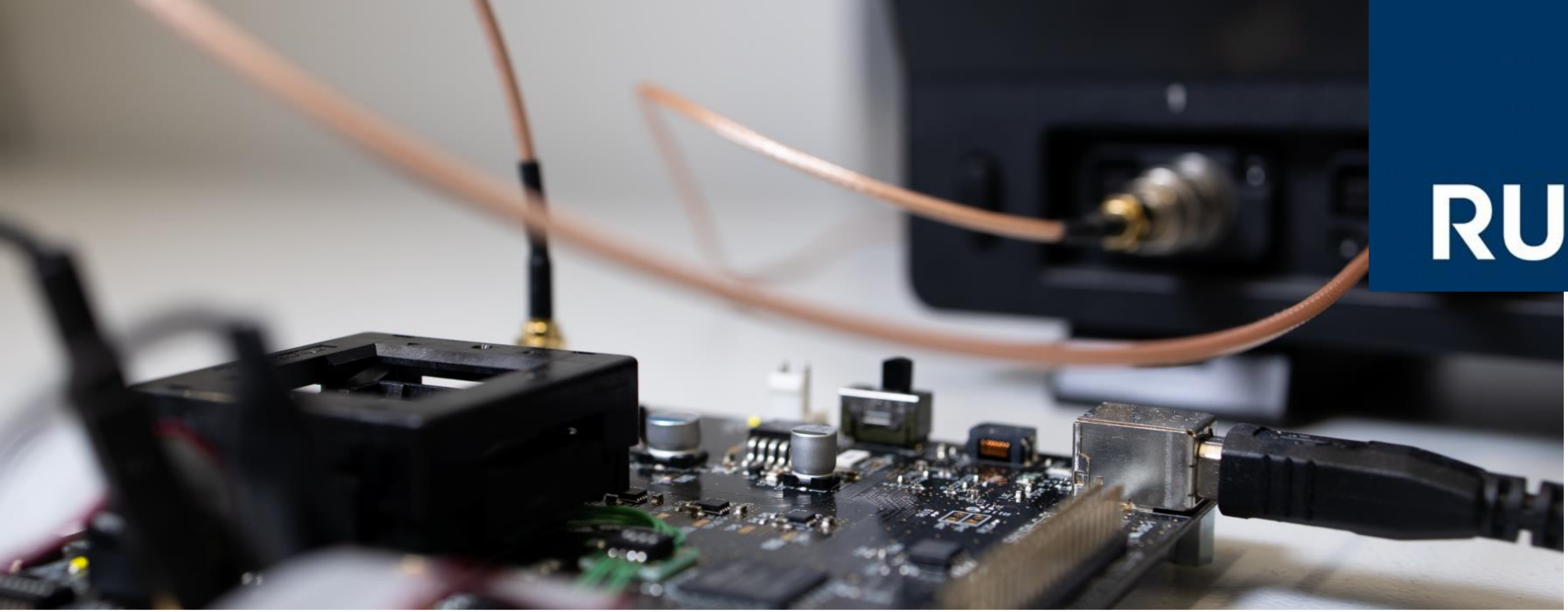


RUHR-UNIVERSITÄT BOCHUM

COMPUTER-AIDED HARDWARE SECURITY VERIFICATION

PASCAL SASDRICH

September 22, 2022

Chair for Security Engineering
Faculty for Computer Science
Ruhr University Bochum

INTRODUCTION

Why do we need to verify hardware security?

CRYPTOGRAPHY IN THEORY AND PRACTICE

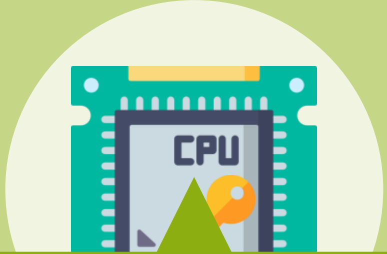
THEORY

Strong and robust cryptographic algorithms.

Implicit secrecy and integrity of computations.

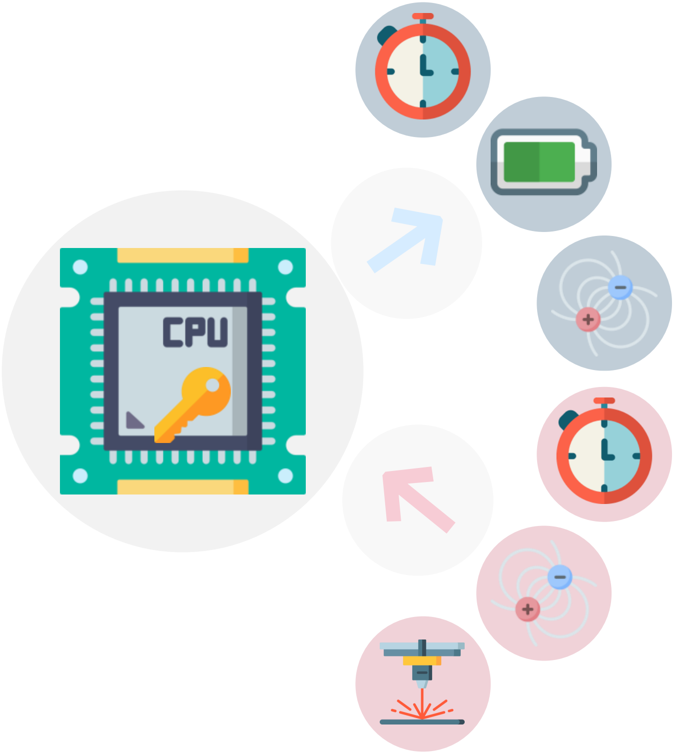
PRACTICE

Cryptography implemented on physical devices.



Attackers observe and manipulate physical characteristics.

THREATS & ATTACKS



SIDE-CHANNEL ANALYSIS

Passive implementation attacks exploiting information leakage:

- *execution time*
- *power & energy consumption*
- *electro-magnetic radiations*

FAULT INJECTION ANALYSIS

Active implementation attacks exploiting information tampering:

- *clock & voltage glitches*
- *electro-magnetic pulses*
- *laser beams*

DESIGN & VALIDATION

HARDWARE
DESIGN TEAM

hand-crafted solutions

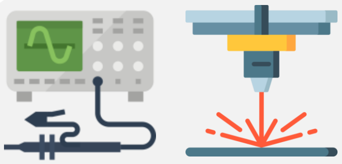
Can remove security features and properties.

- *Classical processing (synthesis, placing , etc.)*
- *Optimization for area, power, performance, etc.*

manual correction

No formal security guarantees.

- *Prototype implementation*
- *Empirical validation through practical attacks*



SECURE
HARDWARE

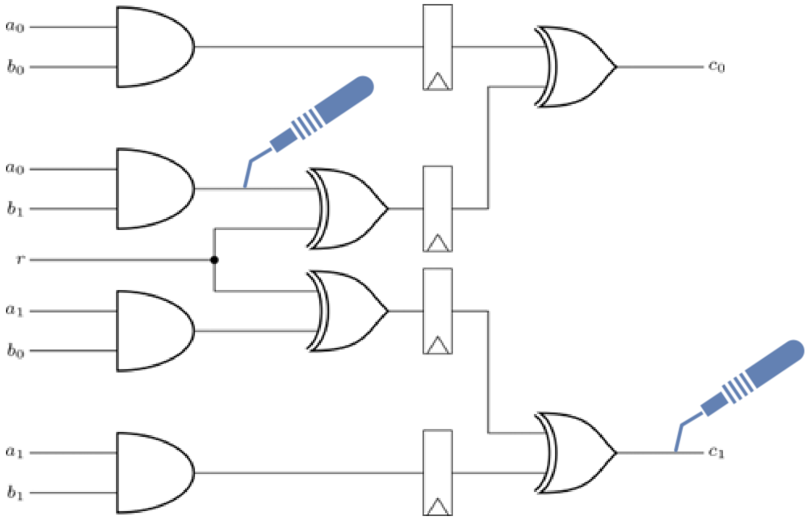
RUHR UNIVERSITÄT BOCHUM RUB

SIDE-CHANNEL ANALYSIS

How do we verify security against passive information leakage?

ADVERSARY | OBSERVING

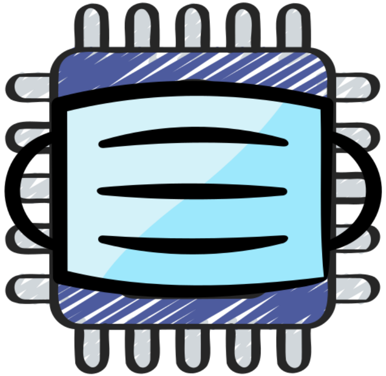
ADVERSARY | MODEL



Threshold t-probing model [ISW03]

- access to up to $t \leq d$ wires of a circuit
- probes are static during circuit invocation
- each probe is:
  - *noise-free, instantaneous & stable*
  - *independent of all other probes*
- probe-extensions [FGMDP+18] to model
  - *combinatorial recombinations (glitches)*

COUNTERMEASURES | MASKING

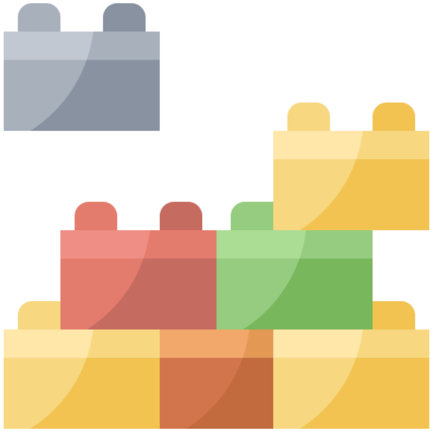


MASKING

BOOLEAN MASKING:

- predominant hardware countermeasure
- formal and sound security foundation:
  - → $X \in \mathbb{F}_n \rightarrow (X^0, X^1, \dots, X^{s-1}) \in \mathbb{F}_n^s$
  - → $X^i \overset{\$}{\leftarrow} \mathbb{F}_n$ for all $0 \leq i \leq s-1$
  - → $X^{s-1} = \left(\bigoplus_{i=0}^{s-2} X^i\right) \oplus X$
- logic operations on shared representation

COUNTERMEASURES | GADGETS



COMPOSING GADGETS

PROBLEM:

Finding efficient masked circuits is hard for:

- higher security orders d
- complex circuits and Boolean functions

SOLUTION:

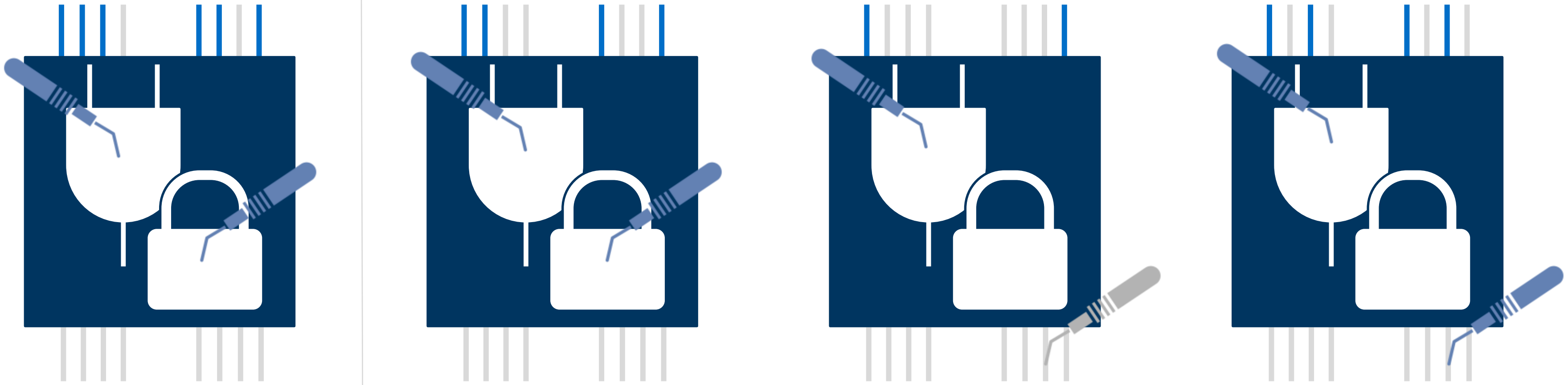
Masked circuits for atomic logic functions:

- mainly focus on masked AND & XOR gates
- special notions ensure secure composition

COMPUTER-AIDED VERIFICATION | NOTIONS

SECURITY

COMPOSABILITY



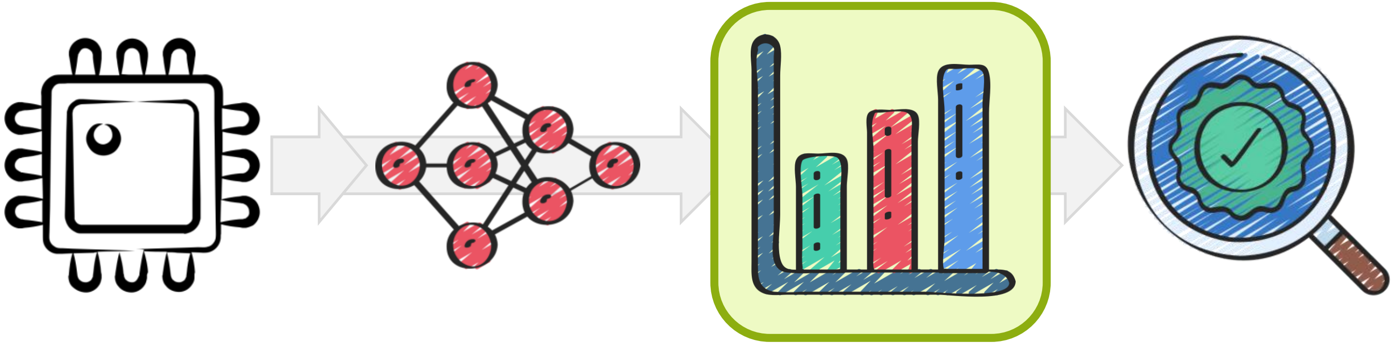
t-PROBING [ISW03]
PROBING
SECURITY

P-NI [BBD+15]
PROBE
NON-INTERFERENCE
$d' \leq d$

P-SNI [BBD+16]
PROBE STRONG
NON-INTERFERENCE
$d_1 + d_2 \leq d$

P-INI [CS20]
PROBE-ISOLATING
NON-INTERFERENCE

VERIFICATION TOOL | APPROACH



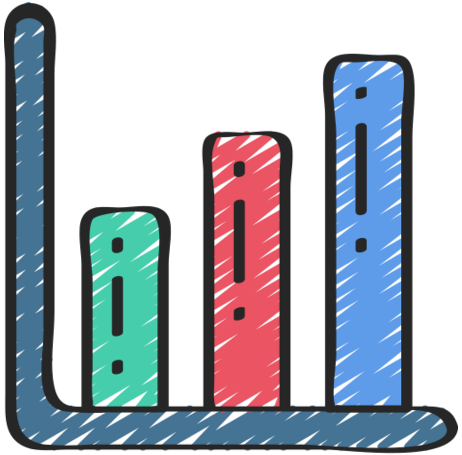
GATE-LEVEL NETLIST

CIRCUIT MODEL (DAG)

STAT. MODEL (BDD)

LEAKAGE VERIFICATION

VERIFICATION TOOL | STATISTICAL MODEL



STATISTICAL MODEL

CONCEPT:

- circuit gates are stored as Binary Decision Diagrams
- BDDs allow counting satisfying solutions
  - identical and independent distributed inputs
  - gate outputs modeled as binary events
- compute statistical independence on binary events

All security and composability notions can be expressed in terms of statistical independence.

VERIFICATION TOOL | SILVER [KSM20]

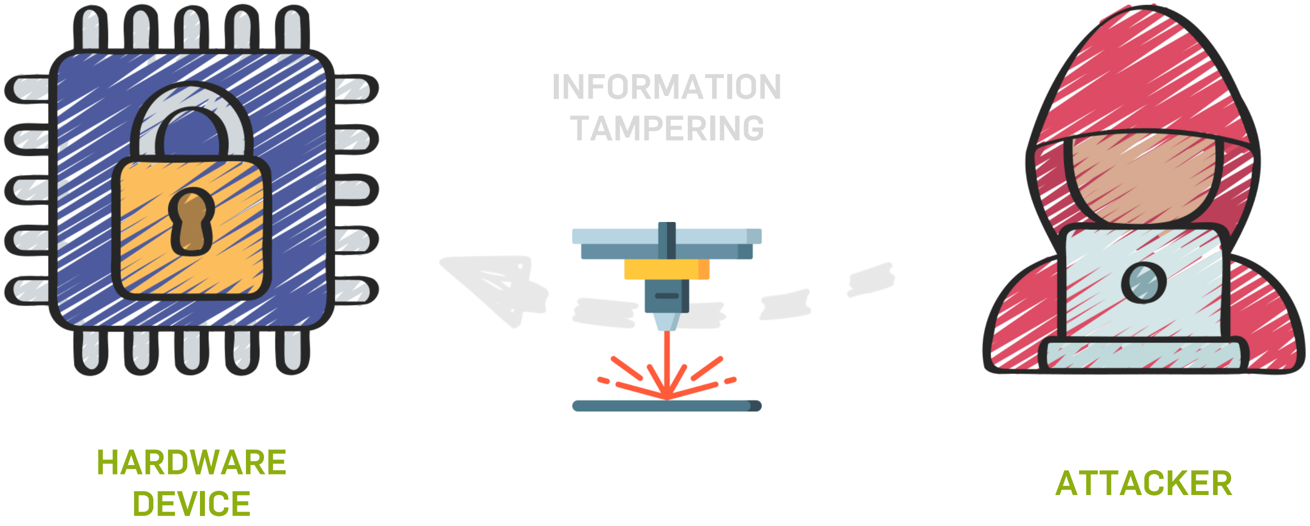
SCAN ME

VERIFICATION OF A FIRST-ORDER DOMAIN-ORIENTED MASKING AND-GADGET

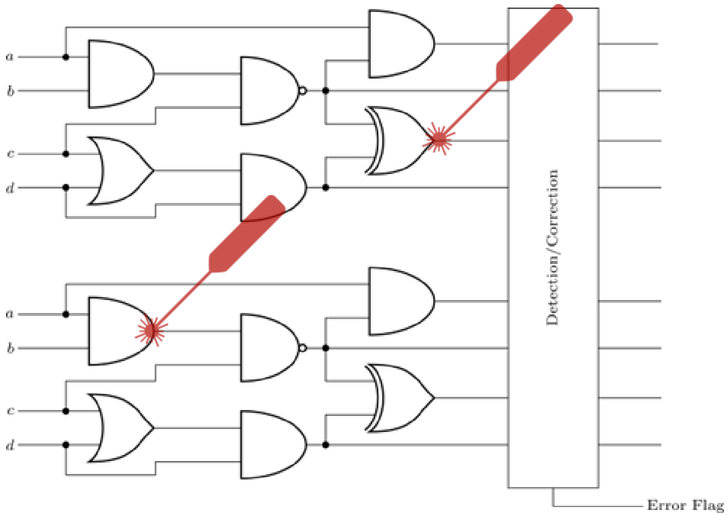
FAULT INJECTION ANALYSIS

How do we verify security against active information tampering?

ADVERSARY | MANIPULATING



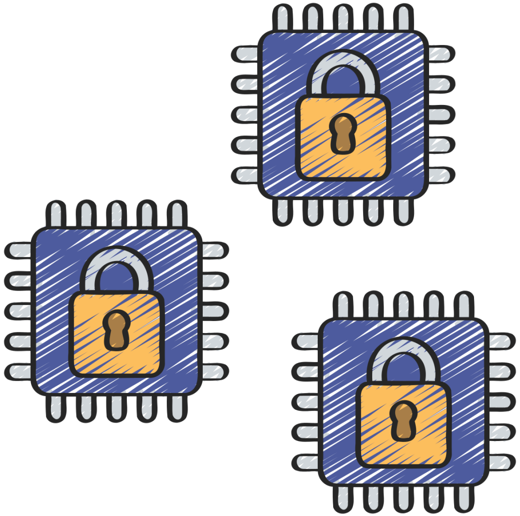
ADVERSARY | MODEL



Parametrized injection model [RBSG22]

- alter to up to $n' \leq n$ gates of a circuit
- each injection is parametrized by:
  - *cardinality (number of faults)*
  - *type (e.g., set, reset, bit-flip, etc.)*
  - *location (comb, or seq. logic)*
- predefined parameters for:
  - *clock/voltage glitches*
  - *EM pulses,*
  - *laser fault injections*

COUNTERMEASURES | REDUNDANCY



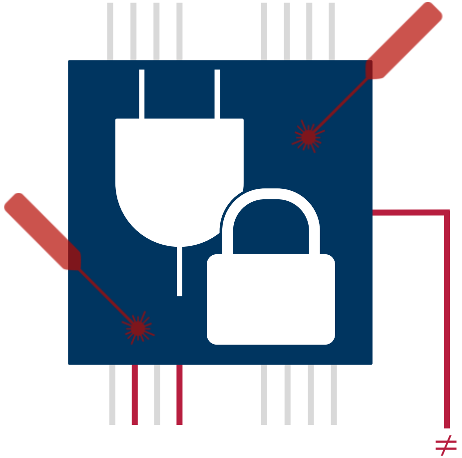
REDUNDANCY

REDUNDANCY:

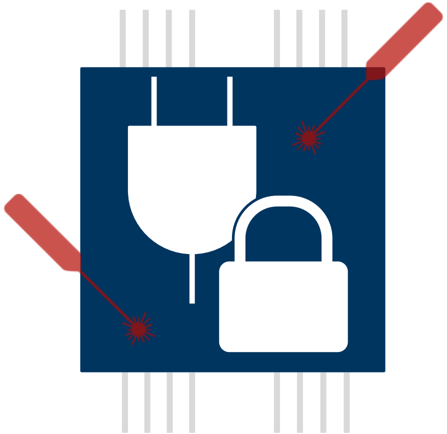
- repeated computation in *space* or *time*
- comparison of $k + 1$ results to:
  - → detect up to k errors
  - → correct up to $\frac{k}{2}$ errors
- can be implemented on gate, component, module, or system level

COMPUTER-AIDED VERIFICATION | NOTIONS

SECURITY



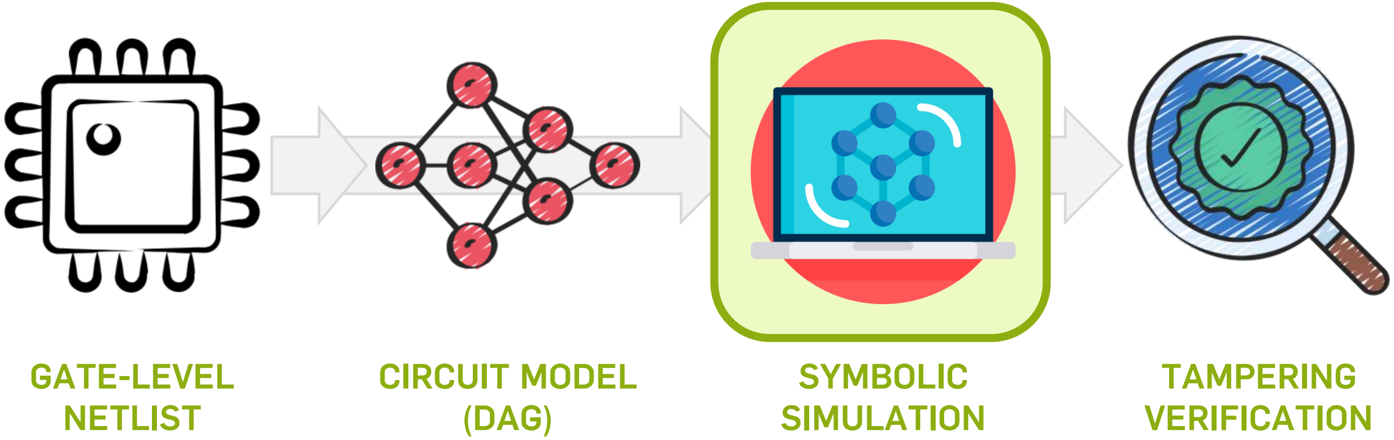
DETECTION



CORRECTION

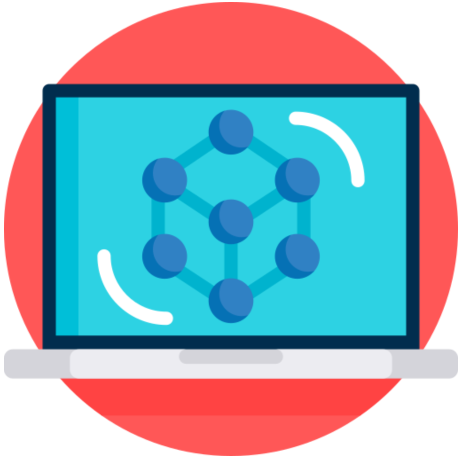
VERIFICATION | APPROACH

SCAN ME



VERIFICATION TOOL | SYMBOLIC SIMULATION

SCAN ME



SYMBOLIC SIMULATION

CONCEPT:

- circuit gates are stored as Binary Decision Diagrams
- Symbolic simulation of golden and faulty circuits
- compute distance function (XOR) of outputs

All detected, effective, and ineffective faults can be computed as satisfying solutions.

VERIFICATION TOOL | FIVER [RBRSS+21]

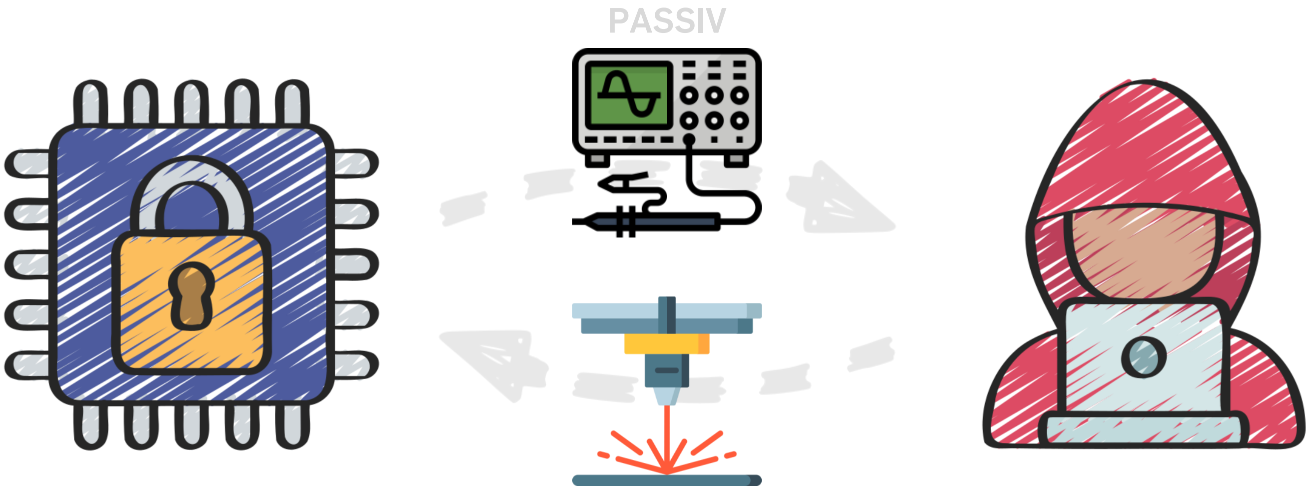
SCAN ME

VERIFICATION OF AN AES-128 ROUND WITH DETECTION (1 FAULT)

COMBINED ANALYSIS

How do we verify security against combined information leakage and tampering?

ADVERSARY MODEL | COMBINED



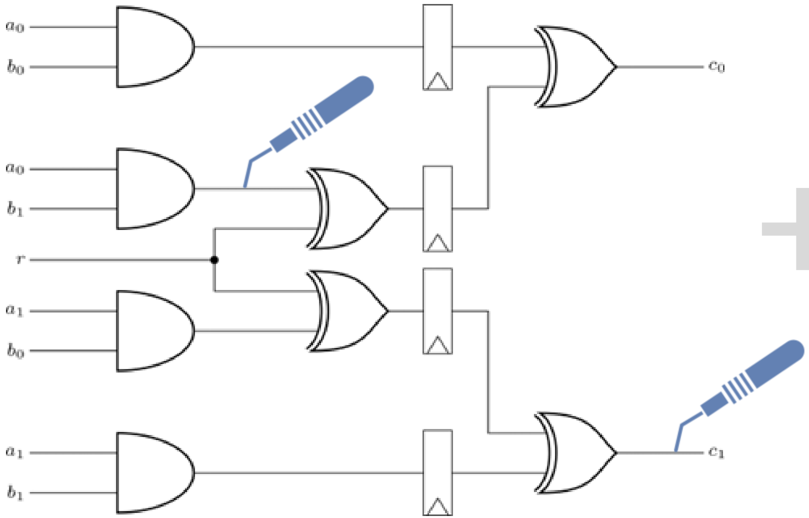
HARDWARE
DEVICE

ATTACKER

RUHR
UNIVERSITÄT
BOCHUM
RUB

ADVERSARY MODEL | MODEL

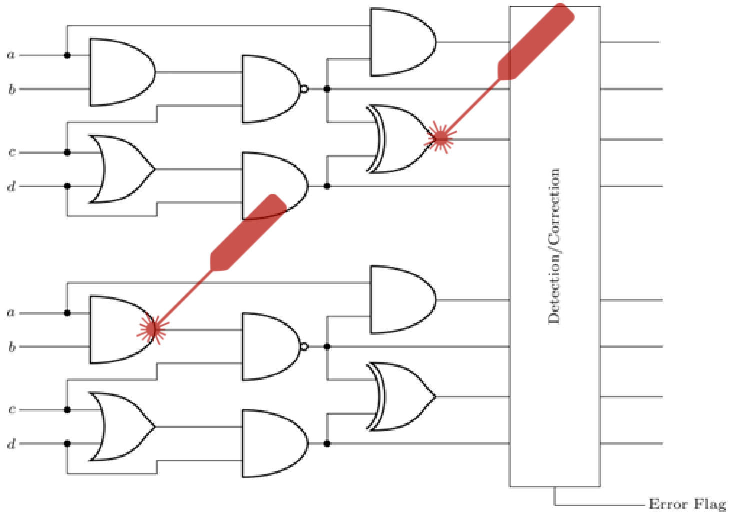
SIDE-CHANNEL ANALYSIS



Threshold t-probing model [ISW03]

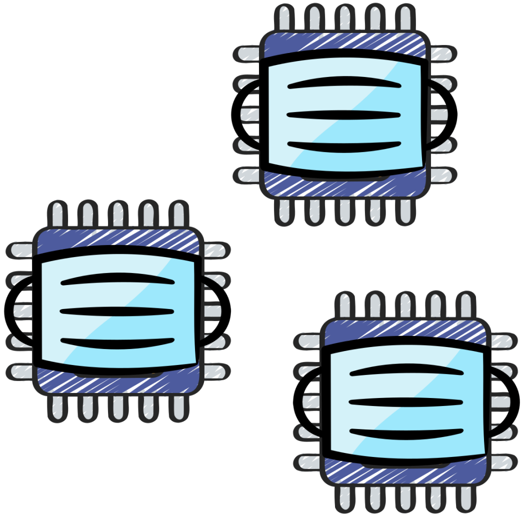
\+

FAULT INJECTION ANALYSIS



Parametrized injection model [RBSG22]

COUNTERMEASURES | MASKED REDUNDANCY



MASKED REDUNDANCY

MASKED REDUNDANCY:

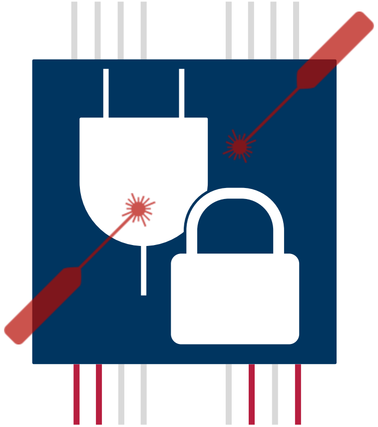
- Boolean sharing, combined with
- Redundancy for detected/correction

CHALLENGES:

- distribution and replication of randomness generation (reciprocal effects).
- shared detection/error flags
- signal (= leakage) amplification

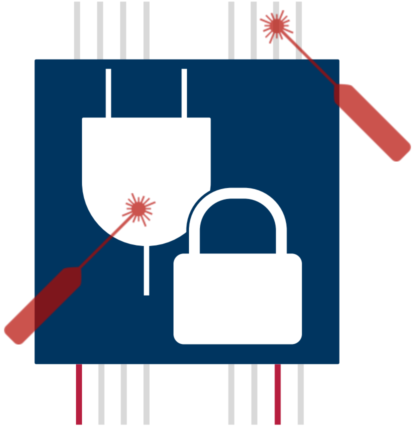
COMPUTER-AIDED VERIFICATION | NOTIONS I/II

COMPOSABILITY



F-NI [DN20]
FAULT
NON-INTERFERENCE

$k' \leq k$



F-SNI [DN20]
FAULT STRONG
NON-INTERFERENCE

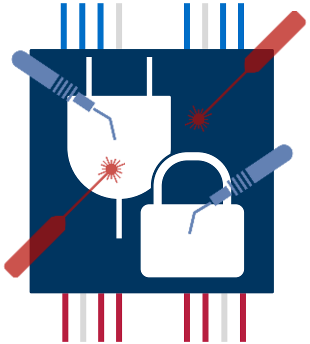
$k_1 + k_2 \leq k$

F-INI [FFRBS+22]
FAULT-ISOLATING
NON-INTERFERENCE

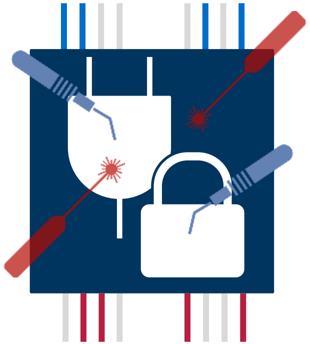
COMPUTER-AIDED VERIFICATION | NOTIONS II/II

SECURITY

COMPOSABILITY



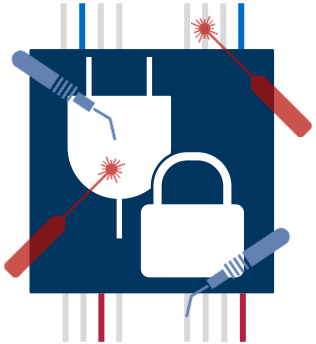
COMB. [RBFS+22]
COMBINED SECURITY



C-NI [DN20]
COMBINED NON-INTERFERENCE

$d' + k_1 + k_2 \le d$

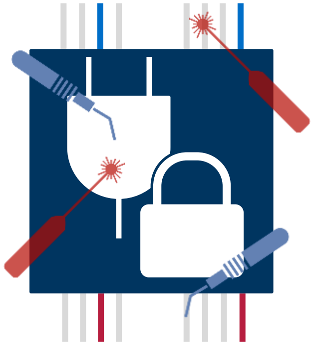
$k_1 + k_2 \le k$



C-SNI [DN20]
COMBINED STRONG NON-INTERFERENCE

$d_1 + d_2 + k_1 + k_2 \le d$

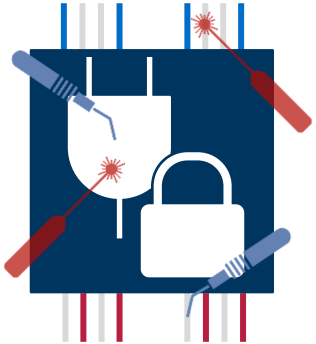
$k_1 + k_2 \le k$



C-SNI$_{ind}$ [DN20]
INDEPENDENT COMBINED STRONG NON-INTERFERENCE

$d_1 + d_2 \le d$

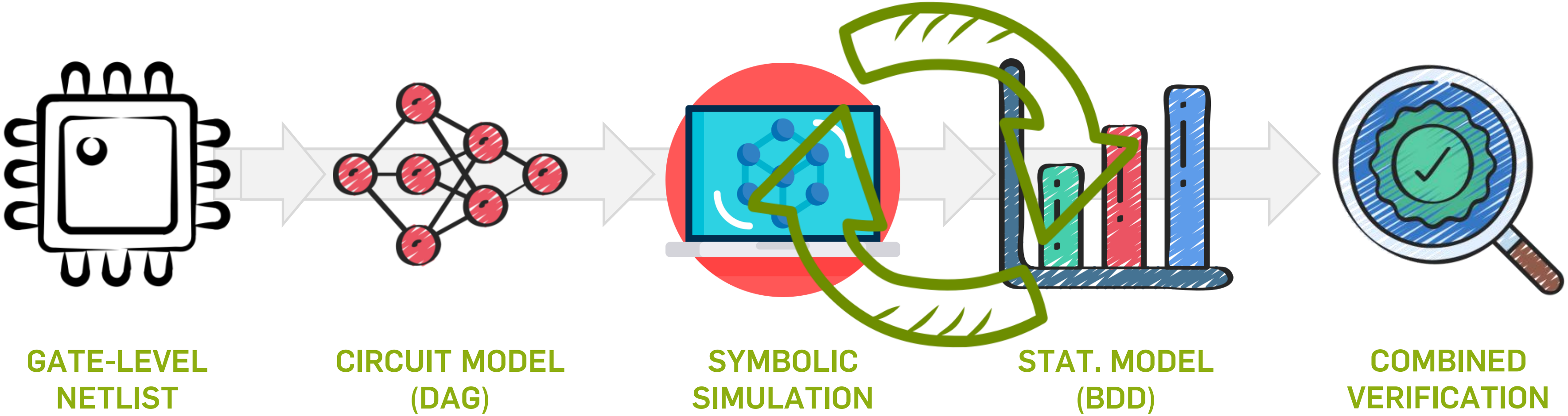
$k_1 + k_2 \le k$



C-INI [FFRBS+22]
COMBINED-ISOLATING NON-INTERFERENCE

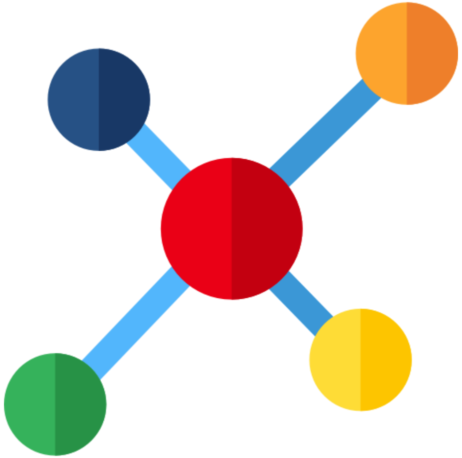
VERIFICATION TOOL | VERICA

SCAN ME



VERIFICATION | RECIPROCAL EFFECTS

SCAN ME



RECIPROCAL EFFECTS

PROBLEMS:

Fault propagation and misbehavior in shared circuits.

1. Faults injected into generated randomness:
   → effective faults but functionally correct behavior
2. Multiple valid sharings for same secret:
   → localization of faulty shares is hard

Reciprocal effects require adjusted definition for the golden (fault-free) shared circuit.

VERIFICATION TOOL | VERICA [RBFS+22]

```
pascal@ubuntu-VM:~/projects/verica$ 
```

SCAN ME

CONCLUSION

Your free takeaway for today :)

CONCLUSION

COMPUTER-AIDED HARDWARE SECURITY VERIFICATION

INFORMATION LEAKAGE
SIDE-CHANNEL ANALYSIS

COMBINATION
COMBINED ANALYSIS

INFORMATION TAMPERING
FAULT INJECTION ANALYSIS

CIRCUITS
SECURITY

GADGETS
COMPOSABILITY

FURTHER DETAILS

Source code, documentation, contact details, references

FURTHER DETAILS | TOOLS

SILVER

https://github.com/Chair-for-Security-Engineering/SILVER
https://eprint.iacr.org/2020/634.pdf
pascal.sasdrich@rub.de

SIDE-CHANNEL ANALYSIS

FIVER

https://github.com/Chair-for-Security-Engineering/FIVER
https://eprint.iacr.org/2021/936.pdf
jan.richter-brockmann@rub.de / pascal.sasdrich@rub.de

FAULT-INJECTION ANALYSIS

VERICA

https://github.com/Chair-for-Security-Engineering/VERICA
https://eprint.iacr.org/2022/484.pdf
jan.richter-brockmann@rub.de / pascal.sasdrich@rub.de

COMBINED ANALYSIS

FURTHER DETAILS | REFERENCES

[ISW03] *Yuval Ishai, Amit Sahai, David A. Wagner: Private Circuits: Securing Hardware against Probing Attacks (CRYPTO 2003).*

[FGMDP+18] *Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, François-Xavier Standaert: Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model (CHES 2018).*

[CS20] *Gaëtan Cassiers and François-Xavier Standaert: Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference (IEEE TIFS 2020).*

[DN20] *Siemen Dhooghe and Svetla Nikova: My Gadget Just Cares for Me – How NINA Can Prove Security Against Combined Attacks (CT-RSA 2020).*

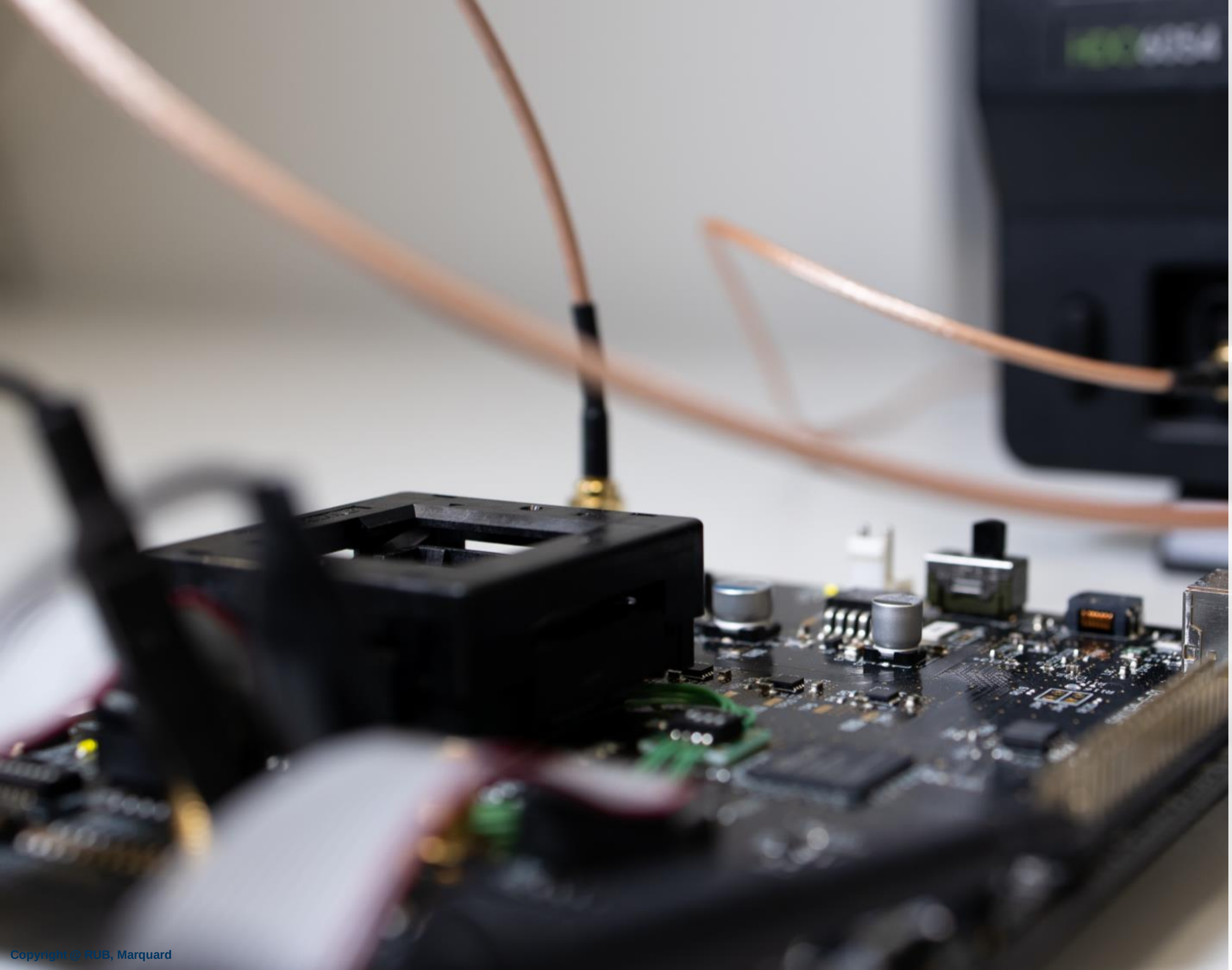
[KSM20] *David Knichel, Pascal Sasdrich, Amir Moradi: SILVER – Statistical Independence and Leakage Verification (ASIACRYPT 2020).*

[RBRSS+21] *Jan Richter-Brockmann, Aein Rezaei Shahmirzadi, Pascal Sasdrich, Amir Moradi, Tim Güneysu: FIVER – Robust Verification of Countermeasures against Fault Injections (CHES 2021).*

[RBFS+22] *Jan Richter-Brockmann, Jakob Feldtkeller, Pascal Sasdrich, Tim Güneysu: VERICA – Verification of Combined Attacks: Automated Formal Verification of Security against Simultaneous Information Leakage and Tampering (CHES 2022).*

[RBSG22] *Jan Richter-Brockmann, Pascal Sasdrich, Tim Güneysu: Revisiting Fault Adversary Models – Hardware Faults in Theory and Practice (IEEE TC 2022).*

[FFRBS+22] *Jakob Feldtkeller, Jan Richter-Brockmann, Pascal Sasdrich, Tim Güneysu: CINI MINIS: Domain Isolation for Fault and Combined Security (ACM CCS 2022).*



RUB

Thank you!

Any questions?

pascal.sasdrich@rub.de

Chair for Security Engineering
Faculty for Computer Science
Ruhr University Bochum