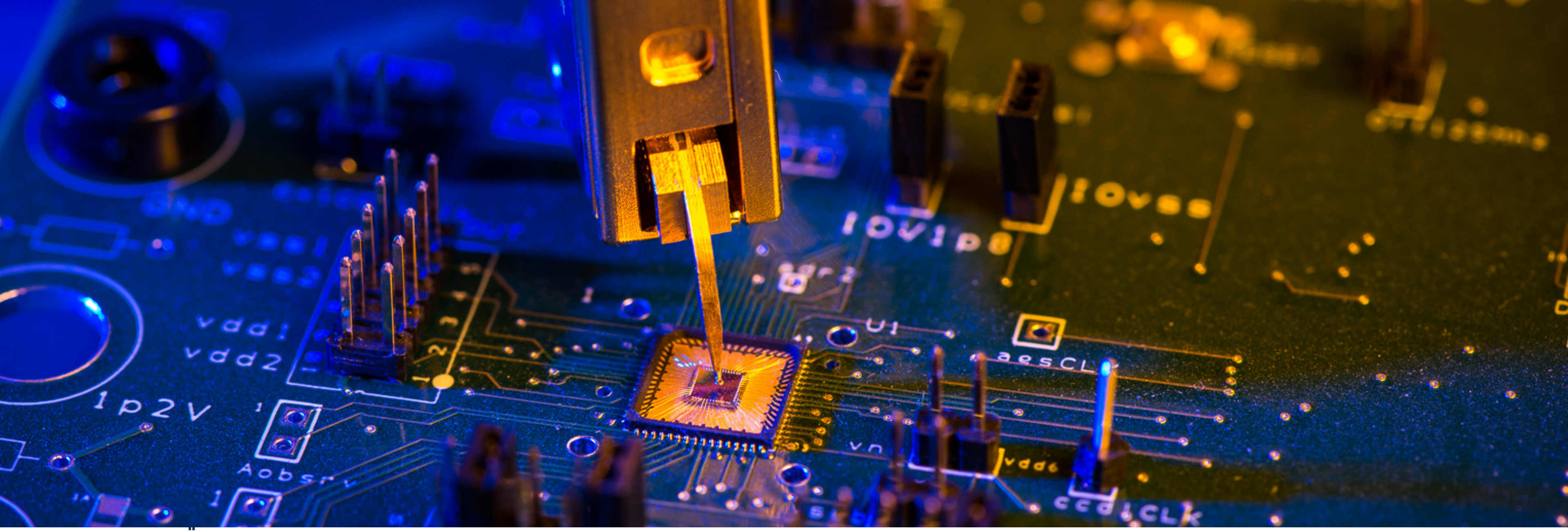


RUHR-UNIVERSITÄT BOCHUM

Hardware Security – Challenges and Directions

Selected Areas in Cryptography (SAC 2023)

Prof. Dr.-Ing. Tim Güneysu

15. Januar 2026

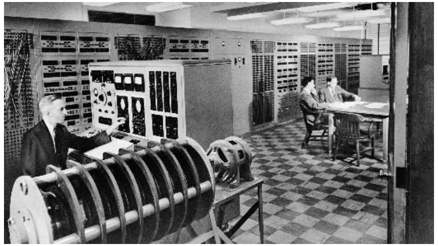
Chair for Security Engineering
Faculty of Computer Science
Ruhr University Bochum

SECURITY
ENGINEERING

Digital Evolution and Security

Granularity

>1950



>1985

>2020

Implementation

Hardware

Software

Hardware

Software

Hardware

Security Demand

How to partition security into hardware & software?
How much security can we achieve in hardware?

Physical Exposure
Privacy Requirements
Networking & Connectivity

Why a Root of Trust in Hardware is Essential

Hardware implementations are more efficient

- Boosts computationally exensive cryptography
- Less energy/area costs for constrained applicaitons

Protections in hardware are more powerful

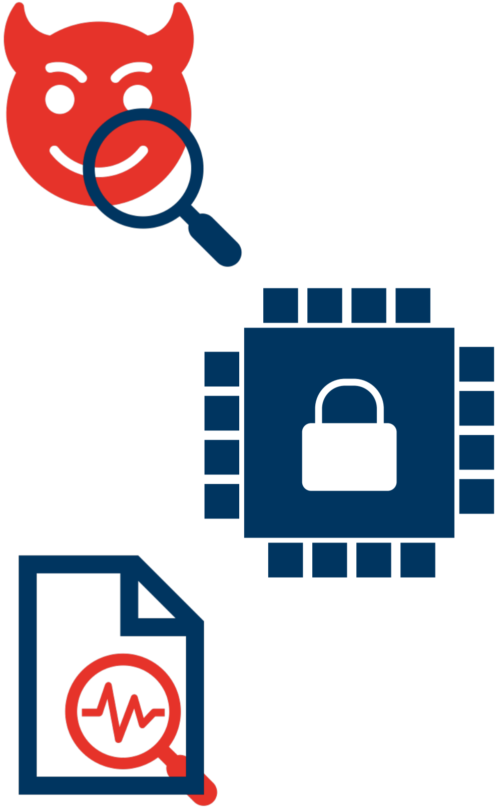
- Opening up a hardware chip is assumed with higher cost compared to a disassembling a binary
- Some advanced countermeasures are only for hardware

Security-relevant components are hardware-only

- True Random Number Generators

0110001010001

Hardware Root of Trust – Directions and Challenges



Challenges

#1 Knowing the Adversary

#2 Preserving your Secrets

#3 Models and Limitations in Implementation Security

Directions

#1 Moving Target Defense in Hardware

#2 New Hardware Constraints for Cryptography

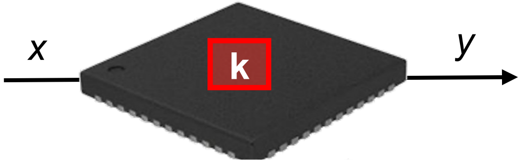
#3 Trusted and Security-oriented Hardware Design

CHALLENGES

Knowing the Adversary

CHALLENGE #1: Knowing the Adversary – Models and Boxes

Black-Box Attacker



- Observing & changing I/O data
- Observing traffic patterns

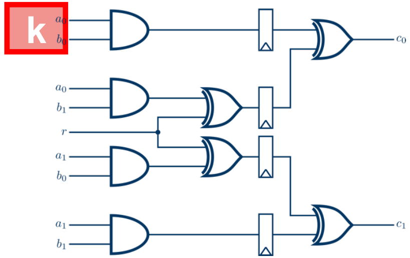
- Observing timing behavior

Gray-Box Attacker

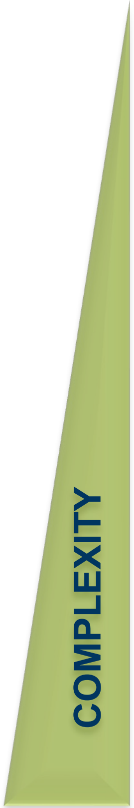
- Faulting security operations

- Observing side-channel information

White-Box Attacker



- Analyzing gate-level components for vulnerabilities
- Probing & modifying circuitry



CHALLENGE #1: Knowing the Adversary – Attacks in SW and HW

…ogical Run-Time Attacks

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

Dennis R. E. Gnad, J… …Mehdi B. Tahoori
Karlsruhe Inst…
{dennis.gnad,jo…

From the bitstream to the netlist

…-Baptiste Note
…ment d'informati…

Éric Rannaud
Département d'informatique
École Normale Supérieure
45 …ue d'Ulm, 75005 Paris
rannaud@ens.fr

Key Extraction Using Thermal Laser Stimulation

A Case Study on Xilinx Ultrascale FPGAs

Heiko Lohrke*,1, Shahin Tajik*,3,†, Thilo Krachenfels2,
Christian Boit1, and Jean-Pierre S…

1Semiconductor…

On the Power of Op…
Attacking Bit…

Shahin Tajik*,1, Heiko…
1Security in Tele…
Technis…
{stajik,jpse…
lohrke@mailbox.tu-b… …rlin.de
…stian.boit@tu-berlin.de
* These autho… contributed equally to this work

Thrangrycat flaw lets attackers plant persistent backdoors on Cisco gear

Most Cisco gear is believed to be impacted. No attacks detected, as of yet.

…m Encryption
…x II

…acilitating Black-Box Analysis using Software Reverse-Engineering

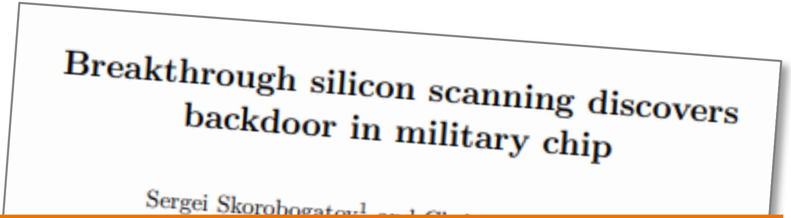
Amir Moradi, David Oswald, Christof Paar, Pawel Swierczynski
Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany
firstname.lastname@rub.de

How much protectio…

- Supply chain

CHALLENGE #1: Knowing the Adversary – Model Mismatches

Even more prominent examples:



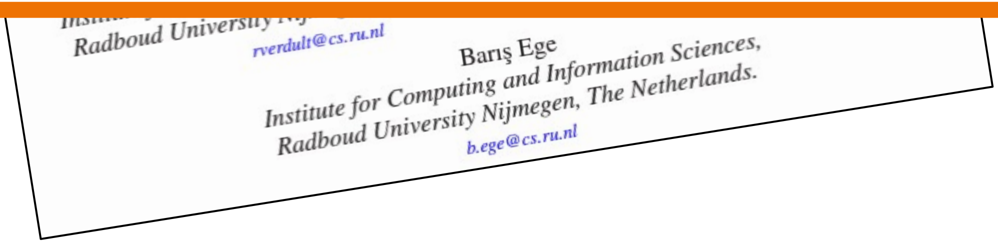
Breakthrough silicon scanning discovers backdoor in military chip

Sergei Skorobogatov

Real attackers do no care about adversary models or laws

(Neither the public when your system is published to be broken)

The selection of choosing a minimal but holistic set of countermeasures is an underresearched topic



Radboud University

rverdult@cs.ru.nl

Barış Ege

Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands.

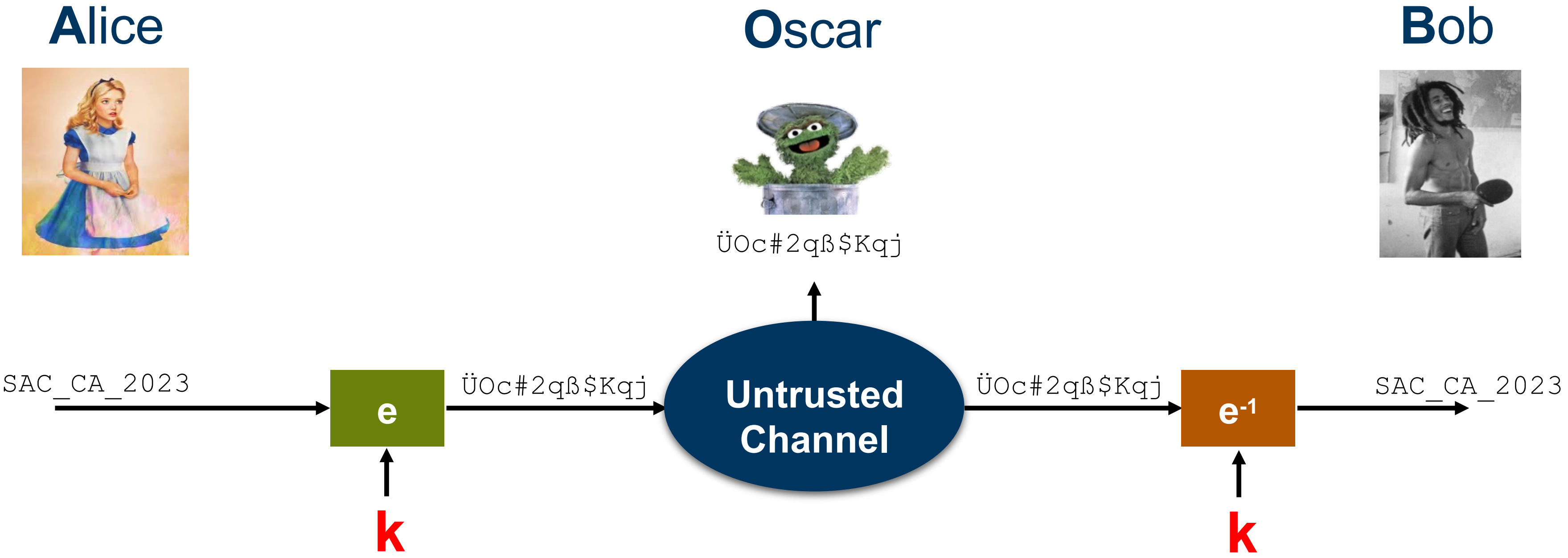
b.ege@cs.ru.nl

Megamos

CHALLENGES

Preserve Your Secrets

CHALLENGE #2: Preserving your Secrets



Alice

Oscar

Bob

ÜOc#2qß$Kqj

SAC_CA_2023

e

ÜOc#2qß$Kqj

Untrusted Channel

ÜOc#2qß$Kqj

e^{-1}

SAC_CA_2023

k

k

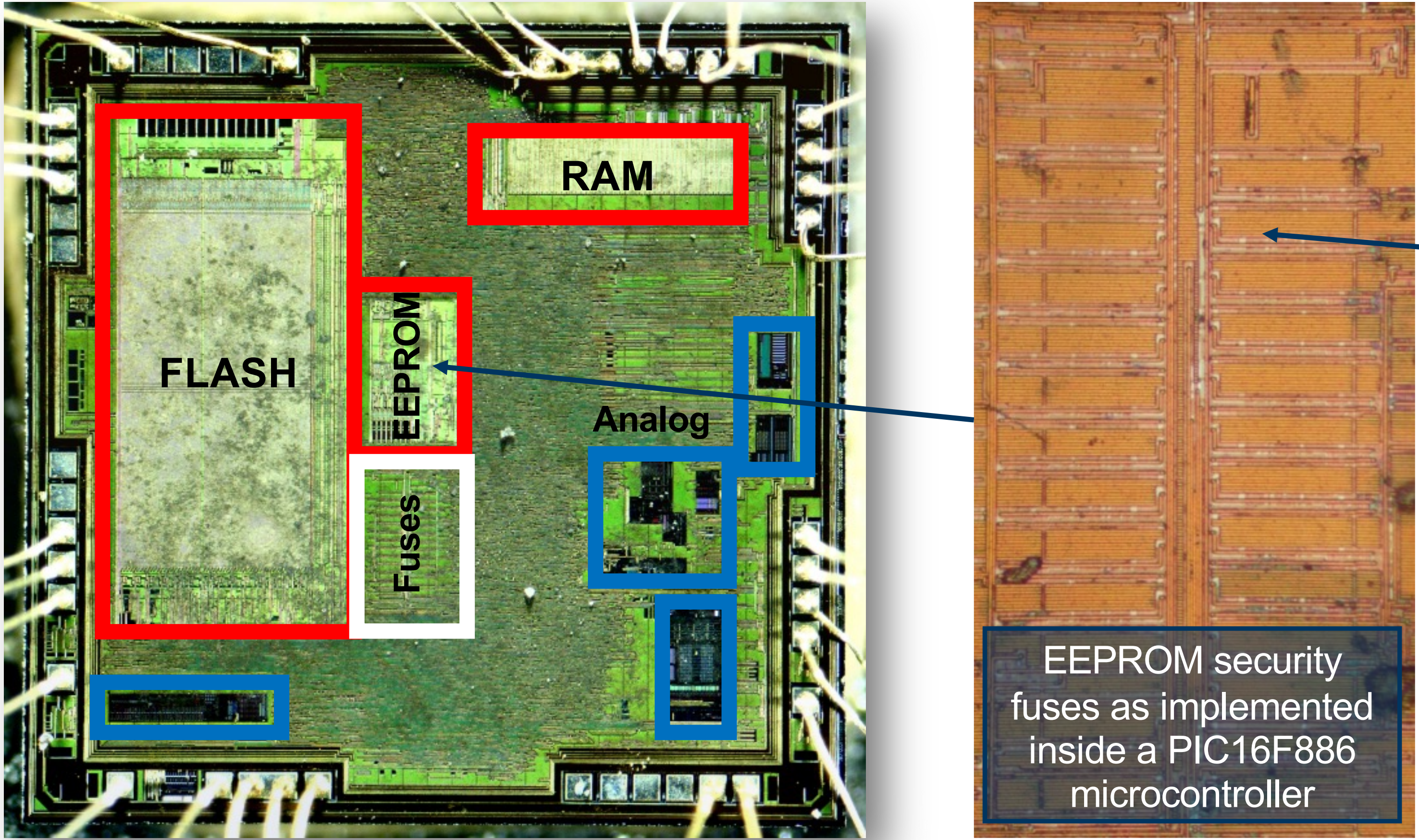
Well-known Kerckhoff's Principle: only key **k** must be kept secret
→ Cryptographer: „Hmm, that's basically **not** my problem!"
→ This needs to be solved by the **implementation**…

CHALLENGE #2: Preserving your Secrets – SW vs. HW

- **Options in Software**
  - Where to put it? In the binary?
  - Who then protects the binary? Using obfuscation?
  - White-Box Cryptography (Fully broken, cf. DES, AES, DCA)
- **Options in Hardware**
  - Secure distributed memories (ROM, Fuse, EEPROM, Flash, battery-backed BRAM)
  - Physically Unclonable Functions (PUF)

Common Approach: Put some trusted components with key storage in hardware

CHALLENGE #2: Preserving your Secrets – Hacking the Trusted HW



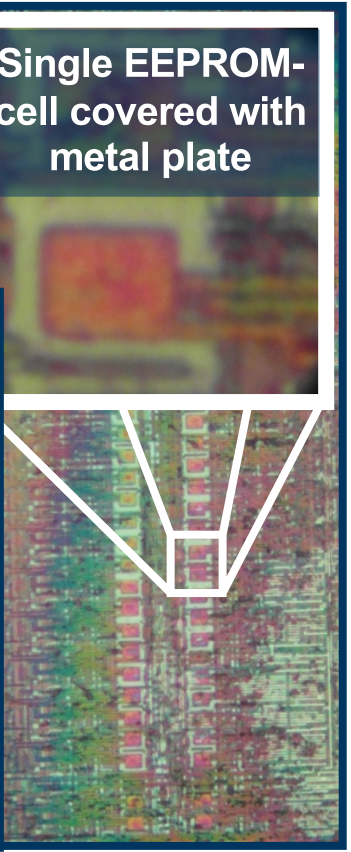
EEPROM security fuses as implemented inside a PIC16F886 microcontroller

EEPROM provides a **security bit** to enable readout protection of Flash memory:

0 = none
1= protected

CHALLENGE #2: Preserving your Secrets – Hacking the Trusted HW

UV-light Erasing Device



Single EEPROM-cell covered with metal plate

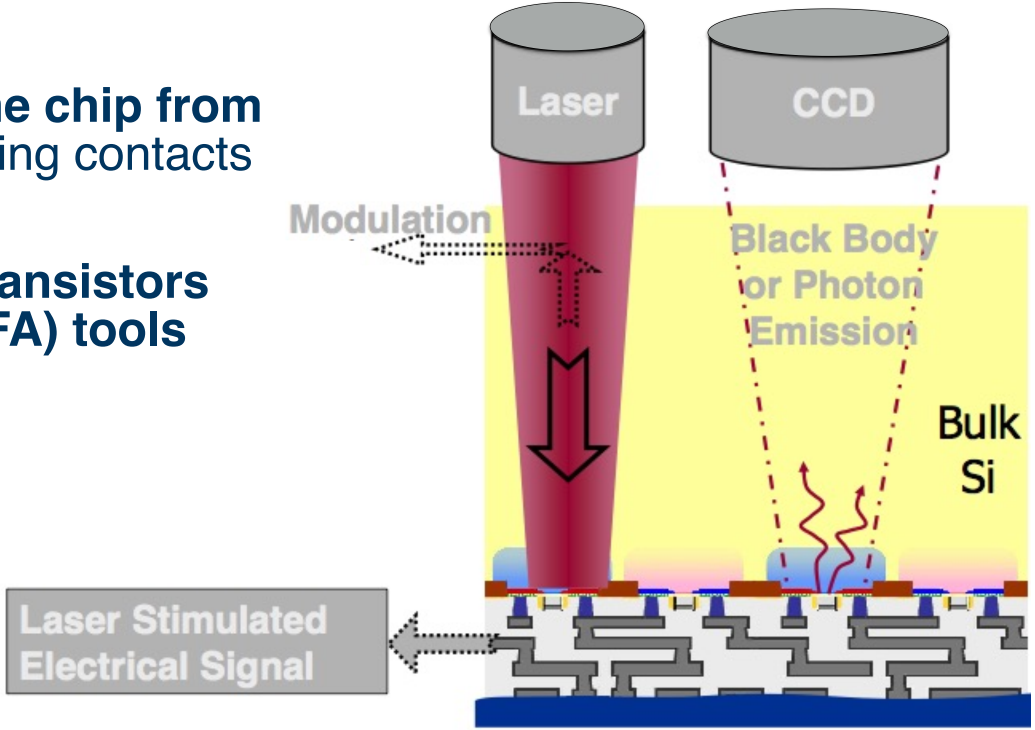
CHALLENGE #2: Preserving your Secrets – Optical Side-Channels

Access to the surface of the chip from IC backside without creating contacts with internal wires

Optical interactions with transistors using Failure Analysis (FA) tools

Optical Techniques:

- Photon Emission
- Laser Stimulation
- Optical Probing



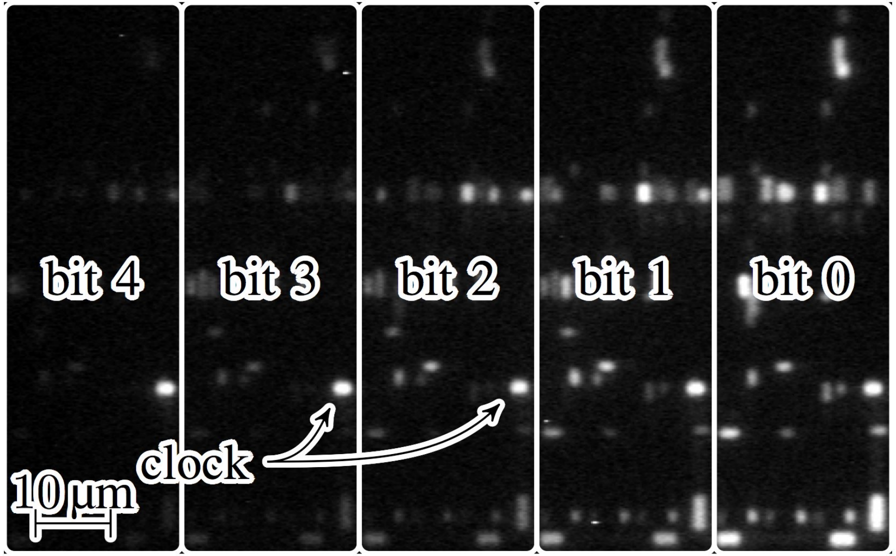
Boit, et al. "From IC Debug to Hardware Security Risk: The power of backside access and optical interaction," IPFA 2016.

Slide courtesy of Shahin Tajik, WPI

CHALLENGE #2: Preserving your Secrets – Photon Emission Example

Assume an n-bit counter in HW:
n clocked registers + some combinatorial logic

- The emission rate is proportional to the switching frequency
- Counter’s LSB is brighter than the MSB
- Clock buffers have the highest switching frequency >> The brightest spots are clock buffers

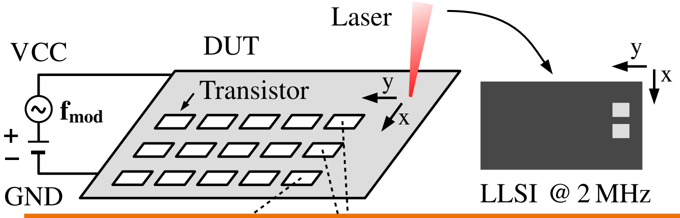


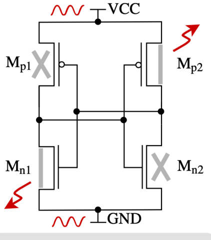
The emission of a counting counter on Intel/Altera MAX V (180 nm)

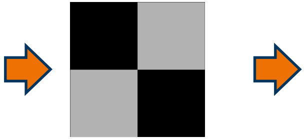
Tajik, et al. ”Emission Analysis of Hardware Implementations," DSD 2014.

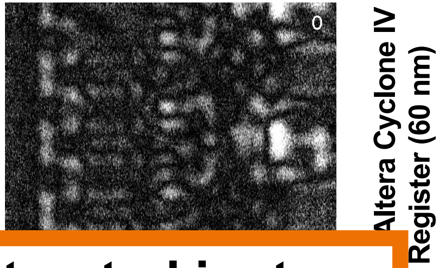
Slide courtesy of Shahin Tajik, WPI

Laser Logic State Imaging (LLSI)



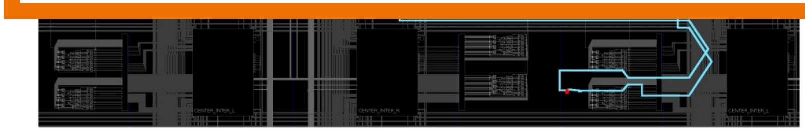


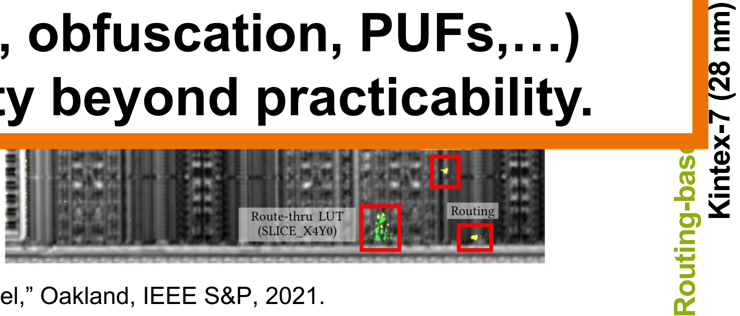




Without any countermeasures, secrets can be extracted just as a matter of time and cost from all conventional CMOS chips.

Combinations of protections (e.g., obfuscation, PUFs,…) can help to increase the complexity beyond practicability.





Krachenfels, et al. "Real-World Snapshots vs. Theory: Questioning the t-Probing Security Model," Oakland, IEEE S&P, 2021.
Krachenfels, et al. "Trojan Awakener: Detecting Dormant Malicious Hardware Using Laser Logic State Imaging," ASHES, 2021.

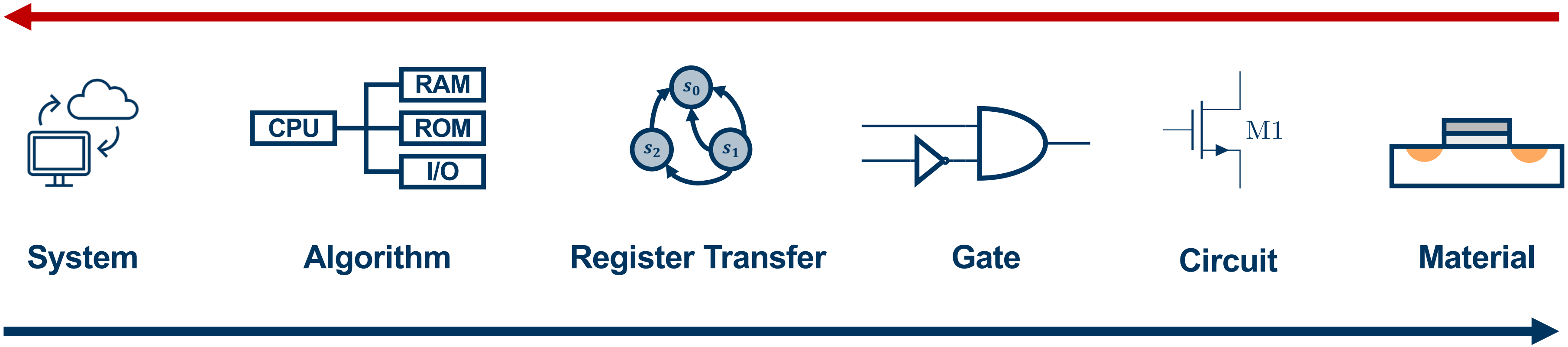
Slide courtesy of Shahin Tajik, WPI

CHALLENGES

Models and Limitations in Implementation Security

Levels of Design Abstraction – Functional Perspective

Abstraction

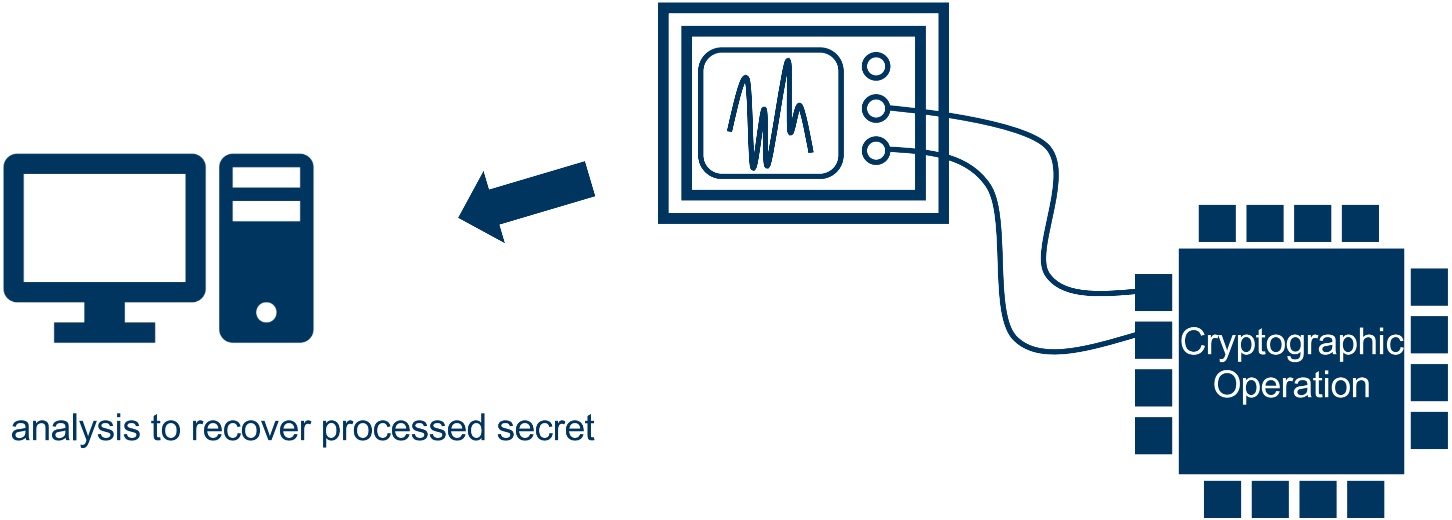


System **Algorithm** **Register Transfer** **Gate** **Circuit** **Material**

Low-Level Detail

How can we this combine with our security requirements, e.g., for SCA?

Side-Channel Attacks in Practice



(Statistical) analysis to recover processed secret

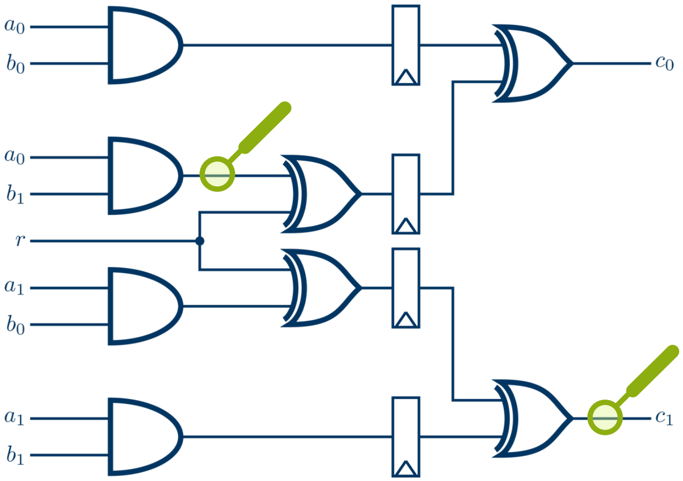
Measure power consumption on execution

Common Empirical Methods:

- Simple Power Analysis (SPA)
- Differential Power Analysis (DPA)
- Correlation Power Analysis (CPA)
- Test Vector Leakage Assessment (TVLA)

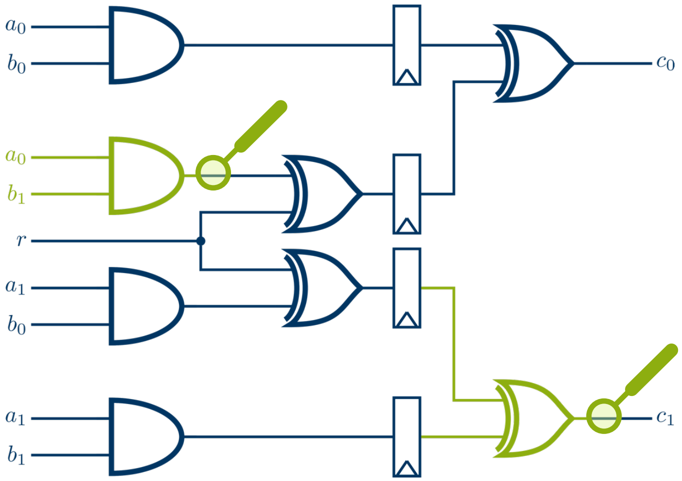
Modeling Side-Channel Attacks

d-probing model [ISW03]



An adversary is given the exact values of up to d wires of a circuit $\mathtt{C}$.

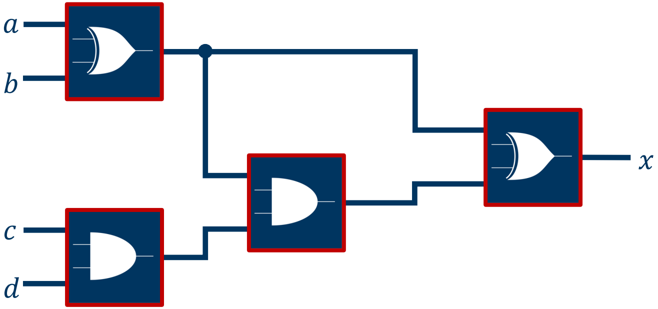
Glitch-extended d-probing model [FGP+18]

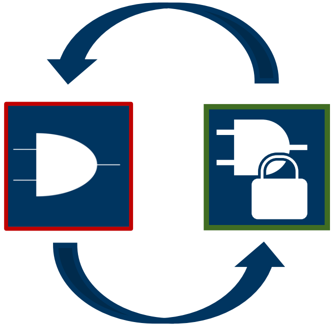


An adversary is given the exact values of all synchronization points influencing up to d wires of a circuit $\mathtt{C}$.

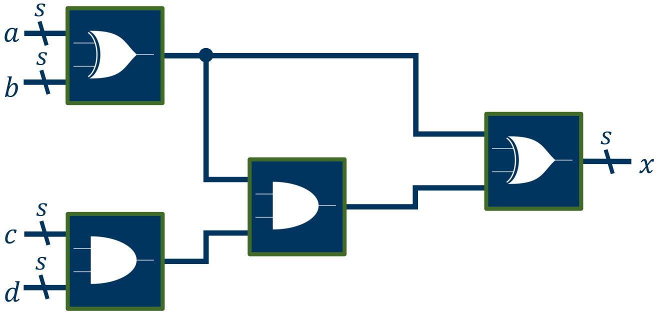
Advanced Models – Protection by Secure Gadgets

Insecure Circuit





Protected Circuit



Replace insecure gates by secure gadgets

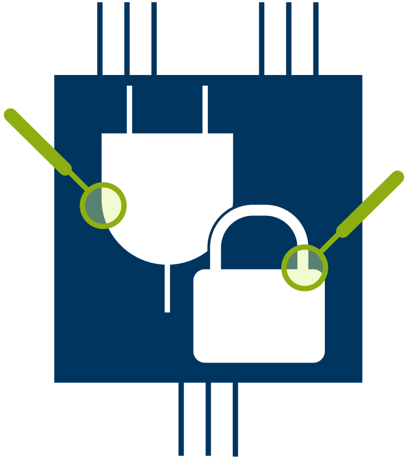
Share inputs and outputs

Maintain timing (pipelining)

Modeling Side-Channel Attacks – Composability

PNI [BBD+15]

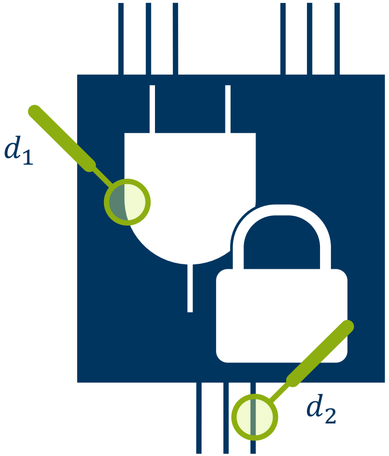
Probe Non-Interference



$d' \leq d$

PSNI [BBD+16]

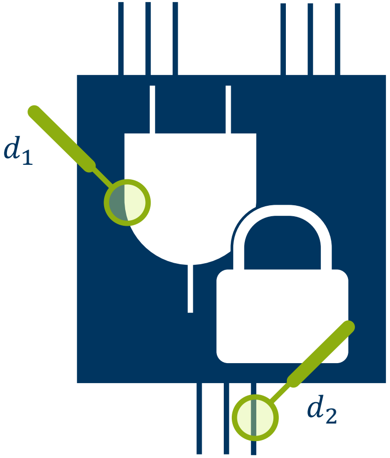
Probe Strong Non-Interference



$d_1 + d_2 \leq d$

PINI [CS20]

Probe-Isolating Non-Interference



$d_1 + d_2 \leq d$

Similiar notions for fault injection attacks and the combined setting: FINI, CINI [FRSG22]

Levels of Design Abstraction and SCA – The Security Gap

Formal Verification

Constant Time

ISW, PNI, PSNI, PINI
FINI, CINI

???

Abstraction

Suitable complexity-aware sub-gate-level modelling is required for holistic verificaiton of implementation security

With a holistic view, root-cause analysis becomes feasible

Low-Level Detail

Testing and Practical Validation

???

SPA
DPA
CPA
TVLA

DIRECTIONS

Moving Target Defense in Hardware

DIRECTION: Moving Target Defense in Hardware

Many attack vectors exploit the static nature of hardware

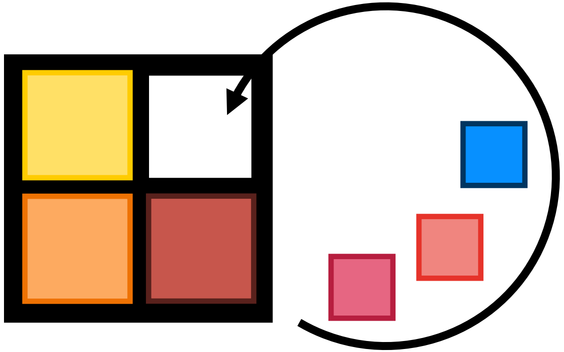
➔ **Idea: *Runtime reconfiguration of HW security components***

Advantages:

- Profiled and location-based attacks could be eliminated, e.g. SCA
- Hardware could be **upgraded** on-the-fly in case of security failures
- Runtime support for **complex crypto portfolios** (cf. NIST PQC standards)

Proof-of-Concept:

Runtime-Dynamic AES Implementation on Hardware (FPGA)

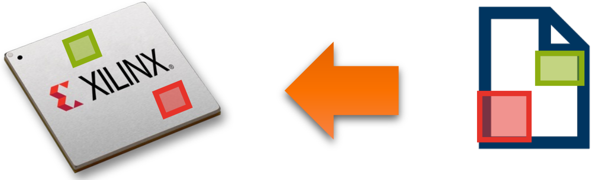


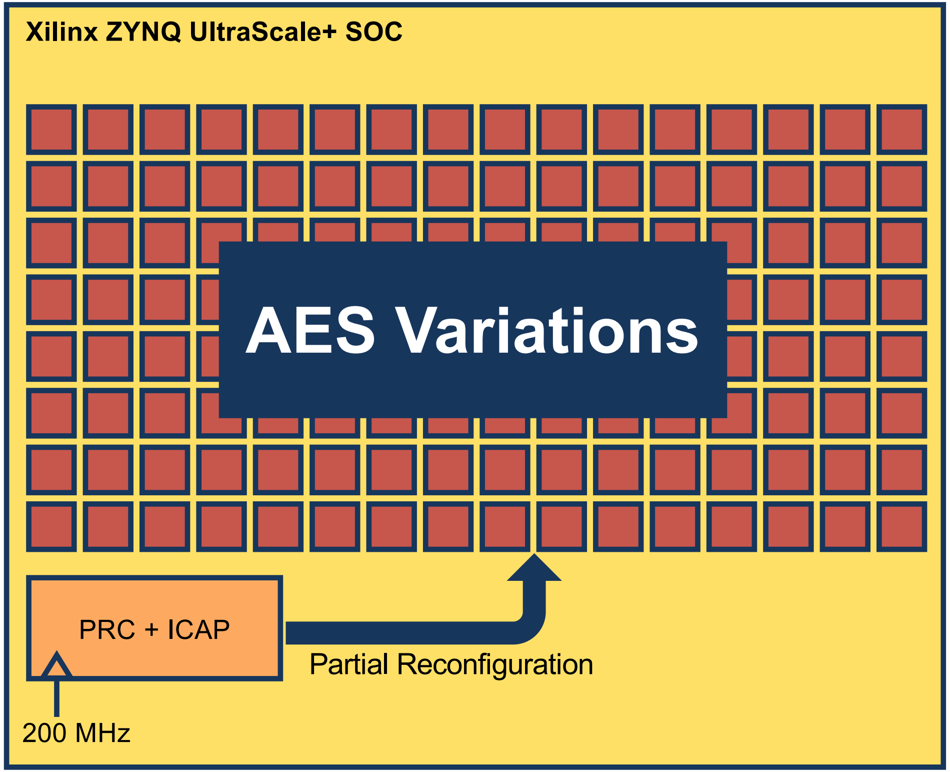
Proof-of-Concept Architecture – Runtime Reconfiguration of AES

FPGAs can be **reconfigured in parts**

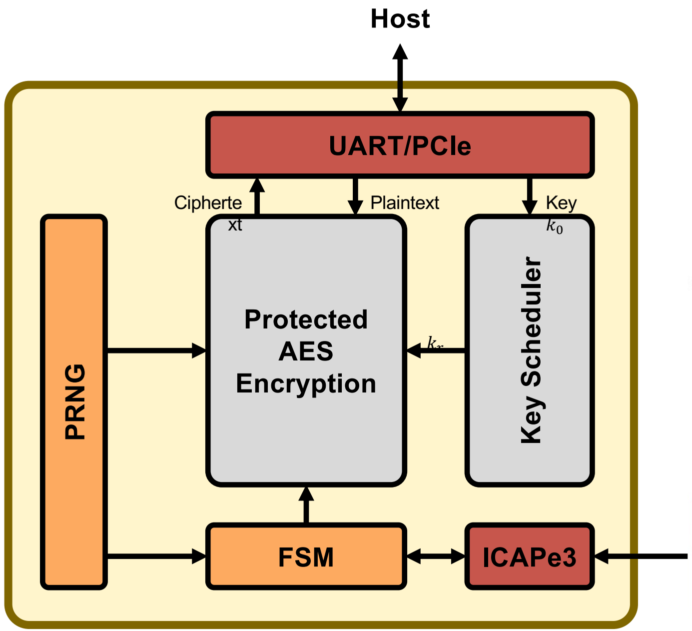
Partial reconfiguration requires prebuilt subconfigurations (e.g. AES rounds)

All configuration data is stored on **external flash** and loaded by the PRC

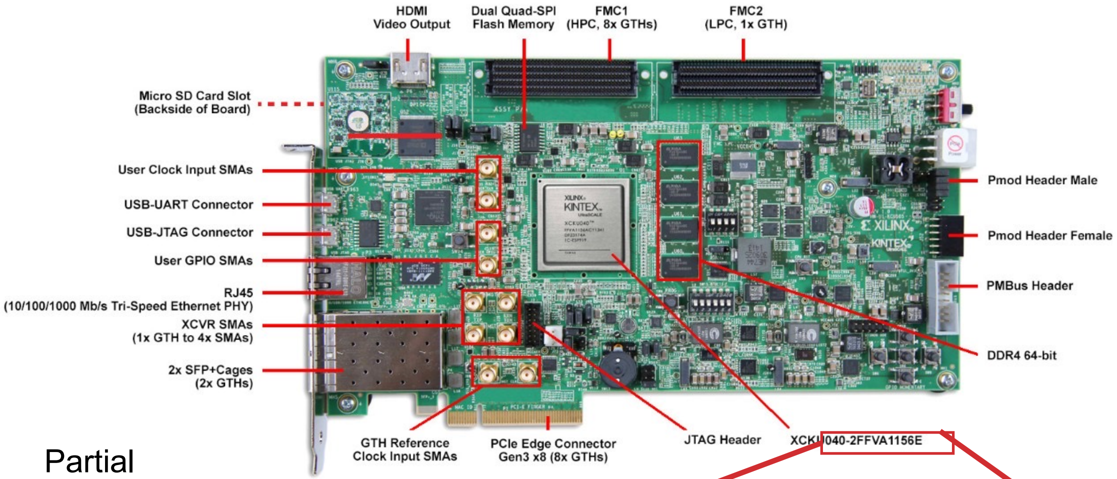




Proof-of-Concept Architecture – Design Concept – Top View



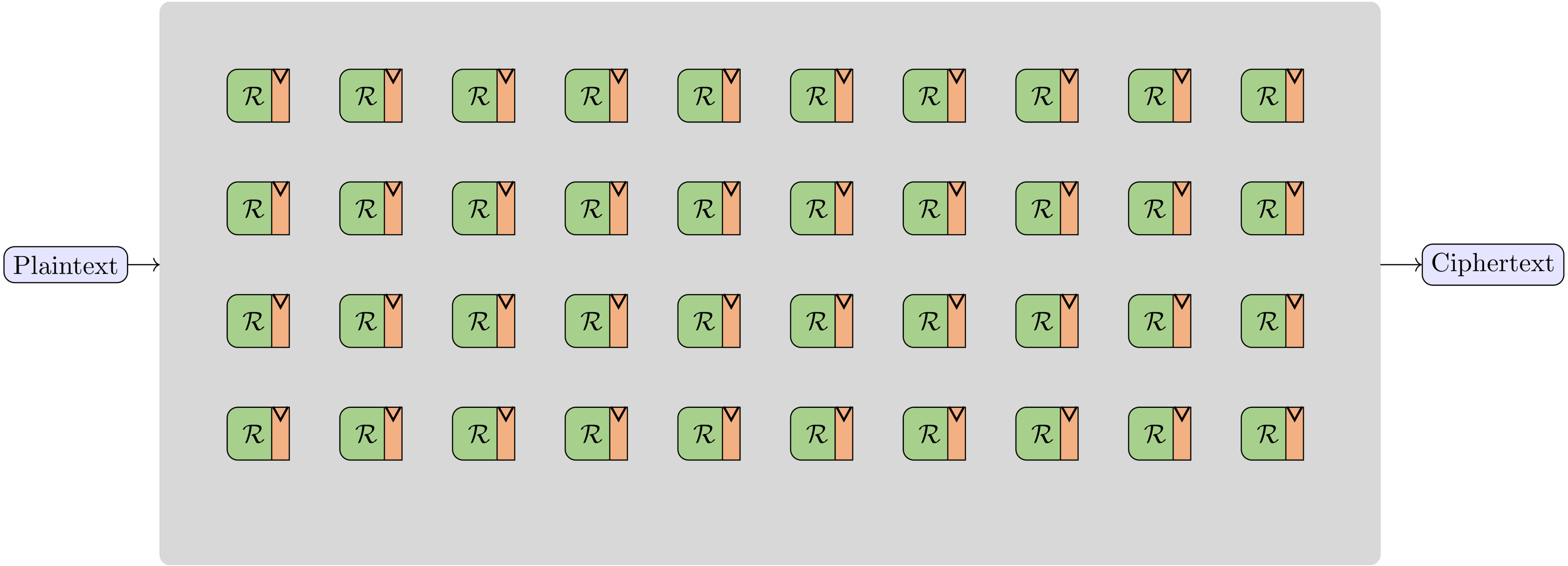
Partial Bitstreams



XCKU040-2FFVA1156E

530 000 Logic Cells • 21.1 Mb BRAM

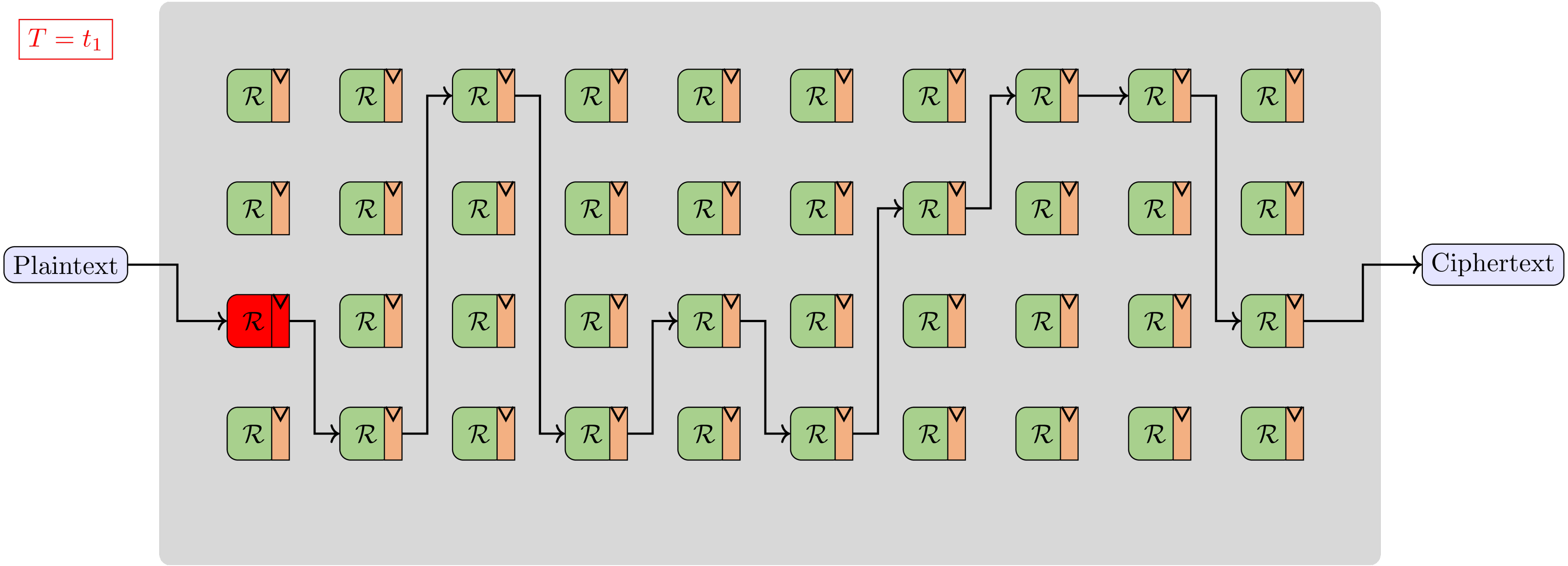
Proof-of-Concept Architecture – Protected AES Encryption



40 differently implemented AES rounds • Arranged in a 4×10 grid • Each column represents one AES round

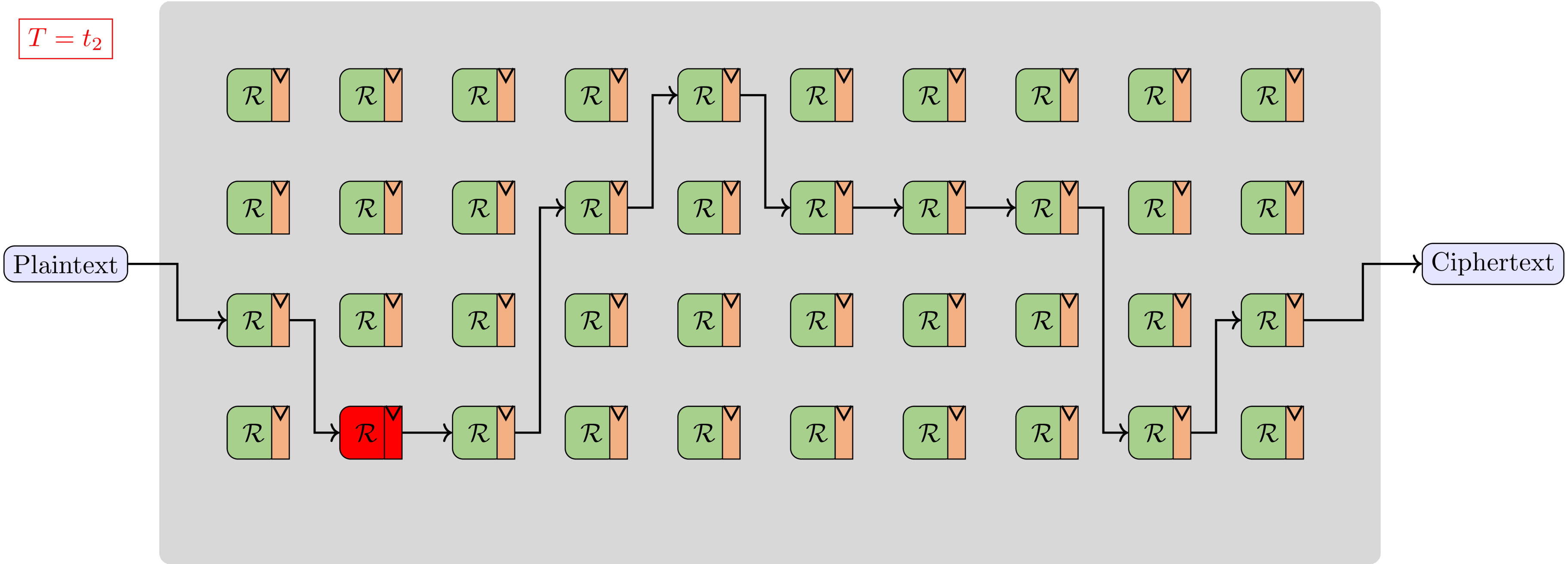
Proof-of-Concept Architecture – Protected AES Encryption

$T = t_1$



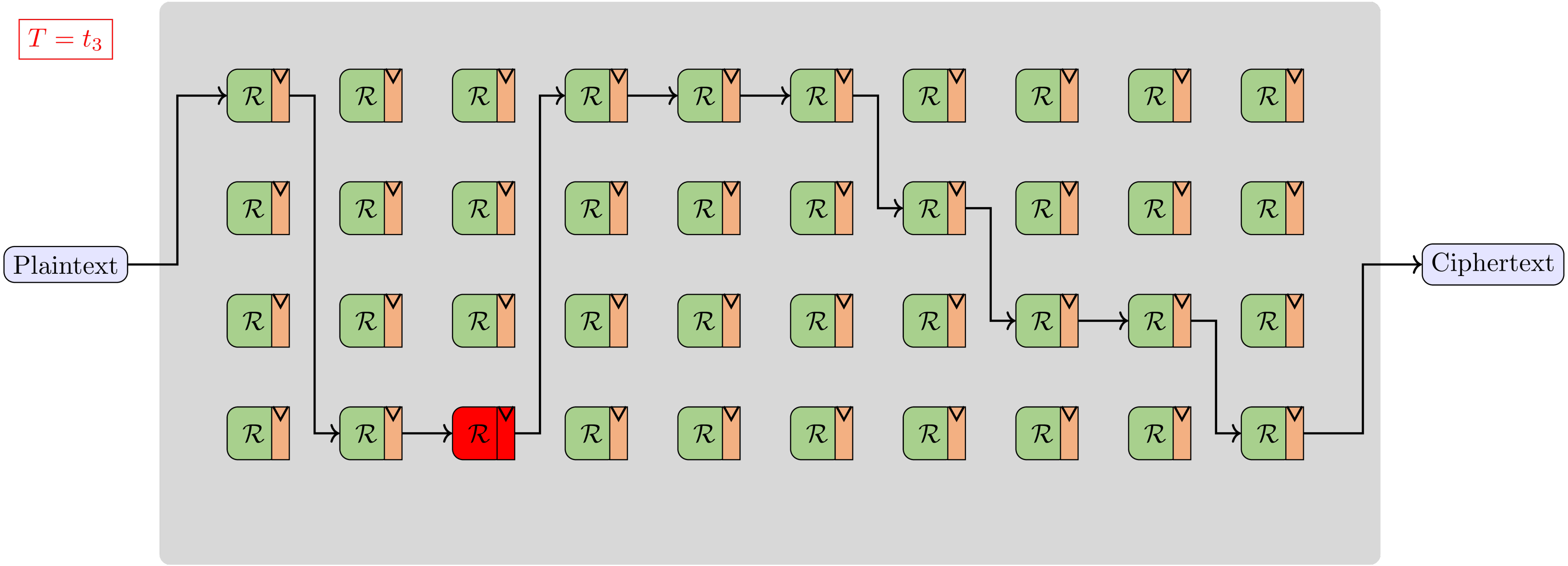
Proof-of-Concept Architecture – Protected AES Encryption

$T = t_2$



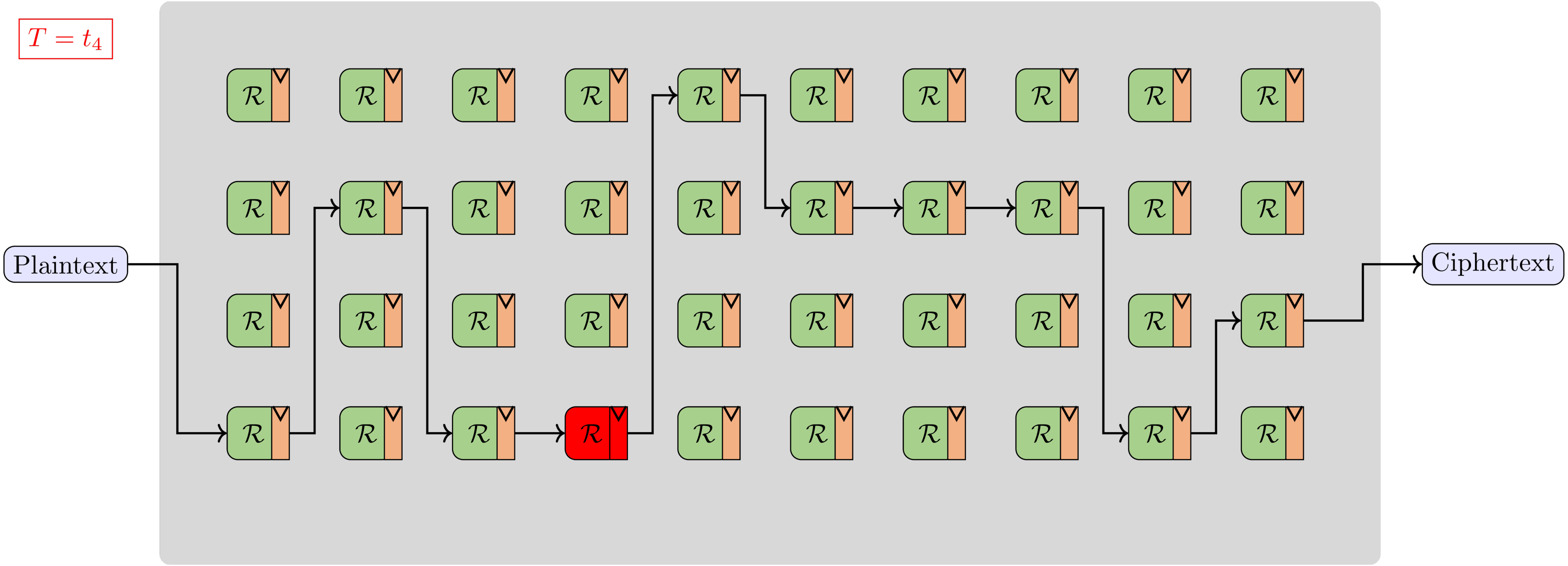
Proof-of-Concept Architecture – Protected AES Encryption

$T = t_3$



Proof-of-Concept Architecture – Protected AES Encryption

$T = t_4$



DIRECTION: Moving Target Defense in Hardware

Many attack vectors exploit the static nature of hardware

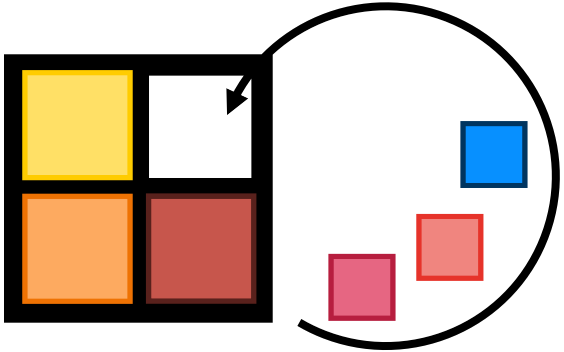
➔ **Idea: *Runtime reconfiguration of HW security components***

Disadvantages :

- Limited countermeasure (x16) with high overhead in time and space
- Runtime reconfiguration on FPGA is currently too slow (>10ms per run)
- Formal security specification, verification and certification with dynamic hardware behavior even more complex

Future Directions:

- Novel highly-reconfigurable FPGAs (as announced by Tabula but never released)
- Neuromorphic computing to the rescue?



DIRECTIONS

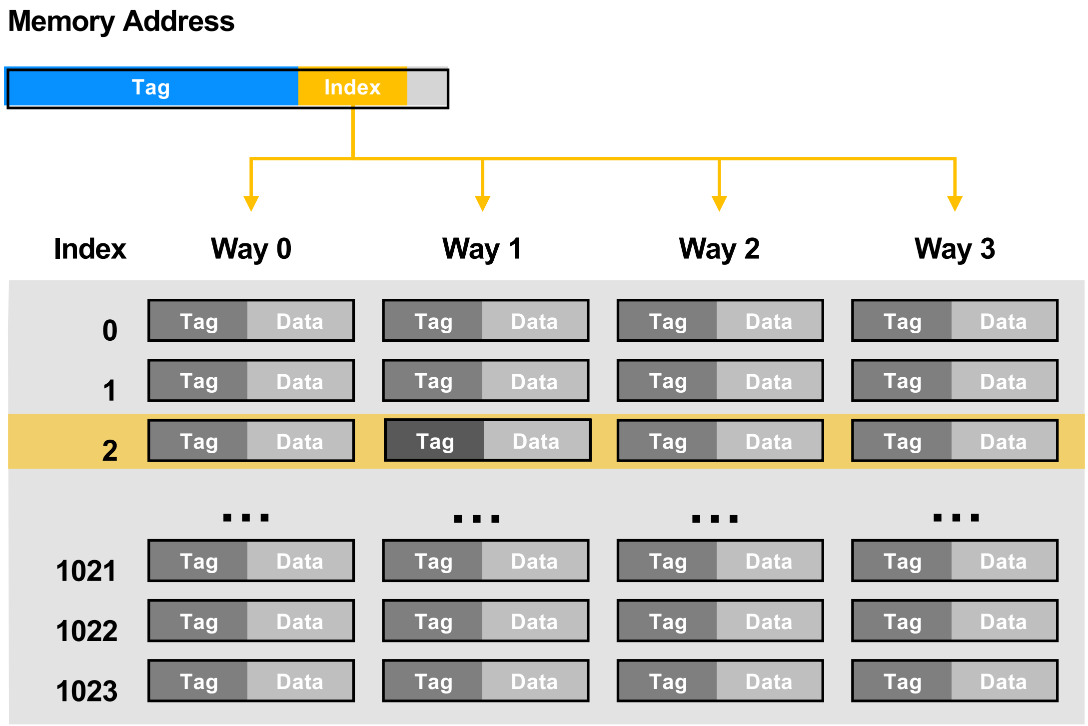
New Hardware Constraints for Cryptography

New Hardware Constraints for Cryptography – Cache Design

Caches are set-associative structures for improving memory access times

Caches are table structure with **ways** and **sets**

- **Set** is determined by part of the address
- **Way** is determined by the replacement policy



Index	Way 0	Way 1	Way 2	Way 3
0	Tag Data	Tag Data	Tag Data	Tag Data
1	Tag Data	Tag Data	Tag Data	Tag Data
2	Tag Data	Tag Data	Tag Data	Tag Data
	…	…	…	…
1021	Tag Data	Tag Data	Tag Data	Tag Data
1022	Tag Data	Tag Data	Tag Data	Tag Data
1023	Tag Data	Tag Data	Tag Data	Tag Data

New Hardware Constraints for Cryptography – Cache Design

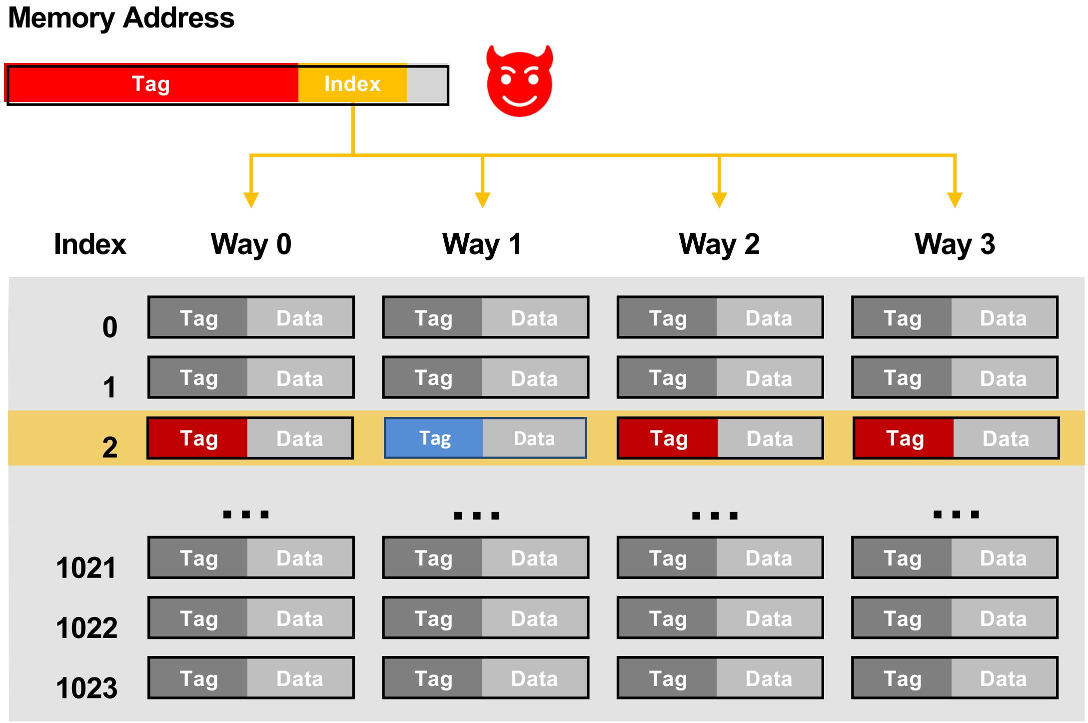
Prime + Probe Attack

An attacker can observe cache accesses

1. Fill a cache set
2. Trigger victim access
3. Re-Access **eviction set**

→ Cache miss = access

- **There are even more sophisticated attacks**
  - Flush & Reload
    Prime Prune & Probe



New Hardware Constraints for Cryptography – Cache Design

Cache randomization

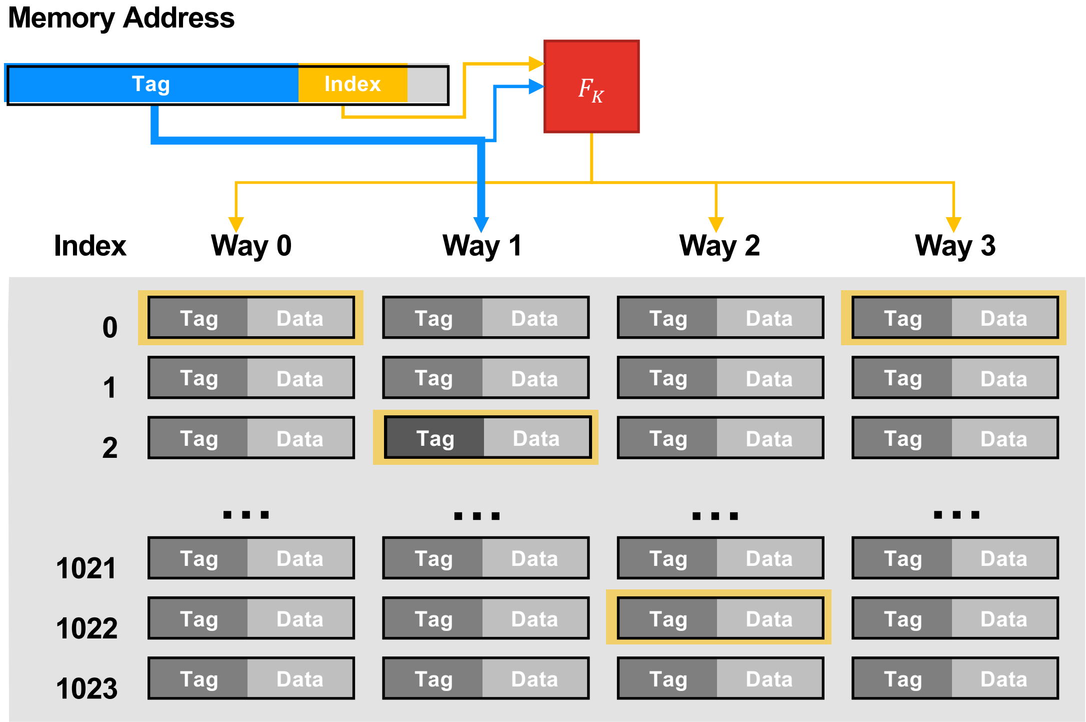
Prevents efficiently
Prime + Probe attacks

Index is pseudorandomly generated from the address

Data is placed in one of the candidate entries

Latest architectural proposal: *ClepsydraCache* [TNF+23]

What do we use as F_K?



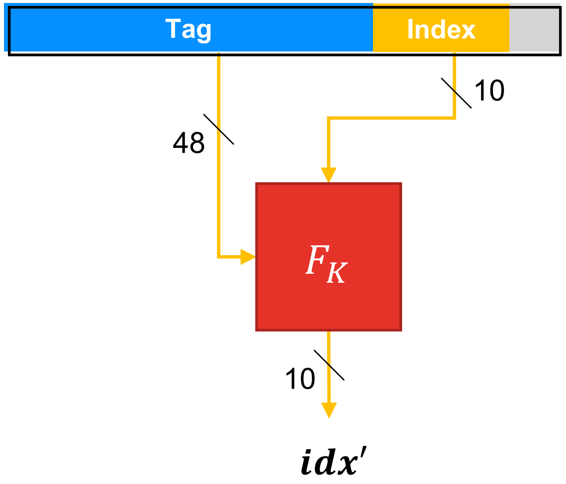
[TNF+23] Jan Philipp Thoma, Christian Niesler, Dominic A. Funke, Gregor Leander, Pierre Mayr, Nils Pohl, Lucas Davi, Tim Güneysu: ClepsydraCache - Preventing Cache Attacks with Time-Based Evictions. USENIX Security Symposium 2023

New Hardware Constraints for Cryptography – Cache Randomization

Functional Requirements

1. **Low Latency**
   Function will be part of cache design and must not decrease the access time
2. **Key Dependency**
   Secret and session-specific cache randomization
3. **Invertibility (with a given tag)**
   This is required to support write-back caches
4. **Designed to match cache architectures (most recent caches have 1024 sets)**
   Map 48-bit tag + 10-bit index to 10-bit randomized index
   Offset bits must be ignored

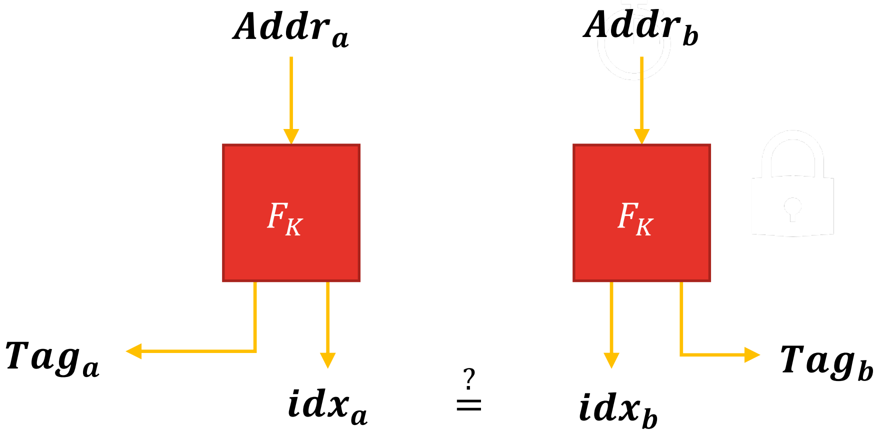
Memory Address



New Hardware Constraints for Cryptography – Attacker Modell

Attacker Model is different from what we usually assume in symmetric cryptography

- The attacker has **never** access to the output of F_K
- The attacker can observe **if two addresses collide**
- The attacker aims to find **colliding addresses**



New Hardware Constraints for Cryptography – Approaches

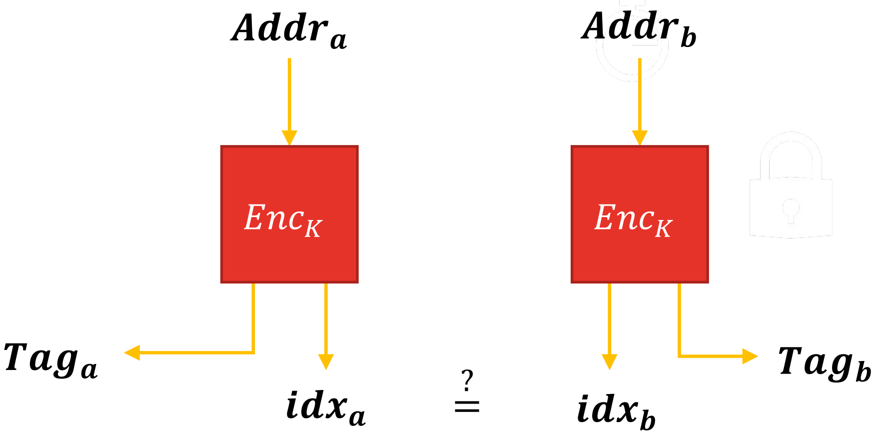
Approach 1: Use a low-latency block cipher (e.g. 64-bit PRINCE)

- Zero-pad 58-bit input and sample index-bits from ciphertext, use remainder as tag
- 6 Bit storage overhead for the tag and comparison logic → expensive

Approach 2: Design a 58-bit block cipher

- Possible, but conventional design processes more (tag) data than required
- Ignores the weaker adversary model (no output available to attacker!)

Approach 3: Design a 10-bit cipher with reduced latency
➔ *SCARF* [CGL+23]



[CGL+23] Federico Canale, Tim Güneysu, Gregor Leander, Jan Philipp Thoma, Yosuke Todo, Rei Ueno: SCARF - A Low-Latency Block Cipher for Secure Cache-Randomization. USENIX Security Symposium 2023

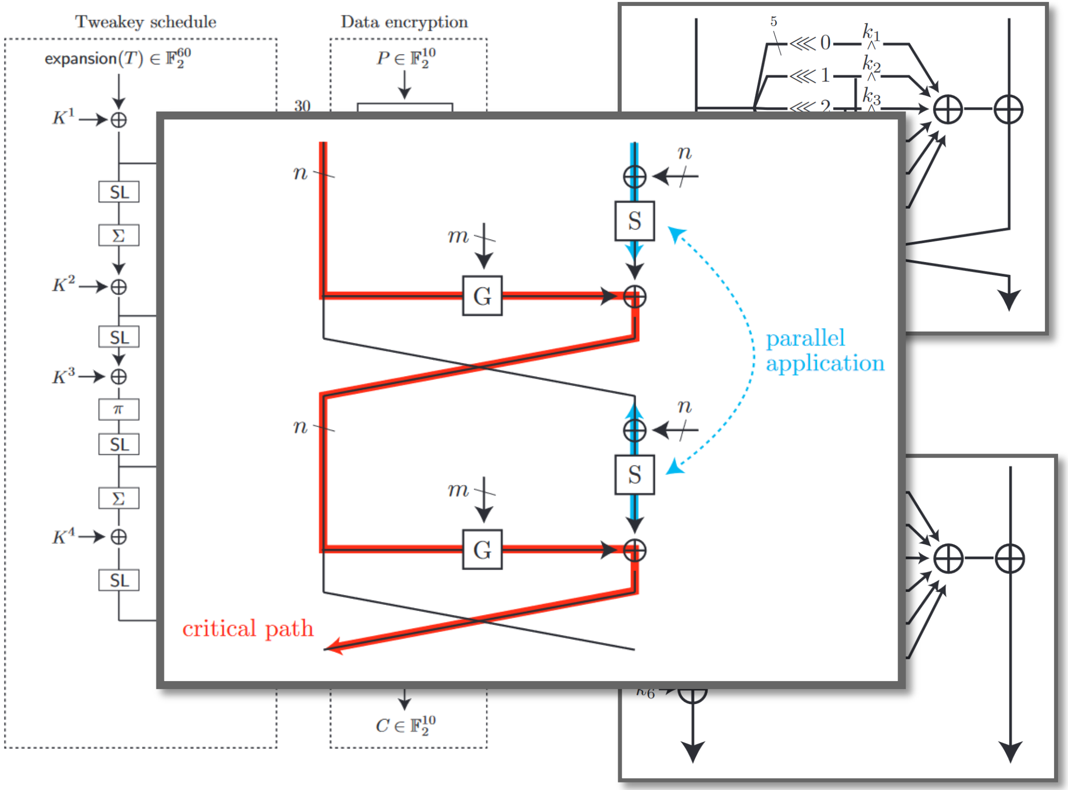
New Hardware Constraints for Cryptography – SCARF Cipher

SCARF is a 10-bit tweakable block cipher with 48-bit tweak and 240 Bit key

Latency optimized combination of SPN and Feistel structure

Designed for 7 + 1 rounds

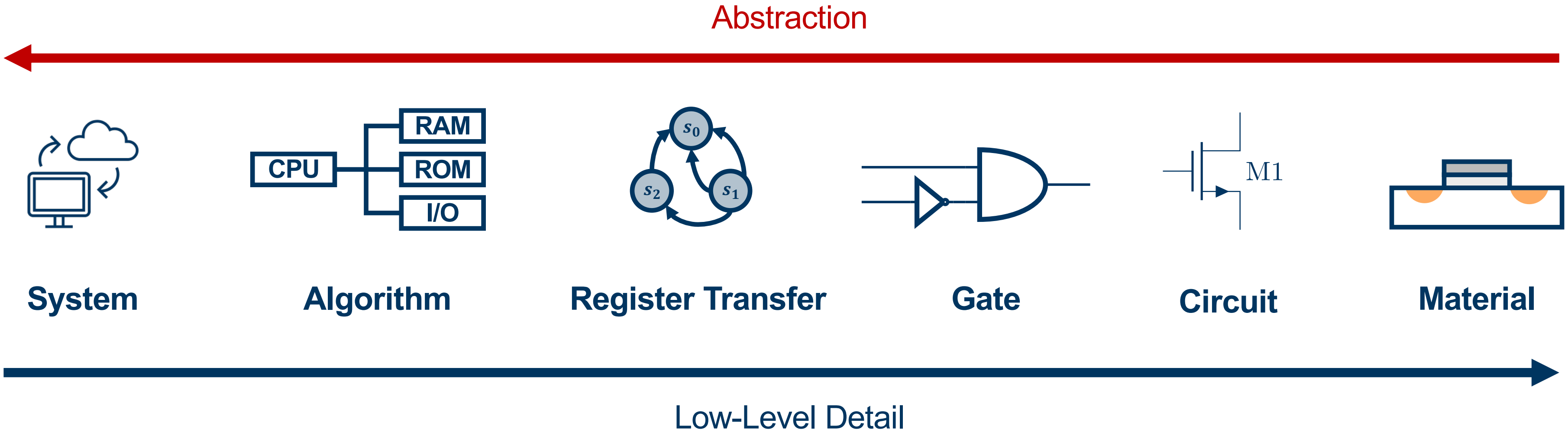
Further cryptanalysis welcome!



DIRECTIONS

Trusted and Security-oriented Hardware Design

Trusted and Security-oriented Hardware Design



Abstraction

System **Algorithm** **Register Transfer** **Gate** **Circuit** **Material**

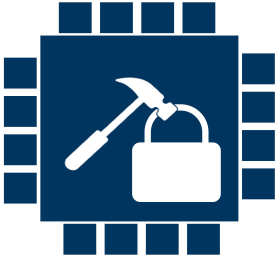
Low-Level Detail

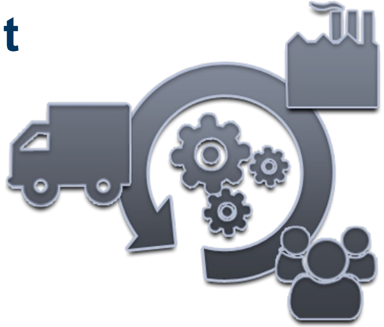
Hardware design EDA support optimization for performance, area and energy.
Support for security-oriented hardware design is required in EDA

Trusted and Security-oriented Hardware Design

Security-oriented design tool support is solely not sufficient

- Trust in development tools (external verification and validation required)
- Trust in manufacturing parties and supply chains
- Trust in delivery services
- Trust in key management and distribution services





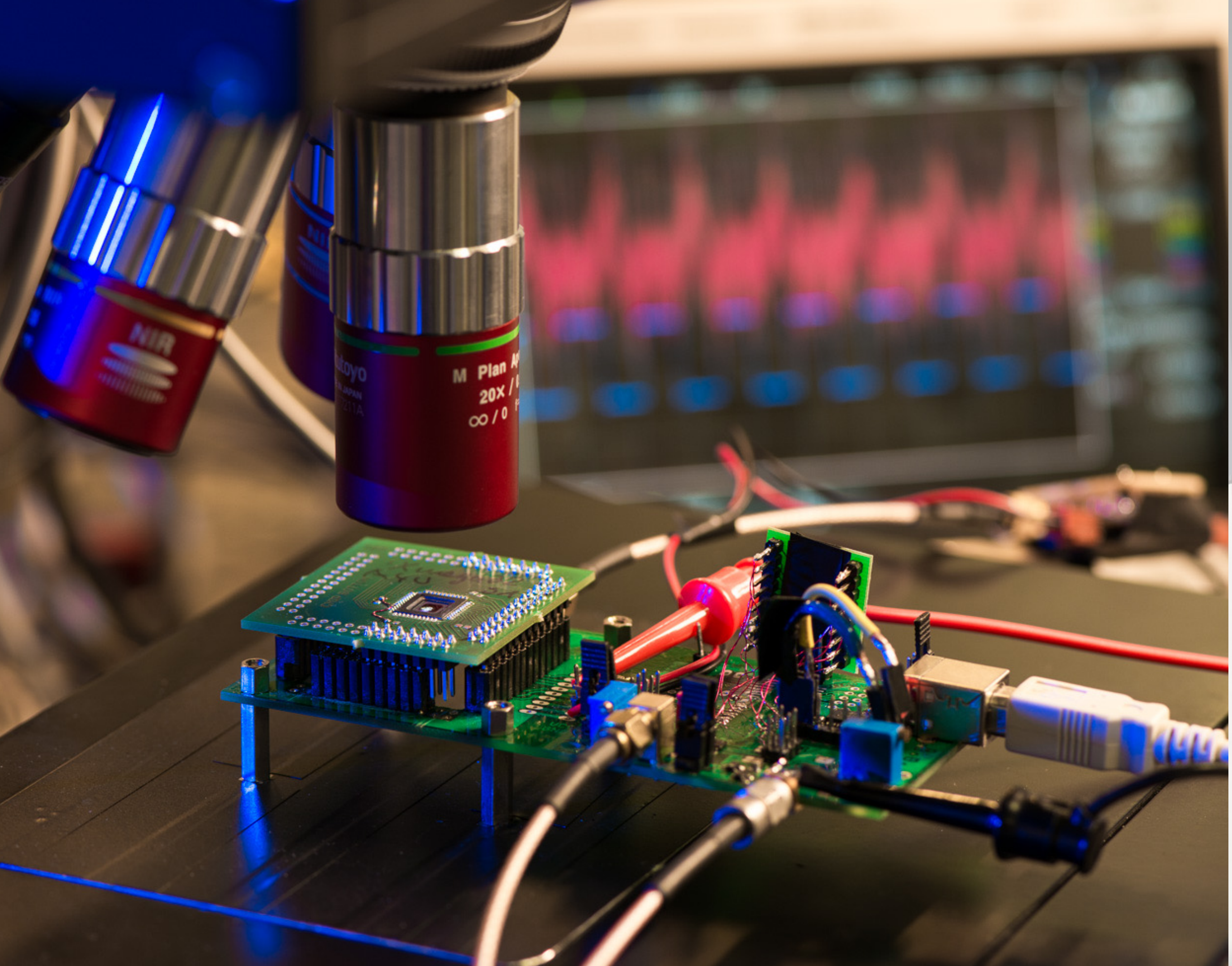
CONCLUSIONS

Conclusions

Lessons Learned

- Models of adversaries are **fuzzy**, better design for stronger ones
- **Secrets are hard to protect** – hardware helps to make it harder
- Models and abstractions are imperfect by nature, but often **still too imperfect for implementation security**

- **Moving target defense in hardware** could disable profiling attacks
- New hardware-oriented use-cases for cryptography emerge (such as **fault-tolerant cryptography**)
- Trusted and secure design requires **trusted and secure-oriented tools**



RUB

Thank you!

tim.gueneysu@rub.de

SECURITY
ENGINEERING

References

[BBD+15] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. *Verified Proofs of Higher-Order Masking.* In EUROCRYPT, pages 457

[BBD+16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. *Strong Non-Interference and Type-Directed Higher-Order Masking.* In SIGSAC, pages 116–129, 2016.

[CS20] Gaetan Cassiers and Fran¸cois-Xavier Standaert. *Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference.* IEEE Trans. Inf. Forensics Secur., 15:2542–2555, 2020.

[DDE+20] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. *Protecting against Statistical Ineffective Fault Attacks.* IACR Trans. Cryptogr. Hardw. Embed. Syst., 2020(3):508–543, 2020.

[DN20] Siemen Dhooghe and Svetla Nikova. *My Gadget Just Cares for Me – How NINA Can Prove Security Against Combined Attacks.* In CT-RSA, volume 12006 of Lecture Notes in Computer Science, pages 35–55. Springer, 2020.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and Fran¸cois-Xavier Standaert. *Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model.* IACR Trans. Cryptogr. Hardw. Embed. Syst. , 2018(3):89–120, 2018.

[FRSG22] Jakob Feldtkeller, Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. *CINI MINIS: Domain Isolation for Fault and Combined Security.* CCS, 2022

[HPB21] Vedad Hadzic, Robert Primas, and Roderick Bloem. *Proving SIFA protection of masked redundant circuits.* In Automated Technology for Verification and Analysis, volume 12971 of Lecture Notes in Computer Science, pages 249–265. Springer, 2021.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. *Private Circuits: Securing Hardware against Probing Attacks.* In Dan Boneh, editor, CRYPTO, volume 2729 of Lecture Notes in Computer Science, pages 463–481. Springer, 2003.

[TNF+23] Jan Philipp Thoma, Christian Niesler, Dominic A. Funke, Gregor Leander, Pierre Mayr, Nils Pohl, Lucas Davi, Tim Güneysu:
ClepsydraCache - Preventing Cache Attacks with Time-Based Evictions. USENIX Security Symposium 2023

[CGL+23] Federico Canale, Tim Güneysu, Gregor Leander, Jan Philipp Thoma, Yosuke Todo, Rei Ueno:
SCARF - A Low-Latency Block Cipher for Secure Cache-Randomization. USENIX Security Symposium 2023

[KSM20] David Knichel, Pascal Sasdrich, and Amir Moradi. *SILVER – Statistical Independence and Leakage Verification.* In ASIACRYPT, volume 12491 of Lecture Notes in Computer Science, pages 787–816. Springer, 2020.

[RBRSS+21] Jan Richter-Brockmann, Aein Rezaei Shahmirzadi, Pascal Sasdrich, Amir Moradi, and Tim Güneysu. *FIVER – Robust Verification of Countermeasures against Fault Injections.* IACR Trans. Cryptogr. Hardw. Embed. Syst., 2021(4):447–473, Aug. 2021.

[RBSG21] Jan Richter-Brockmann, Pascal Sasdrich, and Tim Güneysu. *Revisiting Fault Adversary Models - Hardware Faults in Theory and Practice.* Trans. On Computers, 2022

[RFSG22] Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, and Tim Güneysu. VERICA - Verification of Combined Attacks: Automated formal verification of security against simultaneous information leakage and tampering. IACR Trans. Cryptogr. Hardw. Embed. Syst. , 2022(4), 2022.

[SMG16] Tobias Schneider, Amir Moradi, and Tim Güneysu. *ParTI - Towards Combined Hardware Countermeasures Against Side-Channel and Fault-Injection Attacks.* In CRYPTO 2016, volume 9815 of Lecture Notes in Computer Science, pages 302–332. Springer, 2016.