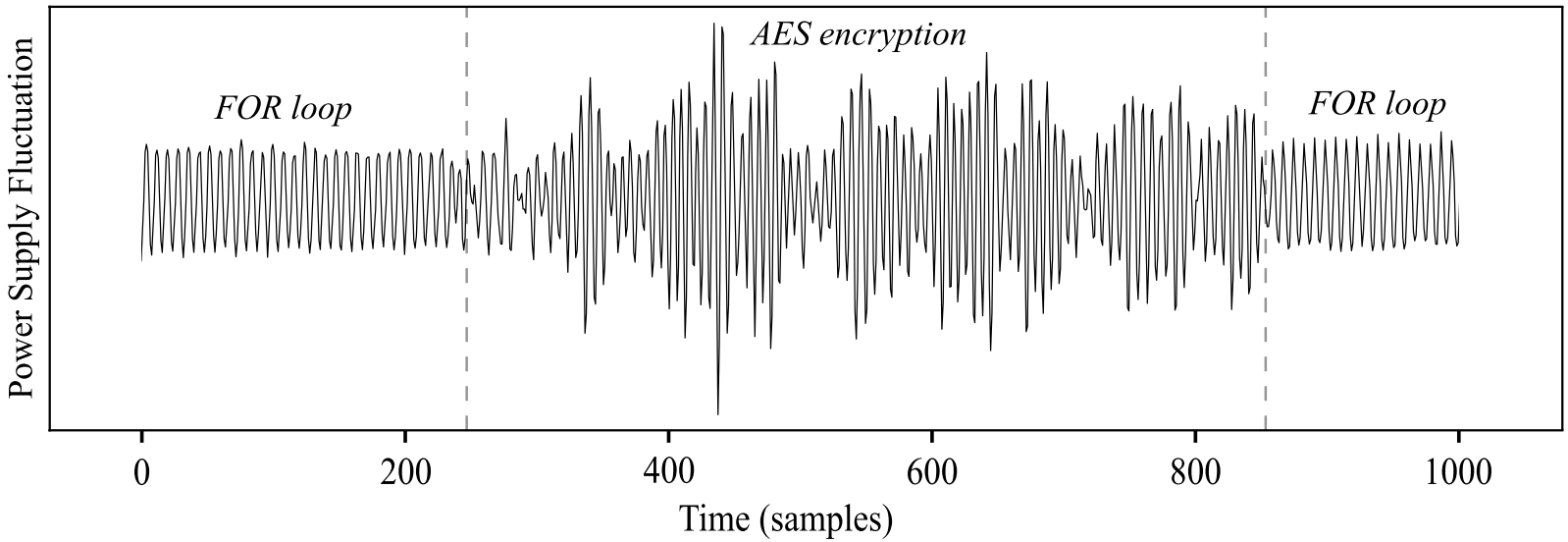


- Crypto library used for secure channels over computer networks.
  - Datapath **32 bit** (T-table).
  - AES encryption time @666MHz $\Rightarrow$ **2.90**μs
  - CPA model $\Rightarrow$ AES First round *Sbox*: $HW[Sbox(k \oplus m)]$



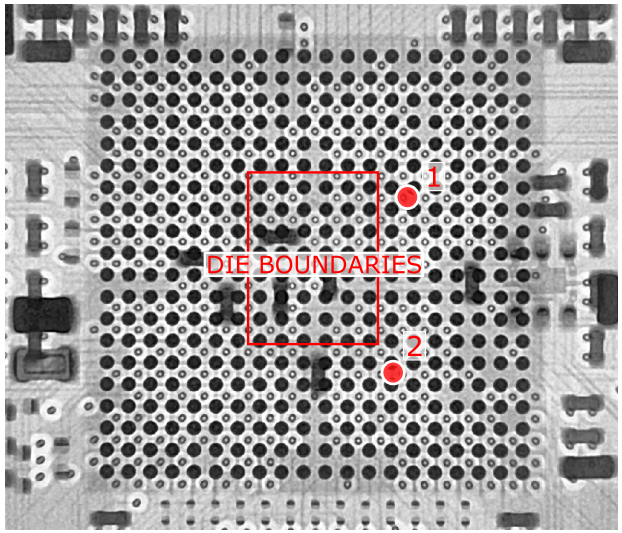
- **Number** of traces required to infer an AES key byte: **127,558**.

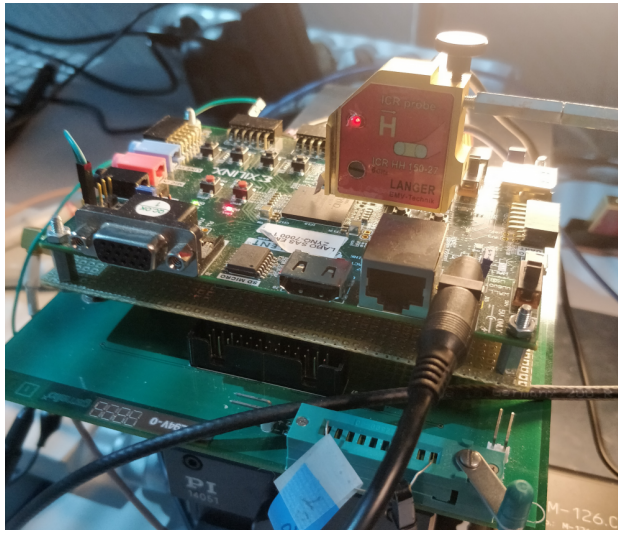
- Crypto library used for secure channels over computer networks.
  - Datapath **32 bit** (T-table).
  - AES encryption time @666MHz $\Rightarrow$ **2.90**μs
  - CPA model $\Rightarrow$ AES First round *Sbox*: $HW[Sbox(k \oplus m)]$



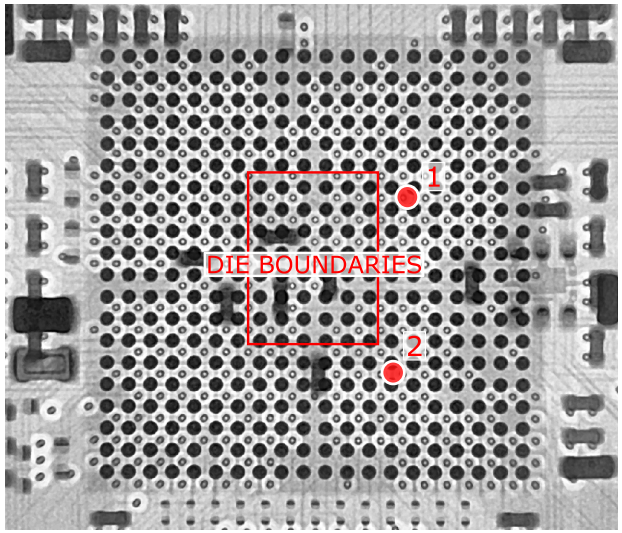
- **Number** of traces required to infer an AES key byte: **127,558**.
- Improved results with T-table model: **87,422** traces.

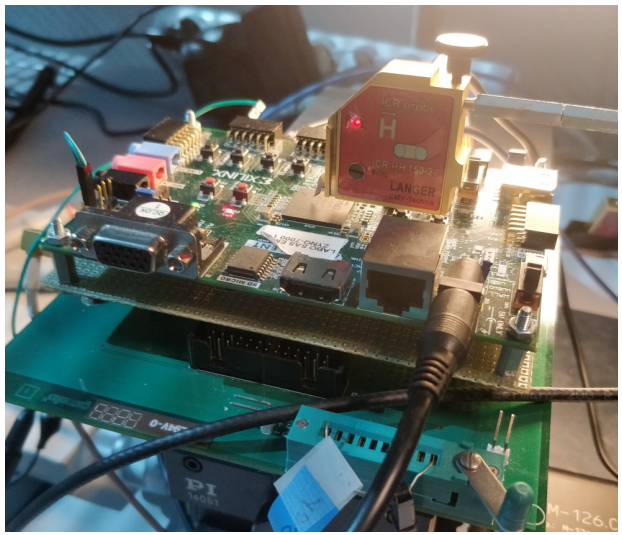
- Goal: **challenge** TDC results regarding **classical SCA**.
- Experimental Setup:
  - Probe: Langer ICR HH 150
  - Oscilloscope Sampling Rate: 5 GS/s





- Goal: **challenge** TDC results regarding **classical SCA**.
- Experimental Setup:
  - Probe: Langer ICR HH 150
  - Oscilloscope Sampling Rate: 5 GS/s
- Two hotspots:
  1. Best results for hardware AES algorithms. (**FPGA**)
  2. Best results for software AES algorithms. (**CPU**)





- CEMA conducted against **each AES studied**.
  - Osc sampling rate (5 GS/s) >> TDC sampling rate (200 MS/s).
  - Osc resolution >> TDC resolution

- CEMA conducted against **each AES studied**.
  - Osc sampling rate (5 GS/s) >> TDC sampling rate (200 MS/s).
  - Osc resolution >> TDC resolution
- Results:

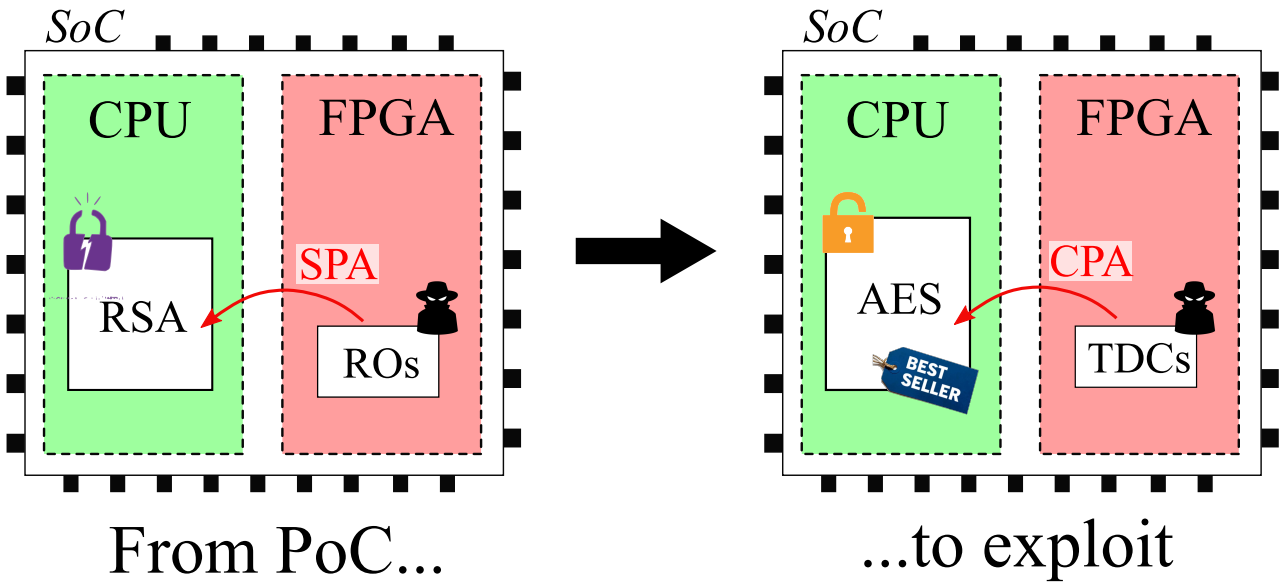
Setup	**HAES**	**Tiny AES**	**OpenSSL 1**	**OpenSSL 2**
EM	**1,021**	**52,438**	**106,225**	88,412
TDC	1,381	111,758	127,558	**87,422**

- CEMA conducted against **each AES studied**.
  - Osc sampling rate (5 GS/s) >> TDC sampling rate (200 MS/s).
  - Osc resolution >> TDC resolution
- Results:

Setup	HAES	Tiny AES	OpenSSL 1	OpenSSL 2
EM	**1,021**	**52,438**	**106,225**	88,412
TDC	1,381	111,758	127,558	**87,422**

- TDCs provide similar results to local side-channel:
  - Side-channel leakage behaviour.
  - TDC calibration (position, delay).

- **FPGA-to-CPU statistical SCA attacks are practicable**.



- To do list:
  - TDC in-depth study (shape, number, chip...)
  - TDC against side-channel countermeasures (shuffling, masking, random delays, jitter, etc).

Thank you! Questions?

joseph.gravellier@emse.fr