Open Hardware and Software Architecture for Safe and Secure IoT Systems Design and Evaluation

David Hély, Cyril Bresch, Athanasios Papadimitriou, Zahra Kazemi and Stéphanie Chollet

Univ. Grenoble Alpes, LCIS, F-26000 Valence, France
david.hely@grenoble-inp.fr

Journées nationales 2018 Pré-GDR Sécurité Informatique

Laboratoire de Conception et d'Intégration des Systèmes

Agenda

I. **Context**
II. Open Platform for Hardware Security Evaluation
III. Hardware Support for Software Security
IV. Conclusions

Serene IoT

Secured & EneRgy EfficieNt hEalth-care solutions for IoT market

- Develop **high quality connected care services** and diagnosis tools based on **Advanced Smart Health-Care IoT** devices thanks to:
  - High healthcare quality service for patients remotely followed by professional caregivers
  - **High level of trust** (Security, Safety, Privacy, Robustness)
  - Good and fast execution of requested tasks
  - Interoperability and compatibility
  - Much lower cost than the traditional care provided today

SERENE-IoT project is a project labelled within the framework of PENTA, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

LCIS Contribution Objectives

- Focus on the IoT node to provide a **safe and secure embedded computing unit** which offers **hardware threats countermeasures and hardware based security features**:
  - Balancing power and security criteria
  - Considering a large attack surface
  - Combining safety and security
  - Accessible to non security experts…

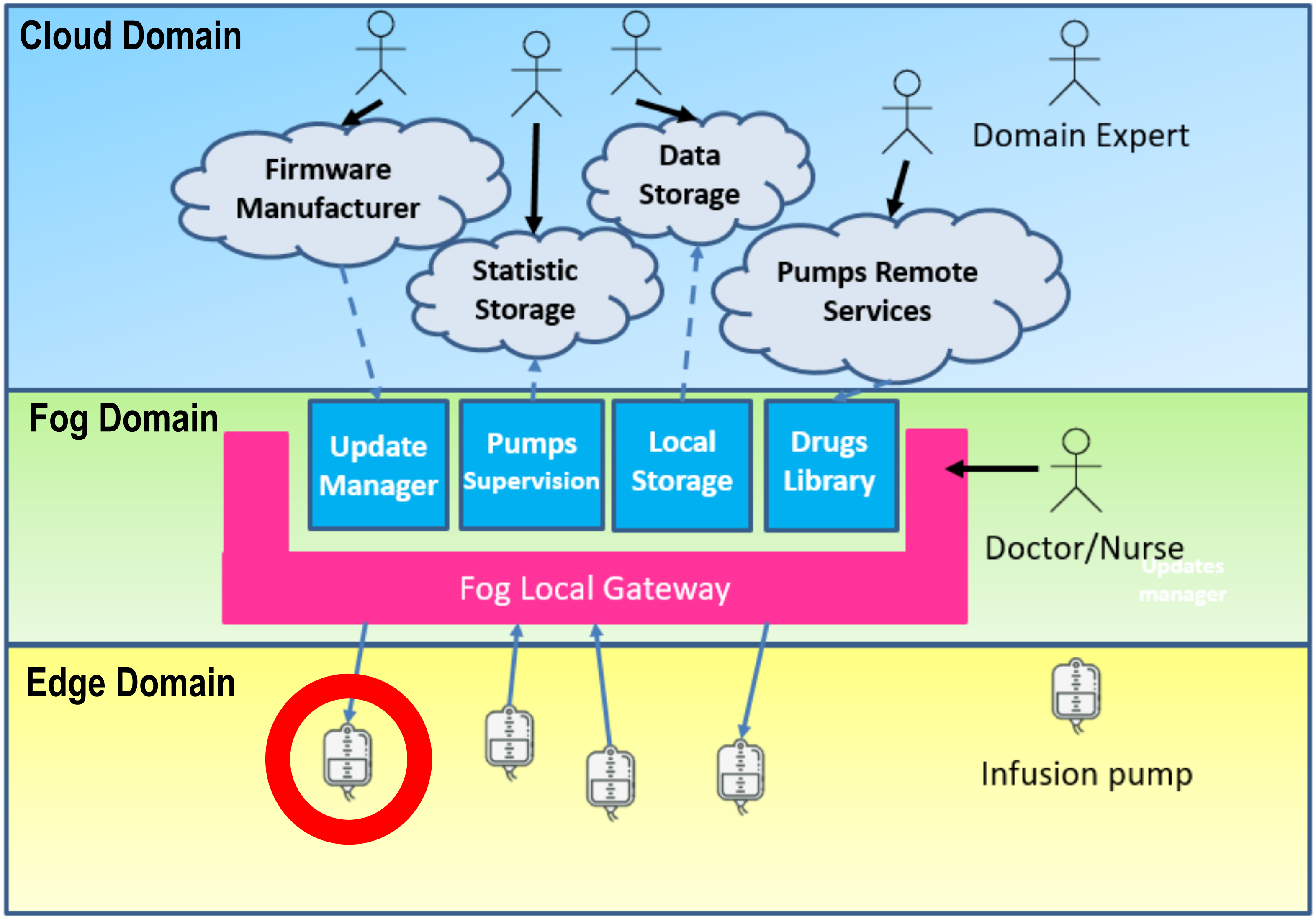
(1) Tools to evaluate security threats against IoT nodes and to validate and optimize the countermeasures

(2) Open architecture to explore HW support for system security

Laboratoire de Conception et d'Intégration des Systèmes

Application to a Medical Use Case

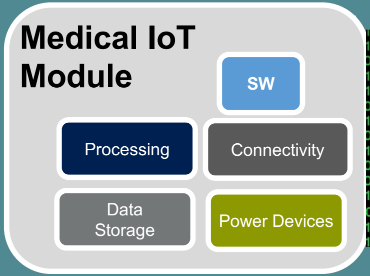
- Infusion Pump



Stressing this use case

Security Evaluation Platform

Hardware Security



Software Security

« Plug and Play » HW and SW attacks set-up for early evaluation

Vulnerable assets: patient data, proprietary firmware, access rights, system safety, aplication data

Security Evaluation, Why?

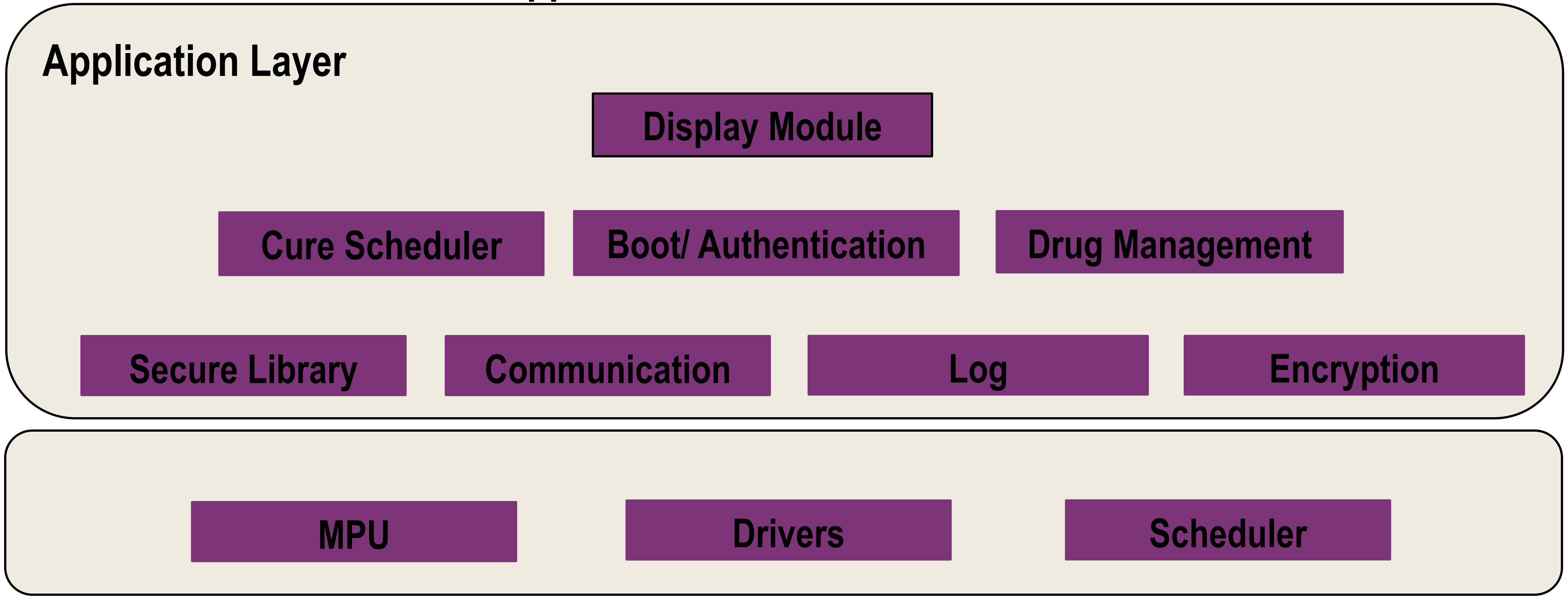
- Threats Identification
- Countermeasures Design and Evaluation

For both **Teaching and Research purposes**

LCIS Laboratoire de Conception et d'Intégration des Systèmes

Infusion Pump Model

- **Software Overview of the application:**



- Infusion Pump Model is running on:
  - STM32 nucleo-board
  - RISC-V based FPGA
- The model is free to use in order to:
  - Demonstrate vulnerabilities
  - Integrate and evaluate countermeasures at any levels

On-going Works

Based on this use case, we are addressing:

- Dynamic authentication of users and Assets through the system lifecycle
- Fault and Side Channel attacks against MCU based systems
  - Dedicated open source evaluation platform
- Memory corruptions based attacks
  - Exploits database
  - Hardware support for secure computing

Laboratoire de Conception et d'Intégration des Systèmes

Agenda

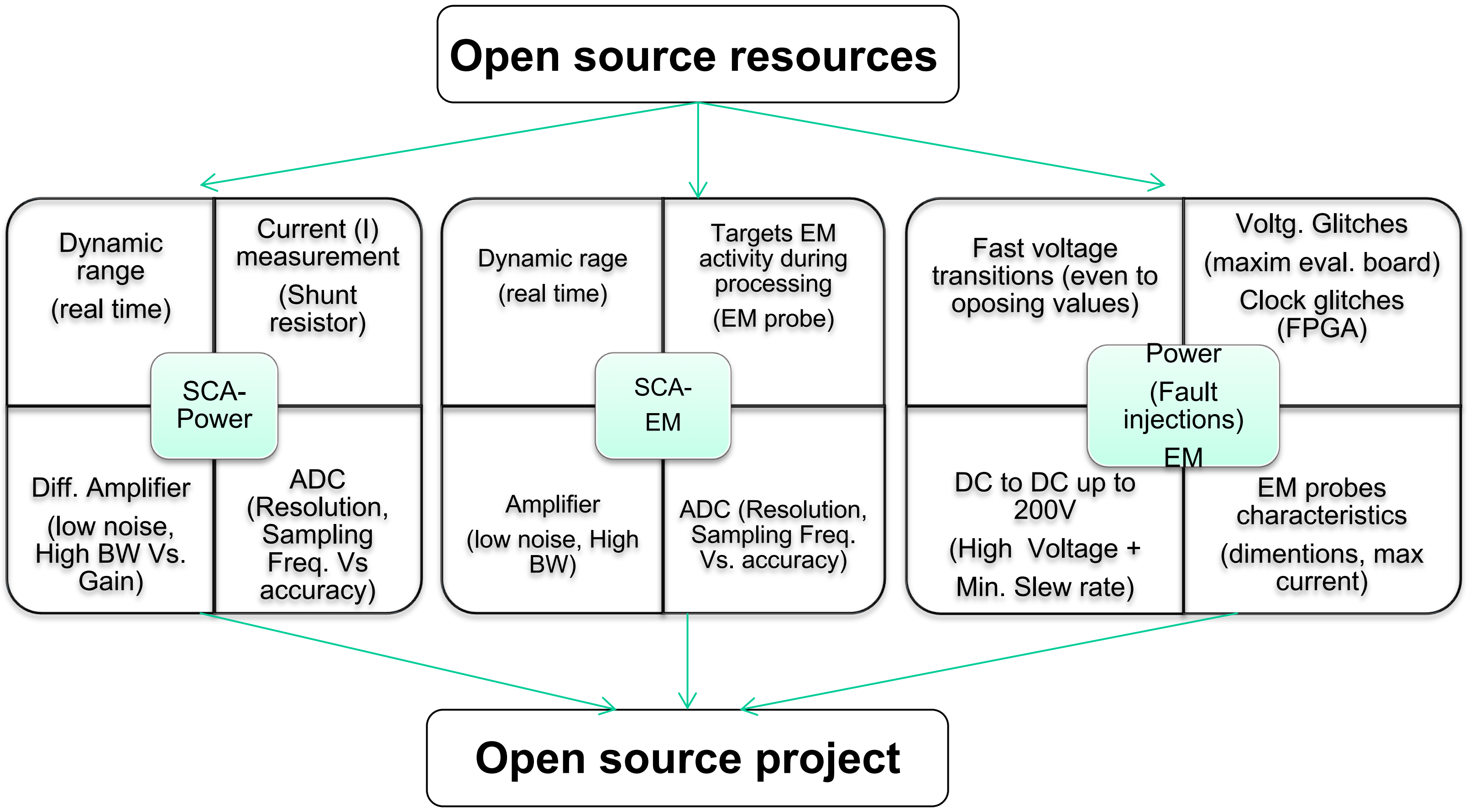
Open Platform for Hardware Security Evaluation

Accurate and accessible evaluation platforms are needed to:

- Make hardware security assessments available to developers which are not hardware security experts
- Minimize the impact of hardware security on design constraints
- Guarantee a specific level of security, depending on the application type

Evaluation platform for Power analysis and Glitch attacks

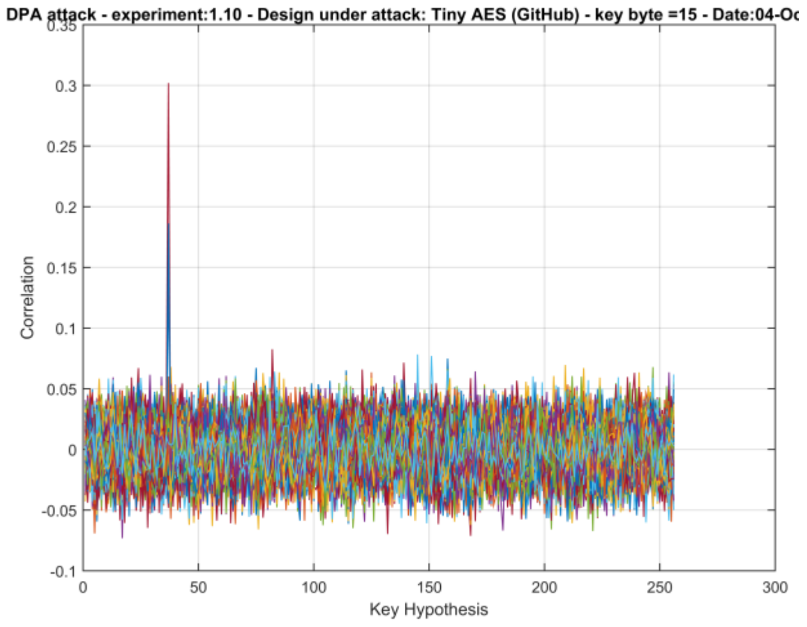
- Security evaluation capabilities to be integrated on the platform:

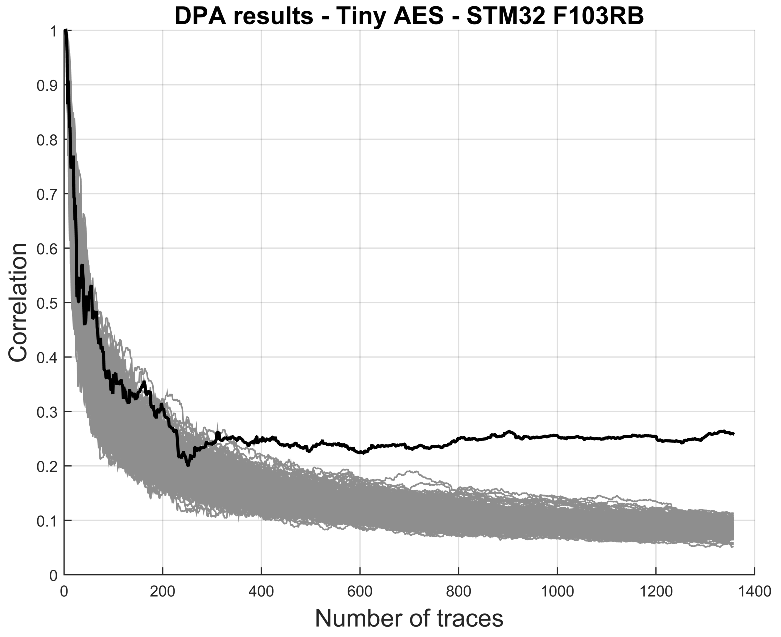


LCIS Laboratoire de Conception et d'Intégration des Systèmes

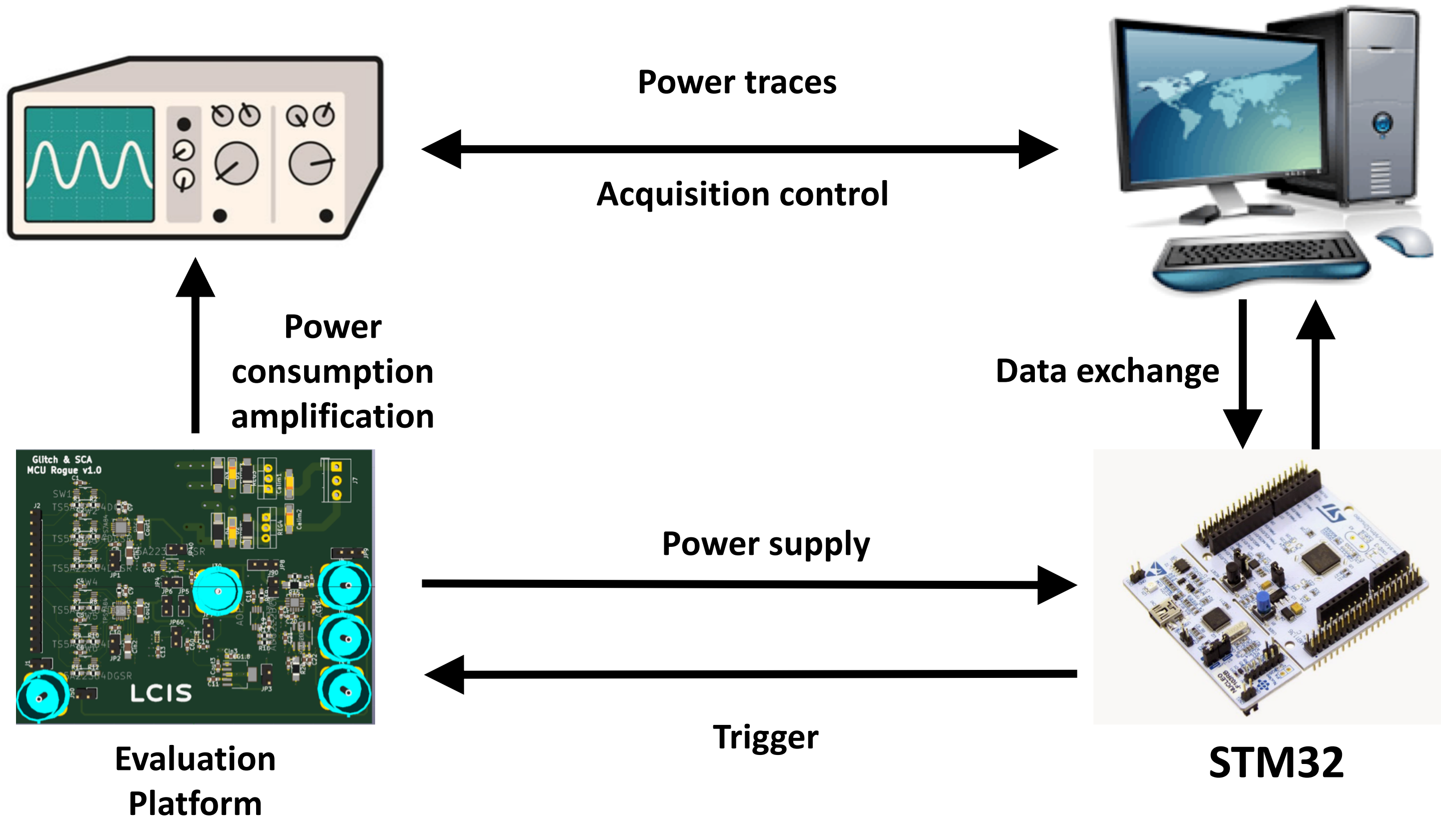
Low-side power analysis attack

- Low-side power analysis attack needs only an oscilloscope and is capable to expose the secret key with just 400 traces for a STM32-F1 series MCU (64 MHz)
- Is a more accurate high-side evaluation platform really needed?





Power analysis evaluation platform

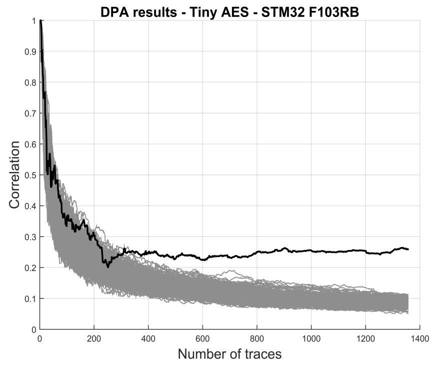


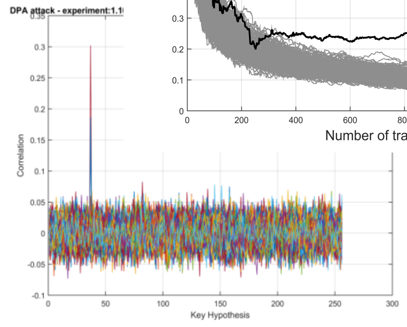
Power analysis attacks

The evaluation board integrates:

- **Low noise, adjustable power supply for MCUs (STM32) and FPGAs**
- **High-side current measurement capabilities for power analysis**
- **The current is converted to voltage by means of a shunt resistor and an instrumentation amplifier**
  - Capabilities: Gain: 1~1000, Bandwidth: 1.2MHz~15kHz
  - Gain was fixed at 100 with a bandwidth of 200kHz

- **Target of the attack:**
  - Open-source unprotected AES





LCIS
Laboratoire de Conception et d'Intégration des Systèmes

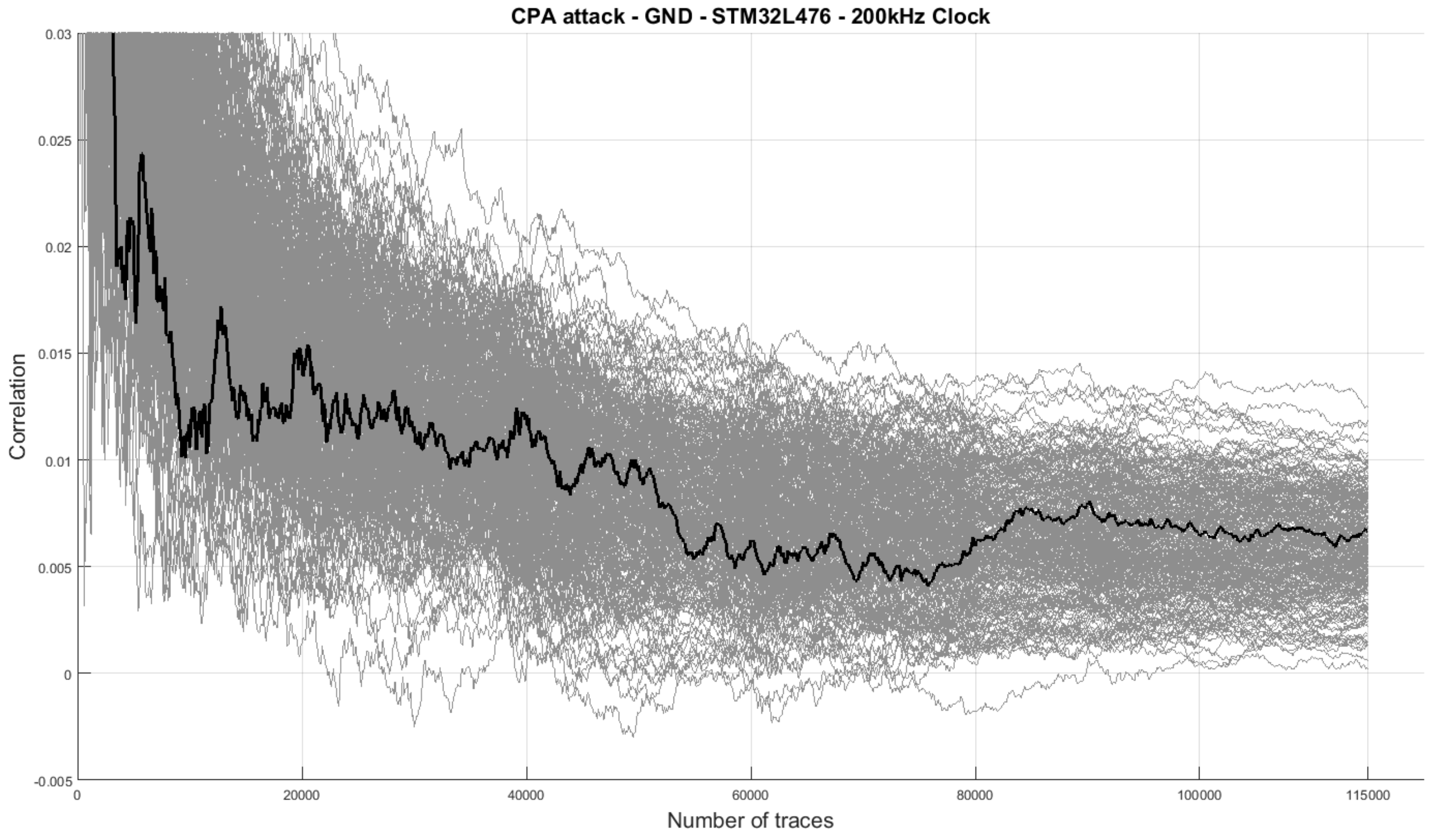
Results

Experiment on a low power STM32-L4 series MCU:

- **Low-side (GND) power analysis**
- **High-side (VDD) power analysis**

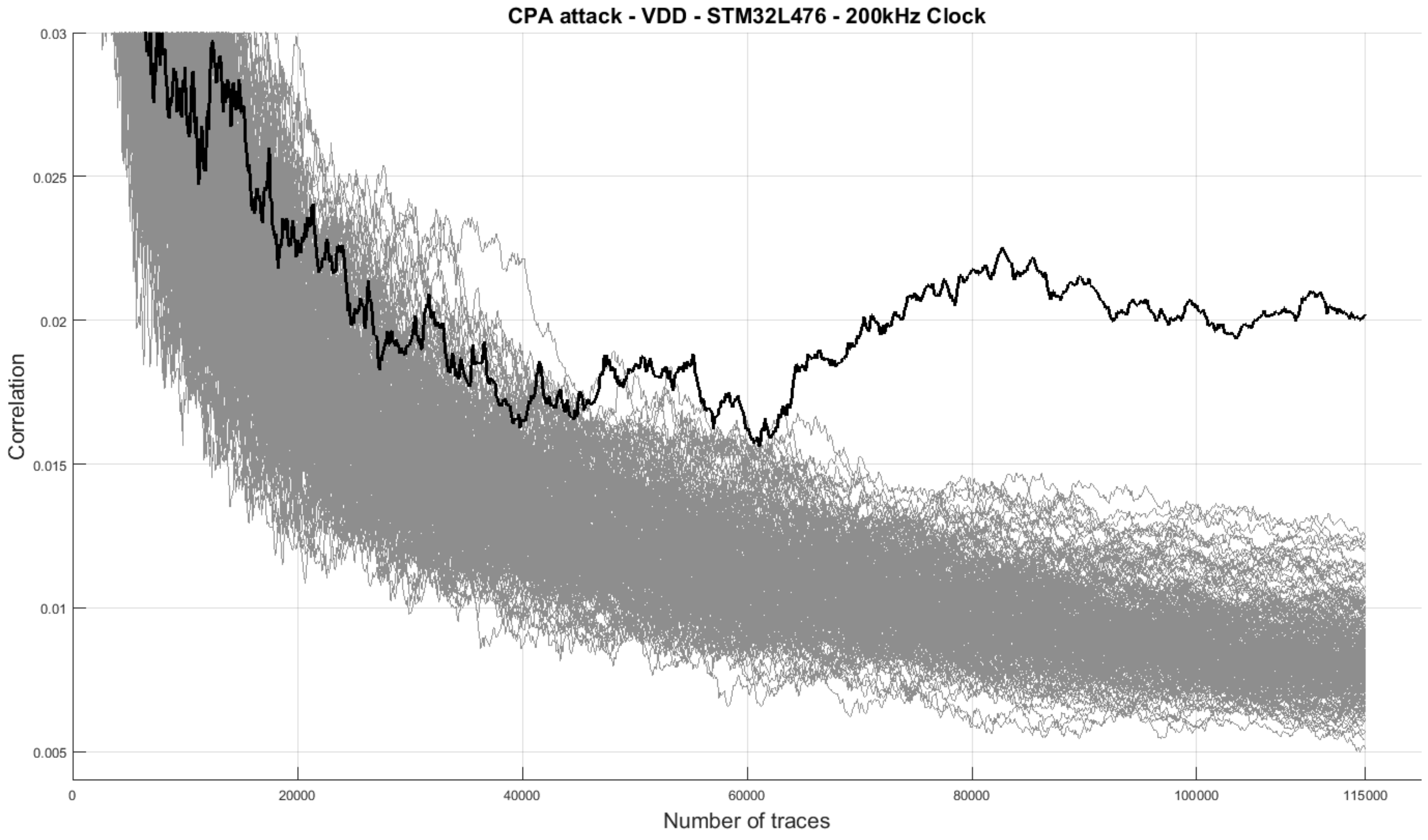
Laboratoire de Conception et d'Intégration des Systèmes

Low-side attack using an oscilloscope



LCIS
Laboratoire de Conception et d'Intégration des Systèmes

High-side attack using the developed platform



LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Platform Integration

Current sensing-
High side

Signal detection

Power fluctuations

Differential input /
Single ended output

Signal conditioning

Topologies for:
High BW-Low Gain
High Gain-Low BW

ADC– Driver
(ADC compatibility)

Signal digitization preperation

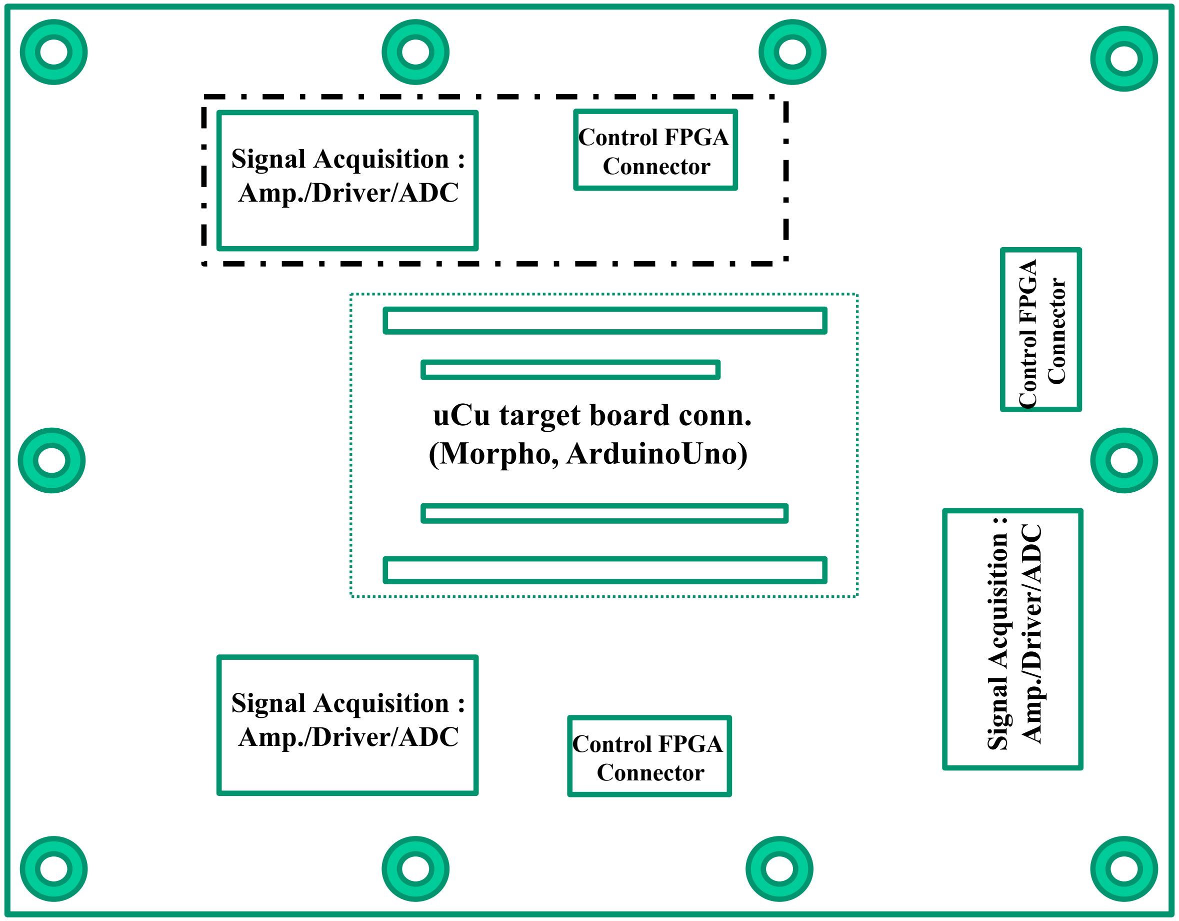
Utilized for:
- Single input to Diff output
- Signal level adjustment

Diff. Analog input
Digital Output
(Parallel/Serial)

Signal digitization

Devices for:
High Bit-Resolution/
High Sampling-Rate

Platform Integration


Signal Acquisition : Amp./Driver/ADC
Control FPGA Connector
Control FPGA Connector
uCu target board conn. (Morpho, ArduinoUno)
Signal Acquisition : Amp./Driver/ADC
Signal Acquisition : Amp./Driver/ADC
Control FPGA Connector

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

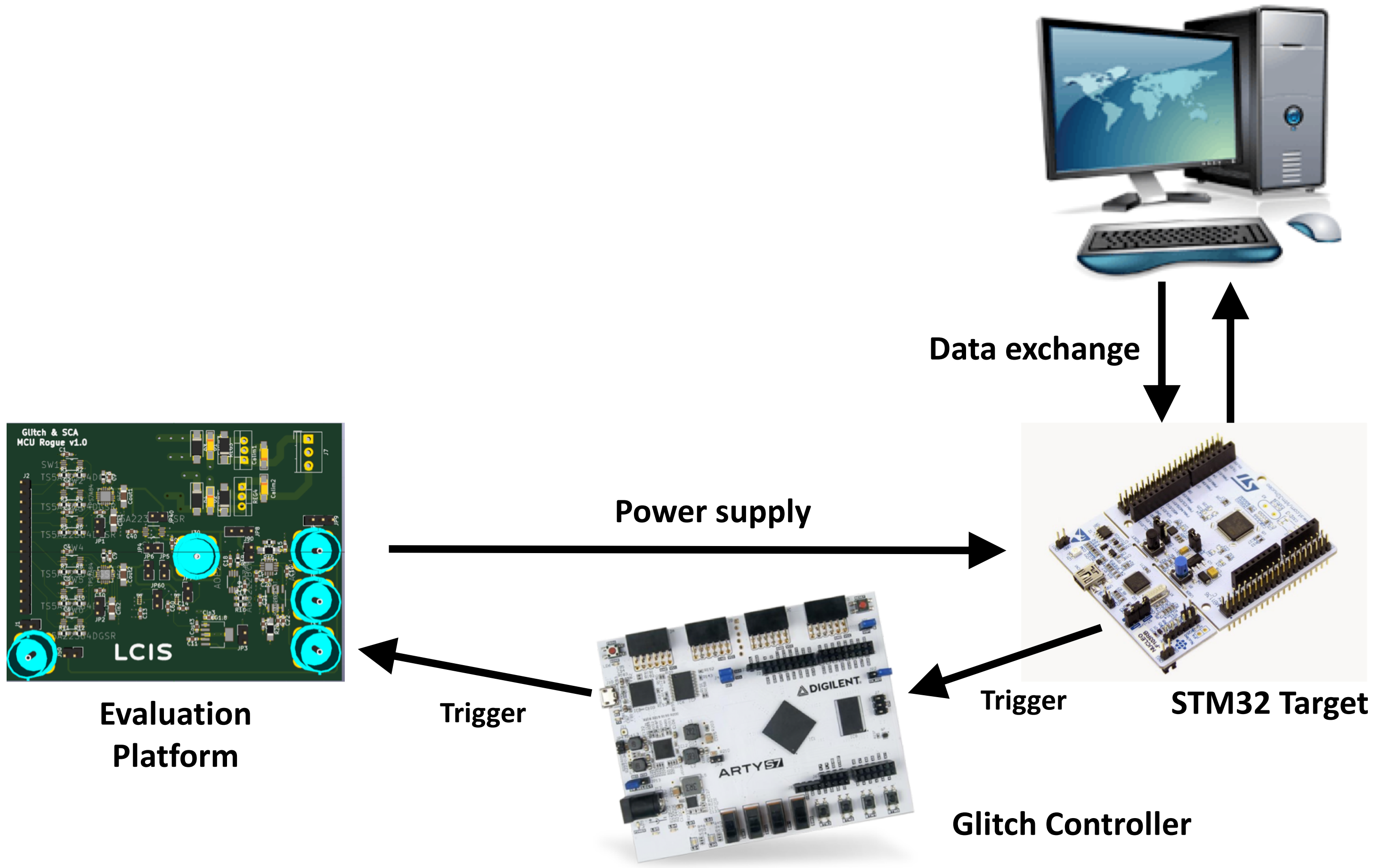
Main results

- **An unprotected AES implementation is vulnerable even when running at a low power MCUs**
- **A low-side analysis @200kHz using just an oscilloscope would not expose the key after 115,000 traces giving a false sense of security**
- **An analysis at higher clock speeds would potentially expose the key with less traces even with the low-side measurement**
  - This could lead to the development of countermeasure which could provide a larger than the necessary level of protection

Power Glitch Attacks

- **The evaluation board integrates:**
  - Two low noise controllable power supplies to control the height of the glitch
    - 0.8V to 5V in 50mV steps, 3A max
  - Switches to control the temporal length of the glitch
    - Turn-on & turn-off time of about 50ns
    - Turn-on & turn-off time of about 30ns

Glitch attack evaluation platform



On-going steps

- Perform glitch attacks on MCUs
- Perform power analysis evaluations on cryptographic implementations including countermeasures
- Improve the platform and make it possible for software developers to perform accurate security evaluations

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Agenda

Hardware Support For Secure Processing

- **Objectives**
  - To develop hardware primitives to allow secure software execution with:
    - **Easy adoption** by software community
    - **Low** hardware **resources**
    - No **impact** on **performances** and **safety**
  - To deliver an open platform for secure MCU architecture exploration:
    - **FPGA** based MCU prototyping **platform**
    - Comprehensive set of **software exploits** to evaluate the countermeasures

Laboratoire de Conception et d'Intégration des Systèmes

The Approach

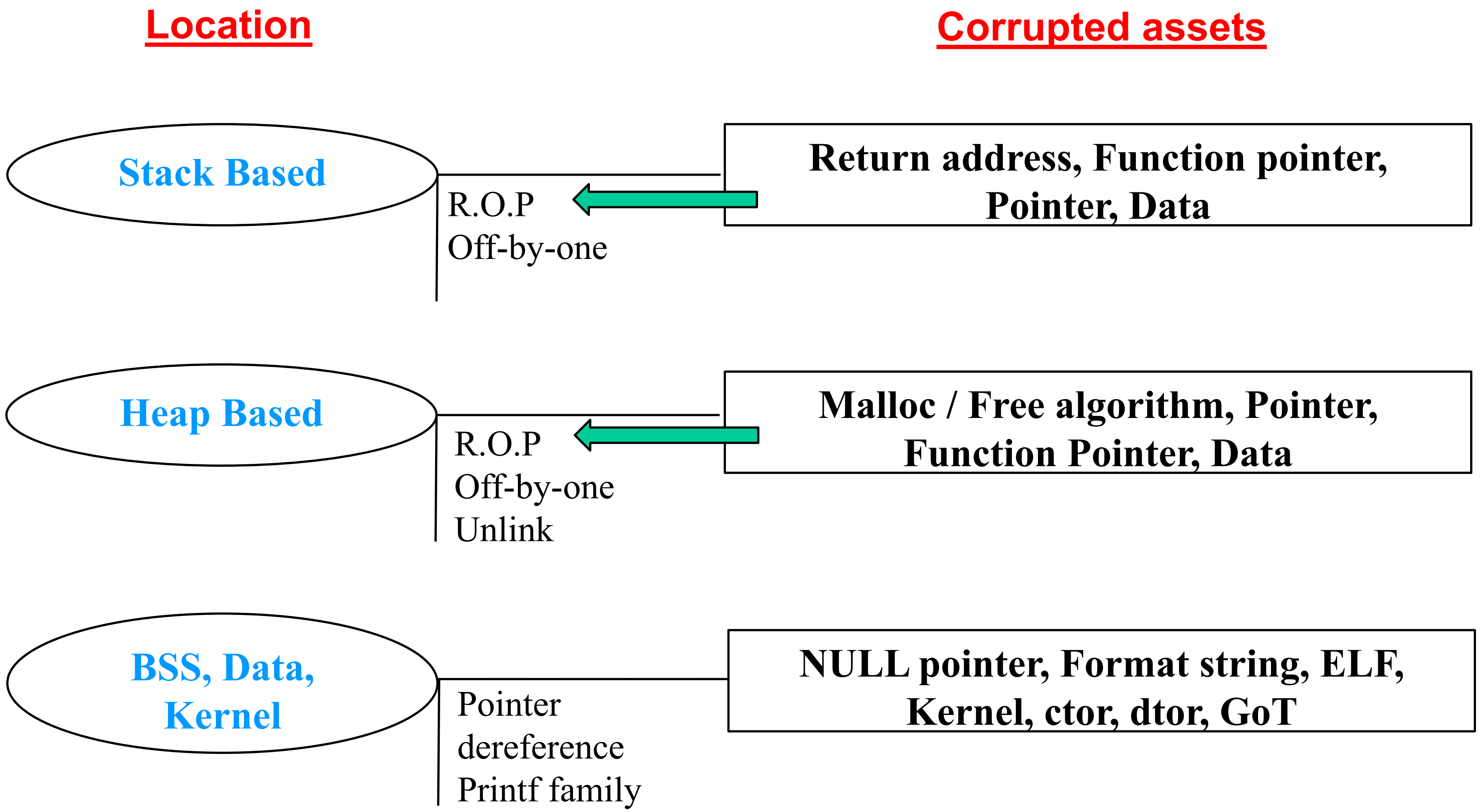
- **Red Team approach : Threats analysis and Software Exploits development:**
  – Leaking sensitive data in memory
  – Corrupting the execution flow
  – Modifying program behavior
  – Denial-of-Service

- **Blue Team approach : Hardware based Countermeasures Development**
  – Targeting one entry point of such attacks.
  – Suggesting hardware based solutions that thwart such exploits.
  – Implementing the design solution on a the OpenRISC core.

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

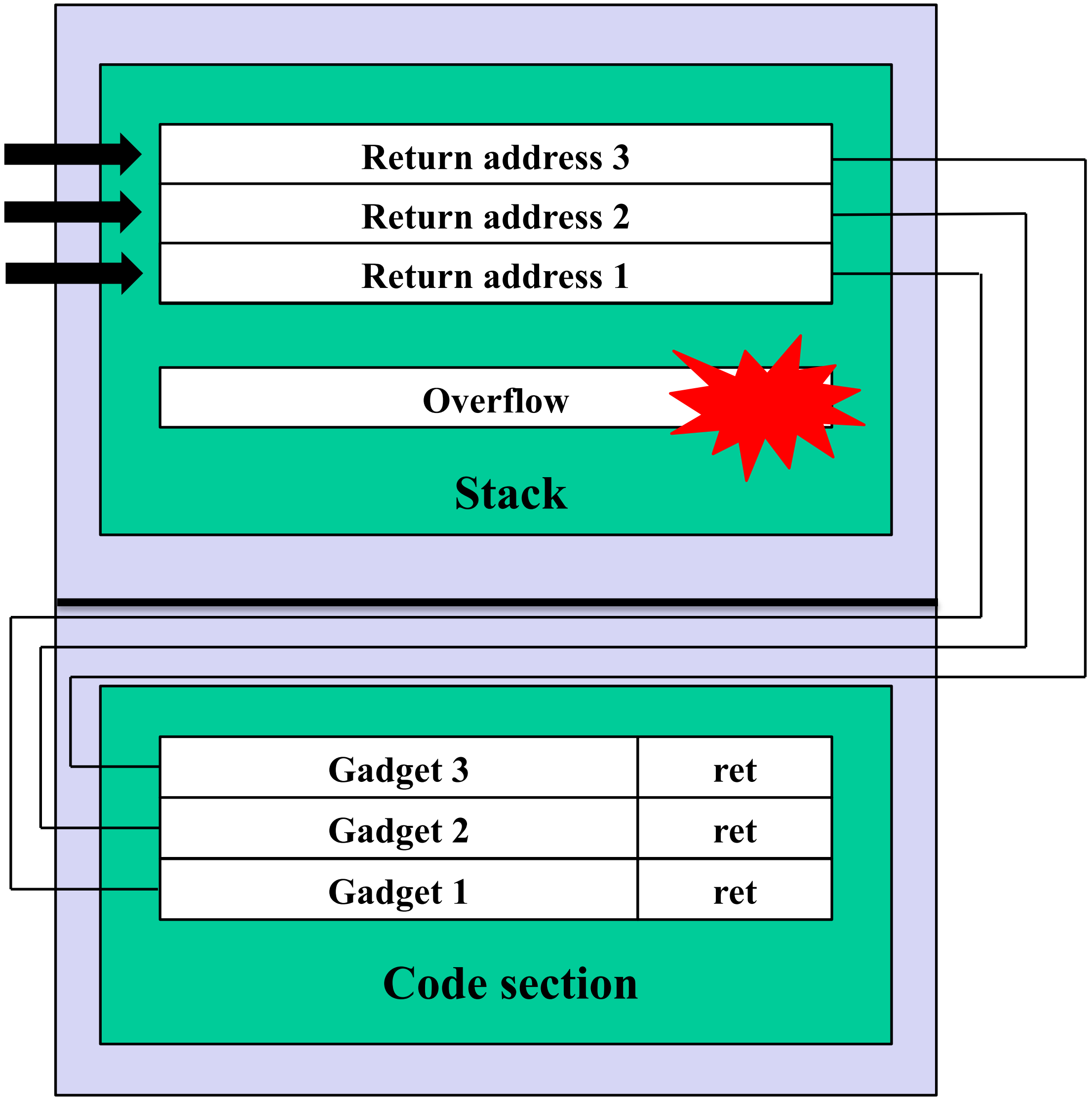
A Red Team Approach

- We are using the current exploitation techniques in order to corrupt the execution flow.


Location
Corrupted assets
Stack Based
R.O.P
Off-by-one
Return address, Function pointer, Pointer, Data
Heap Based
R.O.P
Off-by-one
Unlink
Malloc / Free algorithm, Pointer, Function Pointer, Data
BSS, Data, Kernel
Pointer dereference
Printf family
NULL pointer, Format string, ELF, Kernel, ctor, dtor, GoT

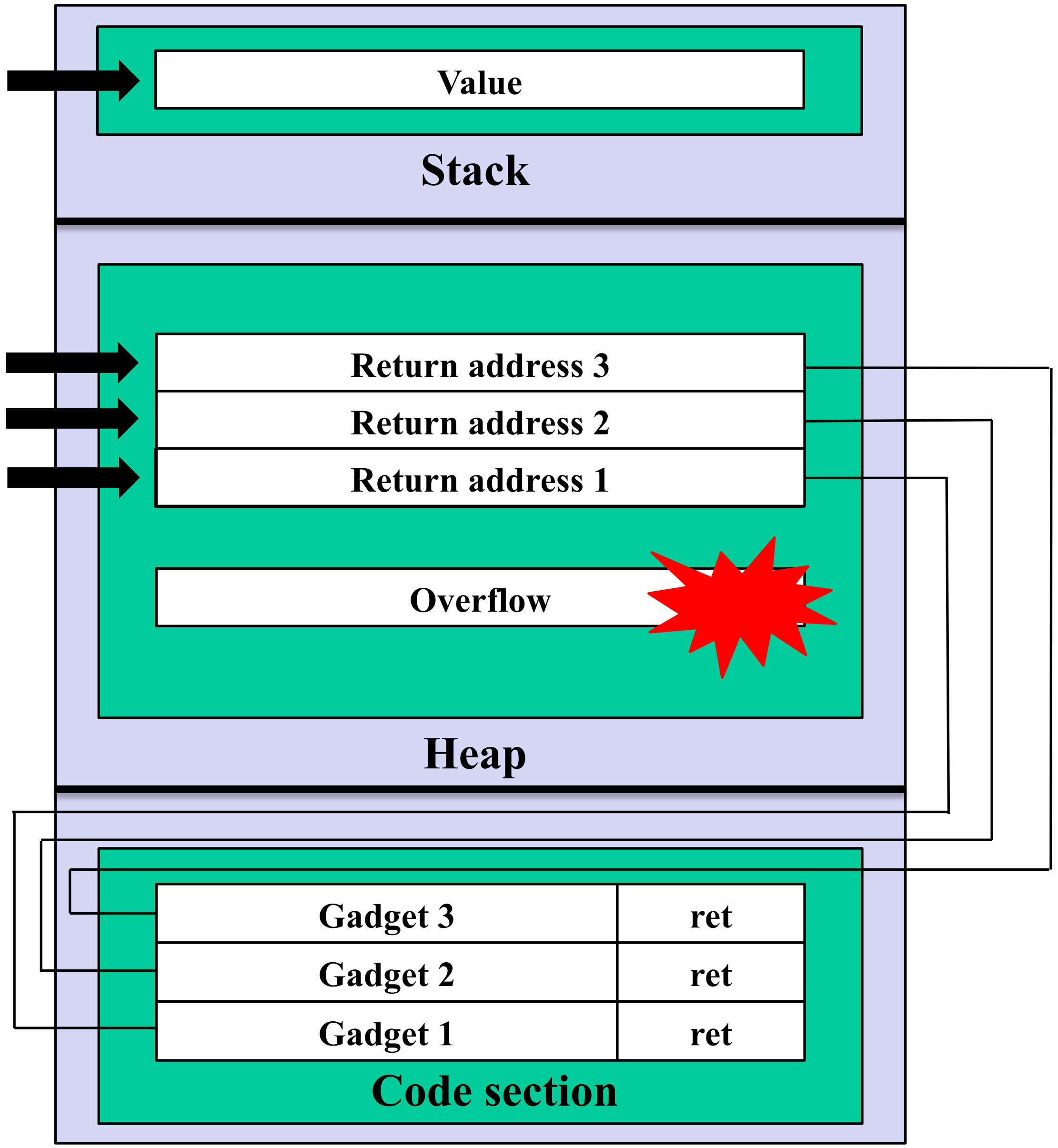
LCIS Laboratoire de Conception et d'Intégration des Systèmes

Stack based R.O.P



- Assuming we control a buffer in the stack
- Gadgets addresses injection
- Return increase stack pointer to the next address in the stack
- Gadgets are located in the code section that is executable
- Chaining gadget in order to execute the malicious function

Heap based R.O.P



- Assuming we control a buffer in the heap
- Gadgets addresses injection
- Modifying the stack pointer value
- Return increase stack pointer to the next address in the stack
- Gadgets are located in the code section that is executable
- Chaining gadget in order to execute the malicious function

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Exploits Library

- **Red Team results**
  - Exploits have been developed and validated : buffer overflow, heap overflow, format string → exploits launching shell
  - Exploits experimentally validated on RISC-V and STM32

Stack corruption is working well

☑ oASLR on 32 bits → 9 bits entropy, easy to bypass

☑ oNo Stack Guard → No return address protection

☑ oNo NX protection → Inject code is executable

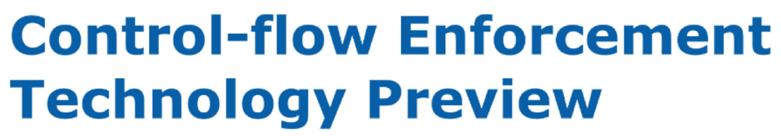
Heap overflow is working well

→ **Return address must be protected**

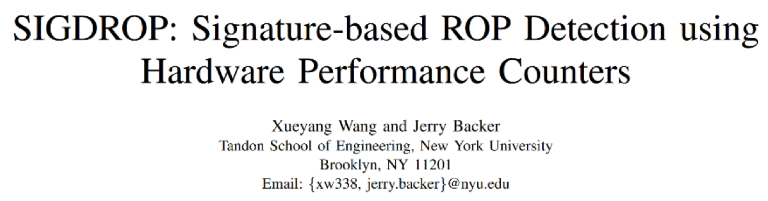
LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Existing countermeasures

- Shadows stack, Return oriented programming signature detection, Adress Space Layout Randomization..

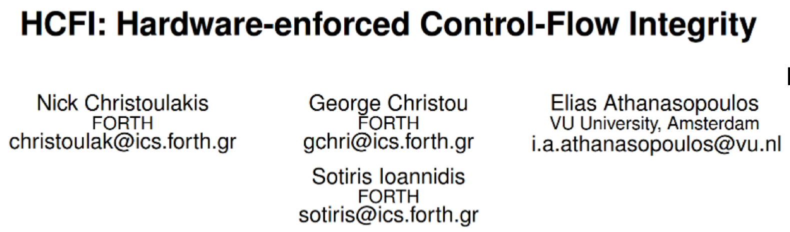

Control-flow Enforcement Technology Preview

→ Secures return address into an inaccessible shadow stack


SIGDROP: Signature-based ROP Detection using Hardware Performance Counters

Xueyang Wang and Jerry Backer
Tandon School of Engineering, New York University
Brooklyn, NY 11201
Email: {xw338, jerry.backer}@nyu.edu

→ Detects R.O.P signature during the execution flow. Too much « ret » → interruption


HCFI: Hardware-enforced Control-Flow Integrity

Nick Christoulakis
FORTH
christoulak@ics.forth.gr

George Christou
FORTH
gchri@ics.forth.gr

Elias Athanasopoulos
VU University, Amsterdam
i.a.athanasopoulos@vu.nl

Sotiris Ioannidis
FORTH
sotiris@ics.forth.gr

→ Assuring that every control-flow transfer points to a legitimate address

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Existing countermeasures

- Shadows stack, Return oriented programming signature detection, Adress Space Layout Randomization..

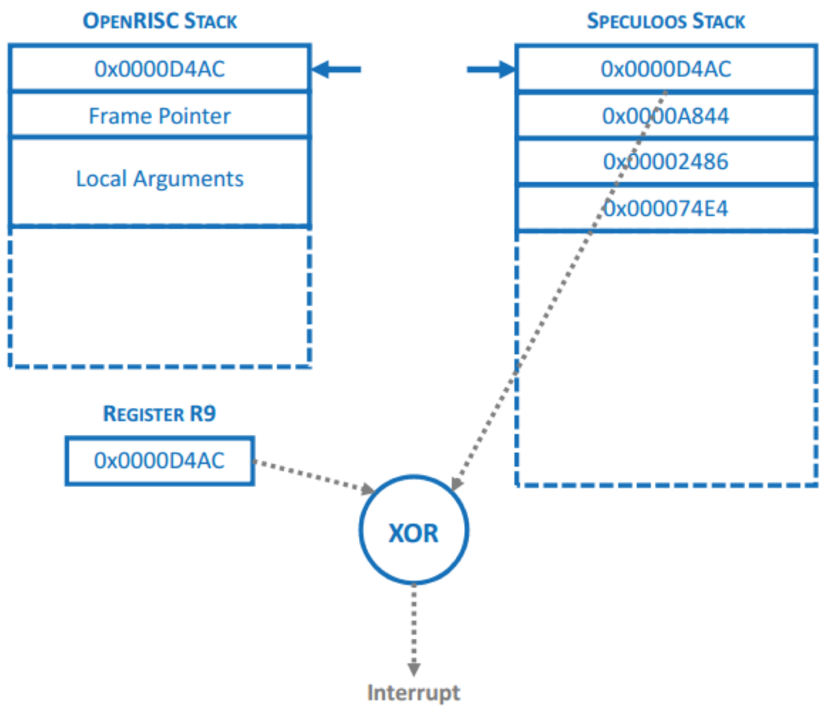
Control-flow Enforcement Technology Preview $\longrightarrow$ **Secures return address into an inaccessible shadow stack**

We decided to implement our own shadow stack (speculoos module) based on Intel CET principles

Laboratoire de Conception et d'Intégration des Systèmes

The Speculoos Stack Properties

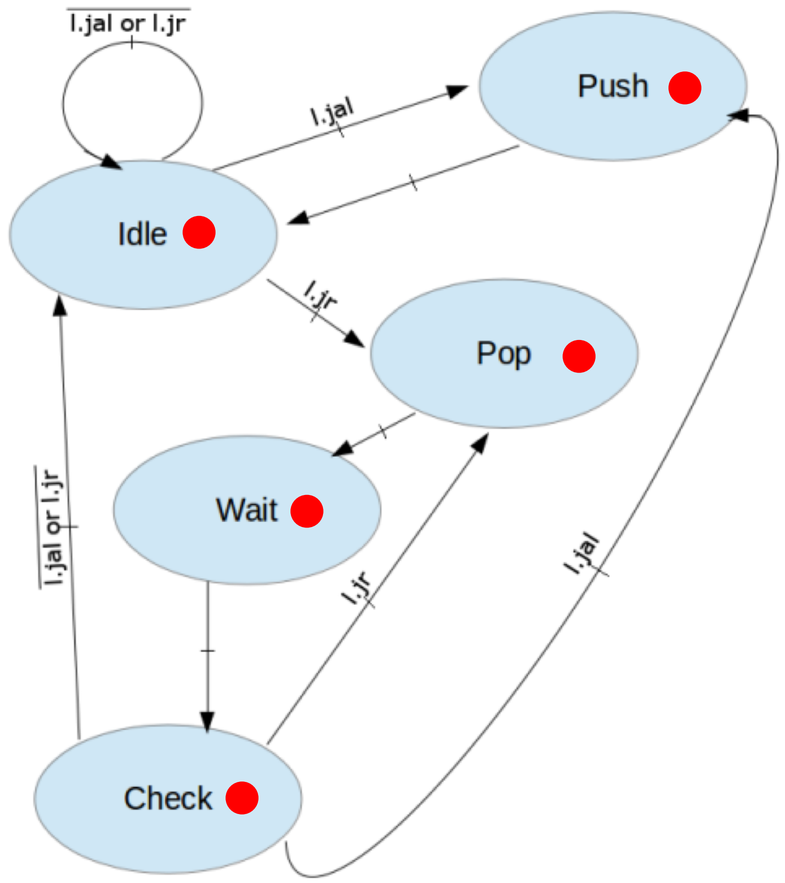
- **Speculoos stack advantages**
  - ✓ This second stack is unattainable from a software point of view, software cannot defeat it.
  - ✓ Transparency for the developer, no coding style required
  - ✓ Hardware optimized.



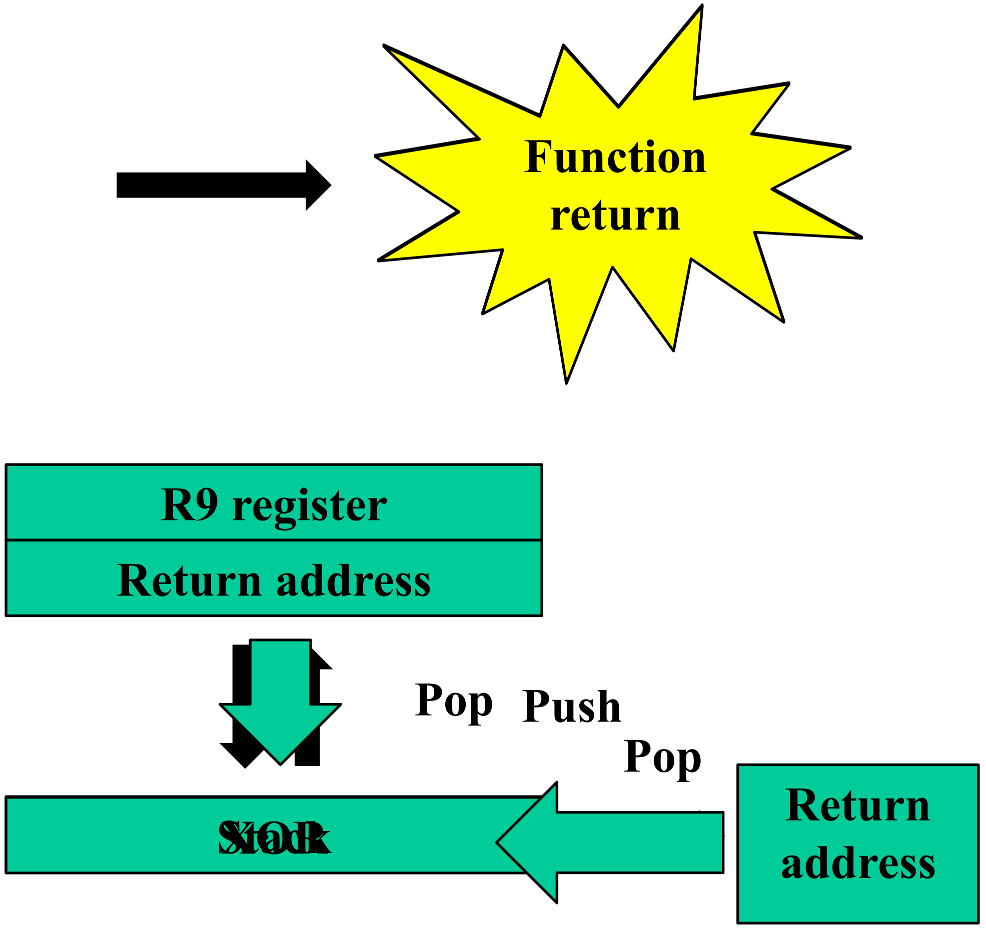
- **Speculoos stack technical challenges**
  - – Get the return address when a function is called.
    - Recognize the calling instruction.
    - Store the return address in the shadow stack.
  - – Recognize return instruction.
    - Getting the corresponding address in the shadow stack.
    - Compare the current return address with the return address that has been stored in the shadow stack. → Interruption if they are different

Shadow Stack Operations

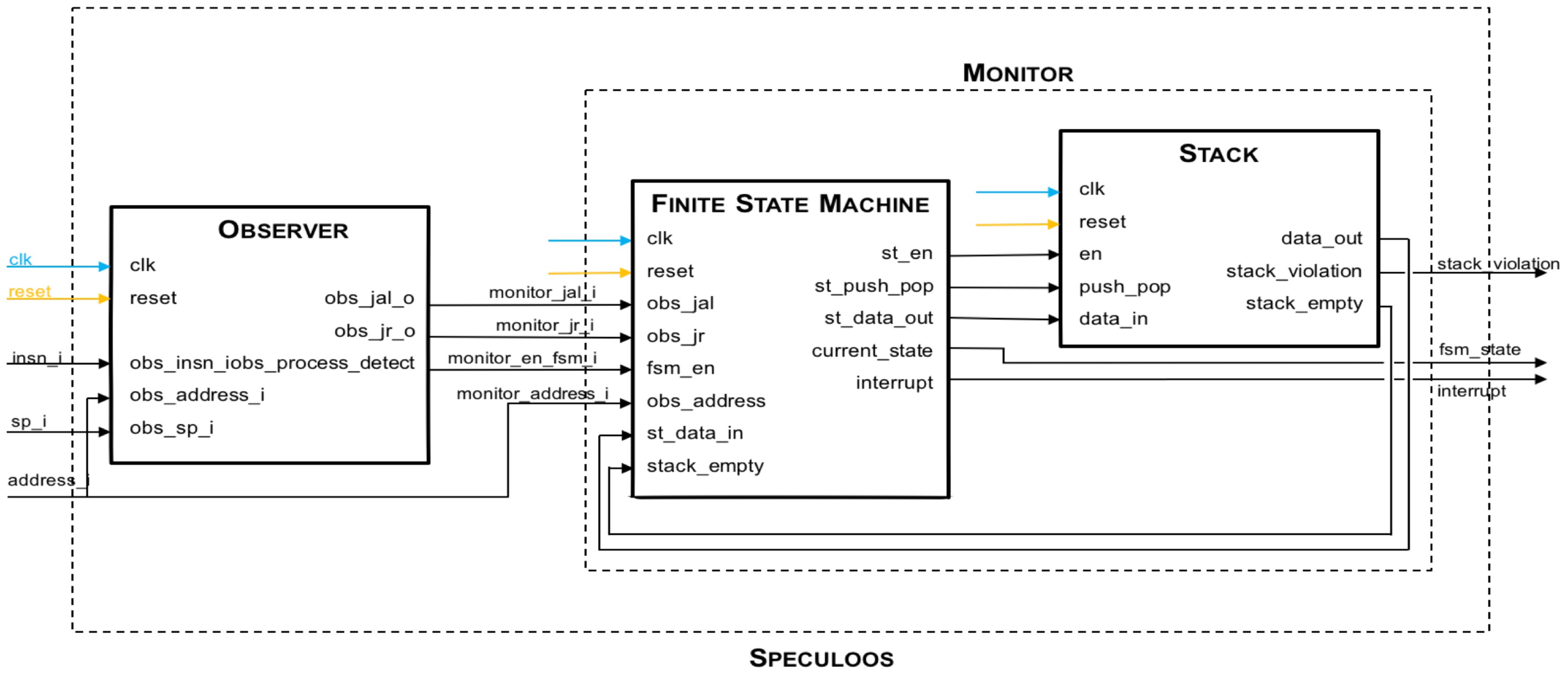
- Monitoring the functioning with a finite state machine



$\overline{\text{l.jal or l.jr}}$
Push
l.jal
Idle
l.jr
Pop
Wait
$\overline{\text{l.jal or l.jr}}$
l.jr
l.jal
Check



The Speculoos module (1)



- The Observer Block: Detects functions call and return instructions. Modify the state of the FSM.
- Stack block: Used for storing return addresses
- Finite state machine module block: Monitor the data that is pushed and poped from the shadow stack. Check if the address from the stack and the one given by the observer are the same

The Speculoos module (2)

- **Validation**
  - Speculoos has been experimentally validated on the **RISC-V based SoC**:
    - **All Function calls and returns are detected**
    - **Return adress corruptions are detected**
  - Hardware Cost is limited: 5 states FSM and few logic ressources, the speculoos stack size can be optimized according the number of cascade calls.
- **Improvements**
  - The solution is working on **baremetal** but requires fine tuning with the OS
    - Context commutation of the Linux kernel
  - A **driver** that monitors the functioning of the hardware part and the software part of the **operating system** has not been implemented yet.
  - The interruption definition is left behind the designer.

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Conclusions

- **Providing an efficient secure computing platform for medical devices.**
  - Facilitate the integration of security when developing embedded applications.
  - Make security concepts explicit and easy to use when developing applications for medical devices.
  - Security properties will be used to secure the system at the lowest cost.

- **Medical devices are ill-prepared for the security challenges of Internet connectivity**
  - Developers are not fully aware about cybersecurity issues.
  - Due to the time to market, design complications related to hardware and software security are often neglected.
  - Medical devices are placed in an environment where employees do not always have cybersecurity skills.

- **To facilitate the integration of security constraints we propose a generative approach from Design-Time to Run-Time.**
  - We propose a development toolchain in which the developer can: define at design time some part of its application with security assets, and then generate secure code adapted to the secure environment.

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

Conclusions

Two platforms **available** for security assesment of MCU based systems:

- RISC-V and STM32
- Application Use Case

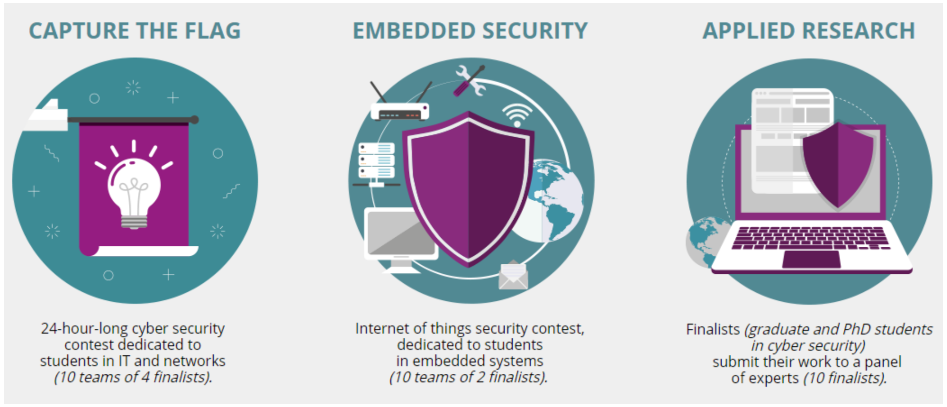
- The platform comes alongside with:
  - An open hardware Attacks platform
  - A set of software exploits
- Feel free to use and to contribute

LCIS
Laboratoire de Conception et d'Intégration des Systèmes

CSAW'18
NOV. 8, 2018 – NOV. 11, 2018
CYBER SECURITY AWARENESS WEEK
Founded by the Department of Computer Science and Engineering at NYU Tandon School of Engineering

Worlwide locations: Tunis(MNEA), New York (North America), Kanpur (India), Valence (Europe)

- registration open now: **All finalists are granted to cover travelling and housing costs**
- Main Students competitions



CAPTURE THE FLAG

24-hour-long cyber security contest dedicated to students in IT and networks *(10 teams of 4 finalists).*

EMBEDDED SECURITY

Internet of things security contest, dedicated to students in embedded systems *(10 teams of 2 finalists).*

APPLIED RESEARCH

Finalists *(graduate and PhD students in cyber security)* submit their work to a panel of experts *(10 finalists).*

- **More Info: CSAW : csaw.engineering.nyu.edu**