SideLine and the advent of software-based hardware attacks

Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia and Philippe Loubet Moundi

SemSecuElec Seminar **DGA INRIA**
19.03.2021

Who am I ?

Past experience

- IUT + Engineering school in Montpellier
- Internship at Gemalto La Ciotat

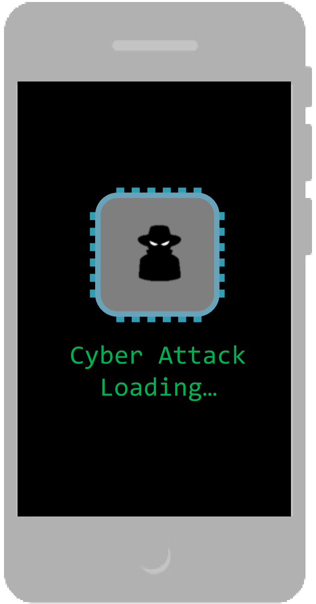
Current Role

- Thales security team member
- 3rd year PhD Student (October 2021)
- Research on new hardware attack vectors

Supervisors:

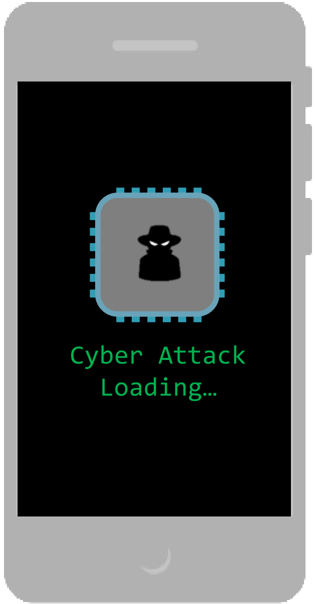
- Thesis Director: Jean-Max Dutertre (EMSE)
- Supervisors: Philippe Loubet Moundi & Yannick Teglia (Thales)

Agenda



- What is a Software-based Hardware Attack ?
- An Overview of the SW-based Power Side-Channel Analysis Works
- Introducing SideLine: SideLine
- Impact for connected devices security

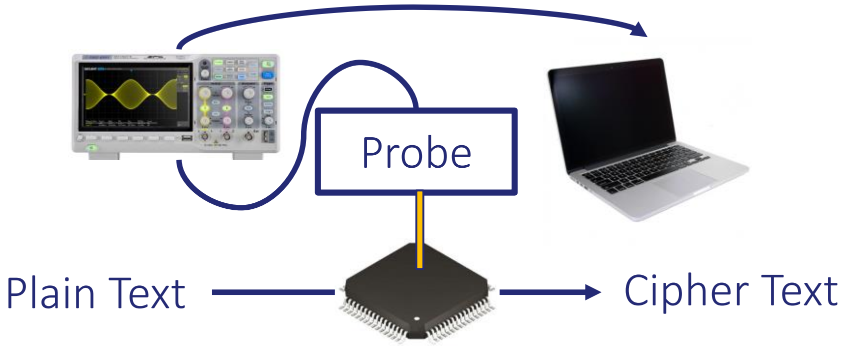
Agenda



- **What is a Software-based Hardware Attack ?**
- An Overview of the SW-based Power Side-Channel Analysis Works
- Introducing SideLine:
- Impact for connected devices security

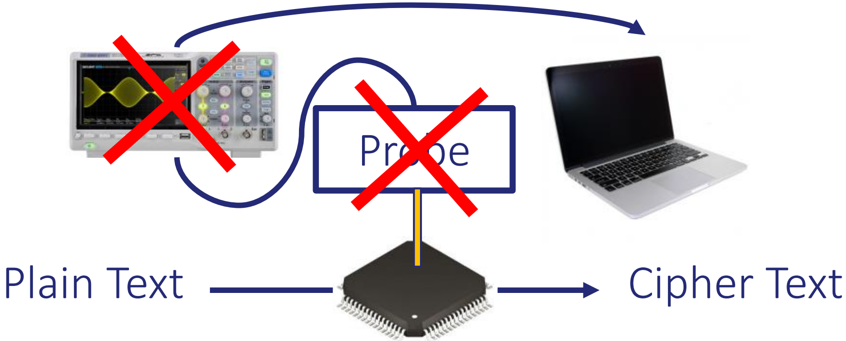
Traditional Hardware attacks

- Type: fault Injection & side-channel analysis
- Means: oscilloscope, laser, EM probe…
- Range: local, **direct physical access required**



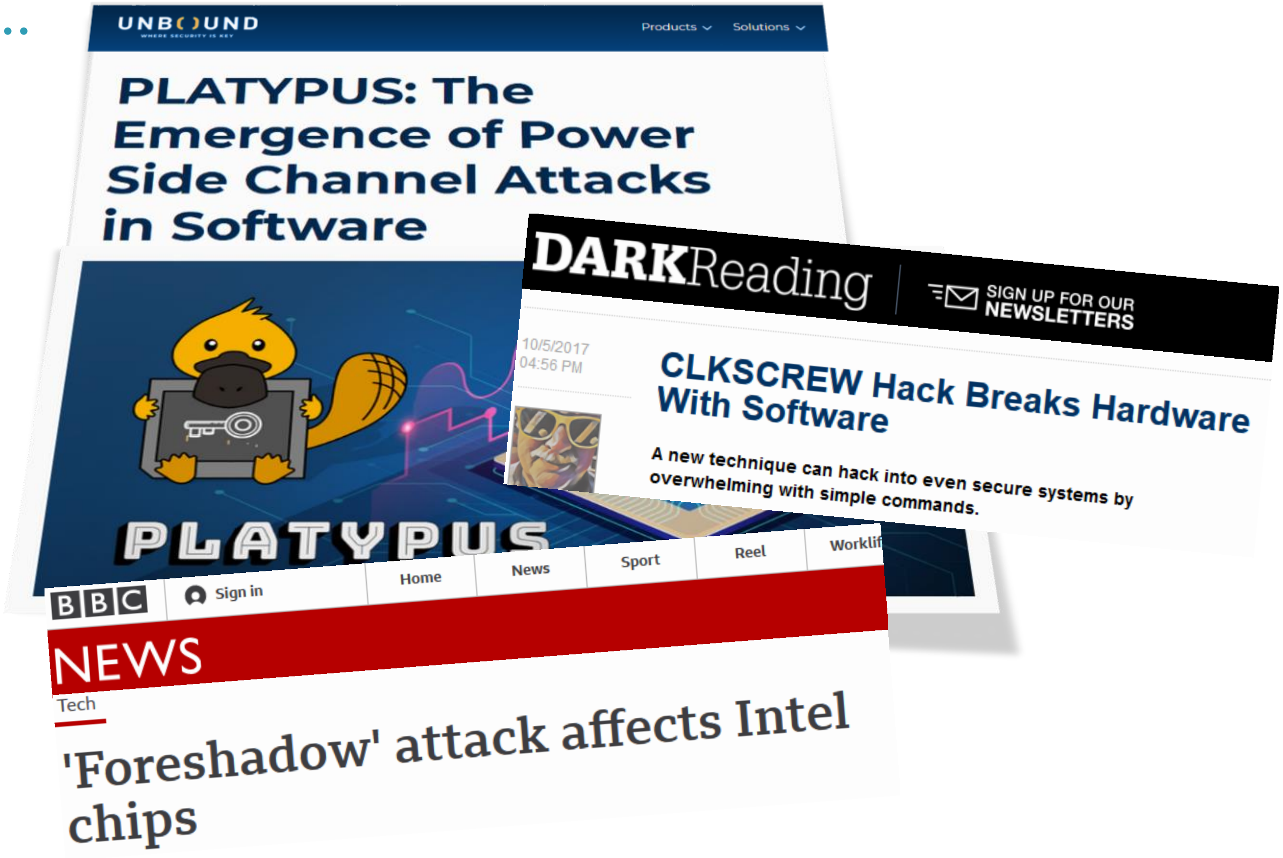
Software-based Hardware attacks

- Type: fault Injection (FIA) & side-channel analysis (SCA)
- Range: remote, **no direct physical access required !**
- Means: **resources available within the target**



Software-based Hardware Attacks

A growing threat…



PLATYPUS: The Emergence of Power Side Channel Attacks in Software

10/5/2017 04:56 PM

CLKSCREW Hack Breaks Hardware With Software

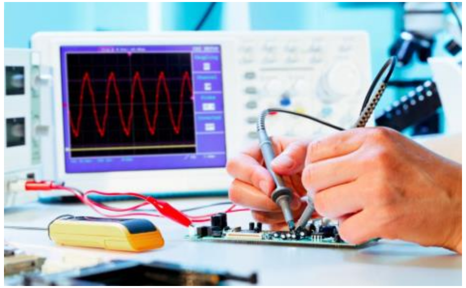
A new technique can hack into even secure systems by overwhelming with simple commands.

'Foreshadow' attack affects Intel chips

Merging two attack families

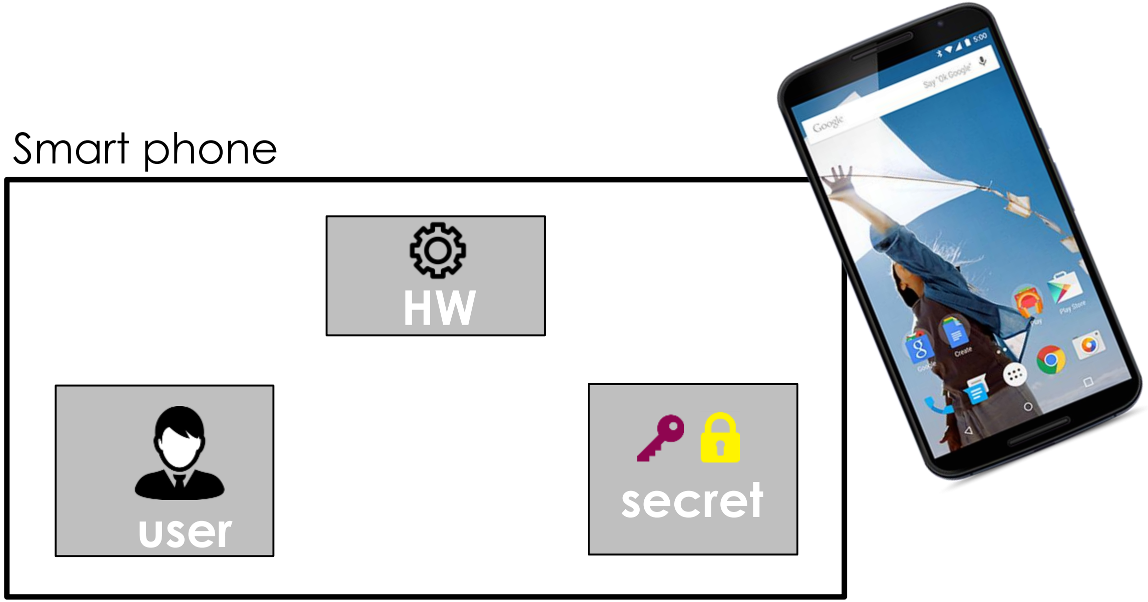
Software Attack

Hardware Attack



Software-Based Hardware Attack

How does it work ?



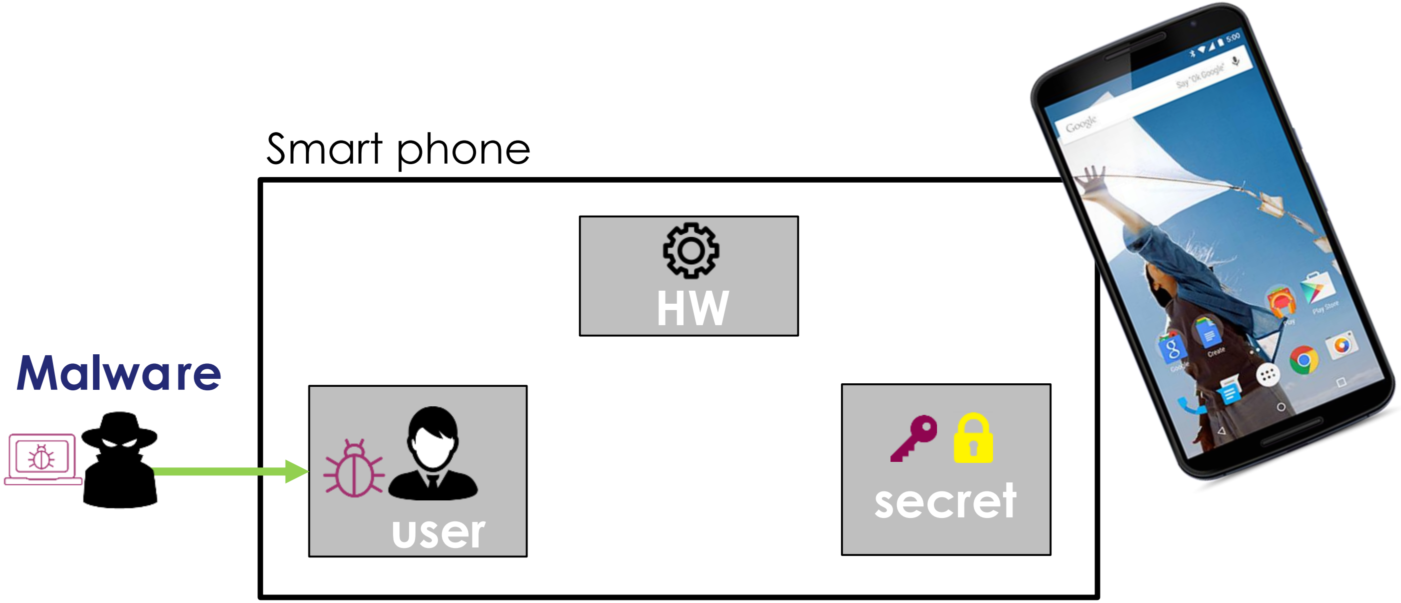
Smart phone

HW

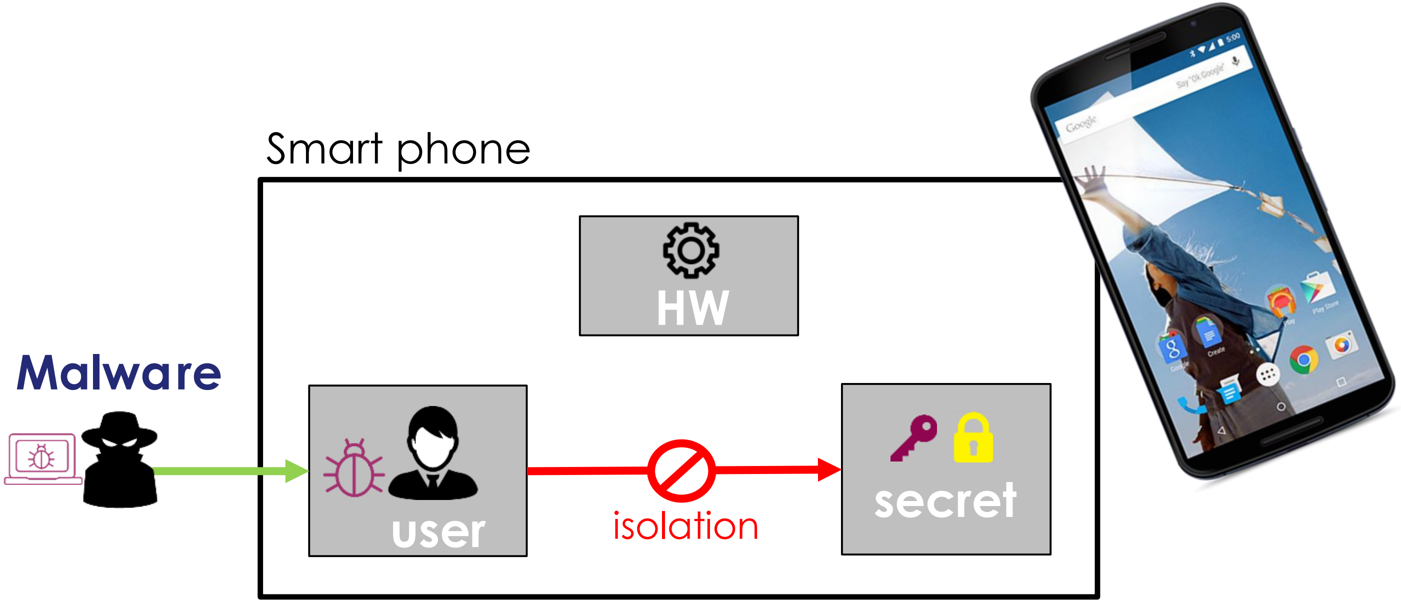
user

secret

How does it work ?



How does it work ?



Smart phone

HW

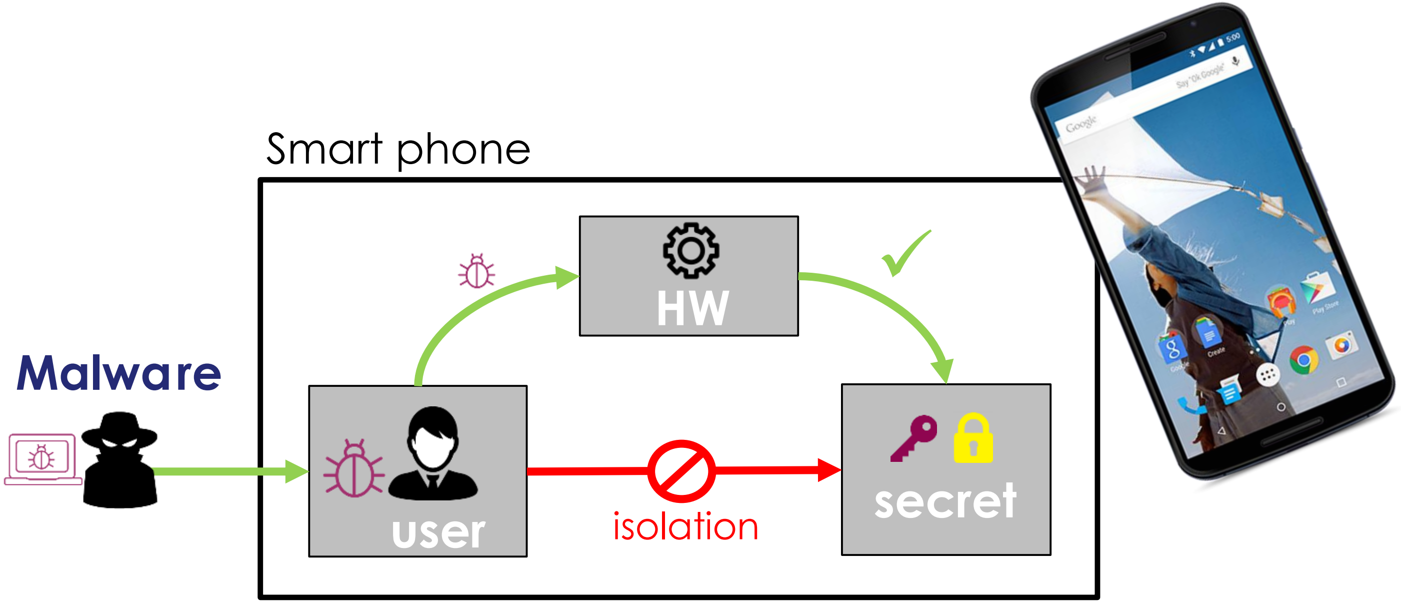
Malware

user

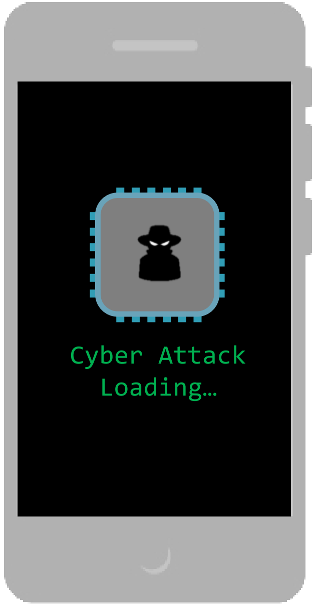
isolation

secret

How does it work ?



Agenda



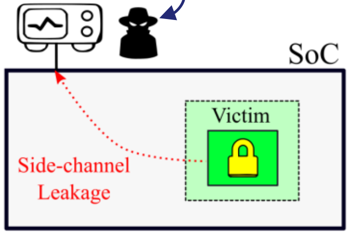
- What is a SW-based Hardware Attack ?
- **An Overview of the SW-based Power Side-Channel Analysis Works**
- Introducing SideLine:
- Impact for connected devices security

Local vs Remote Side-Channel

Local Power Side-channel

- Use an external voltage probe

My team in the lab

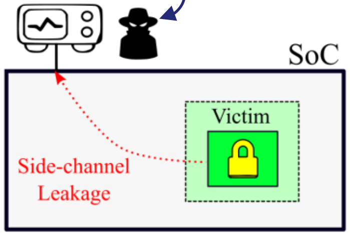


Local vs Remote Side-Channel

Local Power Side-channel

- Use an external voltage probe

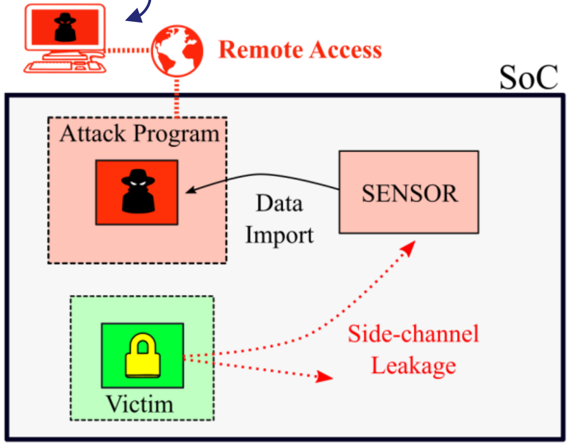
My team in the lab



Remote Power Side-channel

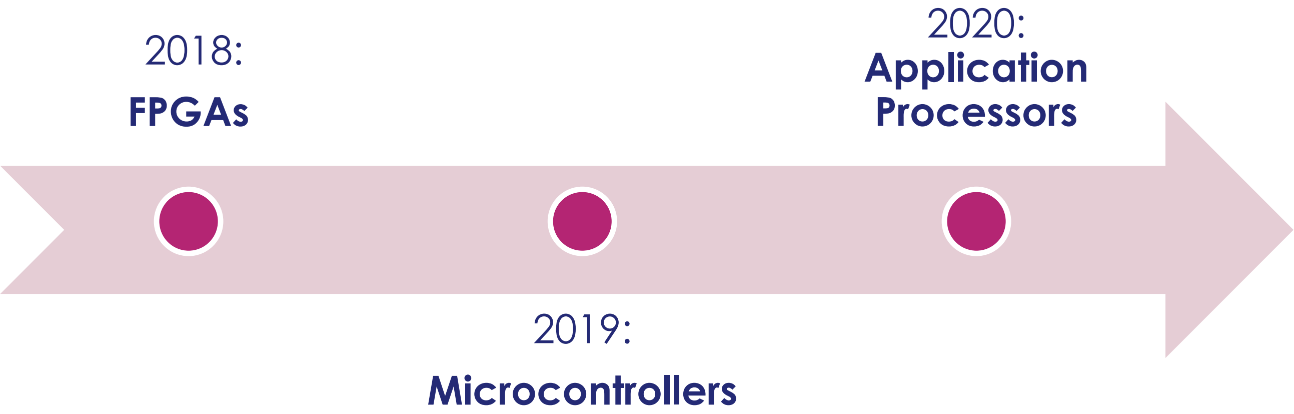
- Requires a **sensor (Hardware)**
- Requires a **malware (Software)**

Me, from my computer!



Software-based Power SCA **Timeline**

From FPGAs to high-end processors



2018:

FPGAs

2019:

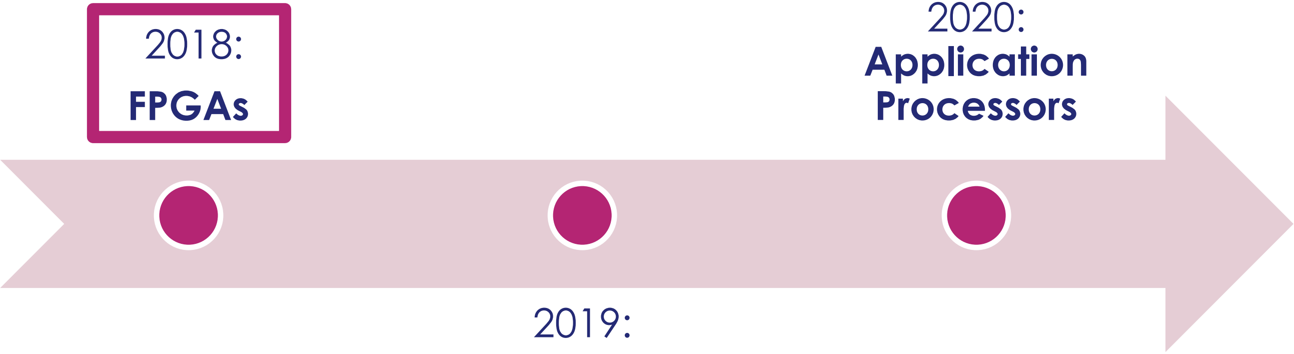
Microcontrollers

2020:

Application Processors

Software-based Power SCA **Timeline**

From FPGAs to high-end processors



2018:
FPGAs

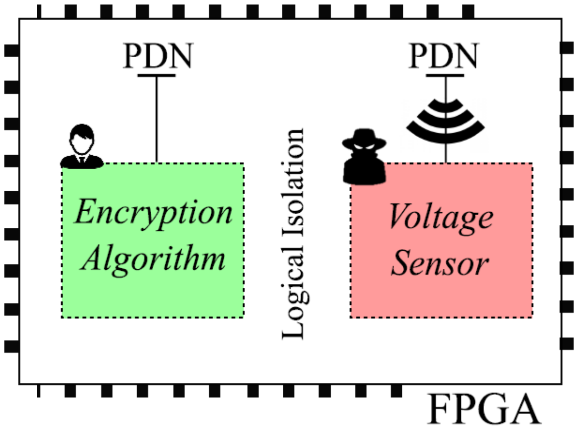
2019:
Microcontrollers

2020:
Application Processors

It all started from FPGAs…

Hardware attacks can be reproduced using FPGA logic

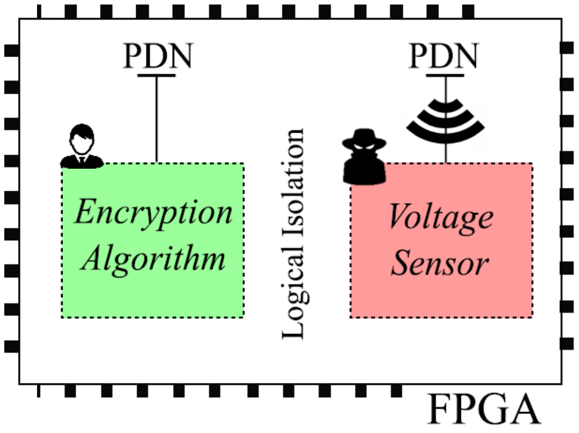
- **Encryption algorithm** implementation.
- **Voltage sensor** implementation.



It all started from FPGAs…

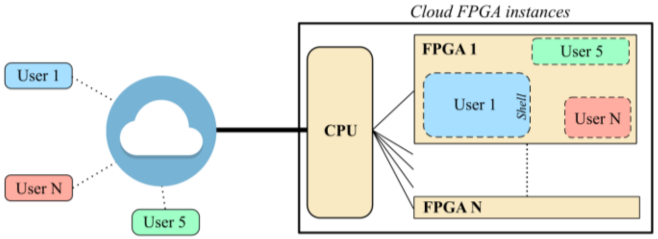
Hardware attacks can be reproduced using FPGA logic

- **Encryption algorithm** implementation.
- **Voltage sensor** implementation.



Recent FPGA adoption in the cloud

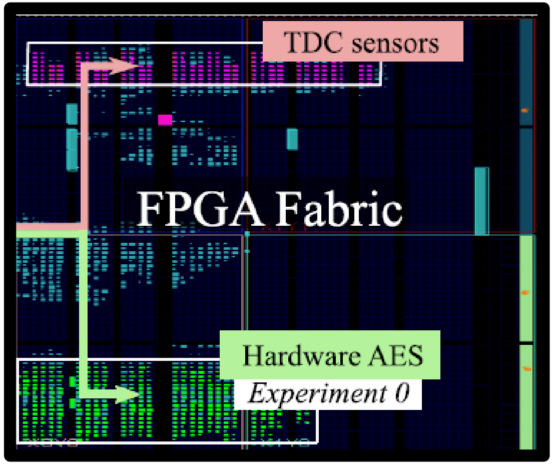
- Amazon, Alibaba + Multi-user FPGAs
- **Security** ?



FPGA-to-FPGA Power SCA

Target: Xilinx Zynq

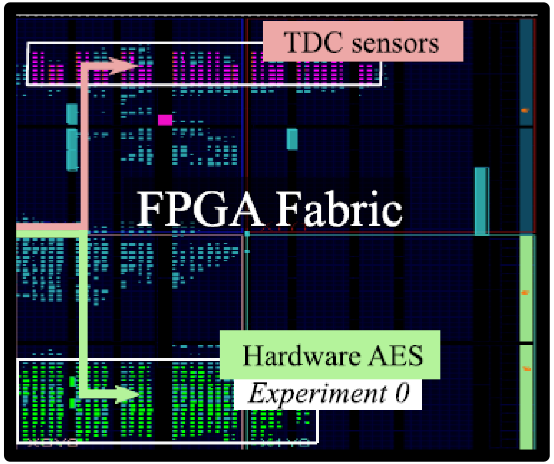
- Adversary: **voltage sensors**
  - *Freq*: **200MHz**
- Victim: **AES algorithm**
  - *Freq*: **10MHz**

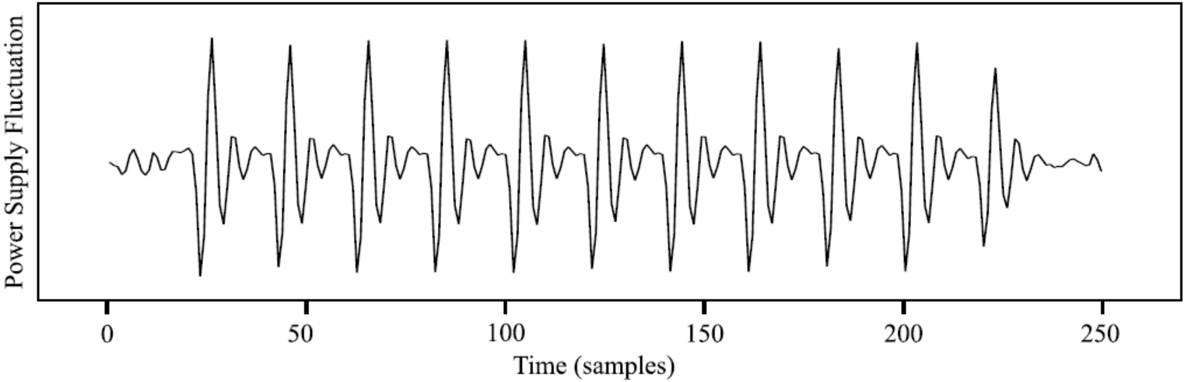


FPGA-to-FPGA Power SCA

Target: Xilinx Zynq

- Adversary: **voltage sensors**
  - *Freq*: **200MHz**
- Victim: **AES algorithm**
  - *Freq*: **10MHz**
- Traces to infer the AES key: **~1,000**

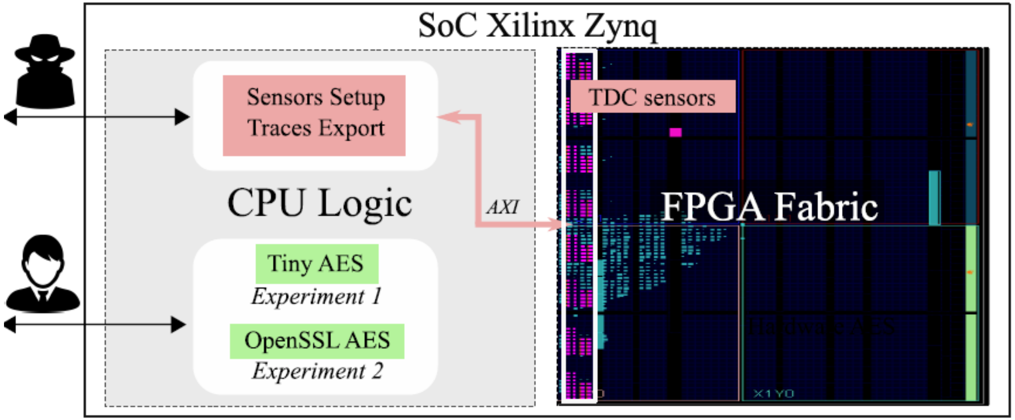




FPGA-to-CPU Power SCA

Target: Xilinx Zynq

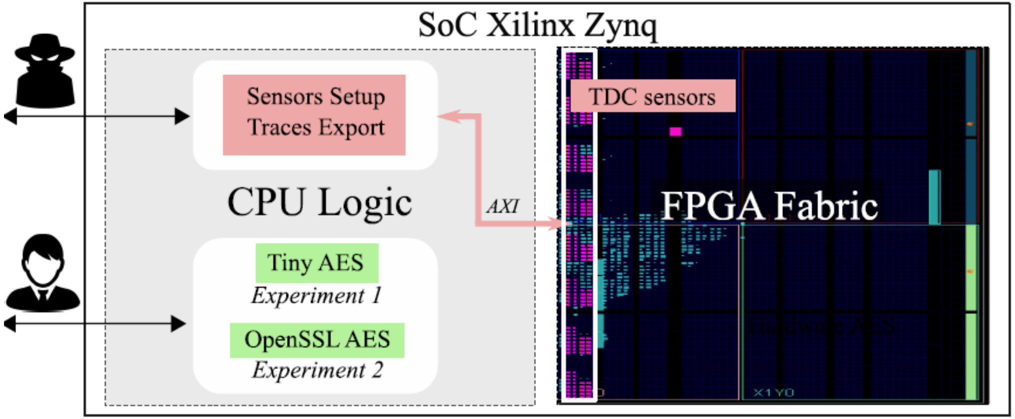
- Adversary: **voltage sensors (200MHz)**
- Victim: **SW AES algorithm (666MHz)**

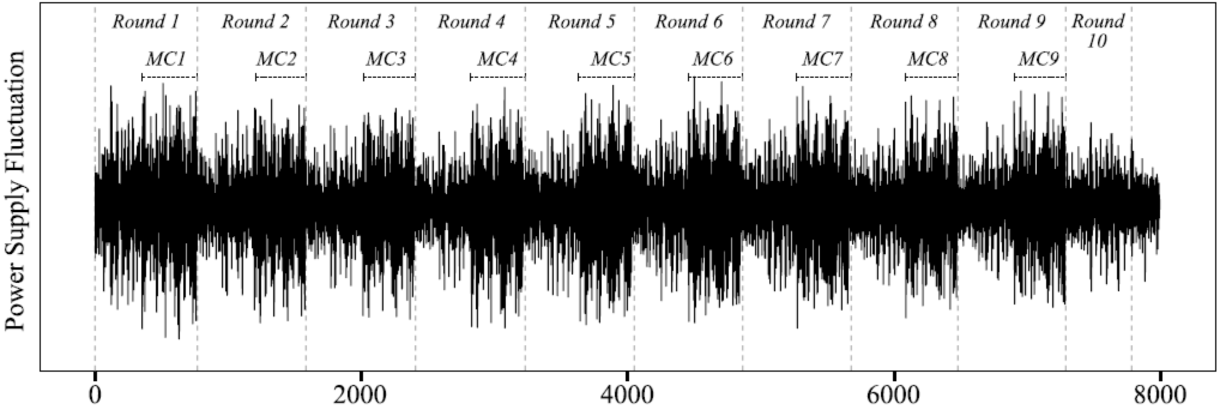


FPGA-to-CPU Power SCA

Target: Xilinx Zynq

- Adversary: **voltage sensors (200MHz)**
- Victim: **SW AES algorithm (666MHz)**
- Traces to infer the AES key: **~90,000**





Results & conclusions on FPGAs

The power consumption leaks through the entire SoC

- SoC architecture is **leaky**
- SCA attacks are feasible **with limited resources**

Results & conclusions on FPGAs

The power consumption leaks through the entire SoC

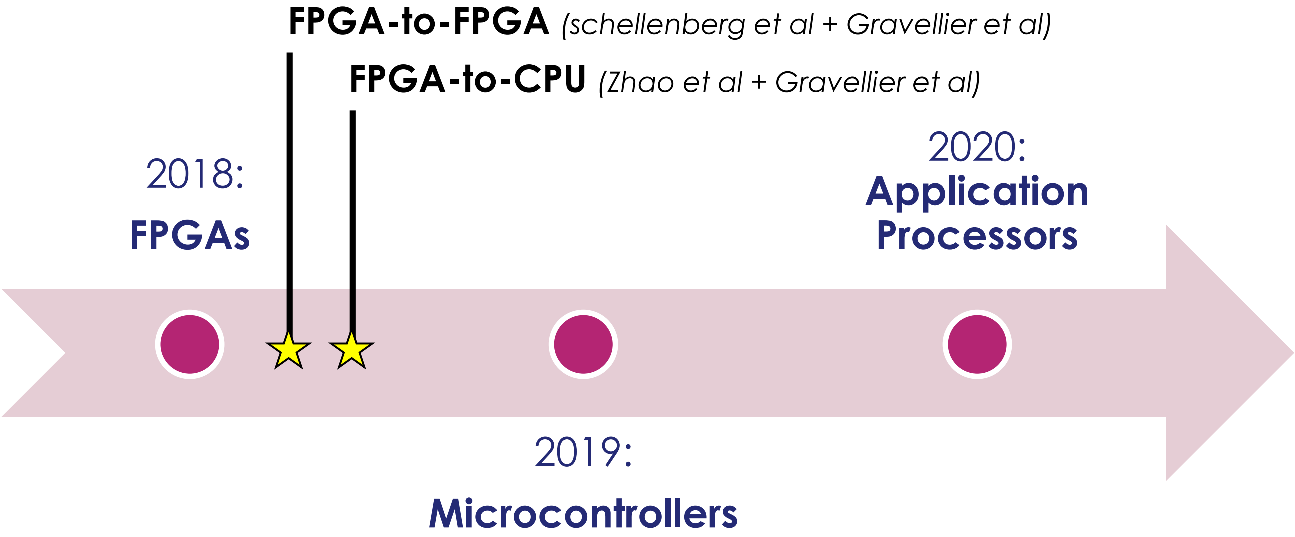
- SoC architecture is **leaky**
- SCA attacks are feasible **with limited resources**

Software security is not enough

- Hardware attacks **bypass software isolation**
- Even if the attacker has **no physical access to the target**

Software-based Power SCA **Timeline**

From FPGAs to high-end processors



FPGA-to-FPGA *(schellenberg et al + Gravellier et al)*

FPGA-to-CPU *(Zhao et al + Gravellier et al)*

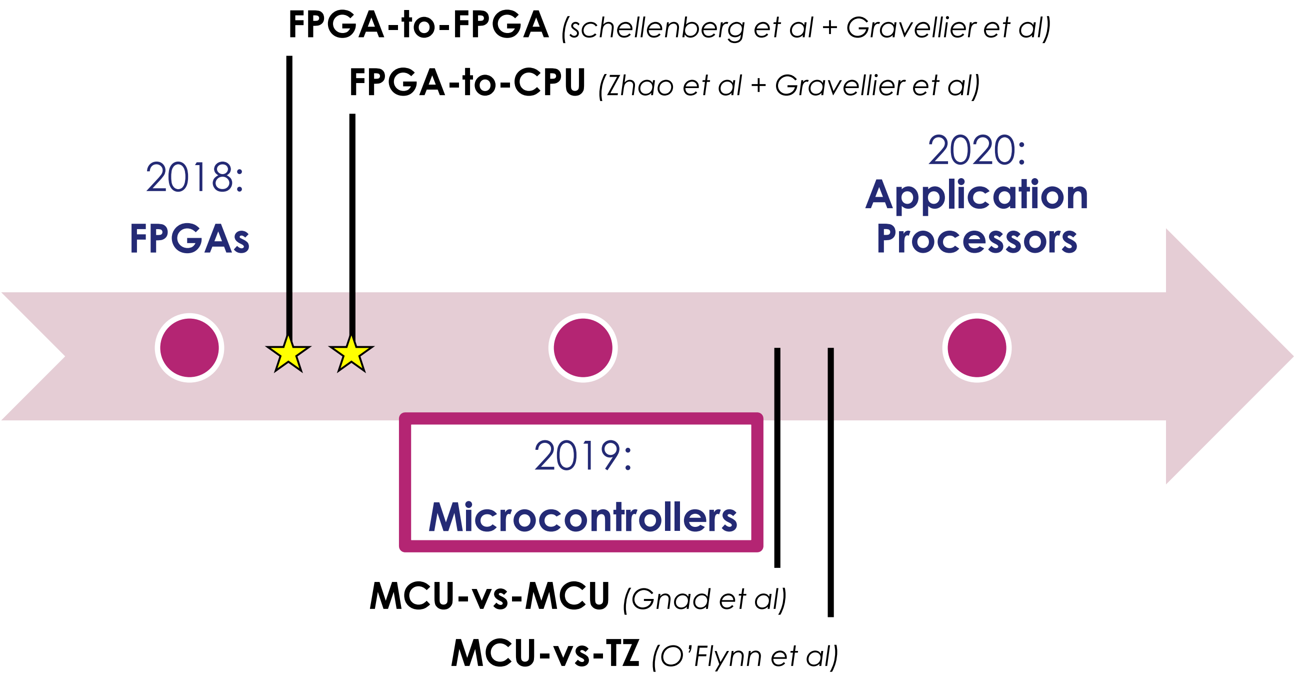
2018:
FPGAs

2019:
Microcontrollers

2020:
Application Processors

Software-based Power SCA **Timeline**

From FPGAs to high-end processors



FPGA-to-FPGA *(schellenberg et al + Gravellier et al)*

FPGA-to-CPU *(Zhao et al + Gravellier et al)*

2018:
FPGAs

2020:
Application Processors

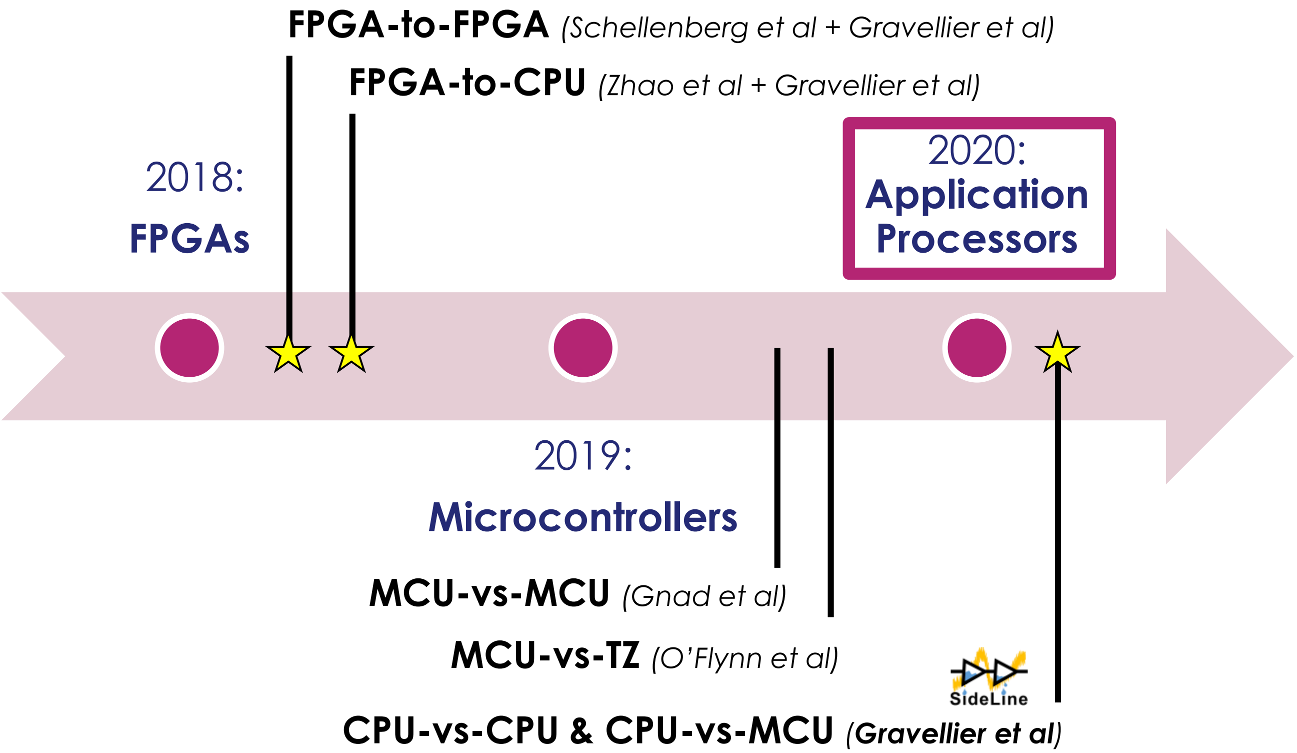
2019:
Microcontrollers

MCU-vs-MCU *(Gnad et al)*

MCU-vs-TZ *(O'Flynn et al)*

Software-based Power SCA **Timeline**

From FPGAs to high-end processors



FPGA-to-FPGA *(Schellenberg et al + Gravellier et al)*

FPGA-to-CPU *(Zhao et al + Gravellier et al)*

2018:
FPGAs

2019:
Microcontrollers

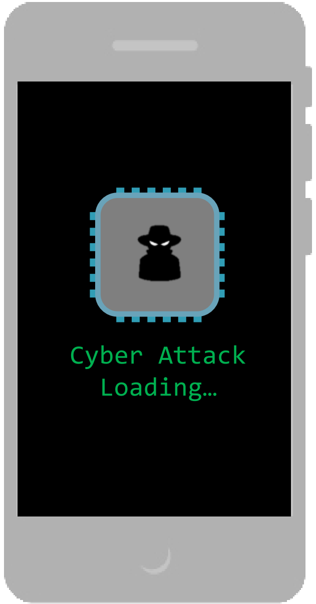
2020:
Application Processors

MCU-vs-MCU *(Gnad et al)*

MCU-vs-TZ *(O'Flynn et al)*

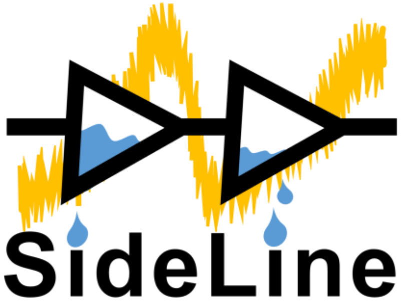
CPU-vs-CPU & CPU-vs-MCU *(**Gravellier et al**)*

Agenda



- What is a SW-based Hardware Attack ?
- An Overview of the SW-based Power Side-Channel Analysis Works
- **Introducing SideLine:** SideLine
- Impact for connected devices security

Introducing SideLine



SideLine enables software-based SCA attack on application processors

It uses **delay-lines** as power meters.

On **Cortex-A ARM** processors

With **Linux OS** implemented

https://josephgravellier.github.io/sideline/

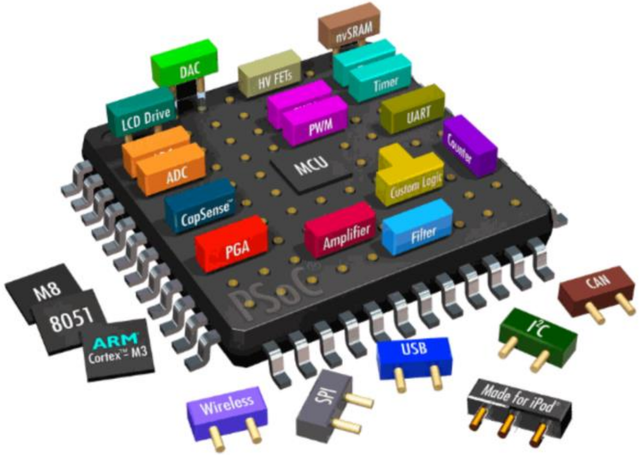
Looking for a new side-channel vector

Among all the HW resources available in SoCs:

- Is there a reliable way to measure power consumption?

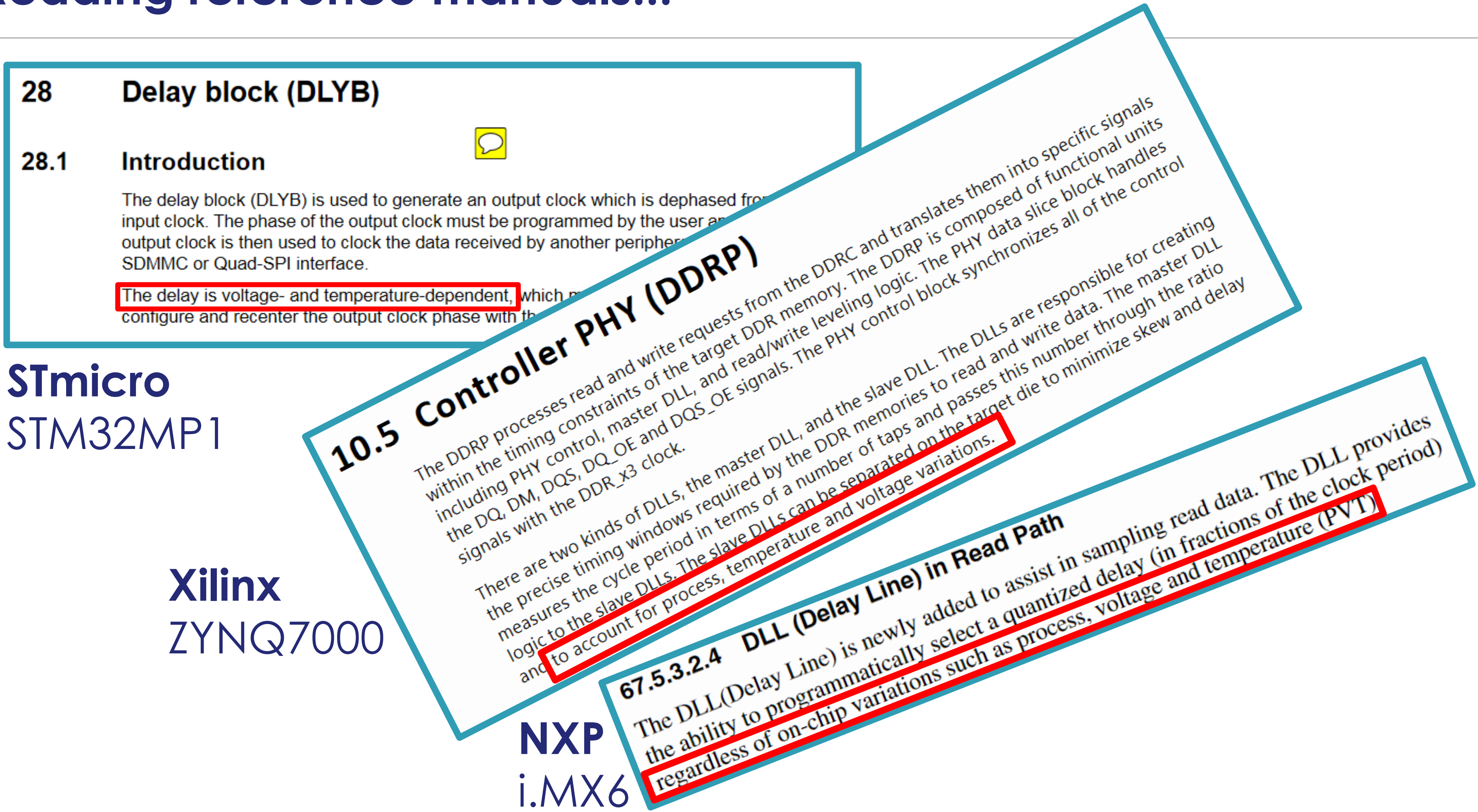
Lets do a benchmark on common SoC devices:

- Results: ADCs, temperature & voltage sensors, **delay lines**



Source: https://www.cypress.com/file/121931/download

Reading reference manuals...



28 Delay block (DLYB)

28.1 Introduction

The delay block (DLYB) is used to generate an output clock which is dephased fro
input clock. The phase of the output clock must be programmed by the user a
output clock is then used to clock the data received by another peripher
SDMMC or Quad-SPI interface.

The delay is voltage- and temperature-dependent, which m
configure and recenter the output clock phase with th

STmicro
STM32MP1

10.5 Controller PHY (DDRP)

The DDRP processes read and write requests from the DDRC and translates them into specific signals within the timing constraints of the target DDR memory. The DDRP is composed of functional units including PHY control, master DLL, and read/write leveling logic. The PHY data slice block handles the DQ, DM, DQS, DQ_OE and DQS_OE signals. The PHY control block synchronizes all of the control signals with the DDR_x3 clock.

There are two kinds of DLLs, the master DLL, and the slave DLL. The DLLs are responsible for creating the precise timing windows required by the DDR memories to read and write data. The master DLL measures the cycle period in terms of a number of taps and passes this number through the ratio logic to the slave DLLs. The slave DLLs can be separated on the target die to minimize skew and delay and to account for process, temperature and voltage variations.

Xilinx
ZYNQ7000

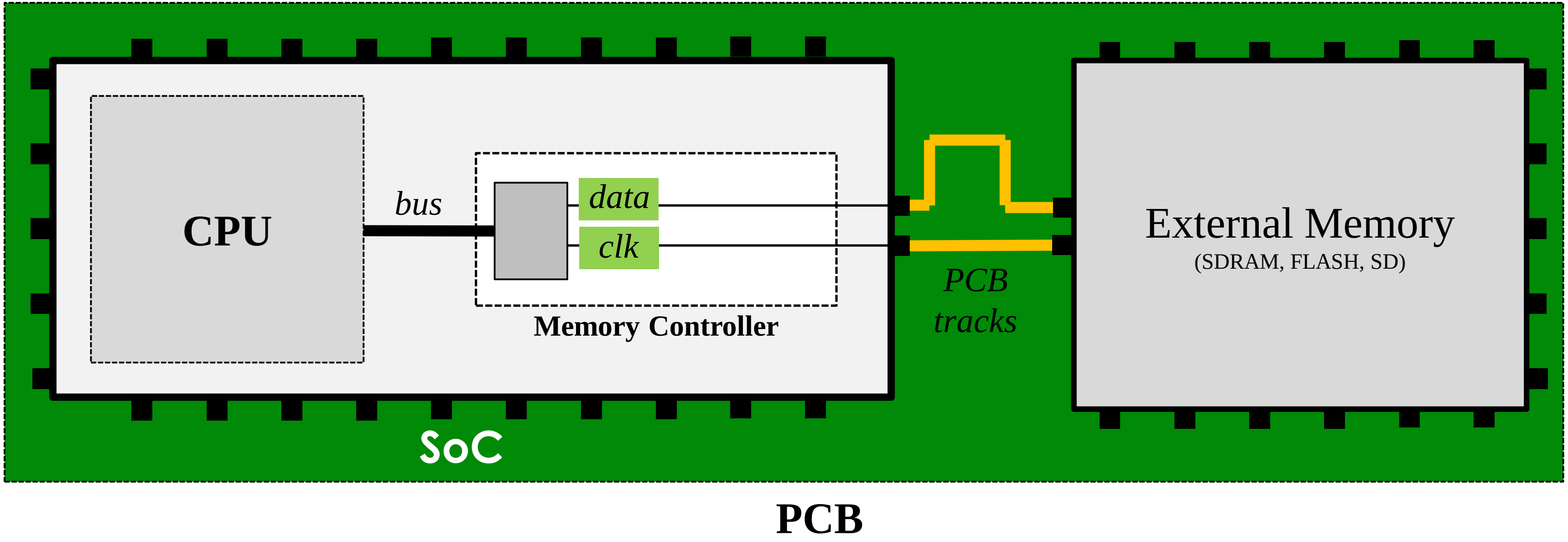
67.5.3.2.4 DLL (Delay Line) in Read Path

The DLL(Delay Line) is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage and temperature (PVT).

NXP
i.MX6

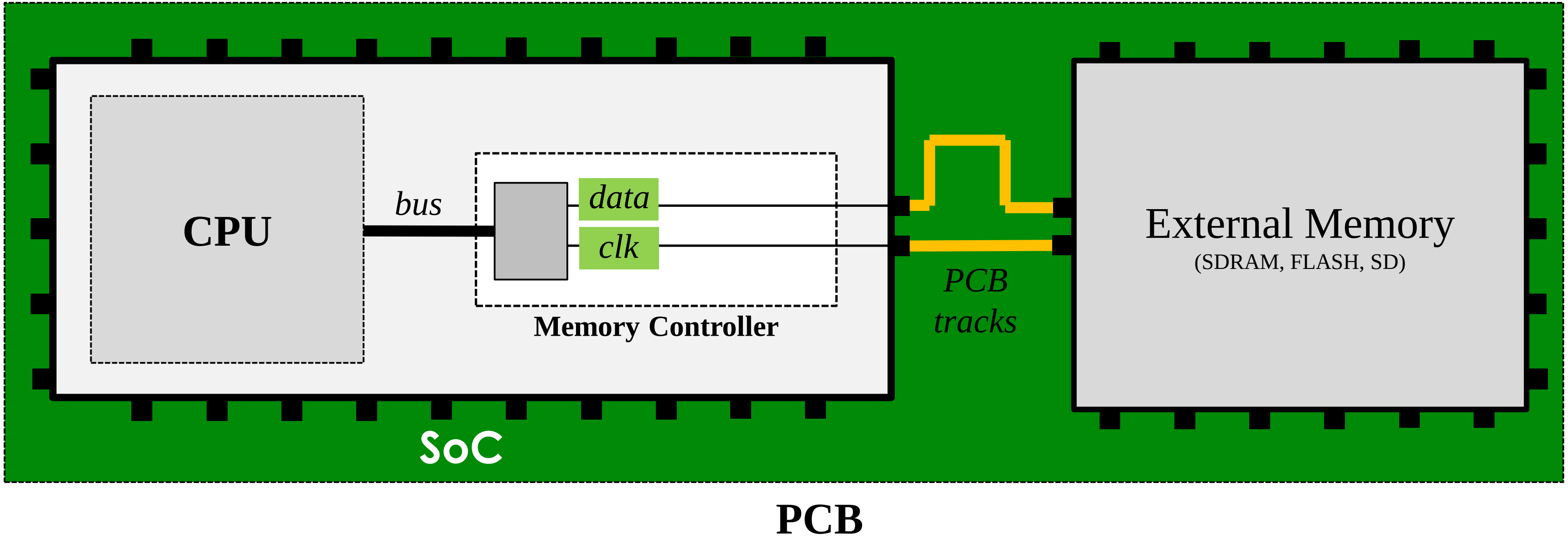
Delay-lines: what for?

- Memory controllers **interface** the SoC with its external memories



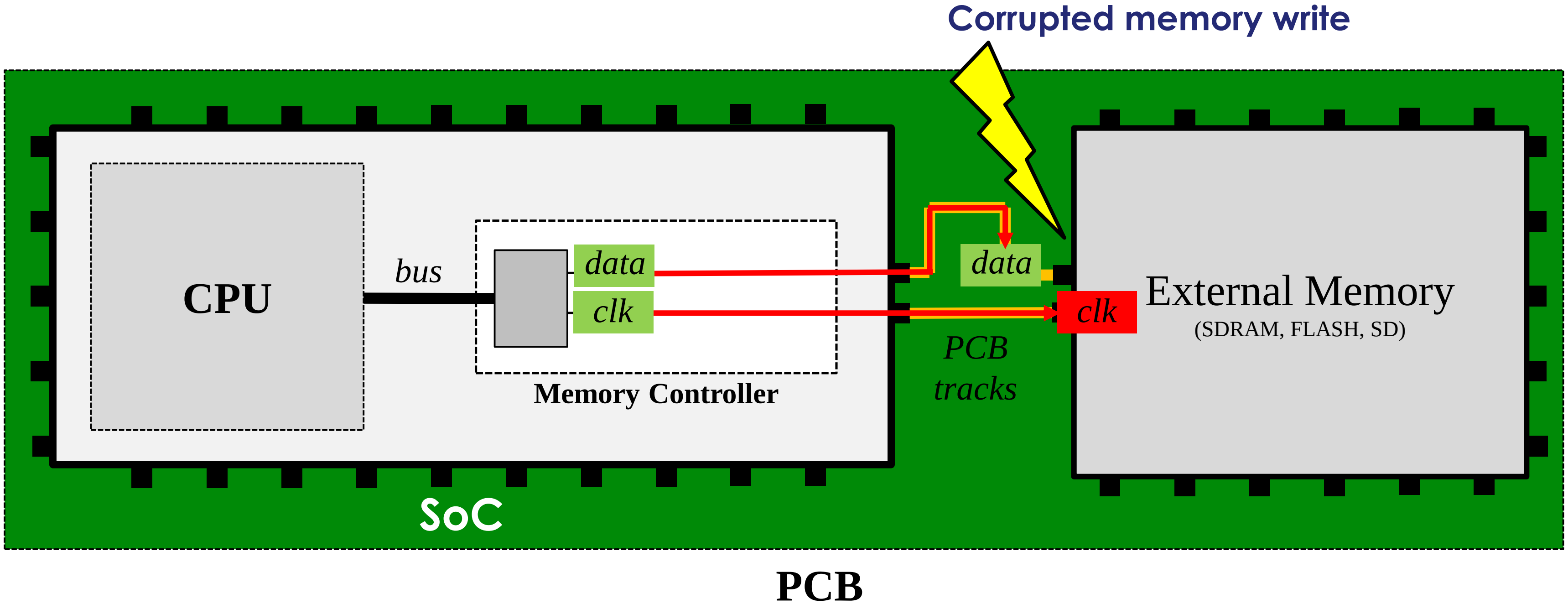
Delay-lines: what for?

- Memory controllers **interface** the SoC with its external memories
- **Potential glitches** with Process, Voltage, Temperature (PVT) **variations**



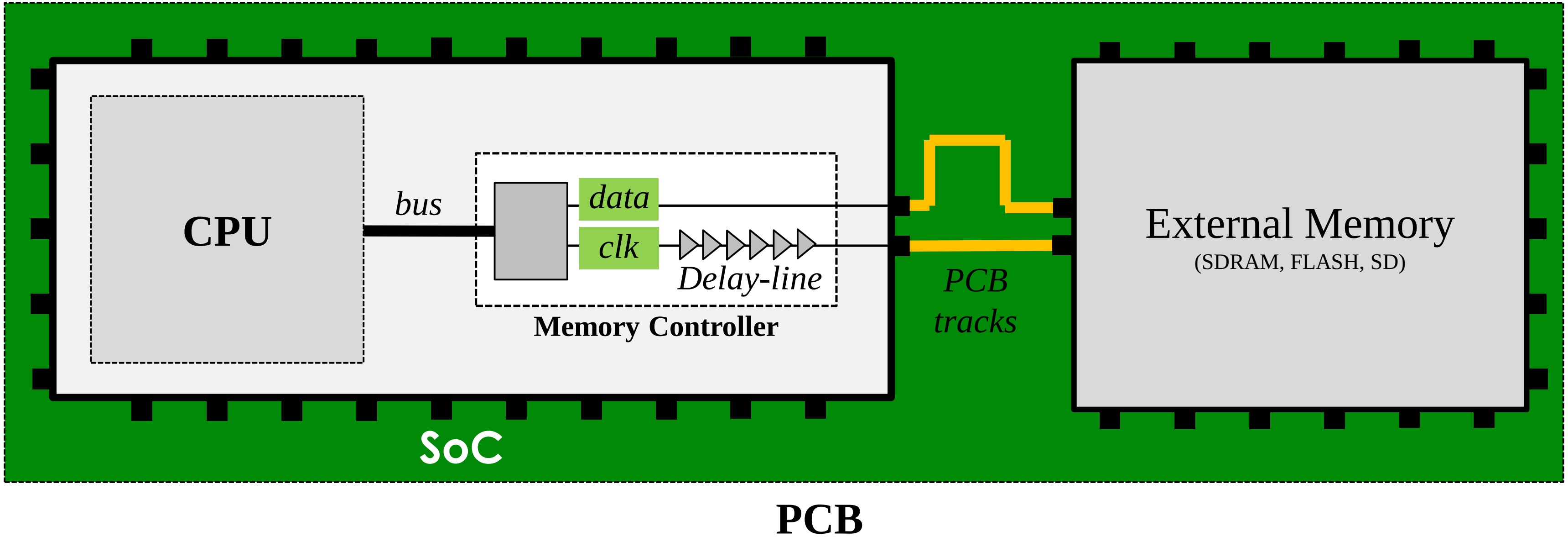
Delay-lines: what for?

- Memory controllers **interface** the SoC with its external memories
- **Potential glitches** with Process, Voltage, Temperature (PVT) **variations**



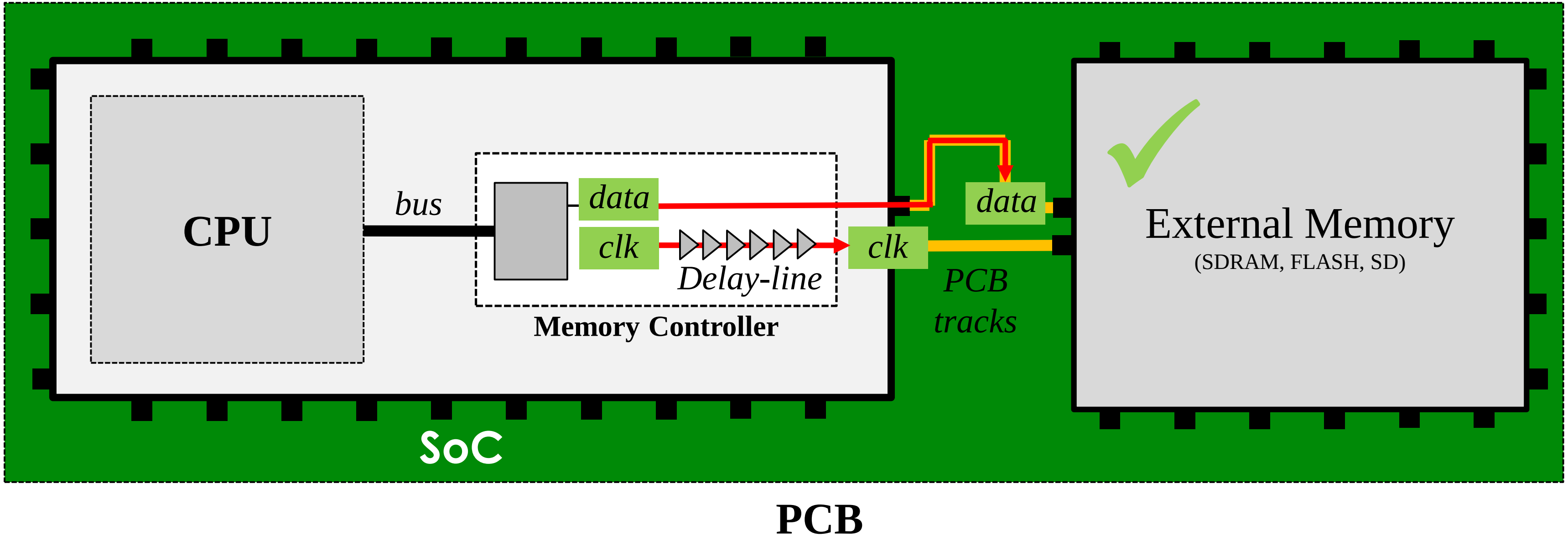
Delay-lines: what for?

- Memory controllers **interface** the SoC with its external memories
- **Potential glitches** with Process, Voltage, Temperature (PVT) **variations**
- Recent SoC use **delay lines** to **counteract** PVT variation.



Delay-lines: what for?

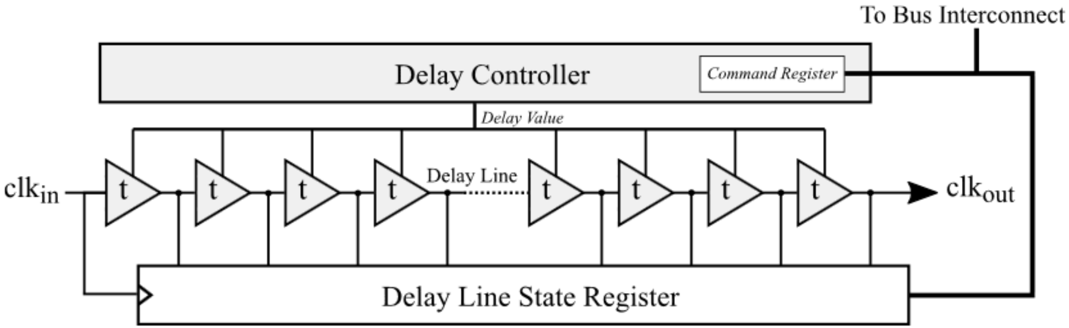
- Memory controllers **interface** the SoC with its external memories
- **Potential glitches** with Process, Voltage, Temperature (PVT) **variations**
- Recent SoC use **delay lines** to **counteract** PVT variation.



Programmable Delay-line Types

The delay-line block

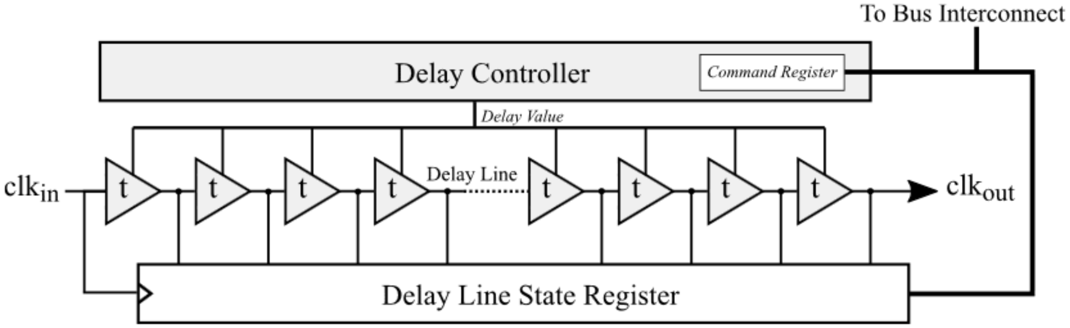
- Delay-line calibration can be programmed **manually**



Programmable Delay-line Types

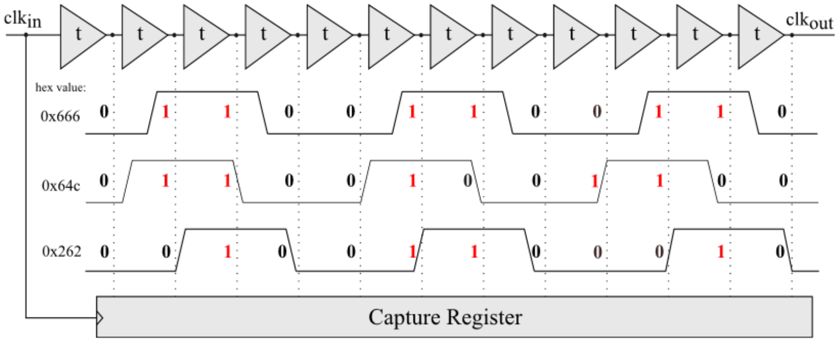
The delay-line block

- Delay-line calibration can be programmed **manually**



Using it as a sensor

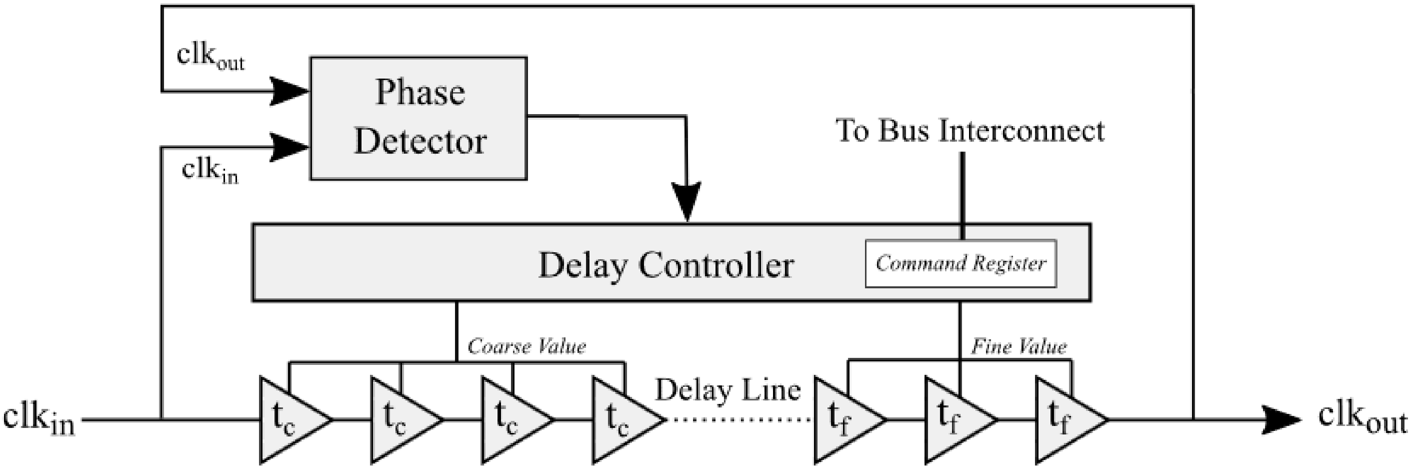
- **The delay-line** stores an image of the clk signal
- This image changes with PVT variations


clk_{in}
t t t t t t t t t t t t
clk_{out}
hex value:
0x666 0 1 1 0 0 1 1 0 0 1 1 0
0x64c 0 1 1 0 0 1 0 0 1 1 0 0
0x262 0 0 1 0 0 1 1 0 0 0 1 0
Capture Register

Programmable Delay-line Types

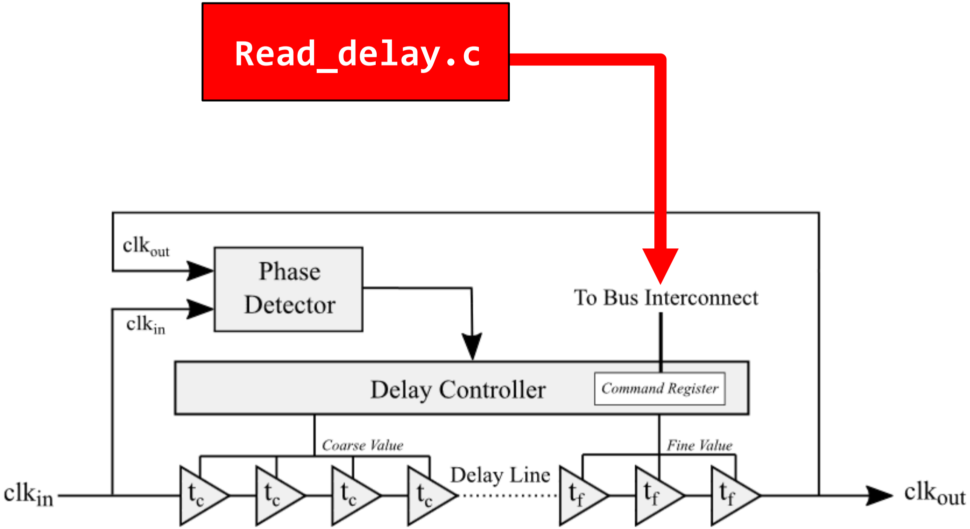
The delay-locked-loop

- Delay-line calibration is updated **dynamically** with PVT variations



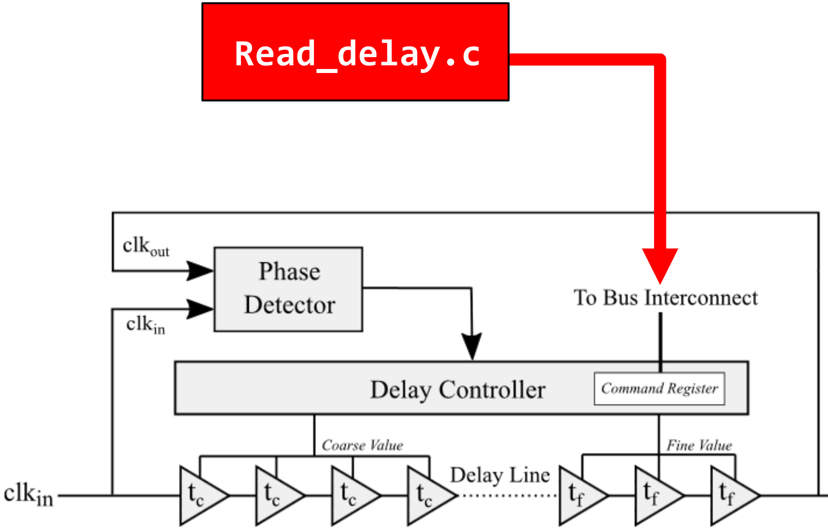
Toward Delay-line-based SCA ?

- Ok so what if we access the delay line state from software
- Does it **change** with **PVT variations**?



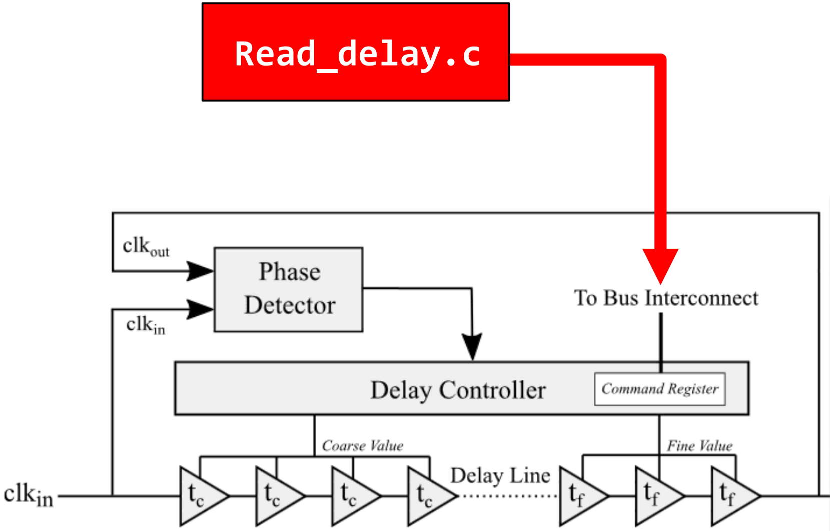
Toward Delay-line-based SCA ?

- Ok so what if we access the delay line state from software
- Does it **change** with **PVT variations**?

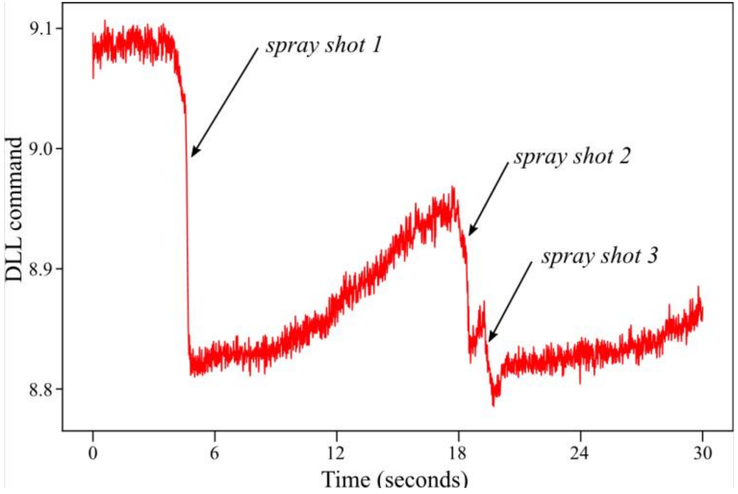

Read_delay.c

Toward Delay-line-based SCA ?

- Ok so what if we access the delay line state from software
- Does it **change** with **PVT variations**?



YES!



SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz

STM32MP1

Zybo-Z7-10

SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz
- **MCU:** Cortex-M4 (STM32MP1)
  - freq: 200 MHz

STM32MP1

Zybo-Z7-10

SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz
- **MCU:** Cortex-M4 (STM32MP1)
  - freq: 200 MHz
- **Algorithms:** OpenSSL AES and custom RSA implems

STM32MP1

Zybo-Z7-10

SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz
- **MCU:** Cortex-M4 (STM32MP1)
  - freq: 200 MHz
- **Algorithms:** OpenSSL AES and custom RSA implems

SideLine App

- C program

STM32MP1

Zybo-Z7-10

SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz
- **MCU:** Cortex-M4 (STM32MP1)
  - freq: 200 MHz
- **Algorithms:** OpenSSL AES and custom RSA implems

SideLine App

- C program
- DMA-based delay line sampling (freq: 16MHz)

STM32MP1

Zybo-Z7-10

SideLine Targets

Victim platform:

- **CPU**: Cortex-A9 (Zybo) and Cortex-A7 (STM32MP1)
  - freq: 600-700 MHz
- **MCU:** Cortex-M4 (STM32MP1)
  - freq: 200 MHz
- **Algorithms:** OpenSSL AES and custom RSA implems

SideLine App

- C program
- DMA-based delay line sampling (freq: 16MHz)
- SideLine launches the encryptions

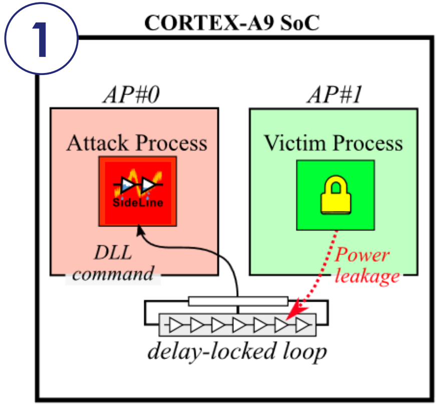
STM32MP1

Zybo-Z7-10

SideLine Attacks Scenarios

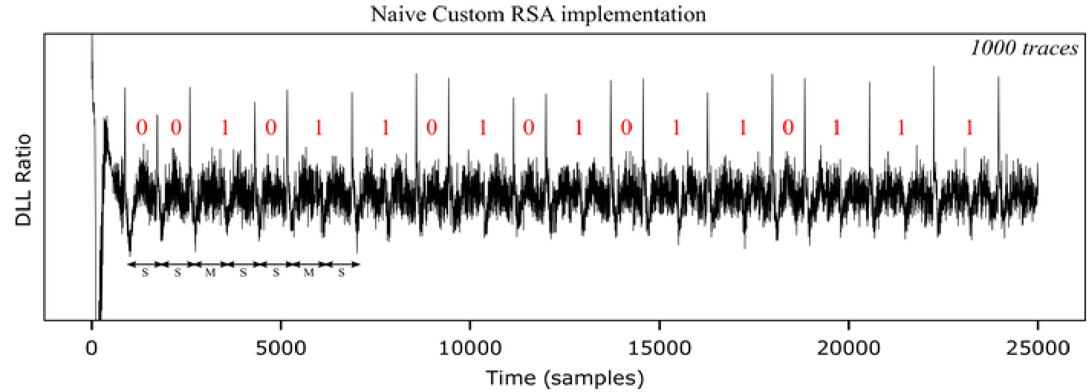
3 core-vs-core attack scenarios

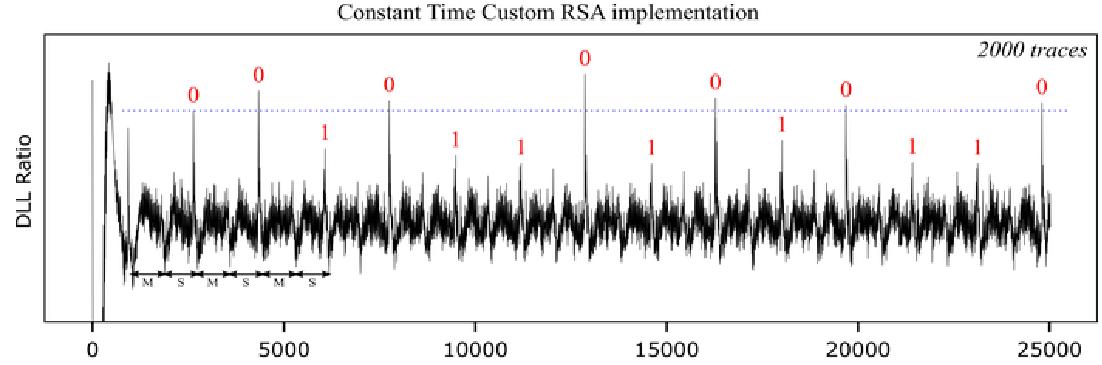
1. **AP-vs-AP attack on Cortex-A9 SoC (Zybo)**
2. MCU-vs-AP attack on Cortex-A7-M4 SoC (STM32MP1)
3. AP-vs-MCU attack on Cortex-A7-M4 SoC (STM32MP1)

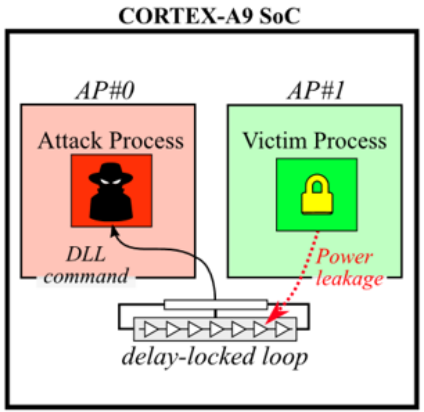


SPA attacks on Zynq

- Custom RSAs based on WolfSSL crypto libraries





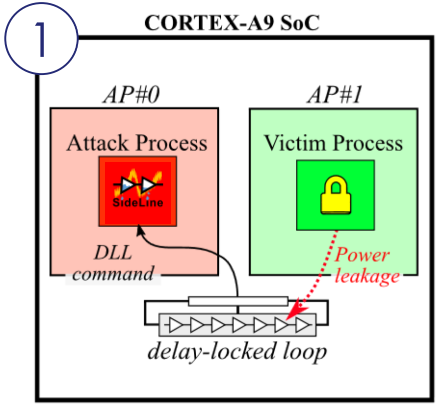


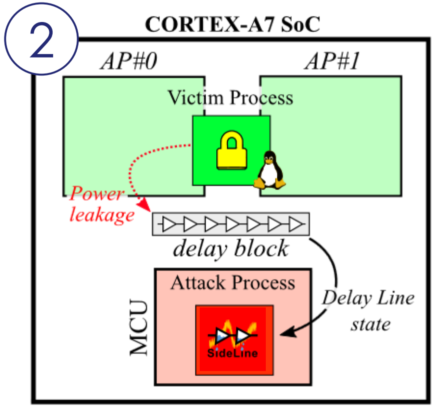
Zybo-Z7-10

SideLine Attacks Scenarios

3 core-vs-core attack scenarios on two distinct SoC:

1. AP-vs-AP attack on Cortex-A9 SoC (Zybo)
2. **MCU-vs-AP attack on Cortex-A7-M4 SoC (STM32MP1)**
3. AP-vs-MCU attack on Cortex-A7-M4 SoC (STM32MP1)

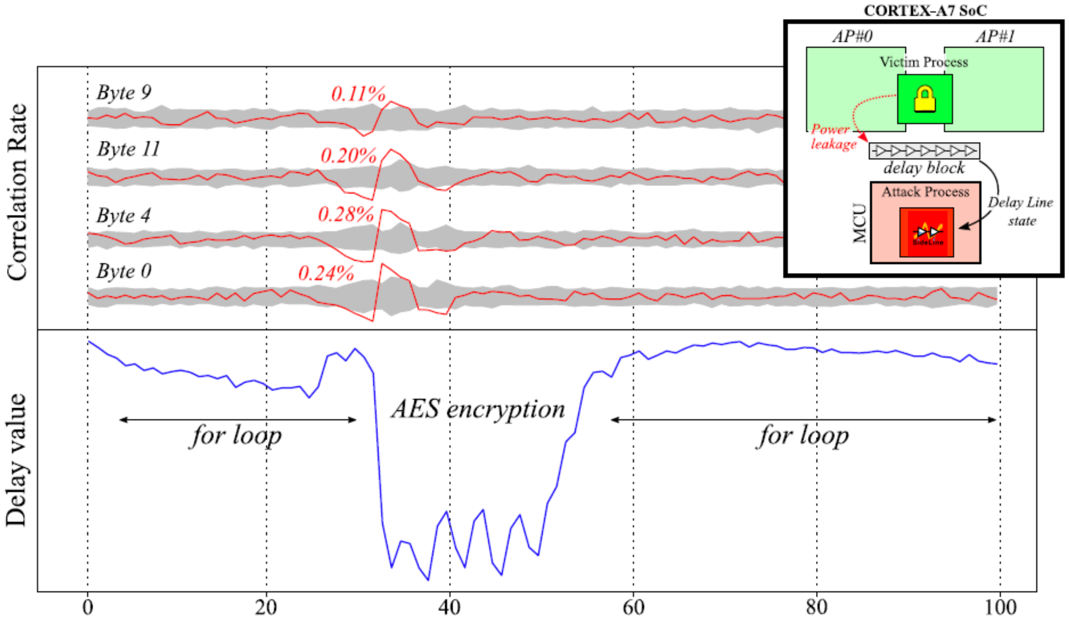




MCU-vs-AP Attack on STM32MP1

- Cortex-M4 MCU runs the **attacker process** (DL sampling)
- Cortex-A7 AP runs the **victim process** (OpenSSL AES encryption)
- Can we perform CPA full key recovery ? **yes:** 40M traces

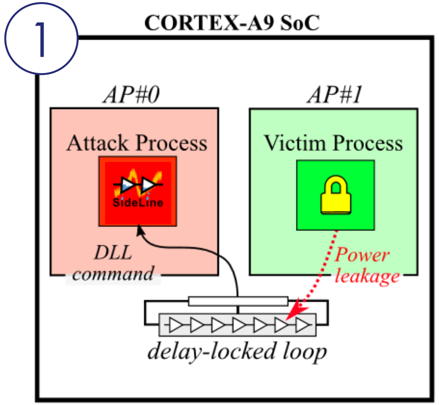
STM32MP1

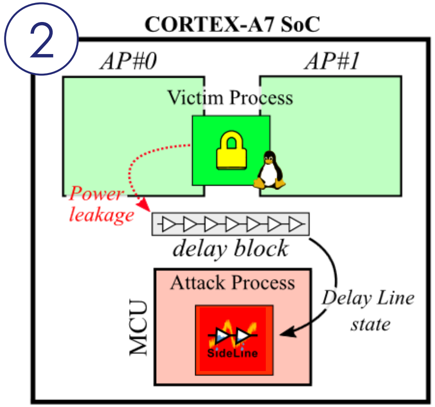


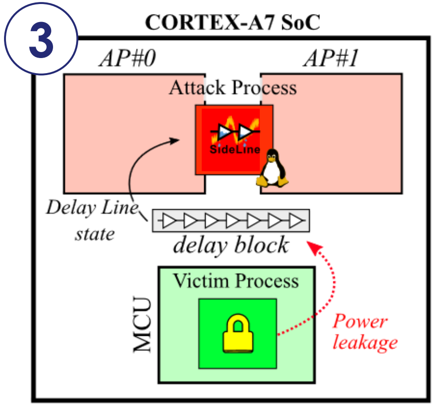
SideLine Attacks Scenarios

3 core-vs-core attack scenarios on two distinct SoC

1. AP-vs-AP attack on Cortex-A9 SoC (Zybo)
2. MCU-vs-AP attack on Cortex-A7-M4 SoC (STM32MP1)
3. **AP-vs-MCU attack on Cortex-A7-M4 SoC (STM32MP1)**



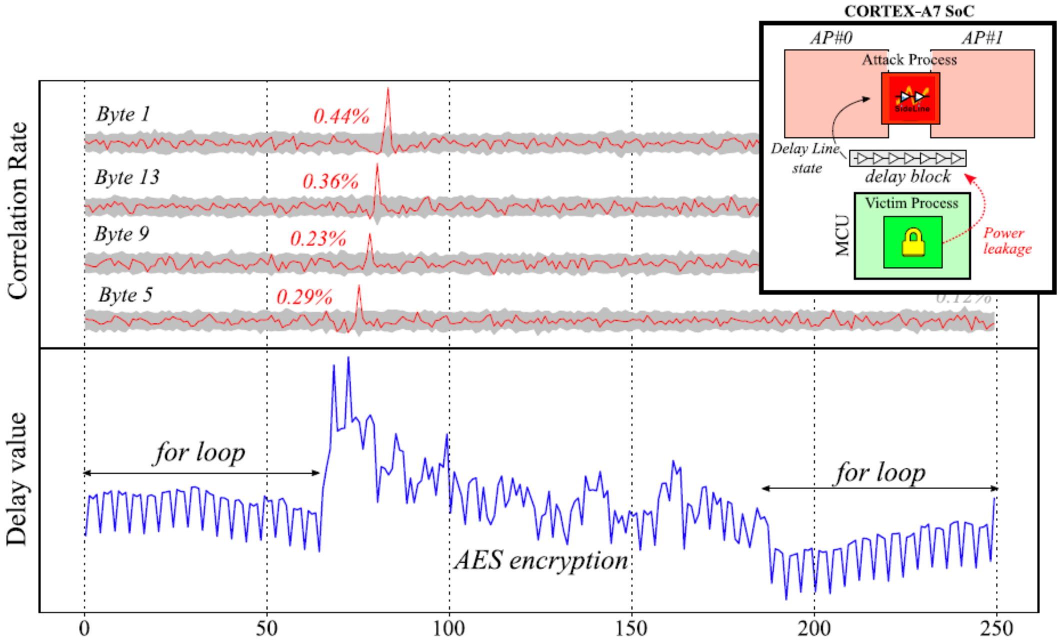




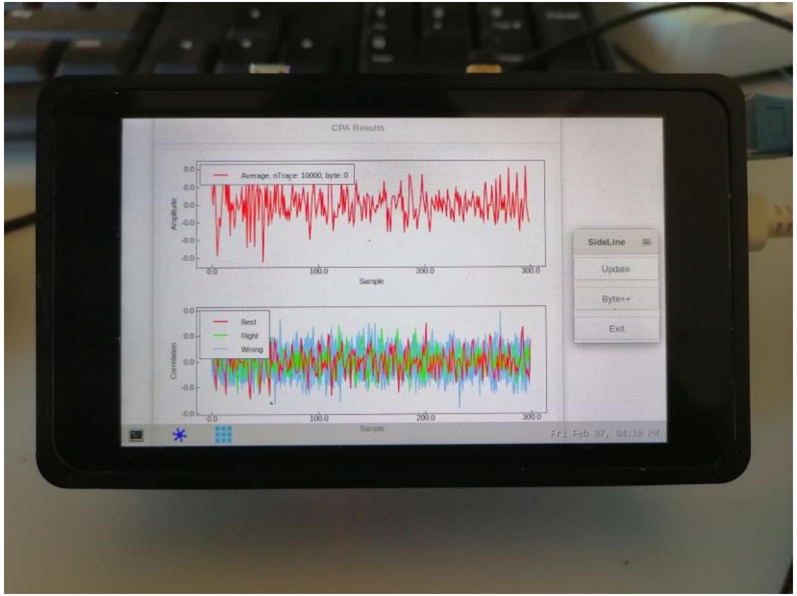
AP-vs-MCU Attack on STM32MP1

- Cortex-A7 AP runs **the attacker process** (DL sampling)
- Cortex-M4 MCU runs **the victim process** (OpenSSL AES encryption)
- Can we perform CPA full key recovery ? **yes: 5M** traces

STM32MP1

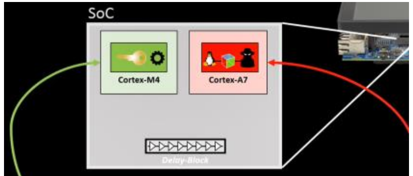


SideLine Demo on STM32MP1



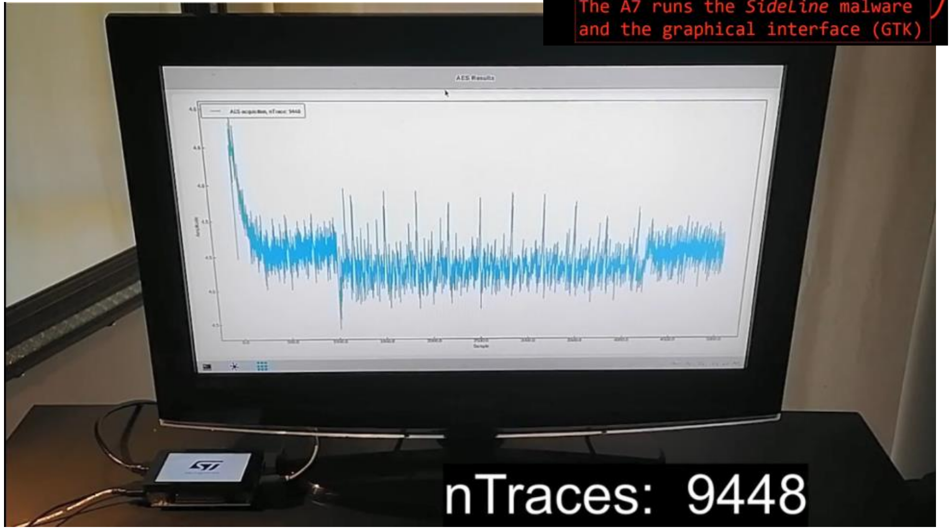
SideLine Embedded CPA

STM32MP1



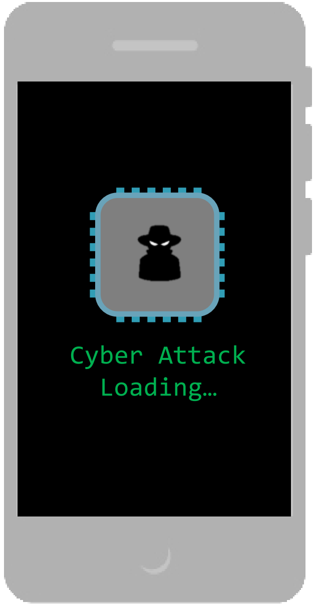
The M4 runs tiny AES encryptions

The A7 runs the *SideLine* malware and the graphical interface (GTK)



SideLine Demo tiny AES

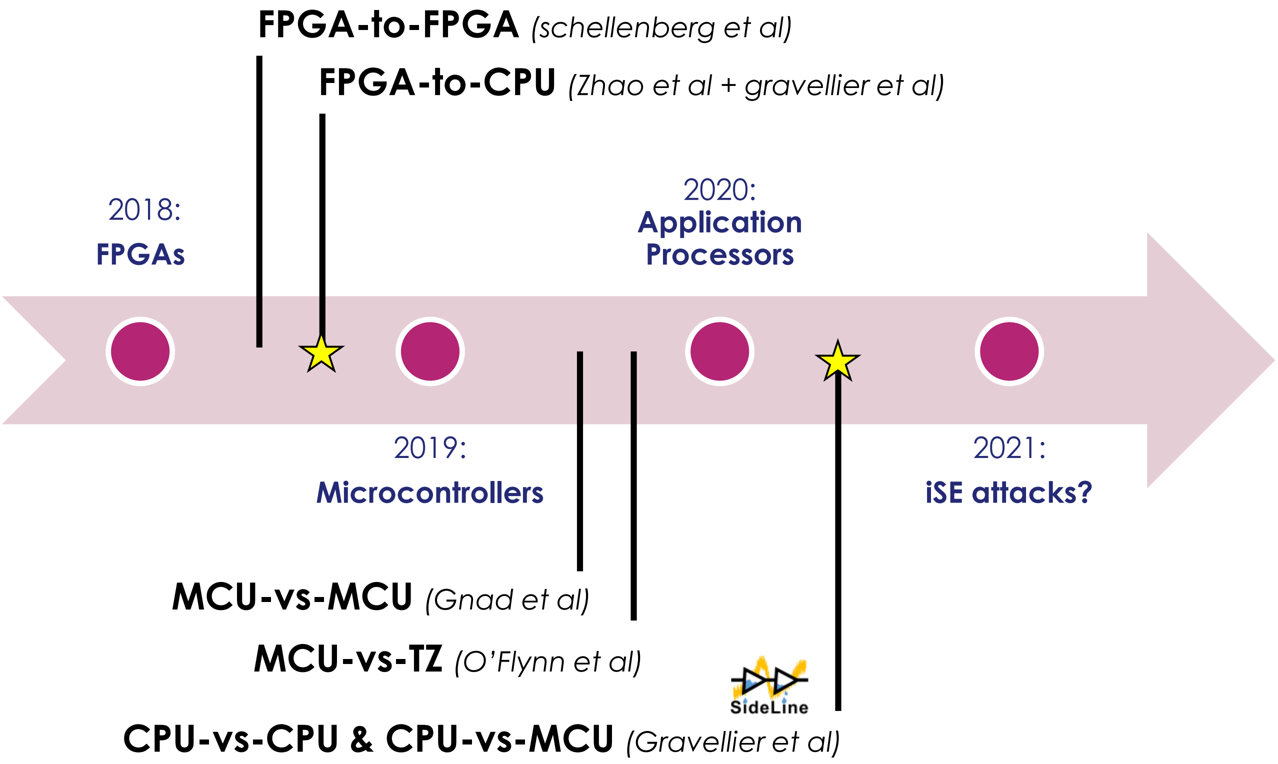
Agenda



- What is a SW-based Hardware Attack ?
- An Overview of the SW-based Power Side-Channel Analysis Works
- Introducing SideLine:
- **Impact for connected devices security**

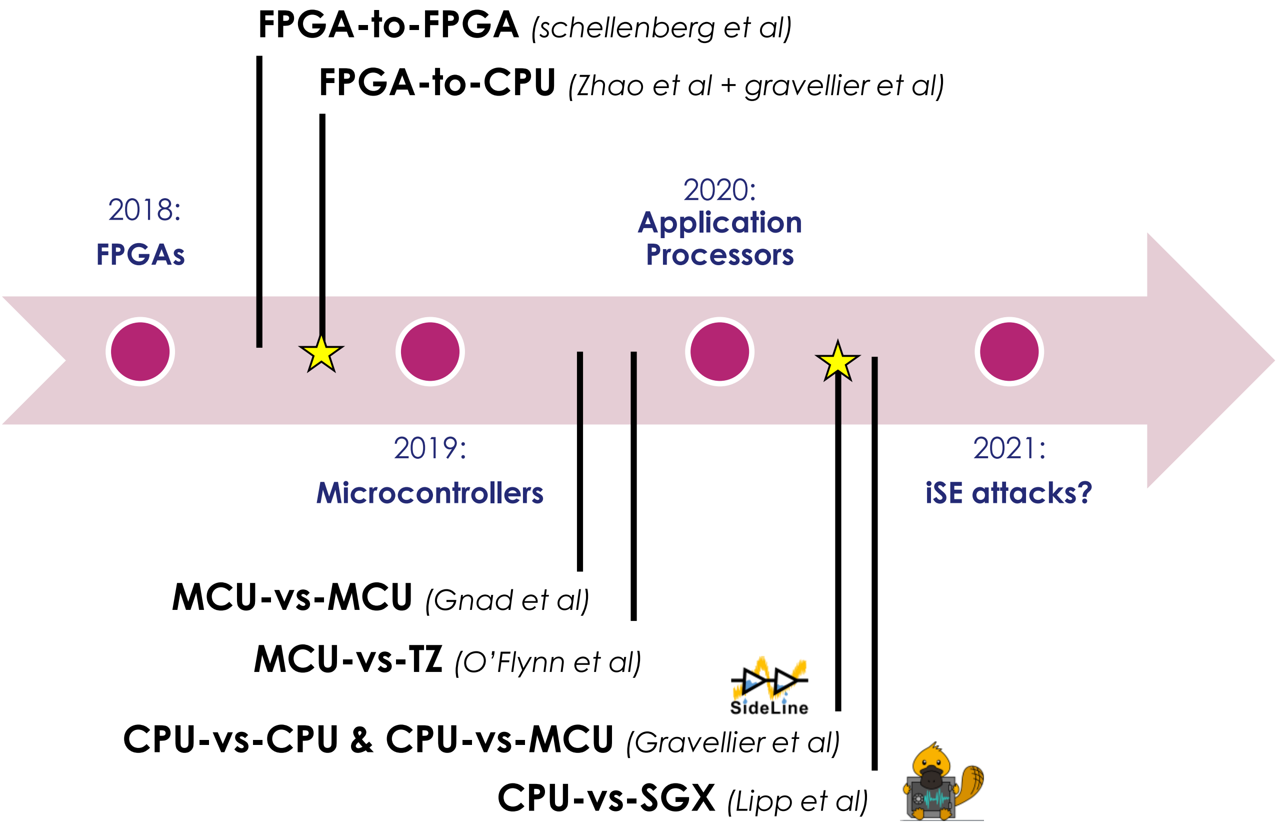
Software-based Power SCA **Timeline**

toward iSE exploit ?


FPGA-to-FPGA *(schellenberg et al)*
FPGA-to-CPU *(Zhao et al + gravellier et al)*
2018:
FPGAs
2020:
Application Processors
2019:
Microcontrollers
2021:
iSE attacks?
MCU-vs-MCU *(Gnad et al)*
MCU-vs-TZ *(O'Flynn et al)*
SideLine
CPU-vs-CPU & CPU-vs-MCU *(Gravellier et al)*

Software-based Power SCA **Timeline**

toward iSE exploit ?



FPGA-to-FPGA *(schellenberg et al)*

FPGA-to-CPU *(Zhao et al + gravellier et al)*

2018:
FPGAs

2019:
Microcontrollers

2020:
Application Processors

2021:
iSE attacks?

MCU-vs-MCU *(Gnad et al)*

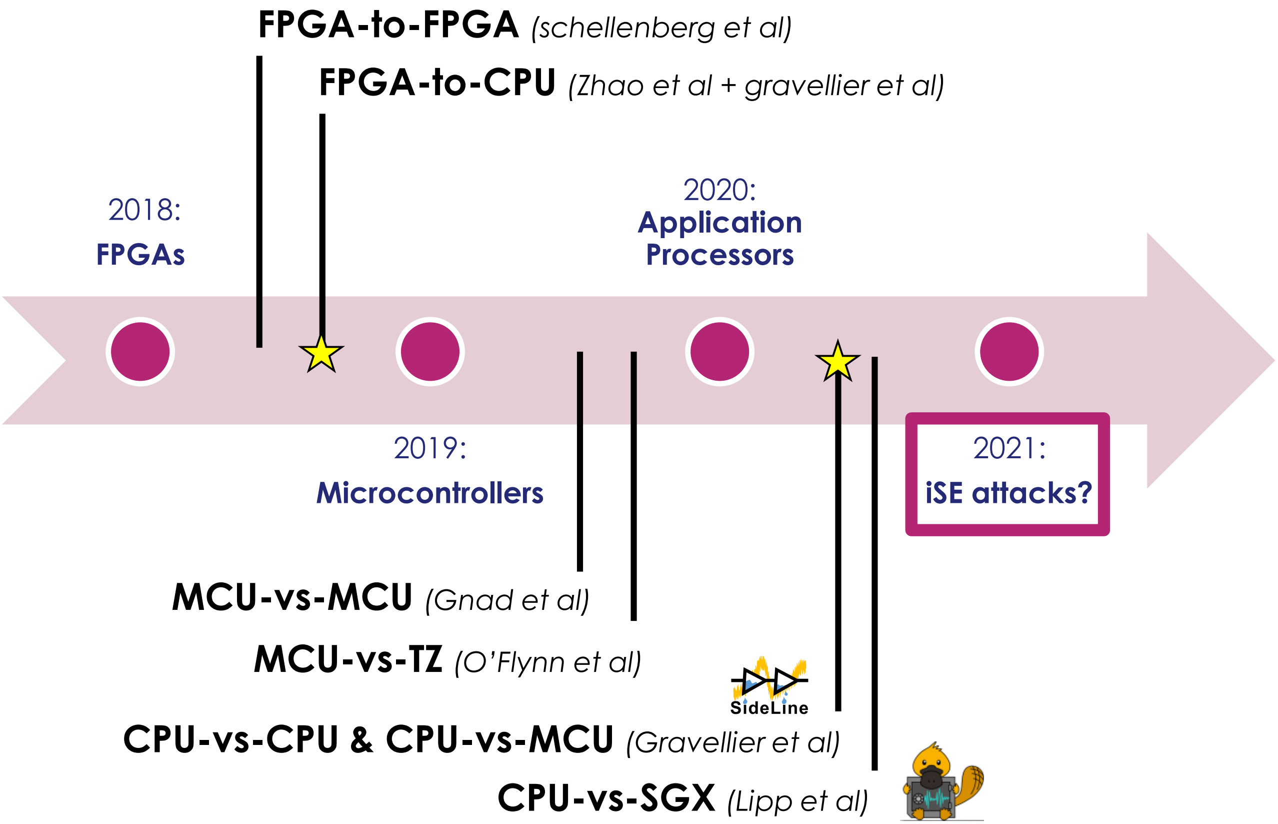
MCU-vs-TZ *(O'Flynn et al)*

CPU-vs-CPU & CPU-vs-MCU *(Gravellier et al)*

CPU-vs-SGX *(Lipp et al)*

Software-based Power SCA **Timeline**

toward iSE exploit ?



FPGA-to-FPGA *(schellenberg et al)*

FPGA-to-CPU *(Zhao et al + gravellier et al)*

2018:
FPGAs

2020:
Application Processors

2019:
Microcontrollers

2021:
iSE attacks?

MCU-vs-MCU *(Gnad et al)*

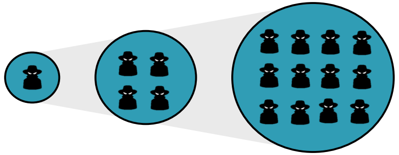
MCU-vs-TZ *(O'Flynn et al)*

CPU-vs-CPU & CPU-vs-MCU *(Gravellier et al)*

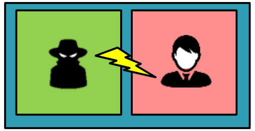
CPU-vs-SGX *(Lipp et al)*

Software-based hardware attacks summary

Remote



Scalable



Hardware

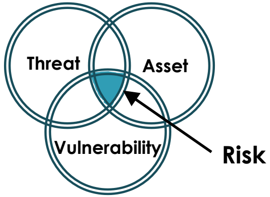
Software-based hardware attacks summary

Remote

Scalable

Hardware

Anticipation



Threat Assessment

Counter measures

Thank you, Questions ?

joseph.gravellier@external.thalesgroup.com

dutertre@emse.fr

yannick.teglia@thalesgroup.com

philippe.loubet-moundi@thalesgroup.com

@joseph_gravel

SideLine
Website

https://josephgravellier.github.io/sideline/

Institut Mines-Télécom

MINES Saint-Étienne
Une école de l'IMT

THALES