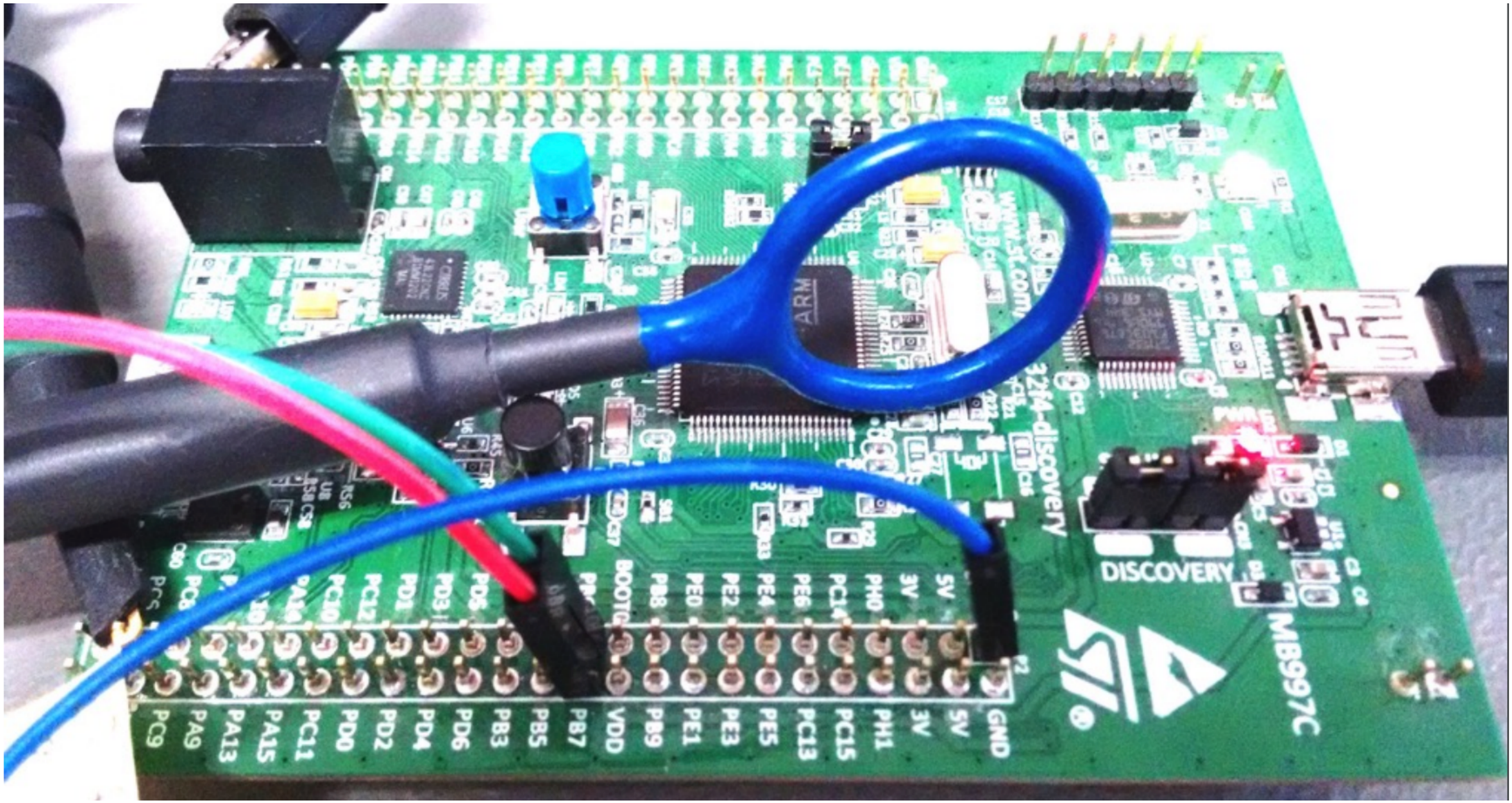
CAPA: the spirit of Beaver against physical attacks

Oscar Reparaz, Lauren De Meyer, Victor Arribas,
Begul Bilgin, Svetla Nikova, Venzi Nikov, Nigel Smart

COSIC KU Leuven
University of Bristol
NXP Semiconductors

Problem statement

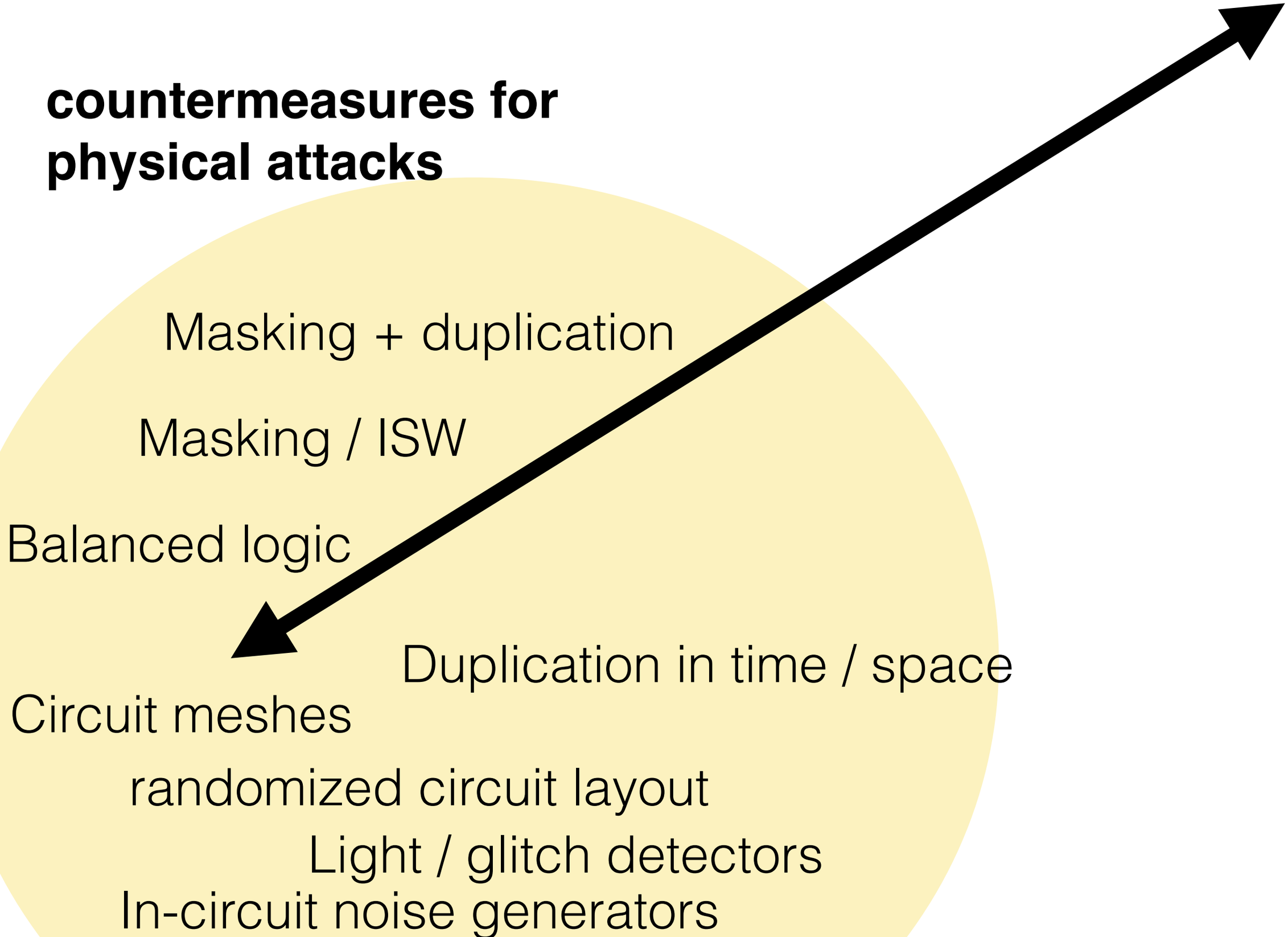


(Johann Heyszl)

Problem statement

- Implementation of crypto in a hostile environment
- **This paper**: adapt MPC protocols to run in hardware

countermeasures for physical attacks



countermeasures for physical attacks

FHE?

SPDZ

BODZ

MASCOT

Tiny-OT

MPC

Masking + duplication

Masking / ISW

Balanced logic

Duplication in time / space

Circuit meshes

randomized circuit layout

Light / glitch detectors

In-circuit noise generators

FHE?

countermeasures for physical attacks

SPDZ
BODZ
MASCOT
Tiny-OT

MPC

Masking + duplication

Masking / ISW

Balanced logic

Duplication in time / space

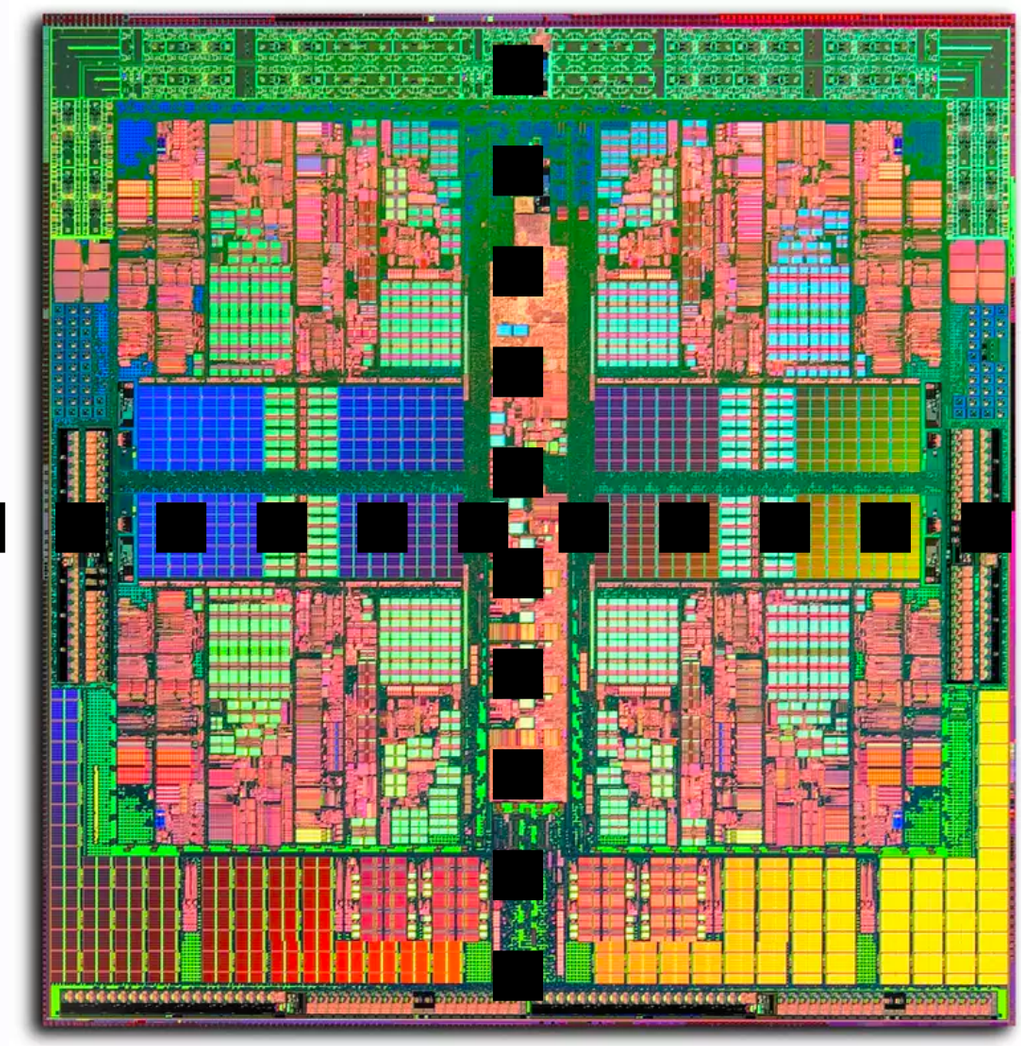
Circuit meshes

randomized circuit layout

Light / glitch detectors

In-circuit noise generators

Adversarial model: **tile** fault-and-probe



Tile IV

Tile I

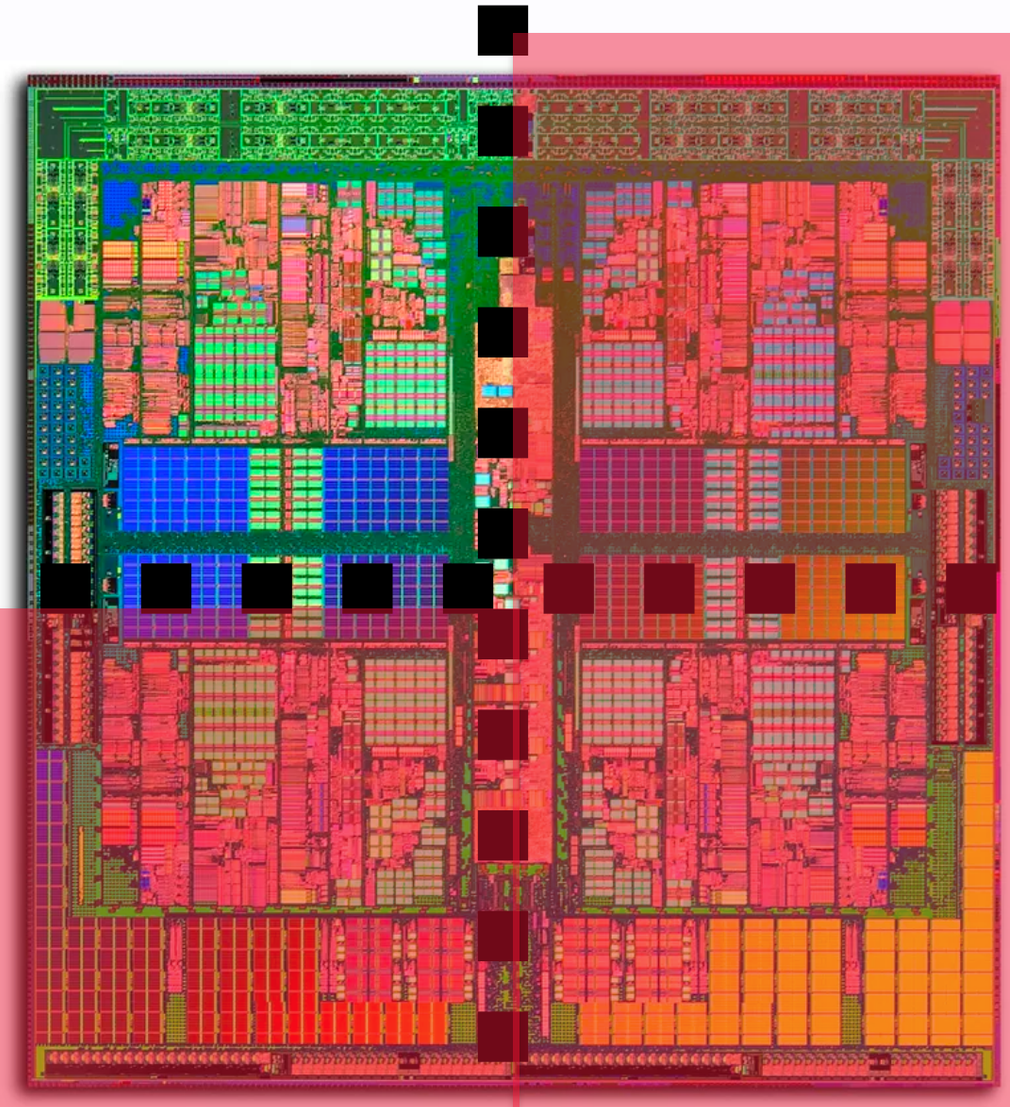
Tile III

Tile II

Tile ≈ party

8

Adversarial model: **tile** fault-and-probe



Tile IV

Tile I

Tile III

Tile II

Adversarially controlled

Tile ≈ party

Adversarial model: SCA

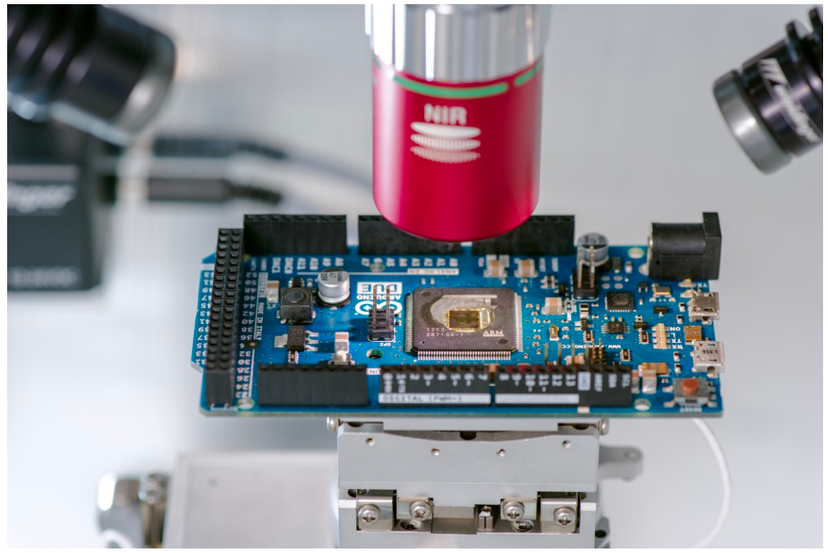
- Adversary is allowed to probe **all** intermediates **within** a set of **tiles** (all except one). Values are disclosed with probability 1

- Related to the **noisy leakage model**

Adversarial model: FA

A. known value fault in any intermediate within up to (d-1)-tiles

≈



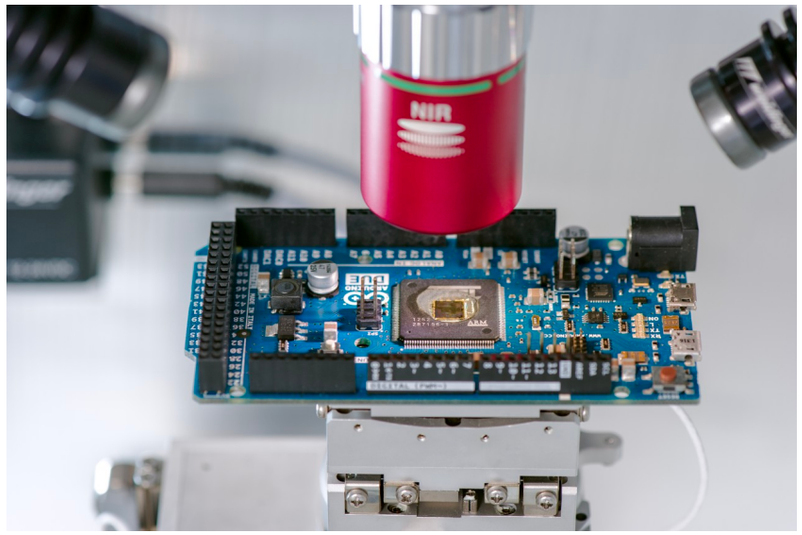
- powerful, inherited by SPDZ

Adversarial model: FA

A. known value fault in any intermediate within up to (d-1)-tiles

- powerful, inherited by SPDZ

≈



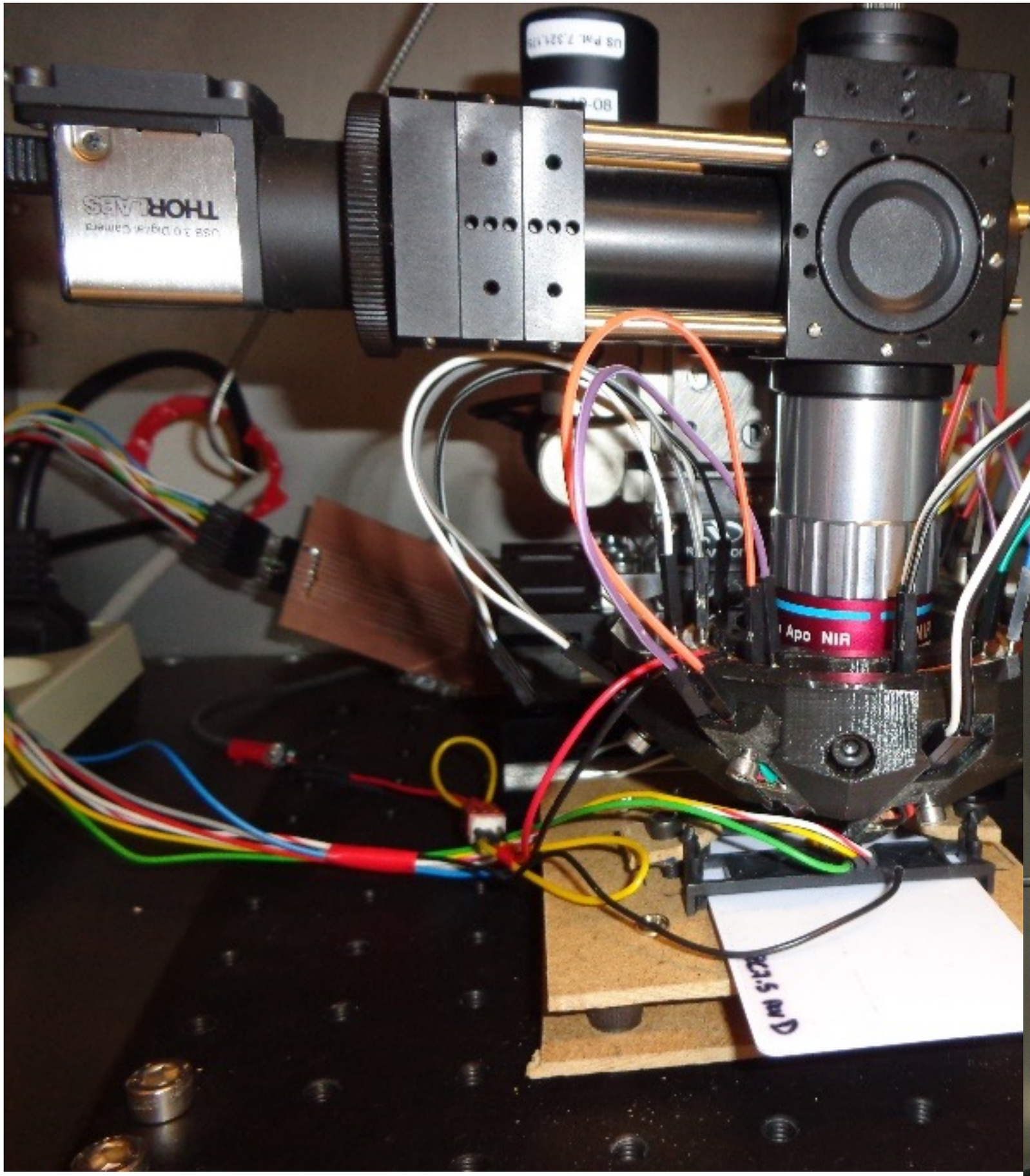
B. random fault everywhere

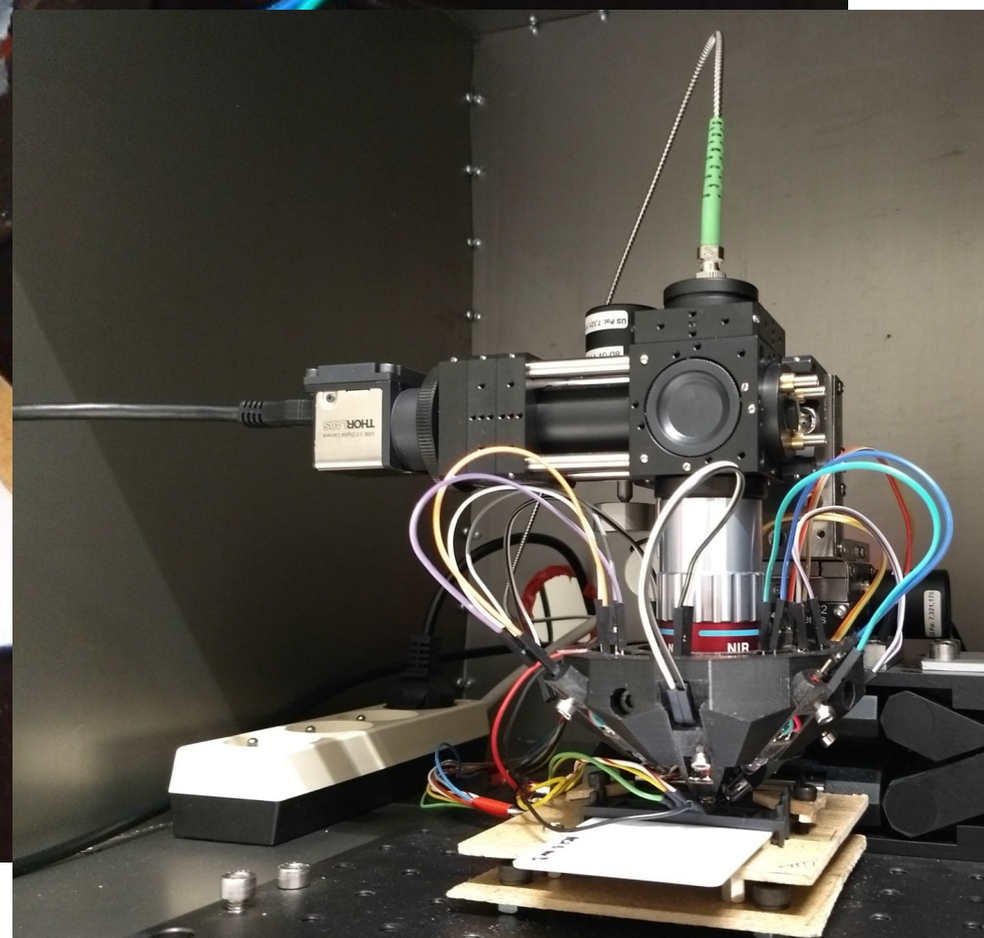
- very relevant for HW

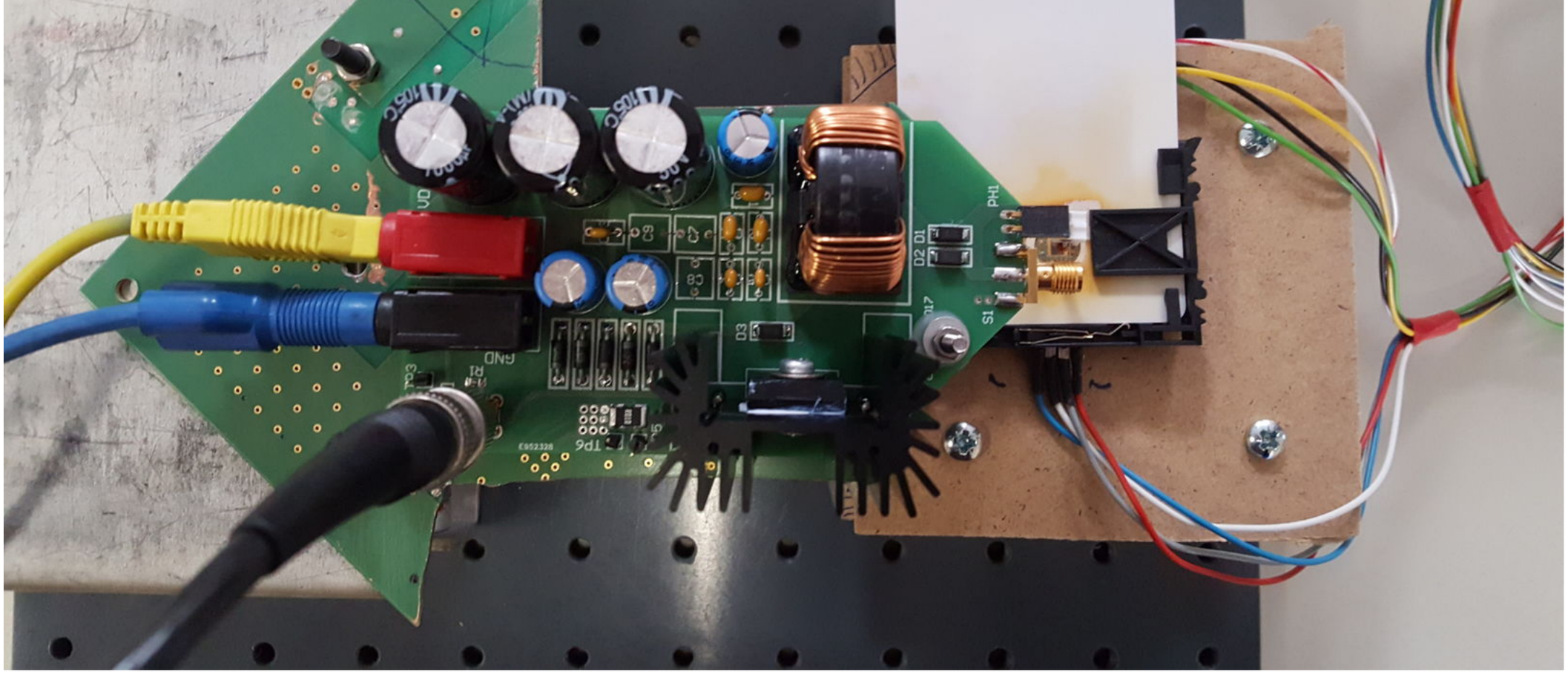
- There is fine print: static adversary. notion of time: computation periods

≈

(J-M Schmidt, M. Hutter)


THORLABS
Apo NIR


THORLABS
NIR


GND
TP3
TP6
D3
D2 D1
PH1
S1
50 v
1000µ

Current countermeasures

- Orthogonal topics: side-channel protection + fault protection

  - A few **combined** attacks (more difficult)

Different worlds - analogies and differences

Party	Tile in the silicon
expensive communication channel	wires on the circuit
local memory cheap	reduced storage
adversary controls arbitrarily some parties, can plot arbitrary attacks	adversary external, controls *somehow* some parties, DFA mostly (bit flips, set, clear)

CAPA

- How to represent data

- How to perform computation

CAPA: data representation

Main idea: attach an info-theoretical MAC to each piece of data

CAPA: data representation

Main idea: attach an info-theoretical MAC to each piece of data

- Handle (shares of data, shares of MAC tag)

$$\langle \boldsymbol{a} \rangle = (\boldsymbol{a}, \boldsymbol{\tau}^{\boldsymbol{a}})$$

  - shares of data = additive secret sharing

$$\boldsymbol{a} = (a_1, \ldots, a_d) \qquad \sum a_i = a$$

CAPA: data representation

Main idea: attach an info-theoretical MAC to each piece of data

- Handle (shares of data, shares of MAC tag)

$$\langle \boldsymbol{a} \rangle = (\boldsymbol{a}, \boldsymbol{\tau}^{\boldsymbol{a}})$$

  - shares of data = additive shares of data

$$\boldsymbol{a} = (a_1, \ldots, a_d) \qquad \sum a_i = a$$

- MAC tag: multiplicative tag

$$\tau^a = \alpha \cdot a$$

CAPA: data representation

Main idea: attach an info-theoretical MAC to each piece of data

- Handle (shares of data, shares of MAC tag)

$$\langle \boldsymbol{a} \rangle = (\boldsymbol{a}, \boldsymbol{\tau}^{\boldsymbol{a}})$$

  - shares of data = additive shares of data

$$\boldsymbol{a} = (a_1, \ldots, a_d) \qquad \sum a_i = a$$

- MAC tag: multiplicative tag

$$\tau^a = \alpha \cdot a$$

- shares of MAC tag: additive shares of the tag

$$\boldsymbol{\tau}^{\boldsymbol{a}} = (\tau_1^a, \ldots, \tau_d^a) \quad \sum \tau_i^a = \tau^a \quad \alpha = \sum \alpha_i$$

CAPA: computation

- **Linear operations are easy**

CAPA: computation

- Linear operations are easy
- **Multiplication**

Inputs (x, τ^x) (y, τ^y)

Auxiliary data $(\langle a \rangle, \langle b \rangle, \langle c \rangle)$

$c = a \cdot b$

A. Blinding

B. Partial unmasking

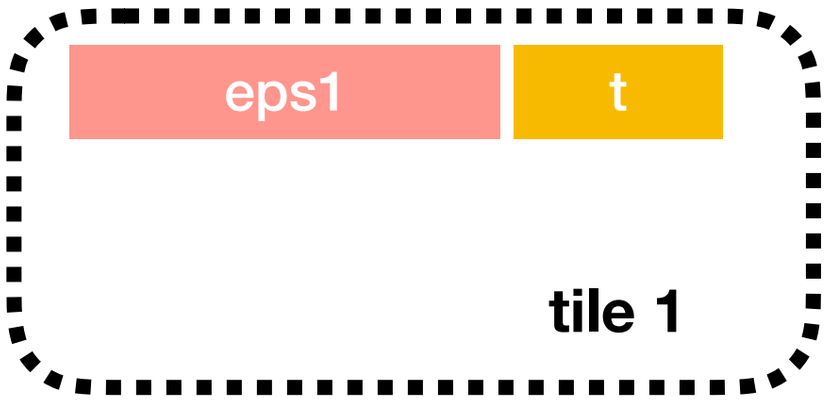
C. MAC tag checking

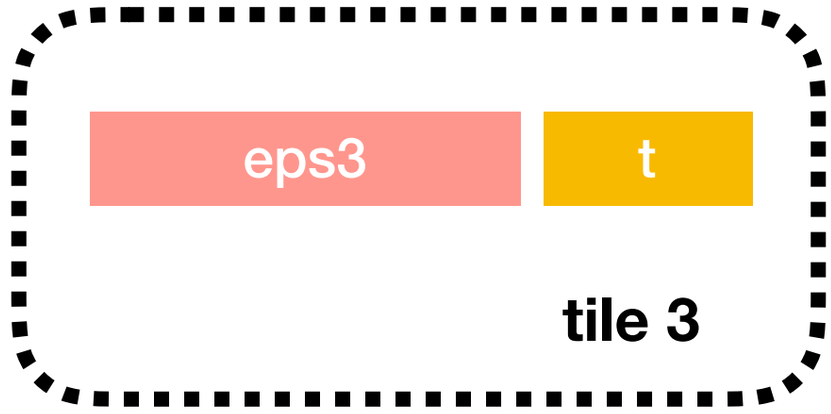
D. Beaver step

CAPA: computation

A. Blinding

$$\varepsilon_i = x_i + a_i$$
$$\eta_i = y_i + b_i$$





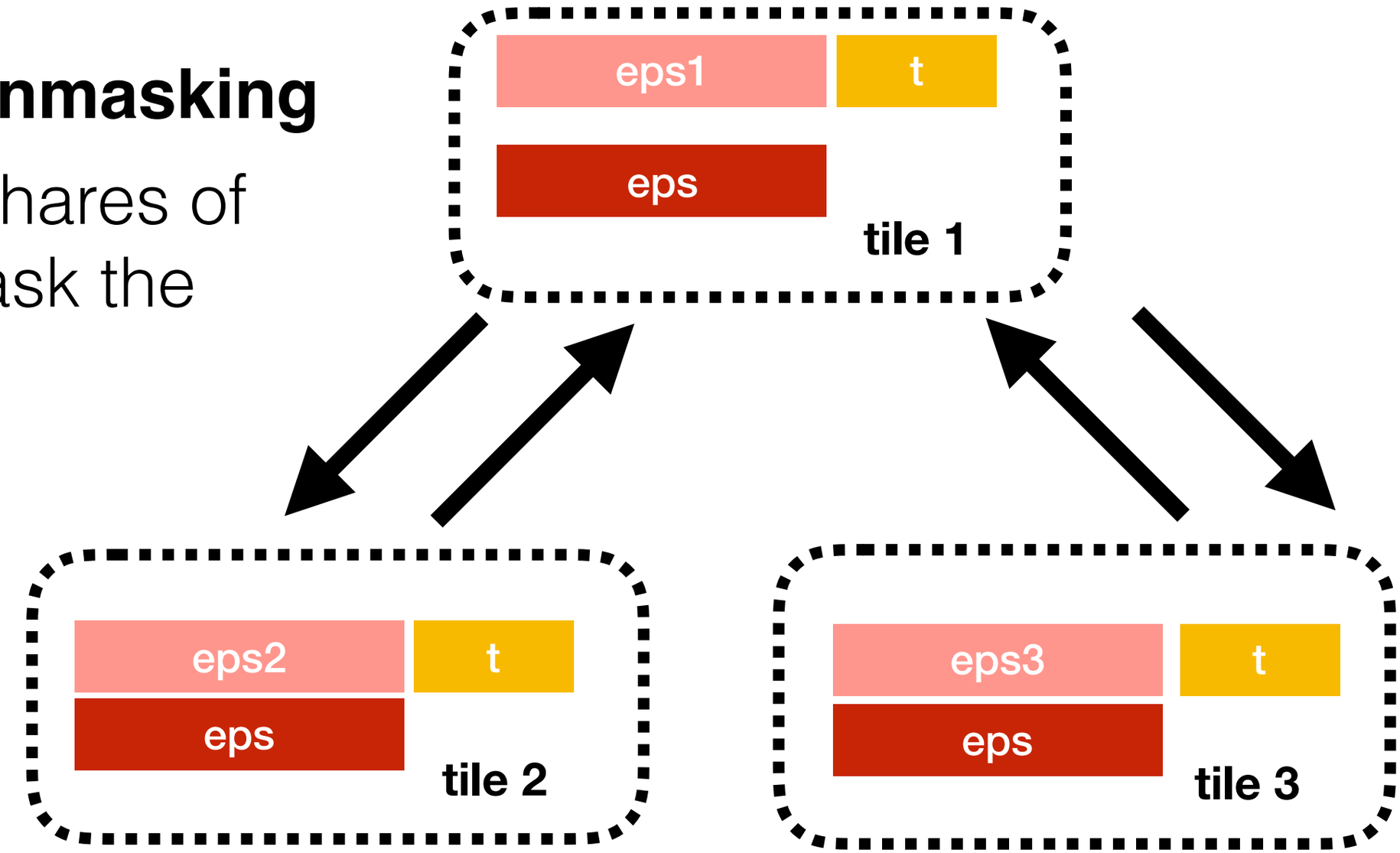
CAPA: computation

B. Partial unmasking

Broadcast shares of eps to unmask the **value**

$$\varepsilon = \sum \varepsilon_i$$

$$\eta = \sum \eta_i$$



each broadcasting needs a synchronization element

CAPA: computation

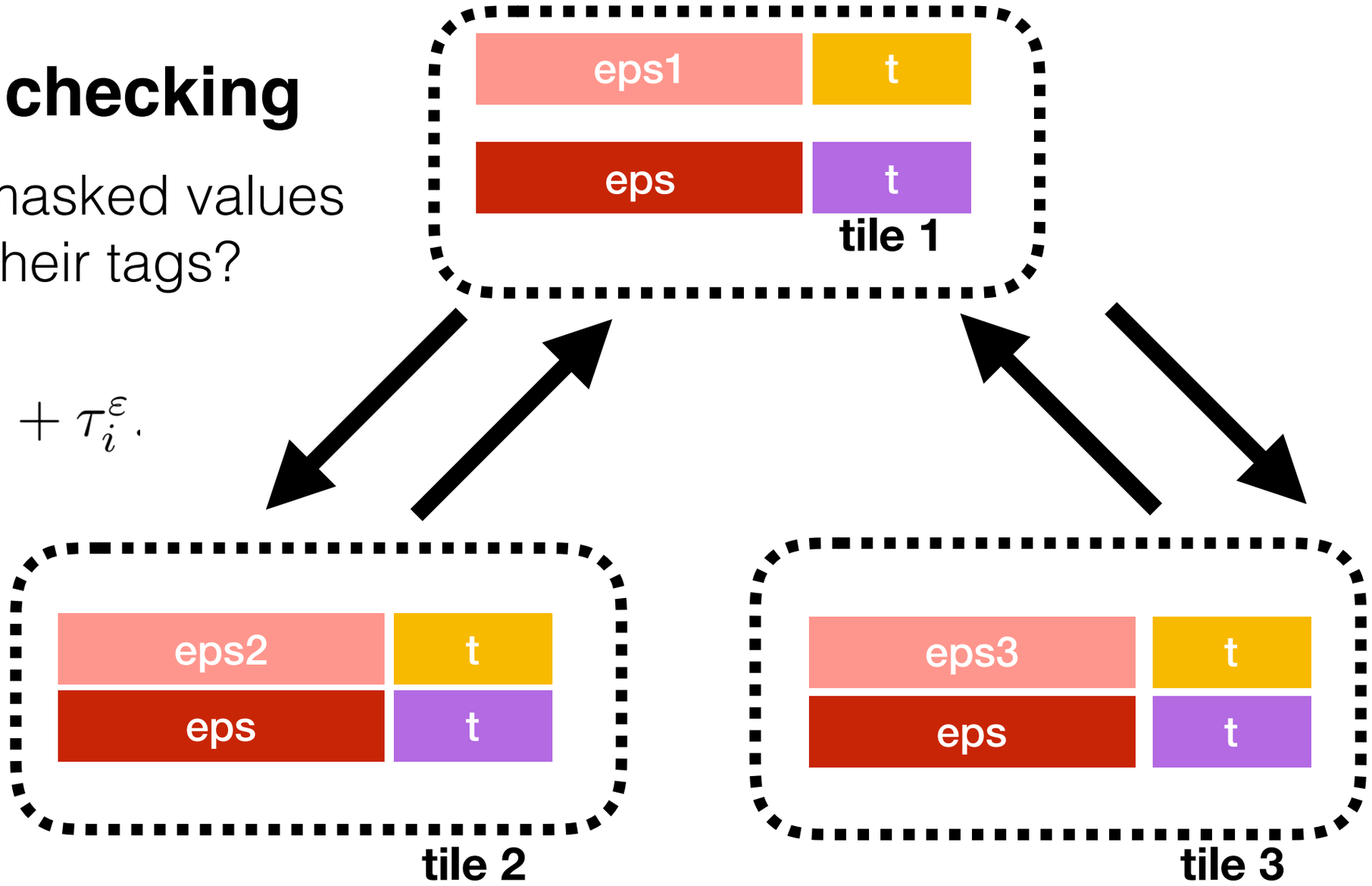
C. MAC tag checking

Are partially unmasked values consistent with their tags?

broadcast $\varepsilon \cdot \alpha_i + \tau_i^{\varepsilon}$.

verify is zero

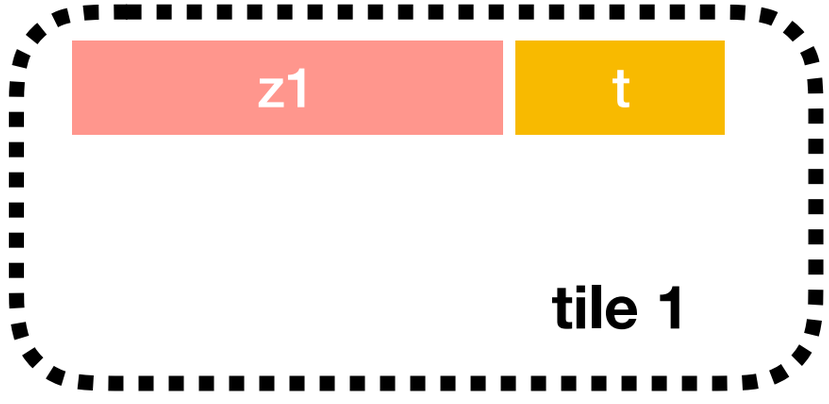
$$\sum(\varepsilon \cdot \alpha_i + \tau_i^{\varepsilon})$$

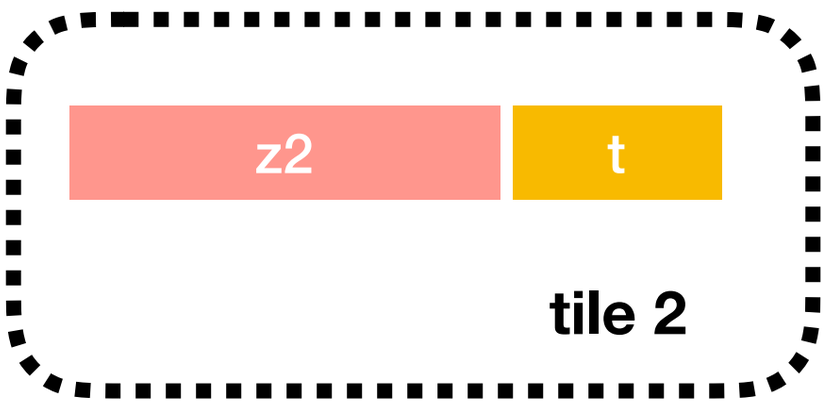


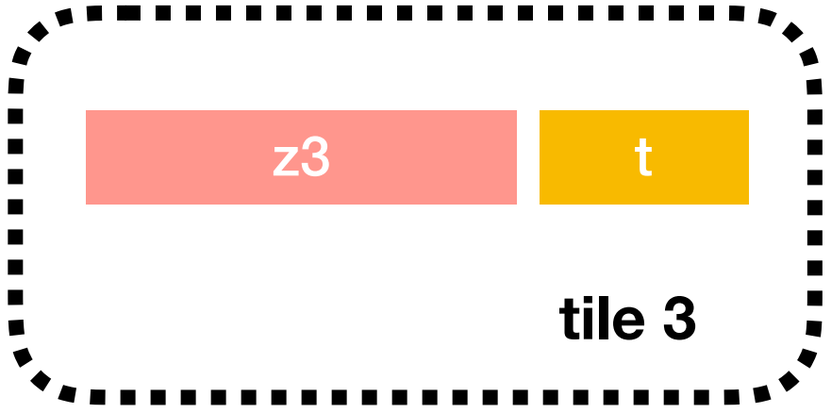
CAPA: computation

D. Beaver computation

The actual multiplication (local)







$$z_i = c_i + \varepsilon \cdot b_i + \eta \cdot a_i + \varepsilon \cdot \eta \cdot \delta_{i,1}$$

$$\tau_i^z = \tau_i^c + \varepsilon \cdot \tau_i^b + \eta \cdot \tau_i^a + \varepsilon \cdot \eta \cdot \alpha_i$$

CAPA: PRE computation

- Auxiliary data needed for multiplication

$(\langle \boldsymbol{a} \rangle, \langle \boldsymbol{b} \rangle, \langle \boldsymbol{c} \rangle) \qquad c = a \cdot b$

- Generate using a passively secure multiplier

- Relation verification step

Security guarantees

- Side-channels: the union of d-1 tiles doesn't disclose any secret -> **(d-1)-order DPA attacks**

- Fault attacks: the fault is undetected if both value and accompanying tag are modified to be consistent.
  Probability that an adversary controlling d-1 tiles is bounded -> **(d-1)-shot FA**

  - Detection probability does not depend on the number of faulty bits or Hamming weight of injected faults

- Combined adversary: inherit from MPC. Not all combined adversaries are covered (we're not using commitments)

Some attacks

- Glitch on power supply or clock line
    - Depends on the underlying HW architecture
- Skipping instructions
    - Detected when checking partiaully unmasked values
- Flipping values
- Safe error attacks

Implementations: AES in HW

$$\text{S-box}(x) = \texttt{0x63} + \texttt{0x8F} \cdot x^{127} + \texttt{0xB5} \cdot x^{191} + \texttt{0x01} \cdot x^{223} + \texttt{0xF4} \cdot x^{239} + \texttt{0x25} \cdot x^{247} + \texttt{0xF9} \cdot x^{251} + \texttt{0x09} \cdot x^{253} + \texttt{0x05} \cdot x^{254}$$

$$x^{254} = x^4 \cdot \left(\left((x^5)^5\right)^5\right)^2$$

Primitives: x^5 $x^4 \cdot y^2$

Inversion: 4 cycles, 3 exponentiation triples and 1 quintuple

Affine: 1 cycle. Total 5-stage pipeline

Implementations: AES in HW

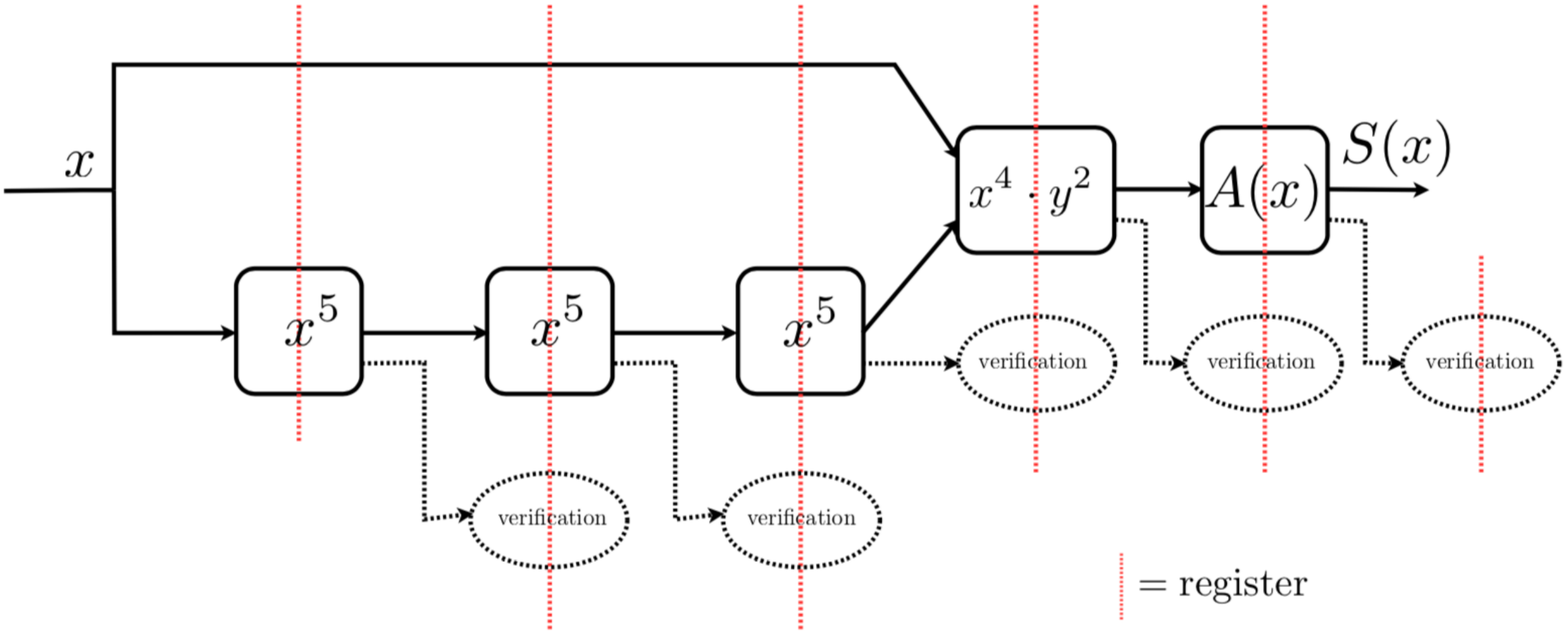


Table 4. Areas for first- and second-order AES implementations with $m = 1$ in 2-NAND Gate Equivalents (GE)

Evaluation	$d = 2$	$d = 3$	Preprocessing	$d = 2$	$d = 3$
S-box	18 810	28 234	Quintuples	29 147	53 212
* Beaver x^5 (x3)	3 914	5 875	* Generation	15 092	32 241
* Beaver x^4y^2	4 944	7 427	* Sacrificing	14 055	20 971
* Beaver Affine	1 563	2 344	Triples (x3)	19 106	34 954
State array	4 962	7 466	* Generation	9 804	21 112
* MixColumns	1 056	1 584	* Sacrificing	9 302	13 842
Key array	3 225	4 835	Affine tuples	7 603	14 657
Others	1 296	1 839	* Generation	4 821	10 444
			* Sacrificing	2 782	4 213
Total	28 293	42 374	Total	94 068	172 731
TOTAL				122 361	215 105

Table 5. The number of randomness in bytes for the initial sharing, shared multiplication and the sacrifice required for AES S-box

	Initial sharing	Shared mult.	Total
Exp. triple	d	$1 + 3m$	$2(d + (1 + 3m)\frac{d(d-1)}{2})$
Quintuple	$2d$	$1 + 5m$	$2(2d + (1 + 5m)\frac{d(d-1)}{2})$
Affine tuple	d	$2m$	$2(d + 2m\frac{d(d-1)}{2})$
Total			$12d + 2(4 + 16m)\frac{d(d-1)}{2}$

KATAN: 2 shares

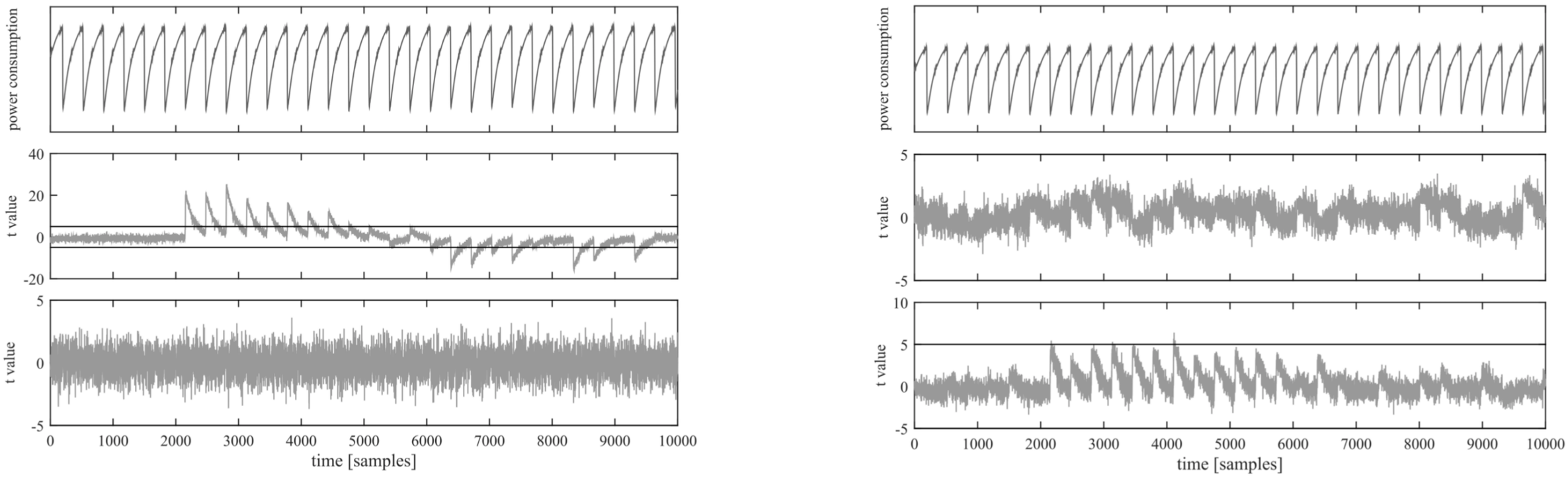


Fig. 2. Non-specific leakage detection on the first 31 rounds of first-order KATAN. Left column: PRNG off (24K traces). Right column: PRNG on (100M traces). Rows (top to bottom): exemplary power trace; first-order t-test; second-order t-test

KATAN: 3 shares

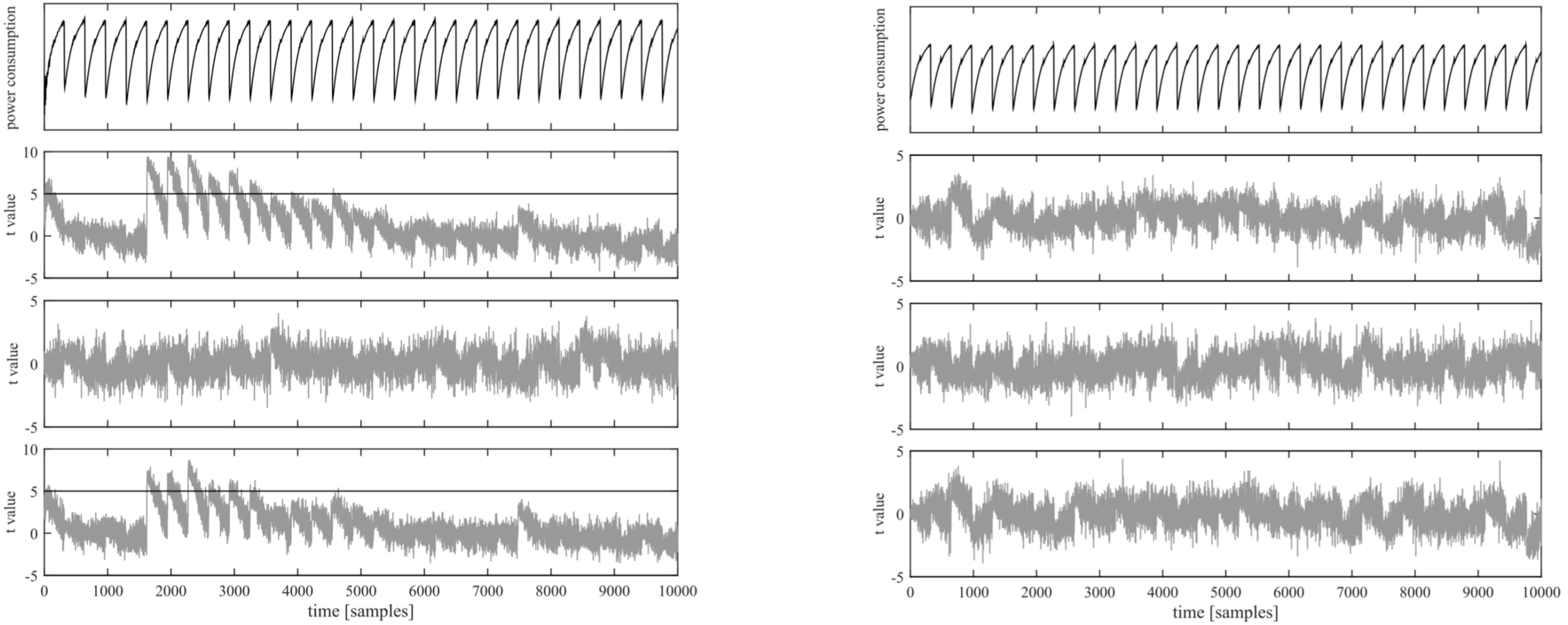
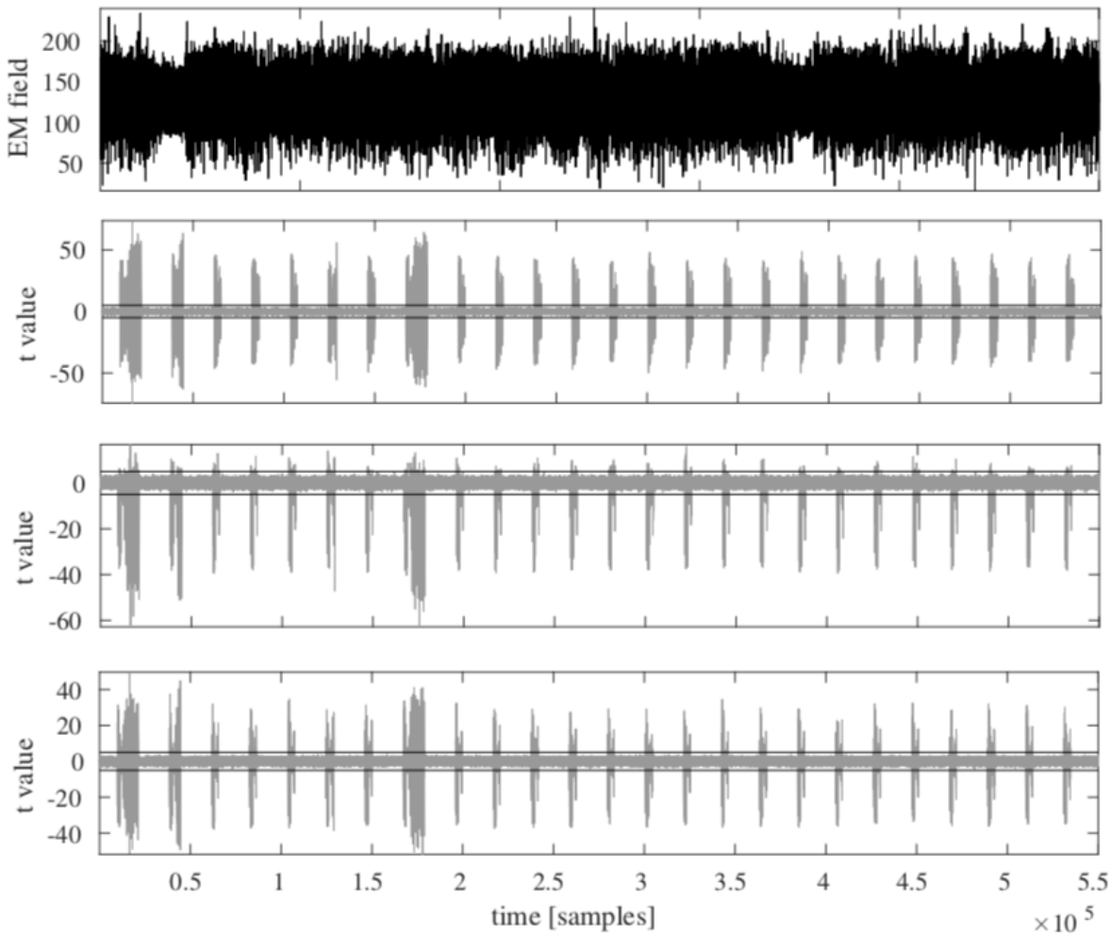
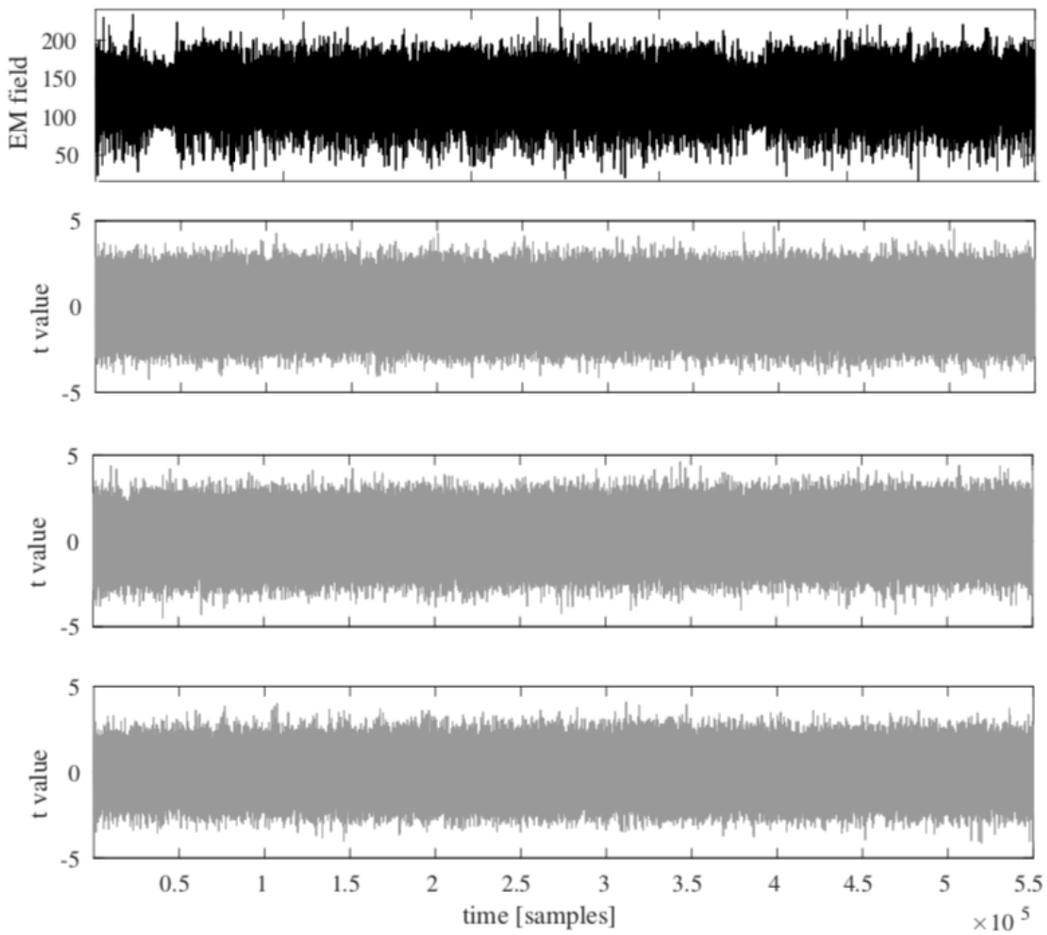


Fig. 3. Non-specific leakage detection on the first 31 rounds of second-order KATAN. Left column: PRNG off (24K traces). Right column: PRNG on (100M traces). Rows (top to bottom): exemplary power trace; first-order t-test; second-order t-test; third-order t-test

Bitsliced AES in SW





Conclusions

- A step towards porting modern MPC to achieve resistance against physical attacks
- Future work
  - Cheaper ways to generate auxiliary data
  - Do not need all machinery of MPC