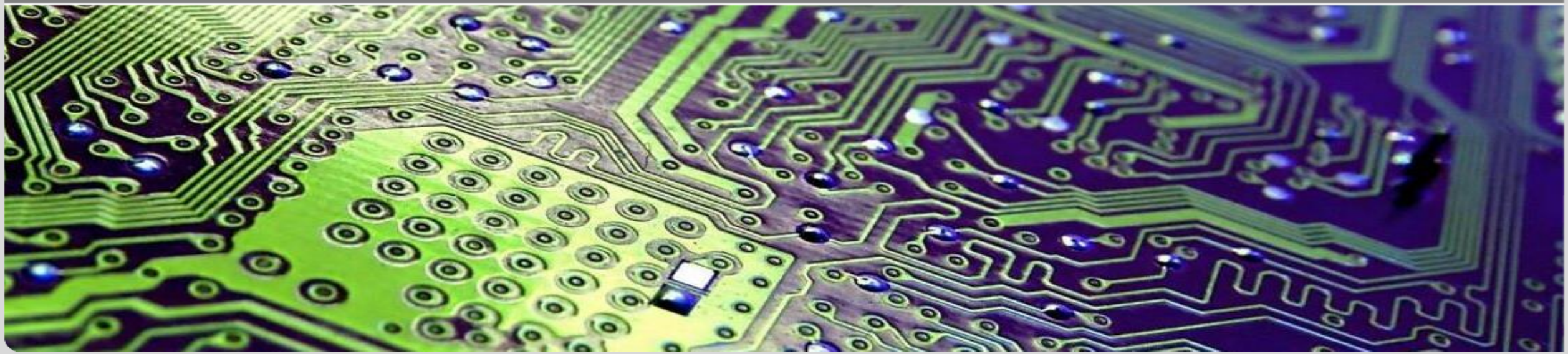
Karlsruhe Institute of Technology

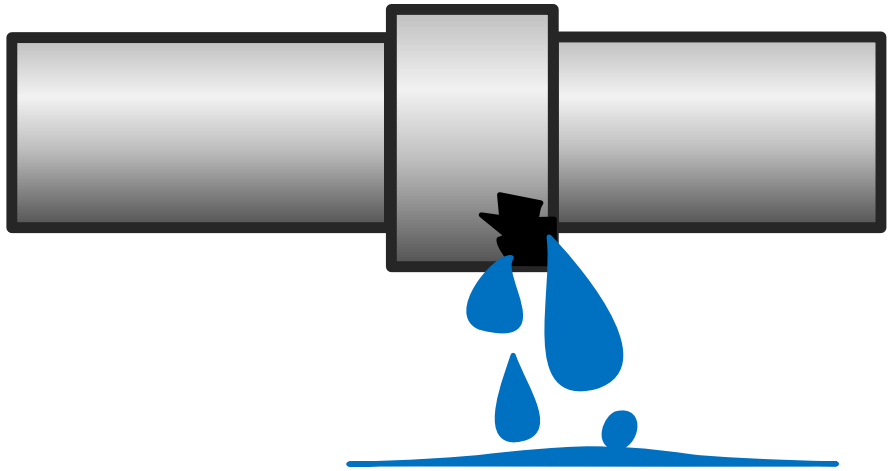
# Leaky Noise: New Side-Channel Attack Vectors in Mixed-Signal IoT Devices

Dennis R. E. Gnad, Jonas Krautter, Mehdi B. Tahoori

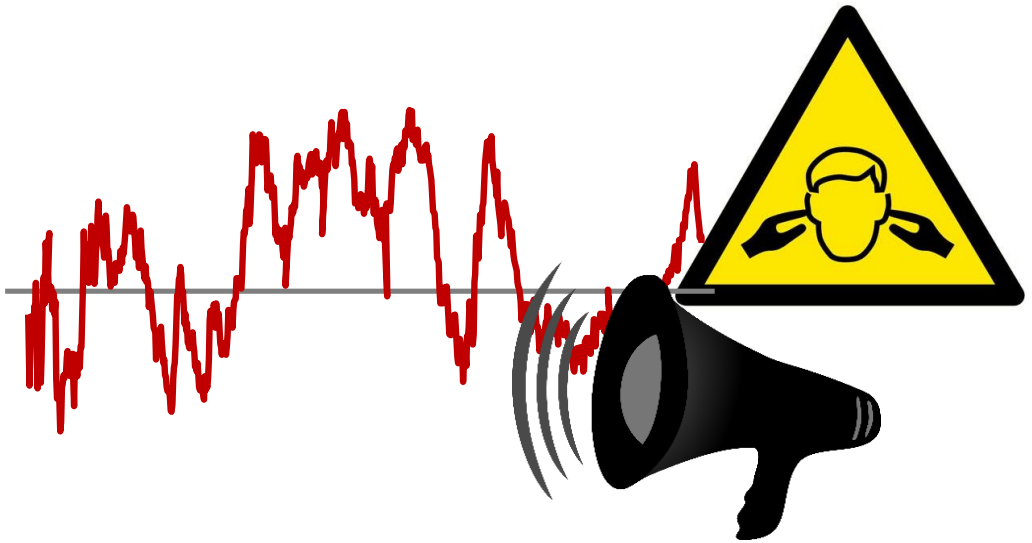
INSTITUT FÜR TECHNISCHE INFORMATIK – CHAIR OF DEPENDABLE NANO COMPUTING



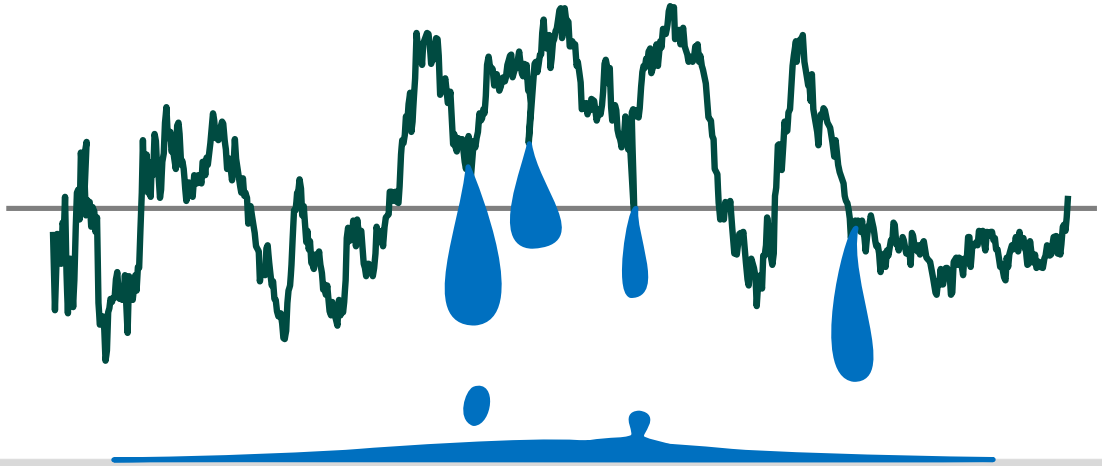
# Leaky Noise ???



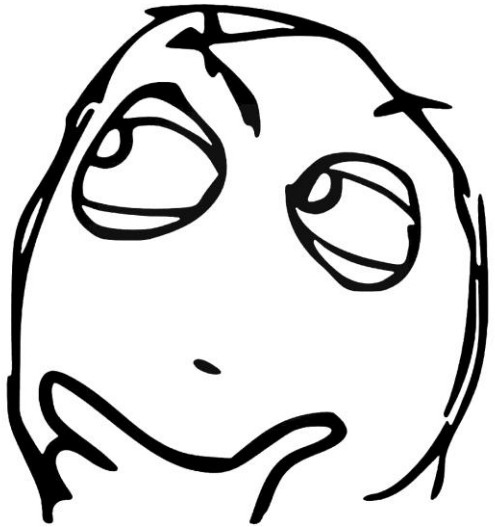
+



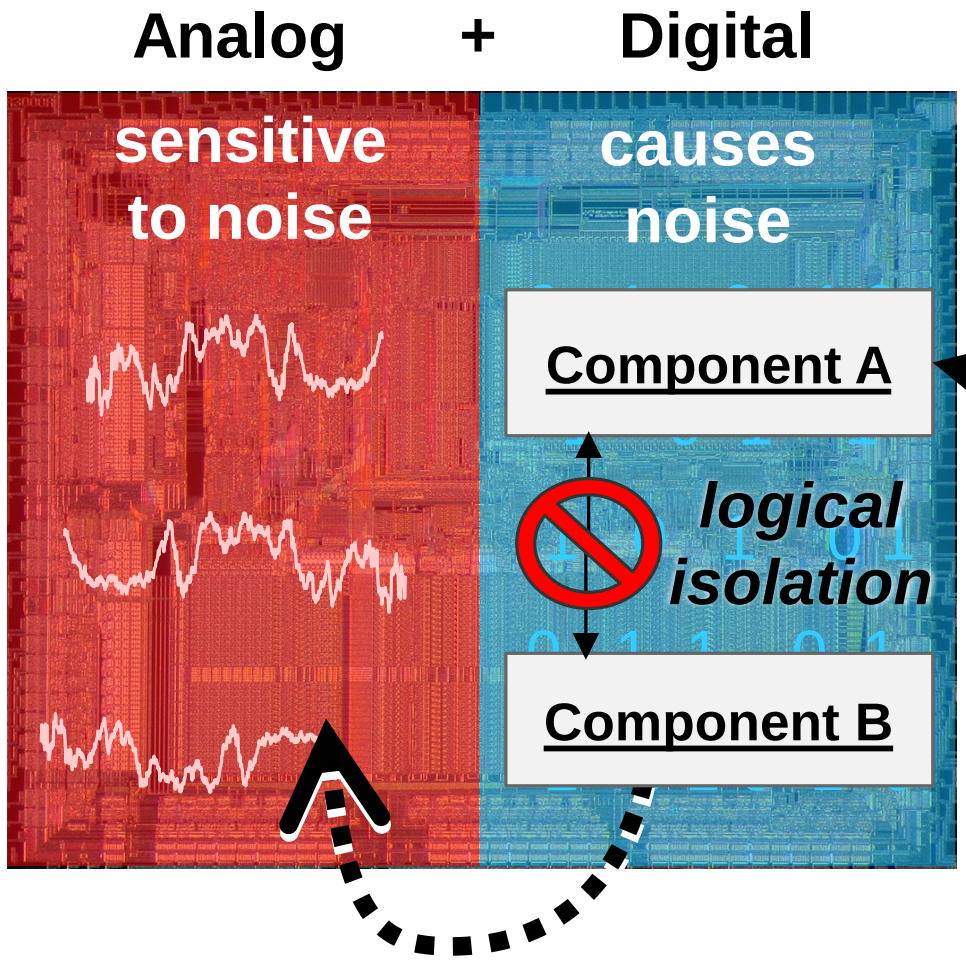
# Noise that leaks information?



*Yes, …*

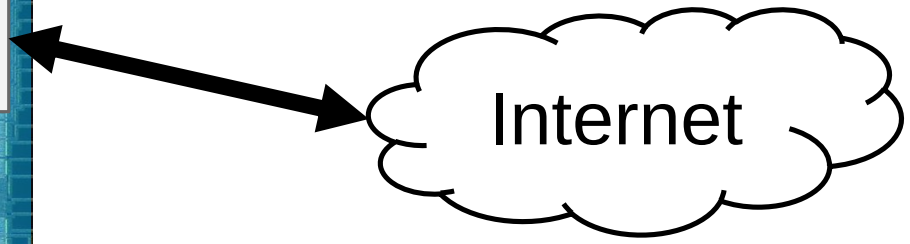


# Motivation




**Future:**

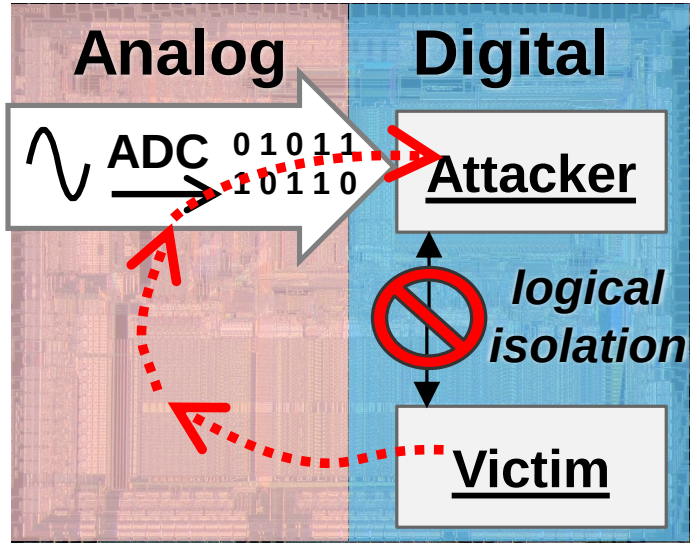
- **Everything Mixed-Signal**
- **Everything Networked / Multi-User**



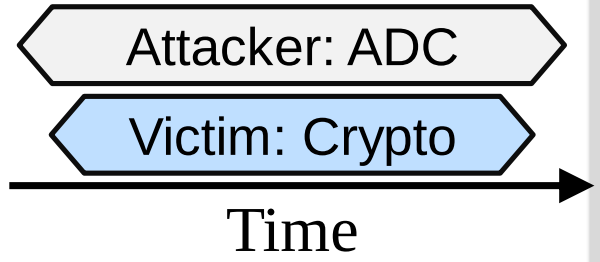

**New security threats?**

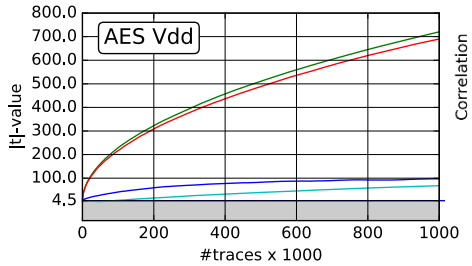
**digital circuits affect analog subsystem**

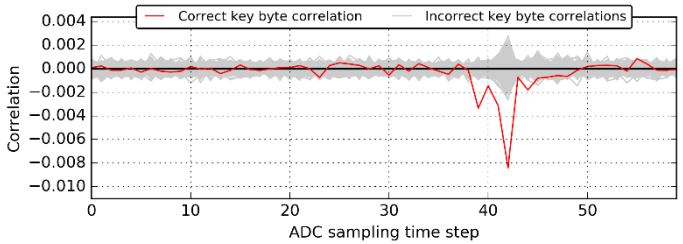
# Paper at a Glance




- Goal: Prove Information Leakage inside Chip:
  - Digital (Attacker) → Analog → Digital (Victim)
- Method:
  - Sample ADC during cryptographic algorithm
  - Leakage Assessment + Correlation Power Analysis (CPA)
- Results:
  - Most tested platforms leak
  - Successful key recovery with CPA




*ADC=Analog-to-Digital Converter*

# Outline

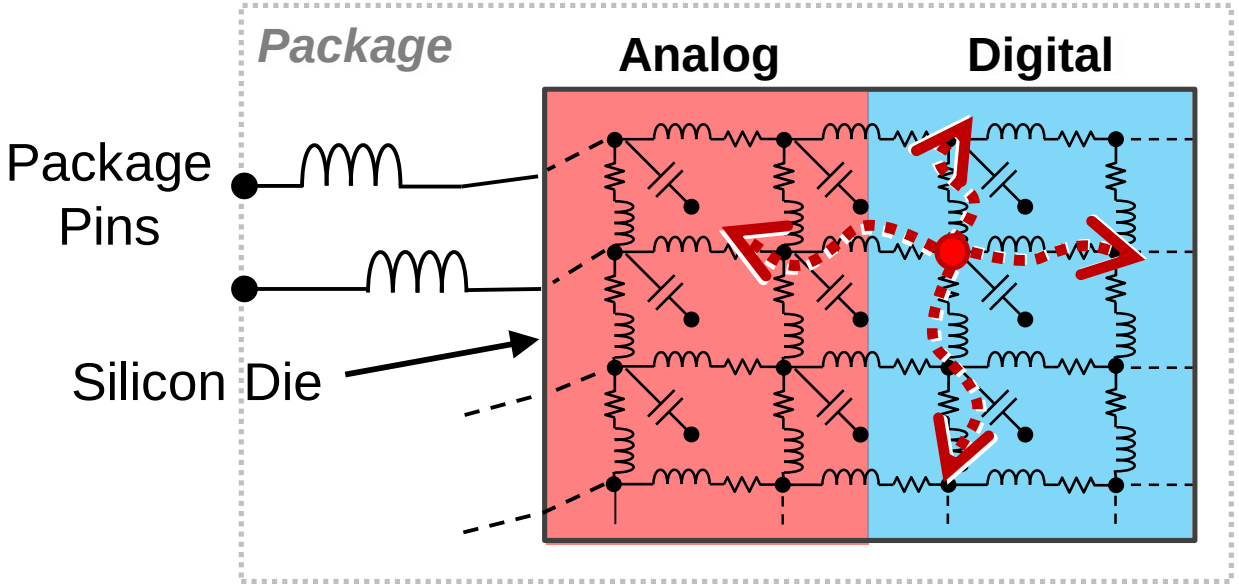
- Background & Related Work
- Experimental Setup
- Results
- Conclusion

# Outline

- Background & Related Work
- Experimental Setup
- Results
- Conclusion

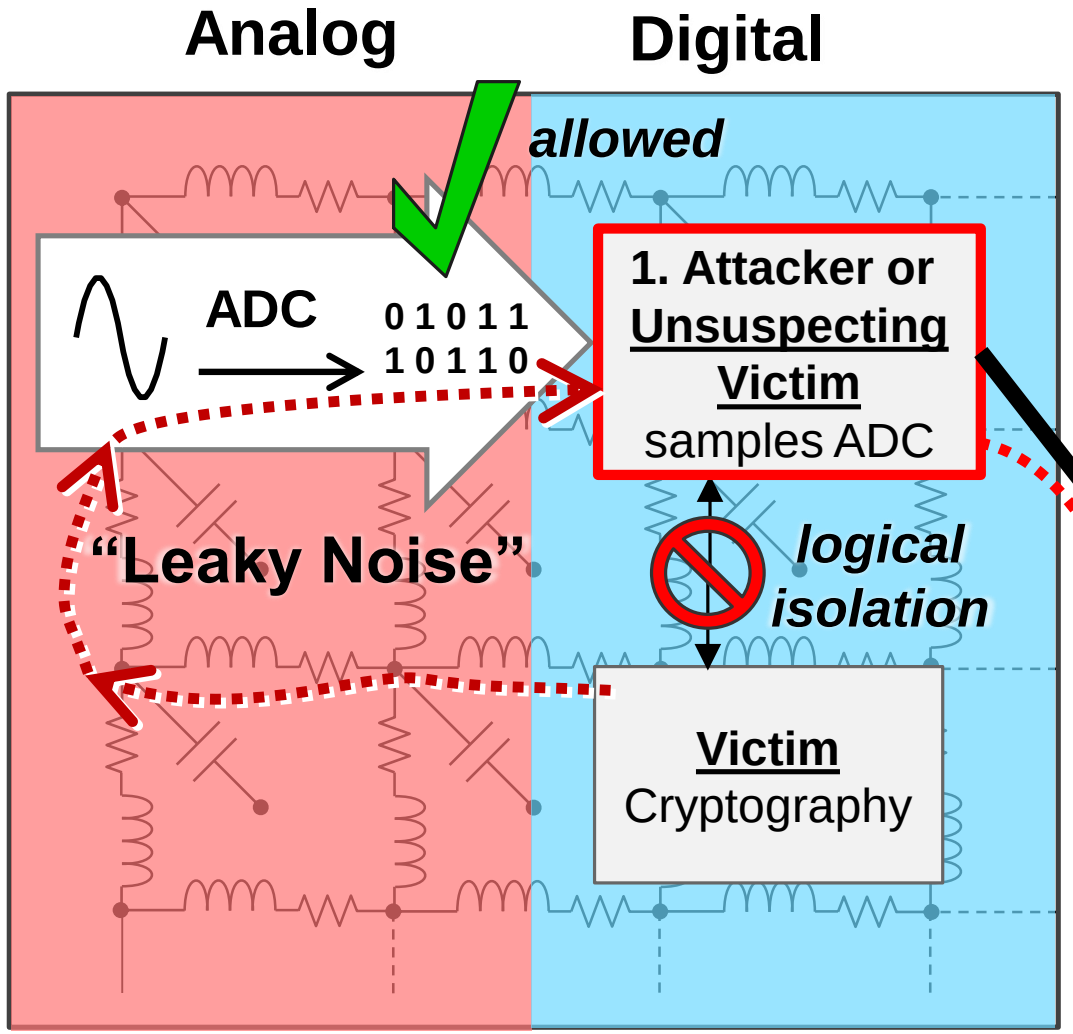
# Background: Power Distribution Networks (PDNs)

- Supplies current to all transistors in a chip
- Complex network: Resistors (R), Capacitors (C), Inductors (L)
  - Some *by design*, others unwanted = *parasitic*
- Circuit activity causes voltage fluctuations by current changes i(t)

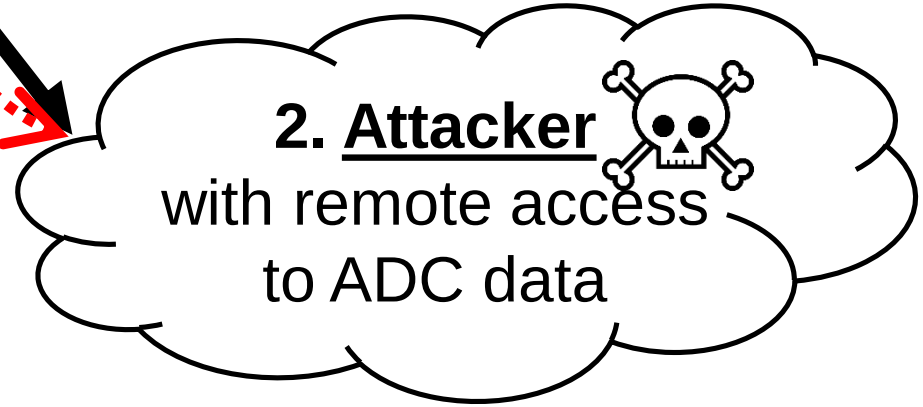



$$V_{noise} = L\frac{di(t)}{dt} + i(t)R$$

# Detailed Adversarial Model – Possible Attack Vectors




- ADC – or any sensor (e.g. Temperature)
- Logical Isolation: Memory Protection, etc.
- **Victim** leaks information into analog part
  - Affects ADC!
- **1. Attacker:** acquires leakage by ADC



# Background: Power Analysis and Leakage Assessment

- Power Analysis Side-Channel Attacks (Kocher et al. 1999)
  - Secret key recovery by analyzing power measurement traces
  - Correlation Power Analysis (CPA), Brier et al. 2004
    - Correlate power measurements with secret key-based hypothesis

- Leakage Assessment (Goodwill et al. 2011, Schneider et al. 2015)
  - Compare:
    - Set of power traces from **random** encryptions
    - Set of power traces from **fixed** (same) encryptions
  - Statistical difference indicates leakage, allow attacks

*Welch's t-test:*

$$t = \frac{\mu_{random} - \mu_{fixed}}{\sqrt{\frac{s^2_{random}}{n_{random}} + \frac{s^2_{fixed}}{n_{fixed}}}}$$

|t| > 4.5 considered sufficient

# Selected related work

- “Inside Job” (Schellenberg et al. DATE’18), extended by (Zhao et al. S&P’18)
  - CPA inside FPGA or FPGA-SoC
  - Indirect voltage measurement


- “Screaming Channels” (Camurati et al. CCS’18)
  - Mixed-Signal Chip, leak over radio, in proximity
  - Digital → Analog


- “Side-channel leakage across borders” (Schmidt et al. CARDIS’10)
  - Successful power analysis on I/O port pins of various chips

➢ Here: **Digital → Analog → Digital** possible on-chip?

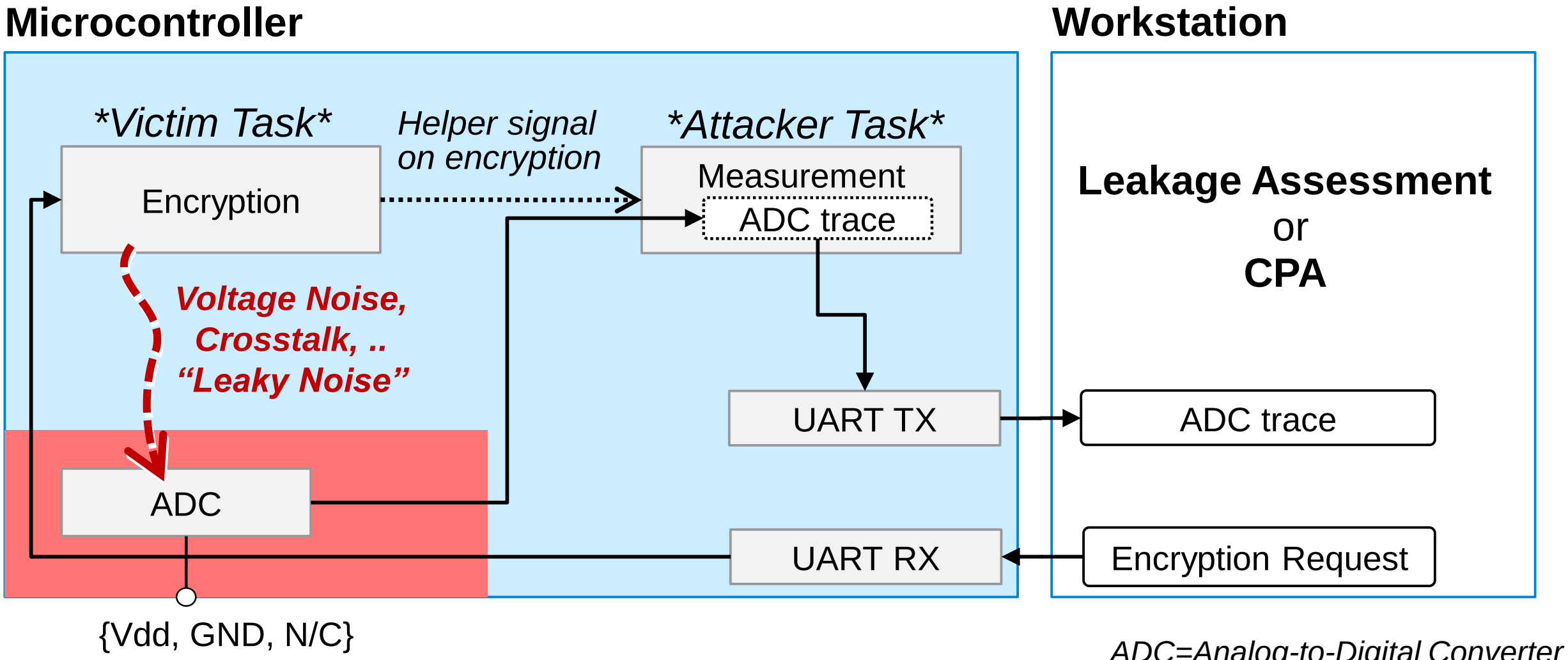
# Outline

- Background & Related Work
- Experimental Setup
- Results
- Conclusion

# Experimental Setup

- Platforms
  - Espressif ESP32
    - ESP32-devkitC – Dual-Core Xtensa CPU, Wifi, .. @ 80MHz
  - ST Microelectronics STM32
    - L4 IoT Node – Single-Core ARM CPU, Wifi On-Board, .. @ 80MHz
    - F407 Discovery – Single-Core ARM CPU, Ethernet @ 168MHz
- Software provided by both vendors:
  - mbedTLS – AES and modular exponentiation (used in RSA, ..)
  - FreeRTOS
  - GCC with standard compiler optimization “-Os”

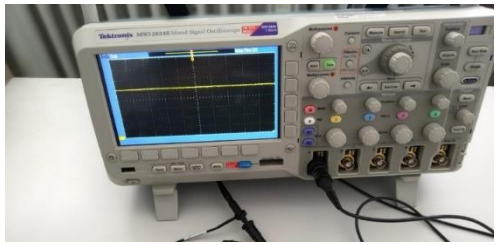
# Experimental Setup



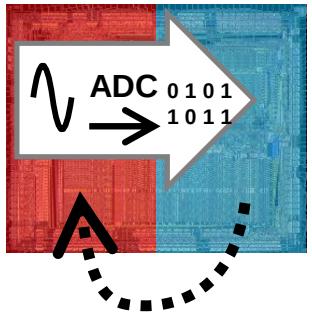
# Outline

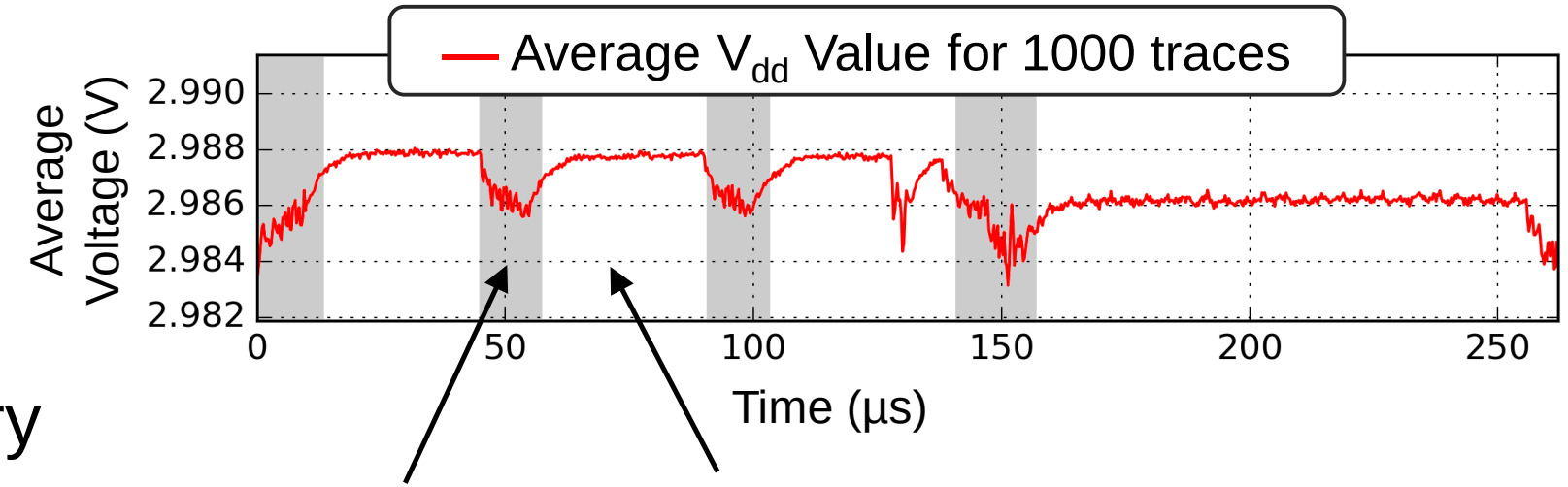
- Background & Related Work
- Experimental Setup
- Results
- Conclusion

# Basic Test: Compare ADC with Oscilloscope

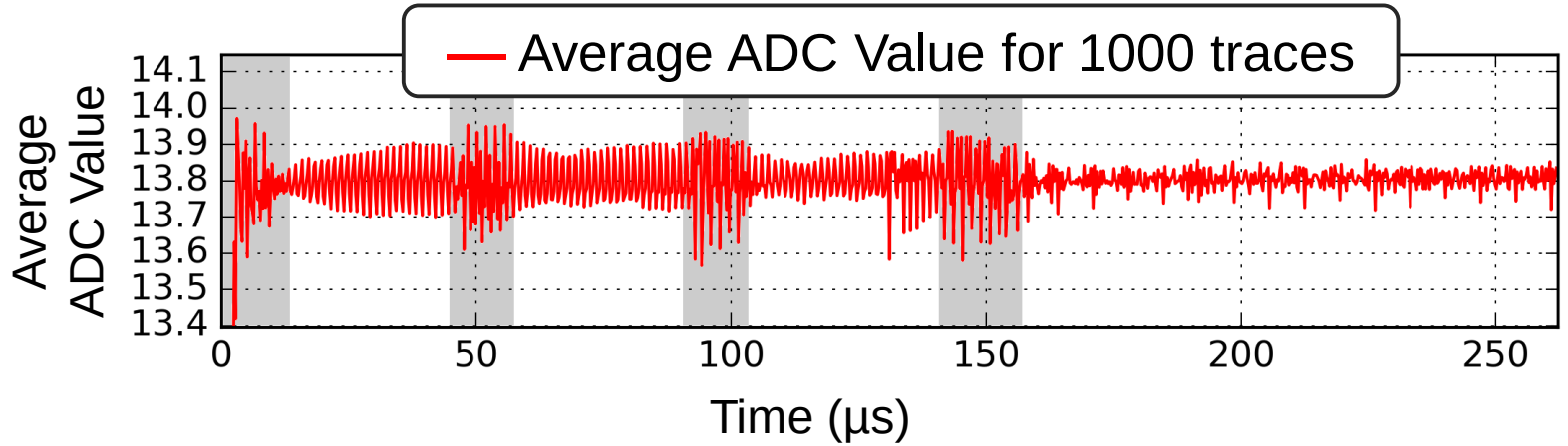


- STM32F407 Discovery
- ADC not connected (‘N/C’)
- 1,000 traces




stress phases idle phases

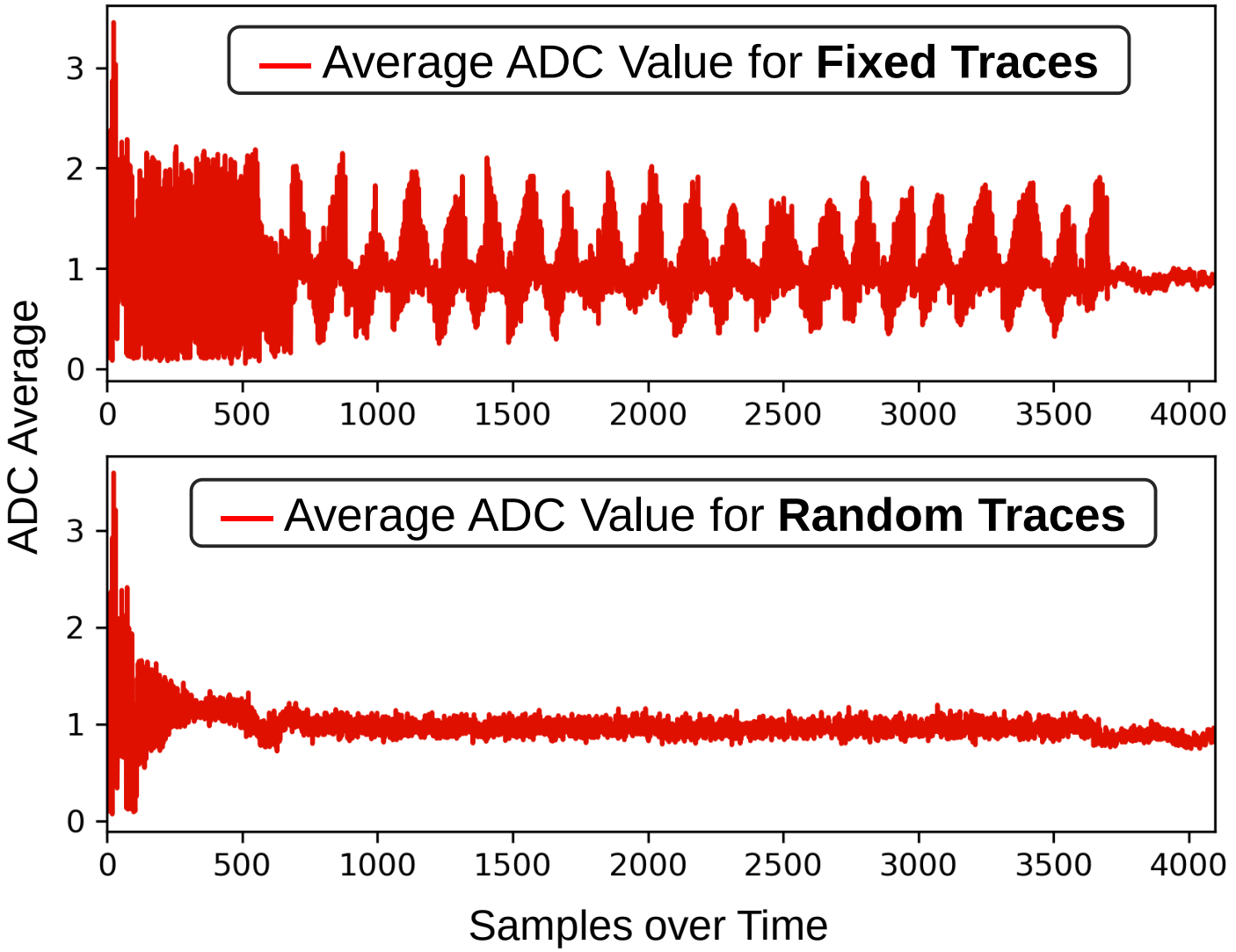


# Leakage Assessment Prerequisites

- Modular Exponentiation
- 1,000 traces averaged
- Fixed + Random Encryptions

*t-test:*

$$t = \frac{\mu_{random} - \mu_{fixed}}{\sqrt{\frac{s^2_{random}}{n_{random}} + \frac{s^2_{fixed}}{n_{fixed}}}}$$



# Leakage Assessment Results Summary

- AES: 1,000,000 traces, Modular Exponentiation: 100,000 traces
- ADC not always noisy (σ=0)
- Most cases with noise leaky, |t| >> 4.5

| Platform | Leakage detected ? | | | | | |
|---|---|---|---|---|---|---|
| | AES-128 (Fast ADC) | | | Modular Exponentiation (Slow ADC) | | |
| | Vdd | GND | N/C | Vdd | GND | N/C |
| ESP32-devkitC | **yes** | σ=0 | **yes** | no | σ=0 | no |
| STM32L4 IoT Node | **yes** | σ=0 | **yes** | **yes** | σ=0 | σ=0 |
| 2x STM32F407 Discovery | **yes** | **yes** | **yes** | **yes** | **yes** | **yes** |

# Correlation Power Analysis Attack on AES

- STM32F407 Discovery
- CPA:
  - 10 Million traces, simple alignment applied
  - Ciphertext-based
- 1. Default setup: ADC@GND, 168MHz, -Os Optimization
  - Less than 25 ADC samples for full AES
  - 2 secret key bytes recovered with high confidence
- 2. Simplified setup: ADC@Vdd, 56MHz, -O0 Optimization:
  - ~60 samples for full AES
  - 6 secret key bytes recovered with high confidence

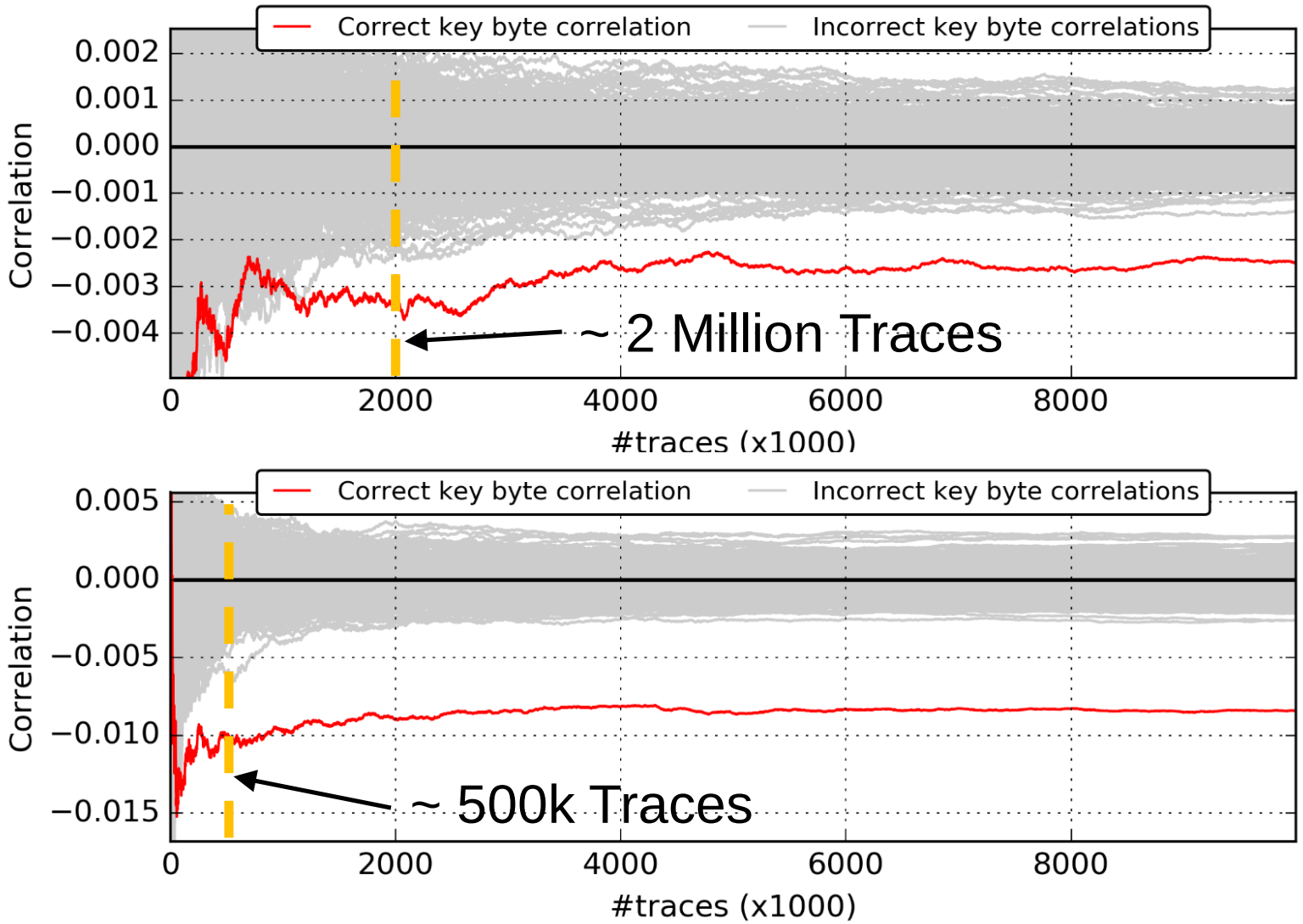
# Correlation Power Analysis results (best bytes)

- GND, -Os Optimization

"Hard"

- Vdd, -O0 Optimization

"Easy"



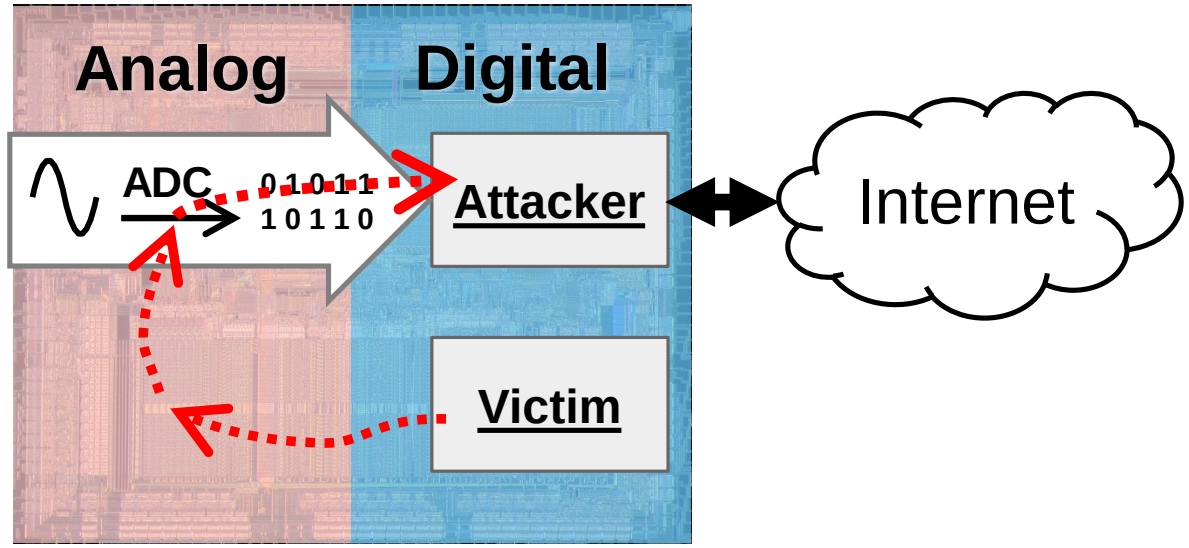
# Outline

- Background & Related Work
- Experimental Setup
- Results
- Conclusion

# “Leaky Noise” – Conclusion

**Data-dependent noise**

**Attacker can recover the data**

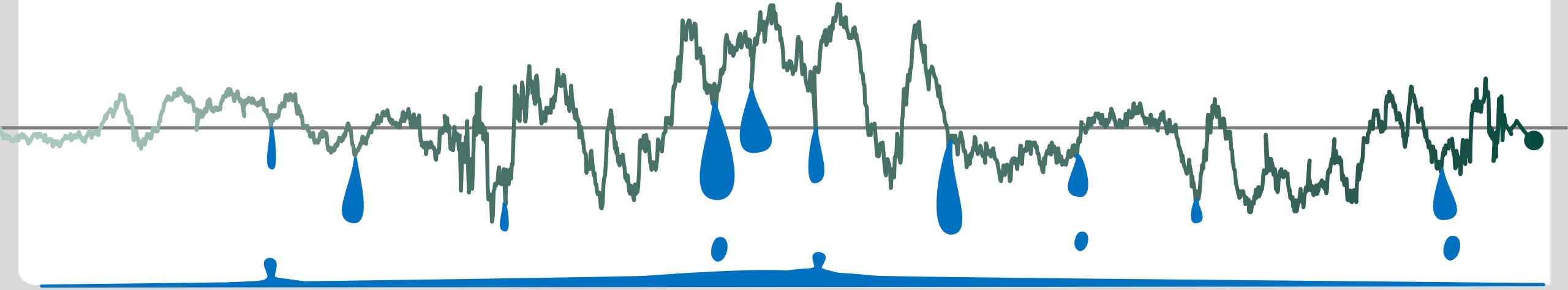



- Feasible:
  - Attacks across security domains in Mixed-Signal Chips
  - Remote power analysis attacks
- Application developers: Prevent ADC-use during cryptography
- SoC integrators: Consider digital noise a security risk
- Potentially: Always apply power analysis countermeasures (?!)

# Thanks for your Attention!

Acknowledgements: Kevin Schäfer from Rutronik & All Reviewers

Questions?



Following: Backup Slides

# Tasks Experimental Setup in FreeRTOS

- Simplified Flow:

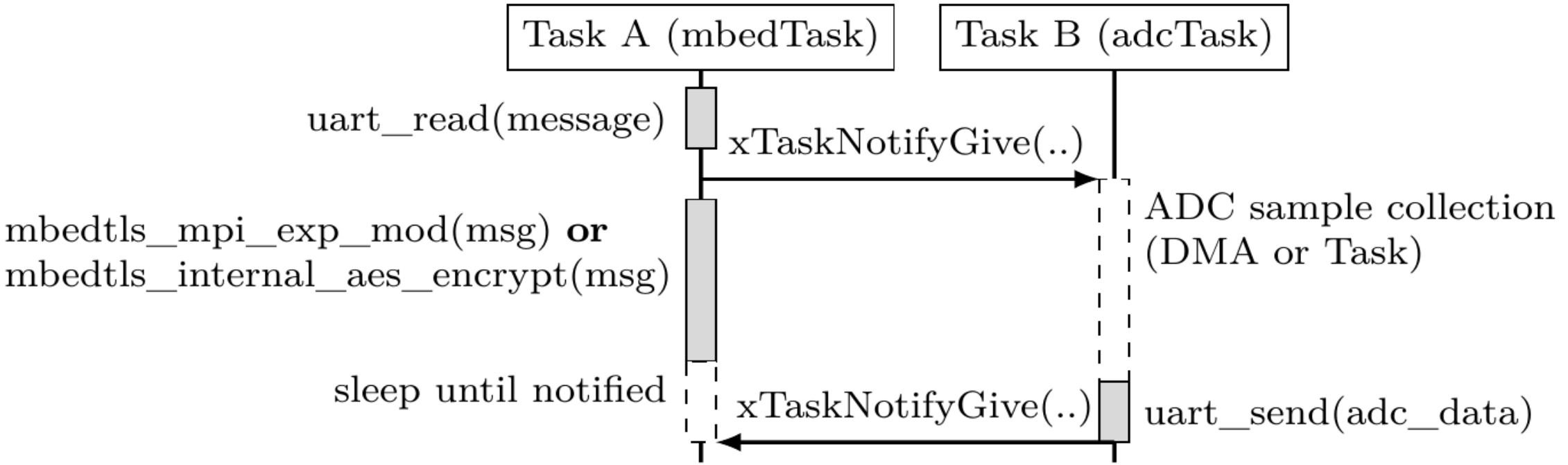



**Figure 3:** Description of one loop iteration of the two FreeRTOS tasks.

# Experimental Setup – Software Details

Table 1: Used vendor toolchain versions and respective library and compiler versions

| Platform | Framework | mbedTLS | FreeRTOS | Compiler(s) |
|---|---|---|---|---|
| Espressif ESP32-devkitC | ESP-IDF 3.1[1] | 2.12.0 | 8.2.0 Xtensa Port[2] | xtensa gcc 5.2.0[3]<br>esp32ulp 2.28.51[4] |
| ST Microelectronics STM32F407VG Discovery | STM32CubeMX[5] 4.26.1, 5.0.1 | 2.6.1 | 9.0.0 | arm gcc 7.3.1[6] |
| ST Microelectronics STM32L475 IoT Node | STM32CubeMX[5] 4.26.1 | 2.6.1[7] | 9.0.0 | arm gcc 7.3.1[5] |

[1] Espressif IoT Development Framework https://github.com/espressif/esp-idf/

[2] Espressif explains the Xtensa Port in https://docs.espressif.com/projects/esp-idf/en/v3.1/api-reference/system/freertos_additions.html, which mainly adds multicore support

[3] crosstool-ng-1.22.0-80-g6c4433a-5.2.0 as linked in https://docs.espressif.com/projects/esp-idf/en/v3.1/get-started/linux-setup.html

[4] v2.28.51-esp32ulp-20180809, as linked in https://docs.espressif.com/projects/esp-idf/en/v3.1/api-guides/ulp.html

[5] STM32CubeMX Eclipse plug in https://www.st.com/en/development-tools/stsw-stm32095.html, 4.26.1 was used for leakage assessment, 5.0.1 was used for the CPA attack in Subsection 4.5.

[6] GNU MCU Eclipse, based on arm-none-eabi-gcc 7.3.1-1.1-20180724-0637 from https://gnu-mcu-eclipse.github.io/blog/2018/07/24/arm-none-eabi-gcc-v7-3-1-1-1-released/

[7] For this platform, none was provided in CubeMX, but the version from STM32F407VG worked directly

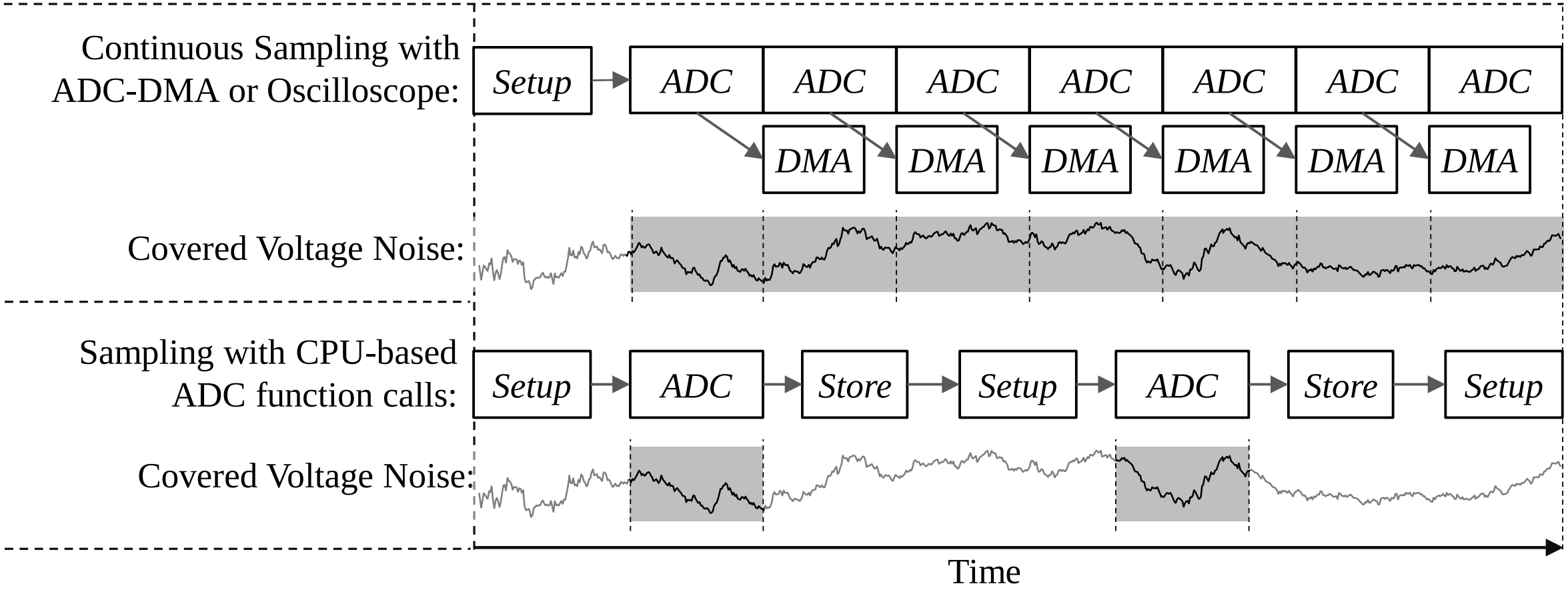
# Experimental Setup – Sampling Details

Table 1: Overview of the Experiments, repeated for ADC Pin = {Vdd, GND, N/C}

| Platform | Sampling Style | Algorithm | Samplerate / #Samples |
|---|---|---|---|
| ESP32-devkitC @80MHz | CPU | AES-128 | 104 kHz / 16 |
| | | RSA-2048 | 20.4 kHz / 2600 |
| STM32L475 IoT Node @80MHz | DMA | AES-128 | 684 kHz / 64 |
| | | RSA-2048 | 40 kHz / 4096 |
| STM32F407VG Discovery @168MHz | DMA | AES-128 | 980 kHz / 32 |
| | | RSA-2048 | 88 kHz / 4096 |

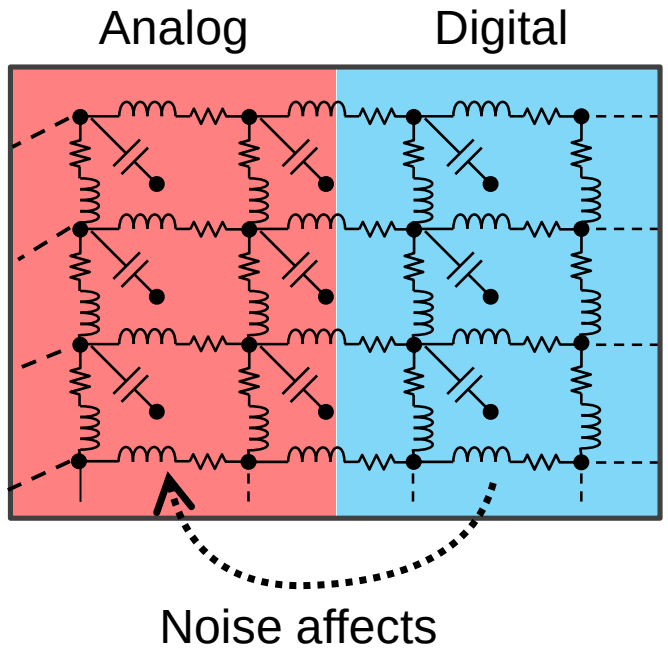
# ADC Sampling DMA/CPU

- Different ADC sampling styles covering less or more voltage noise in the ADC data. DMA needs to be used for continuous sampling, while CPU-based will always introduce gaps



# Background: Mixed-Signal and Analog

- Digital & Analog in one chip: Mixed-signal
  - Often shared PDN
  - Well-known: Digital circuits cause noise in analog part
- Analog Components integrated with Digital
  - Analog-to-Digital Converters (ADCs), DACs, …
  - Noise typically analyzed in signal processing terms
    - i.e. not considered data-correlated, security-relevant



# Leakage Assessment

- Tries to prove a statistical dependency
- Method:
  - Acquire two sets of side-channel traces:
    1. Encryption with the same fixed message
    2. Encryption with various random messages
  - Pearson's correlation between the two sets (Welch's t-test)
- Goal:
  - Show that it is possible to distinguish them using the side-channel
- If the test succeeds, we can speak of *leakage*

*t-test:*

$$t = \frac{\mu_{random} - \mu_{fixed}}{\sqrt{\frac{s^2_{random}}{n_{random}} + \frac{s^2_{fixed}}{n_{fixed}}}}$$

# Leakage

- Existing leakage shows that an attack probably exists
- No information on:
  - Easiness/hardness of an attack
  - How the attack can be done (used intermediate values, ..)

## Order of Leakage

- Higher-order statistical moments can be used
- Sometimes only leakage in a higher order can be assessed

# Formulas Power Analysis and Leakage Assessment

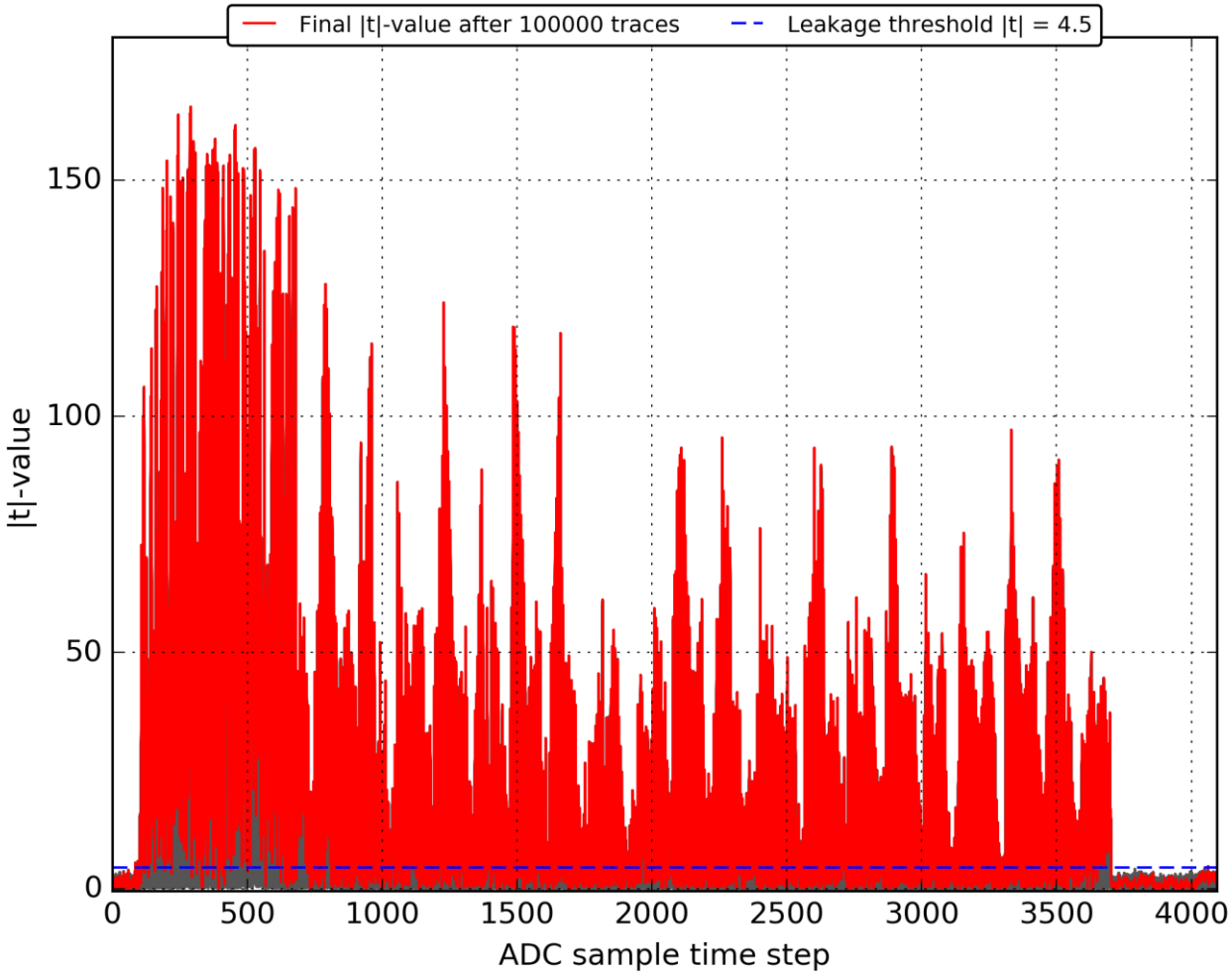
$$P_{hyp} = HW(SBox^{j}(K_{hyp} \oplus S_i))$$

$$t = \frac{\mu_r - \mu_{fixed}}{\sqrt{\frac{s_r^2}{n_r} + \frac{s_{fixed}^2}{n_{fixed}}}}$$

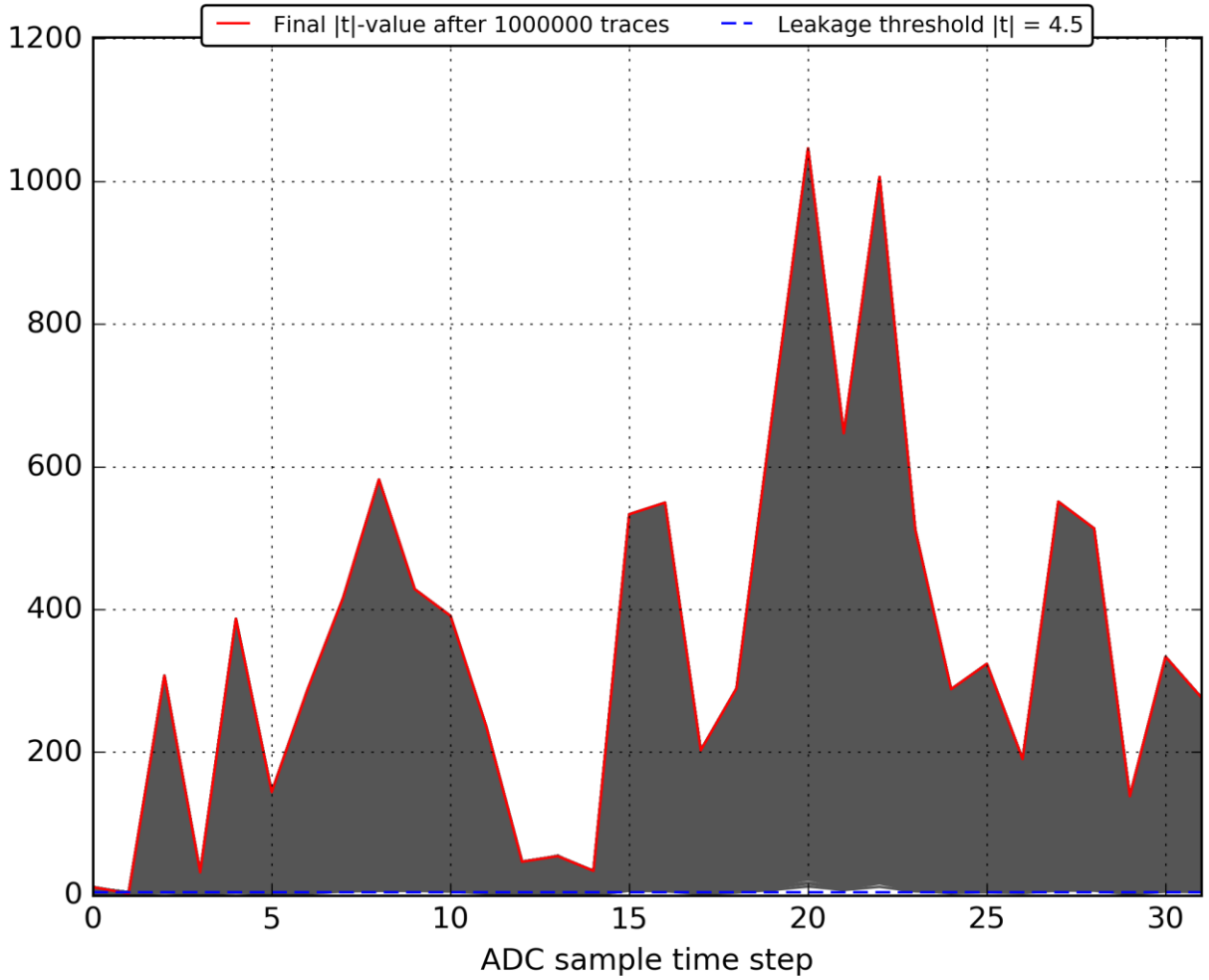
# **Leakage Assessment Trace** (ADC on GND, STM32F407)

Karlsruhe Institute of Technology

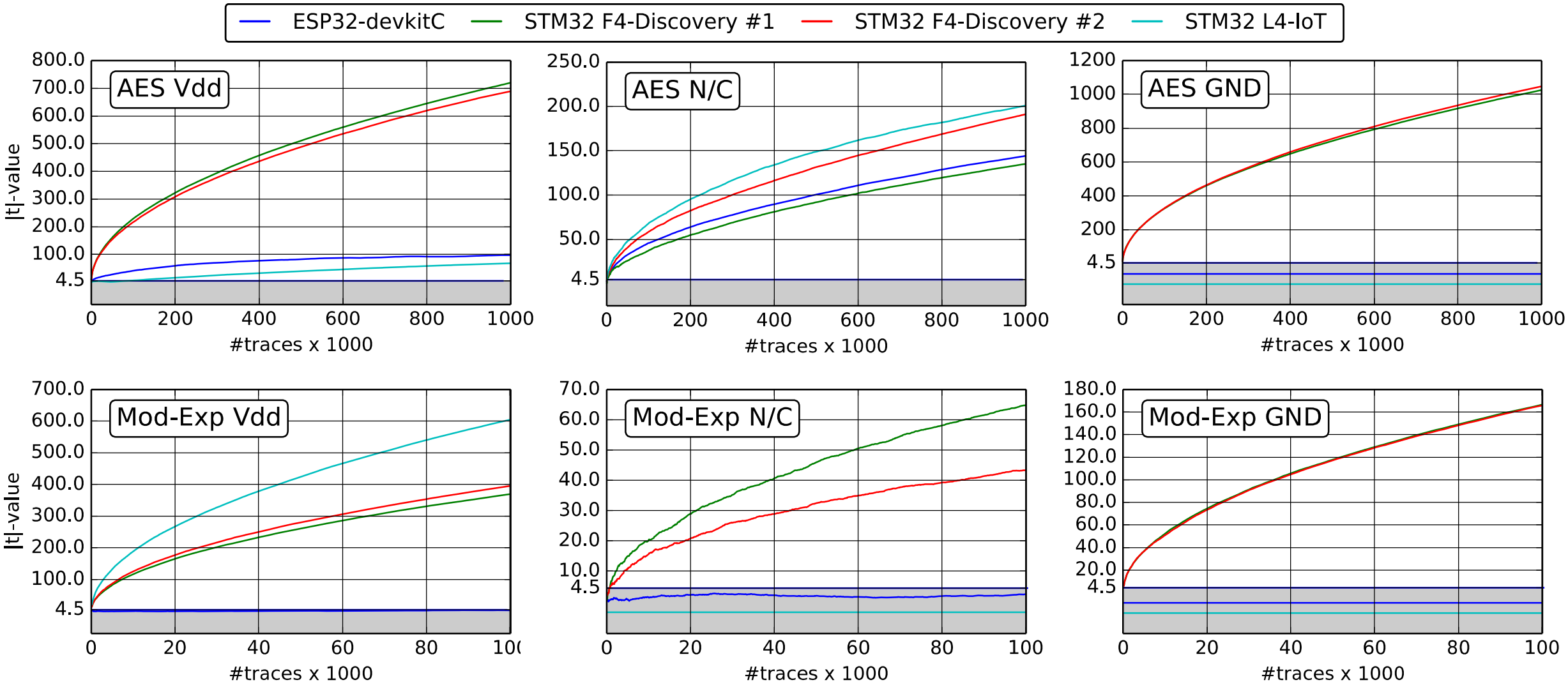
## Modular Exp.




## AES

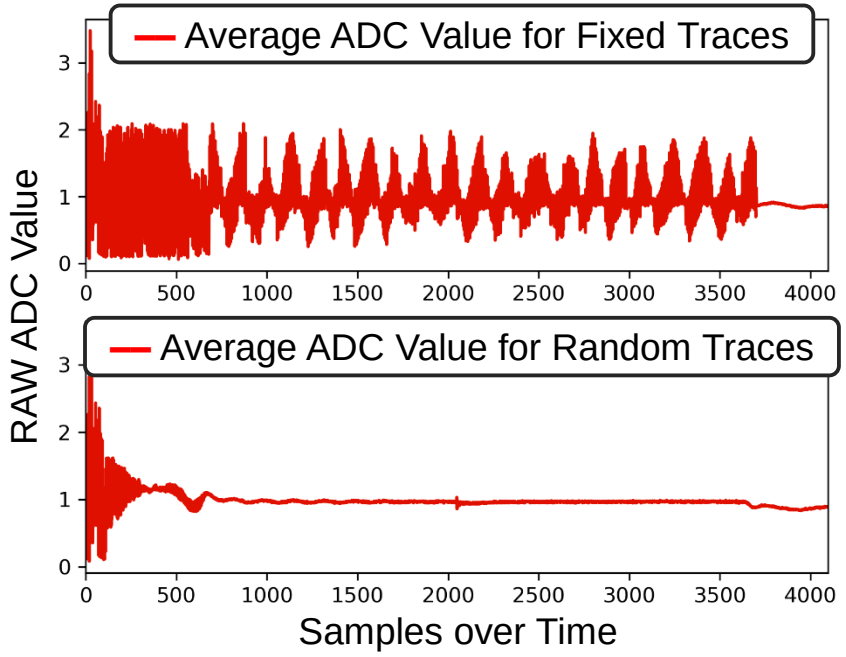


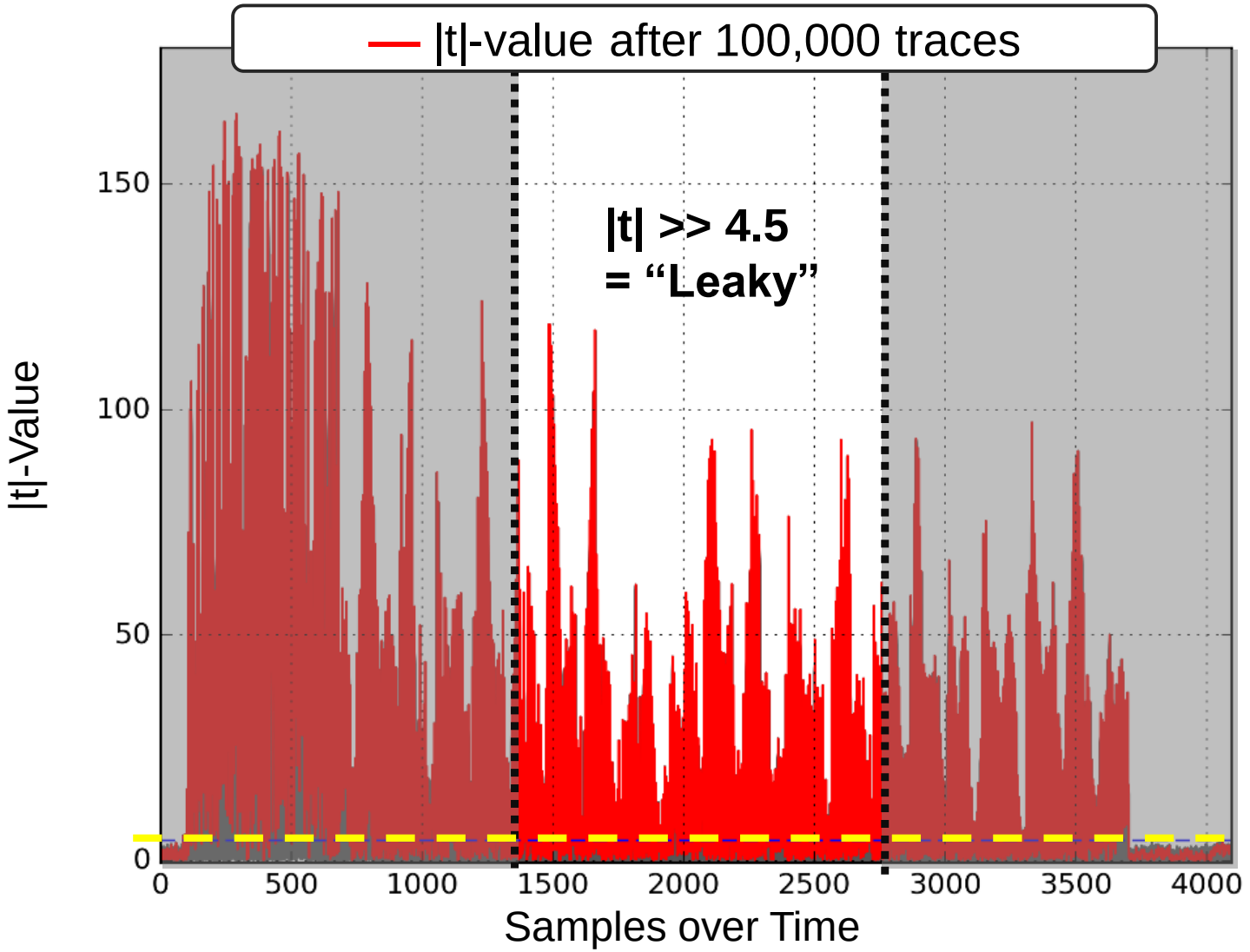
# All Leakage Assessment Results



# Leakage Assessment Example

- Modular Exponentiation
- STM32F407 Discovery
- ADC connected to GND
- 1,000 →**100,000 Traces**

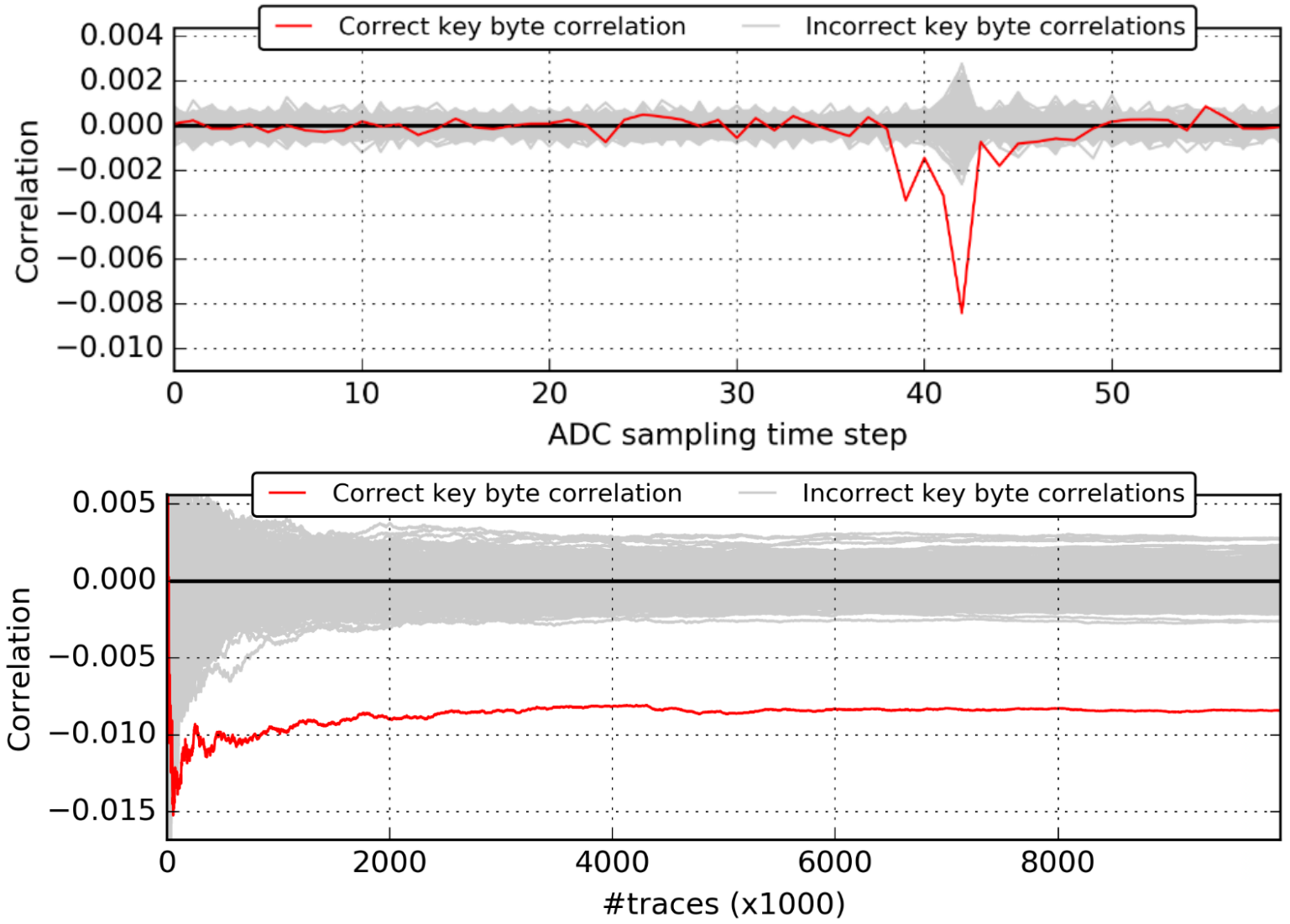


# Correlation Power Analysis (best byte)

- Vdd, -O0 Optimization

"Easy"



# Correlation Power Analysis (best byte)

- GND, -Os Optimization

"Hard"