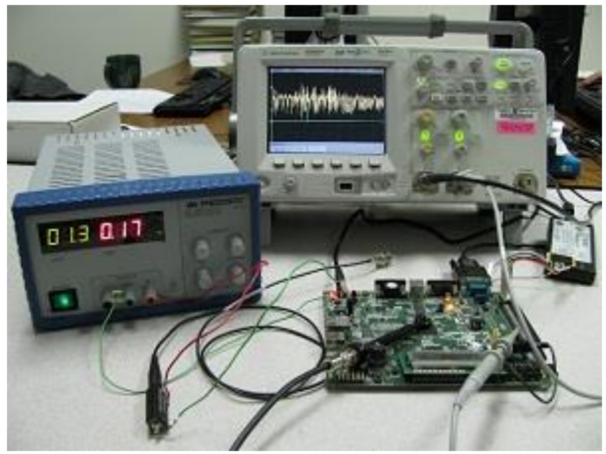
Parallel Implementations of Masking Schemes and the Bounded Moment Leakage Model



G. Barthe, F. Dupressoir, S. Faust,
B. Grégoire, **F.-X. Standaert**, P.-Y. Strub

IMDEA (Spain), Univ. Surrey (UK), Univ. Bochum (Germany), INRIA Sophia-Antipolis (France), UCL (Belgium), Ecole Polytechnique (France)

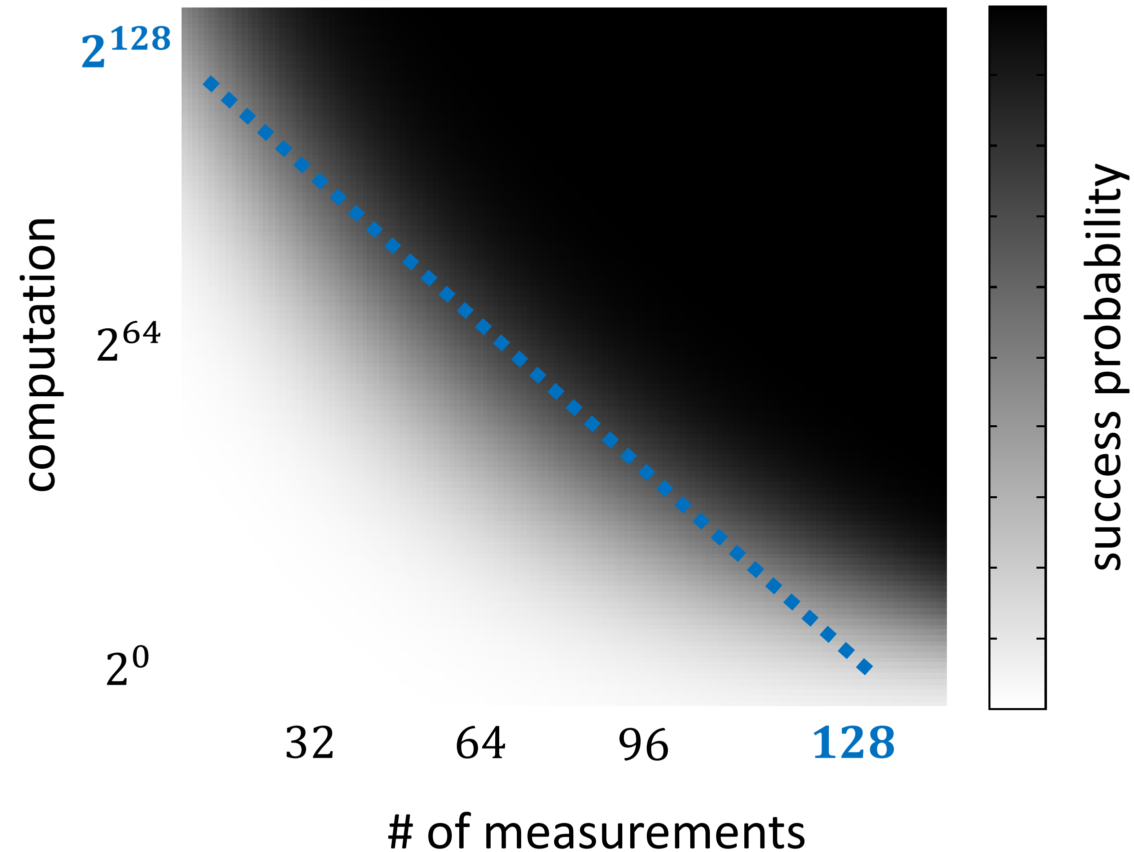
EUROCRYPT 2017, Paris, France

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

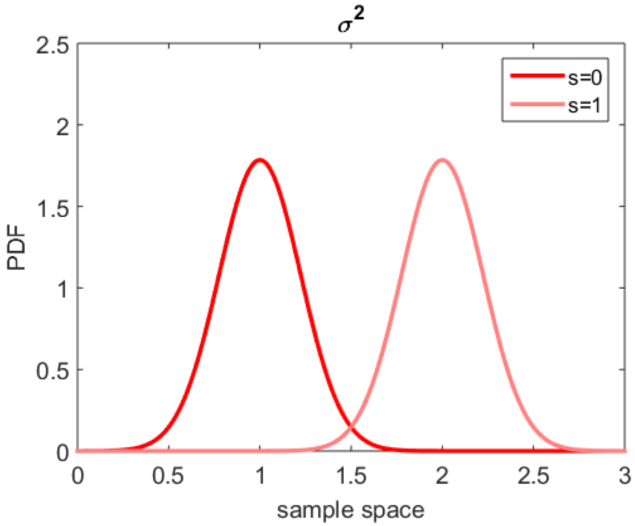
Outline

- Introduction / motivation
    - Side-channel attacks and noise
    - Masking and leakage models
- Bounded moment model
    - Masking intuition & BMM definition
    - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
    - Inner product masking (with “non-mixing” leakages)
    - Continuous security & refreshing gadgets
- Conclusions

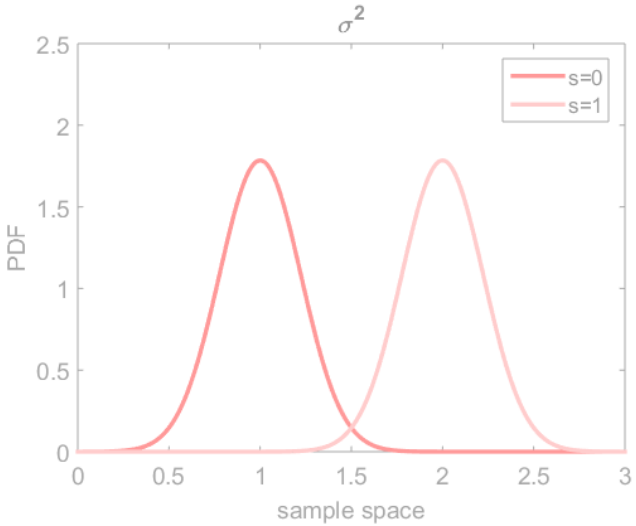


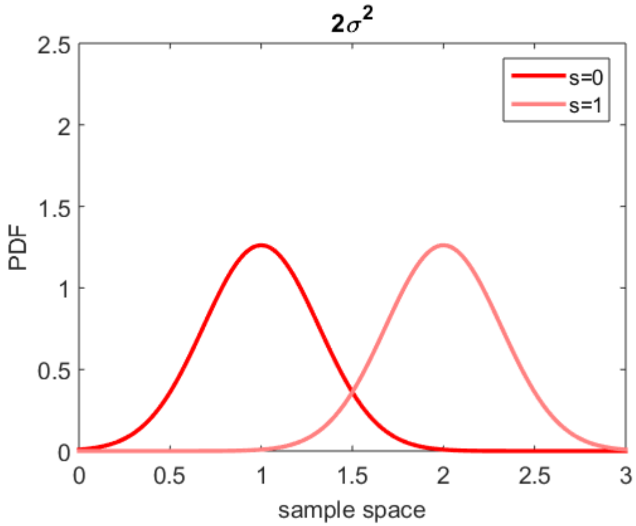
- $\approx$ physical attacks that decreases security exponentially in the # of measurements

Noise (hardware countermeasures)

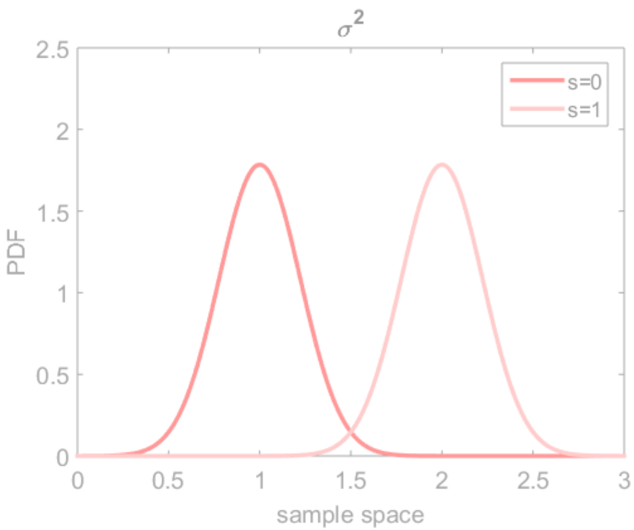


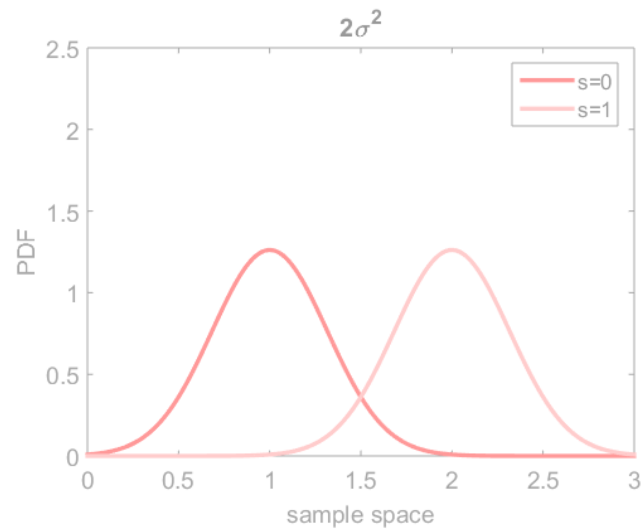
Noise (hardware countermeasures)





Noise (hardware countermeasures) 2





- Additive noise $\approx$ cost $\times$ 2 $\Rightarrow$ security $\times$ 2
  $\Rightarrow$ not a good (crypto) security parameter

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

Masking ($\approx$ noise amplification)

- Example: Boolean encoding

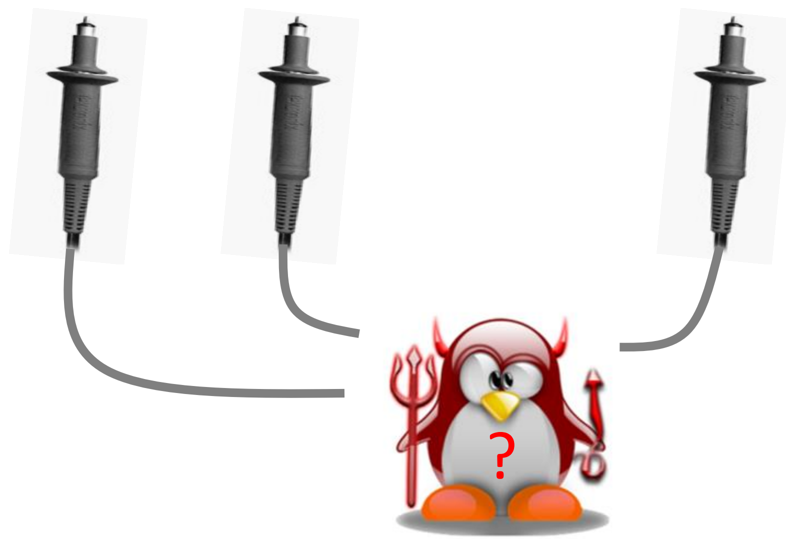
$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$

- With $y_1, y_2, \ldots, y_{d-2}, y_{d-1} \leftarrow \{0,1\}^n$

Masking (abstract view)

- Probing security (Ishai, Sahai, Wagner 2003)

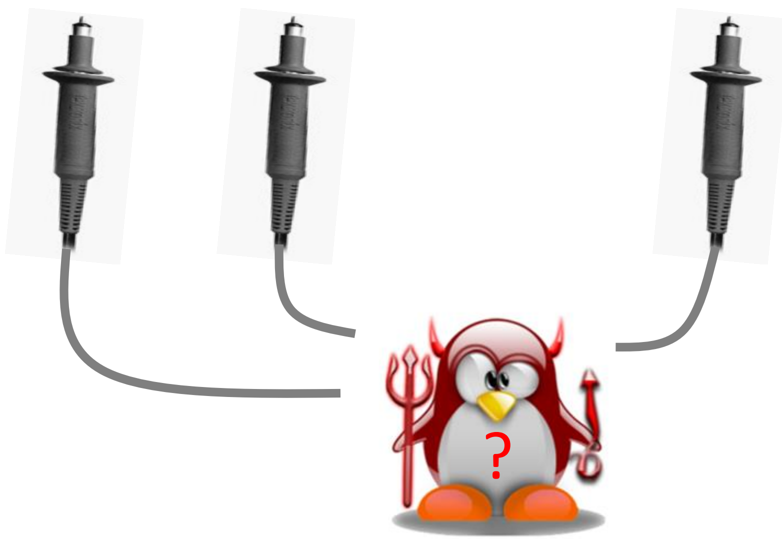
$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$



Masking (abstract view)

- Probing security (Ishai, Sahai, Wagner 2003)

$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$

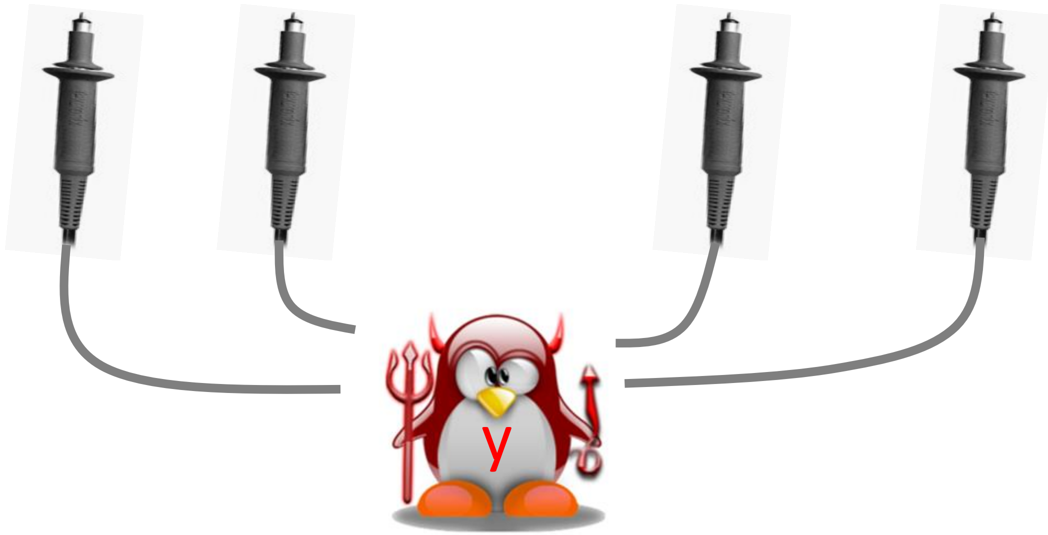


- $d-1$ probes do not reveal anything on y

Masking (abstract view)

- Probing security (Ishai, Sahai, Wagner 2003)

$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$

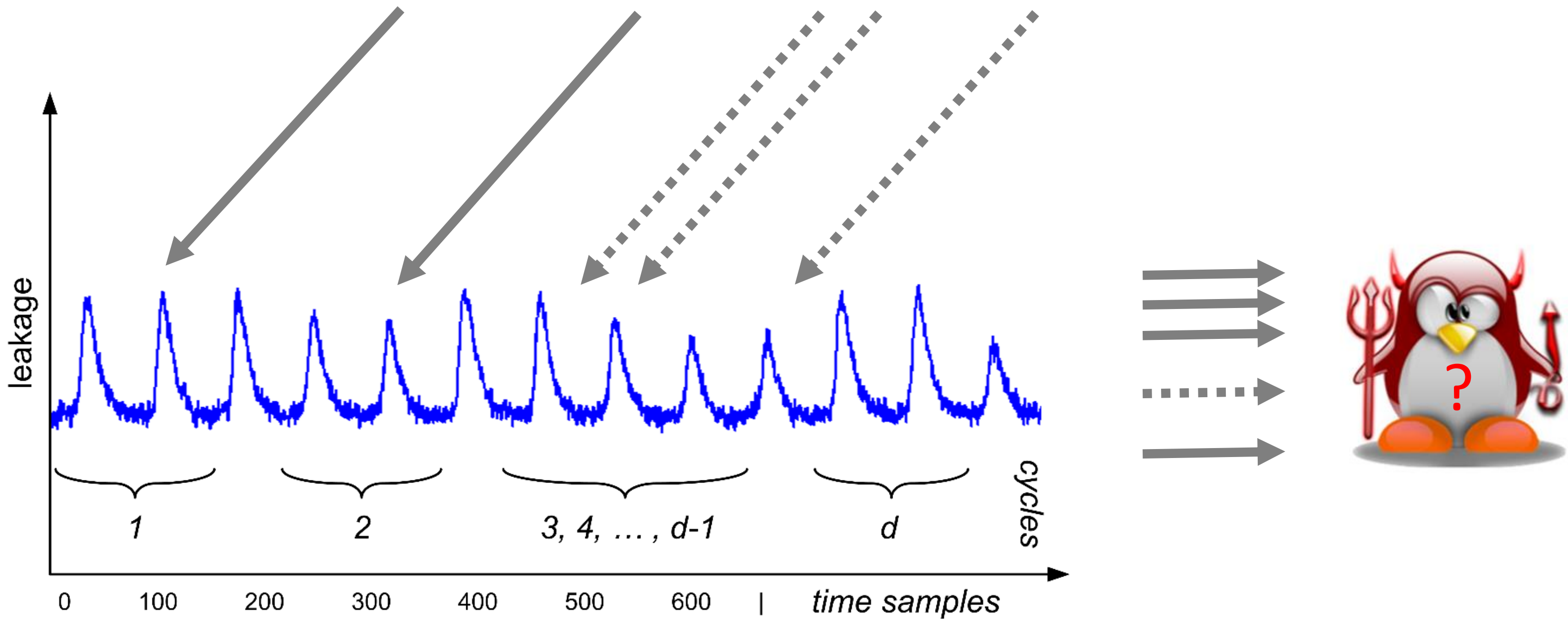


- But d probes completely reveal y

Masking (concrete view)

- Probing security (Ishai, Sahai, Wagner 2003)

$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$



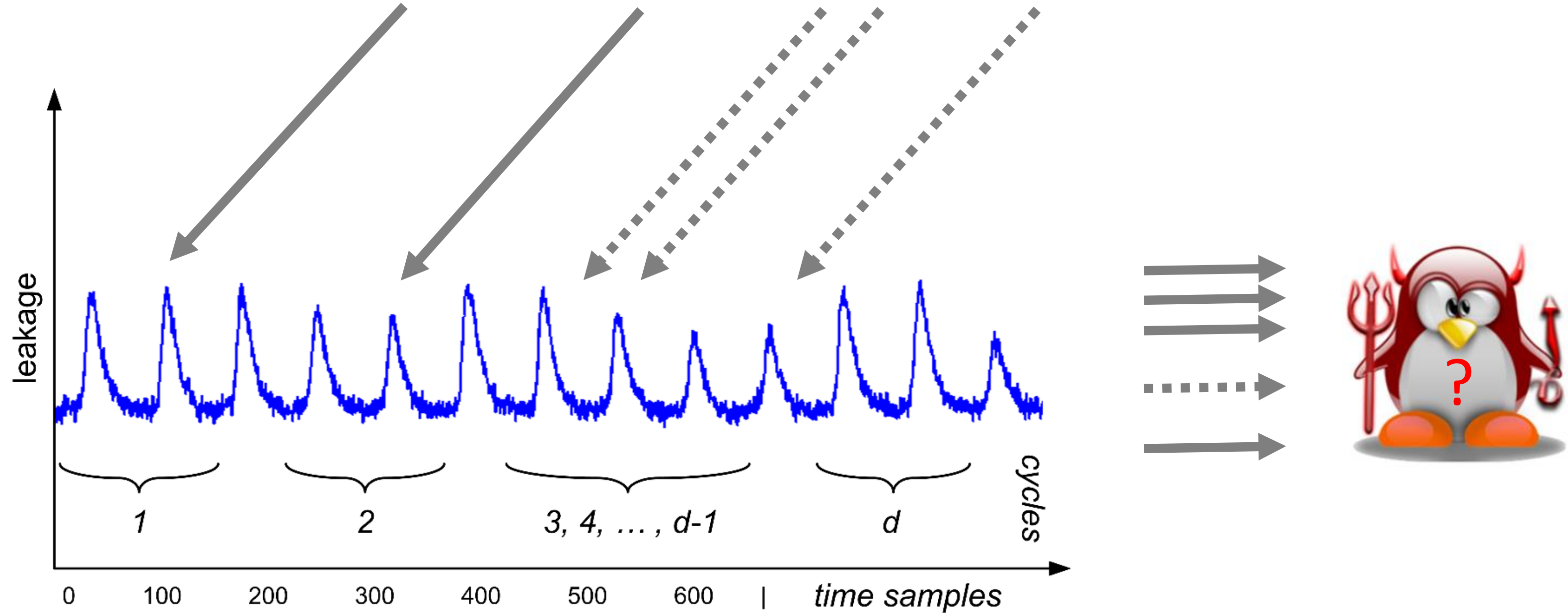
(a) serial implementation.

- Bounded information leakage $\mathrm{MI}(Y_i; L)^d$

Masking (concrete view)

- Probing security (Ishai, Sahai, Wagner 2003)

$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$



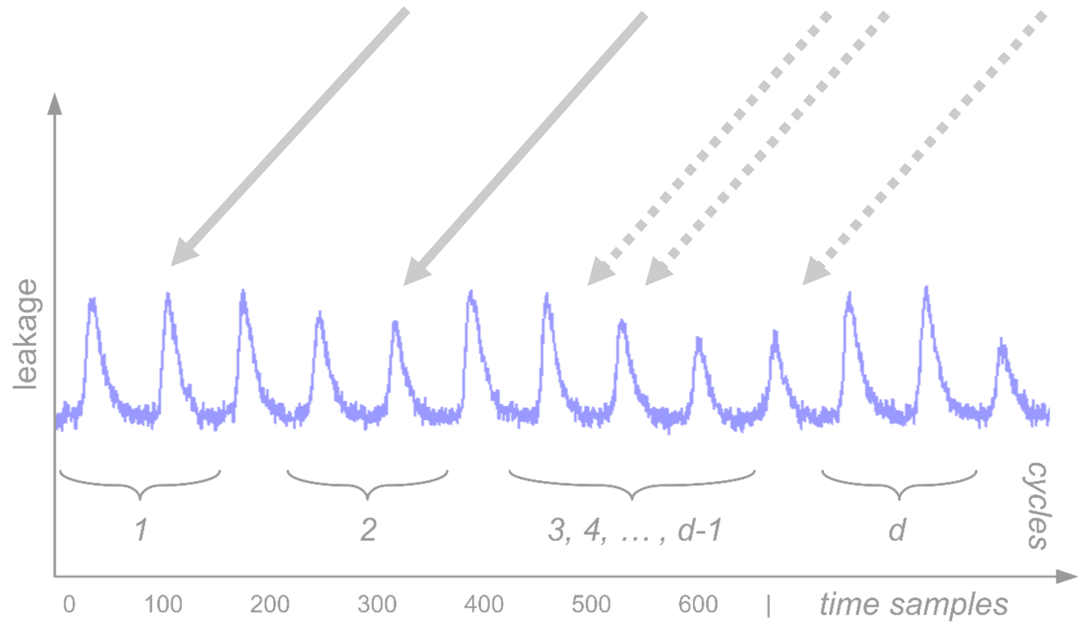
(a) serial implementation.

- Noisy leakage security (Prouff, Rivain 2013)

Masking (concrete view)

- Probing security (Ishai, Sahai, Wagner 2003)

$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$

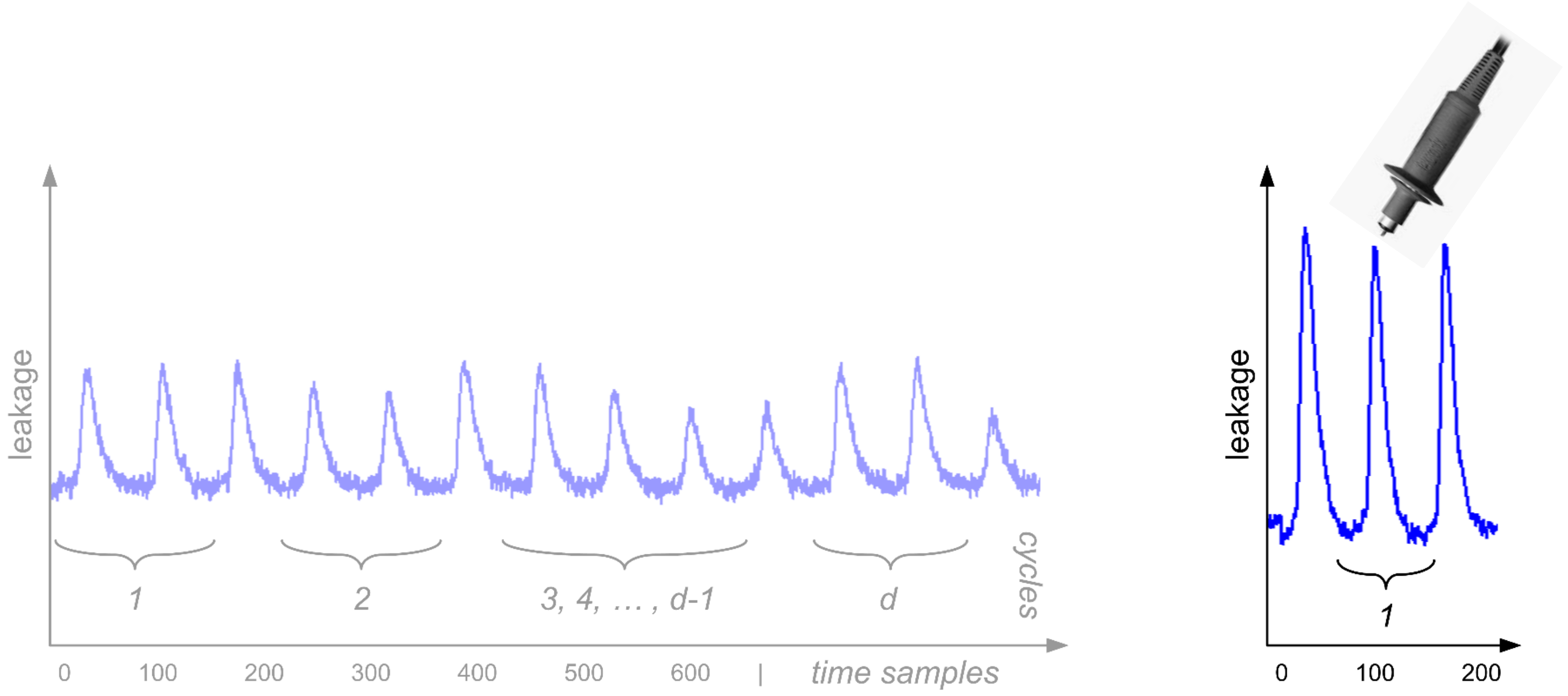


(a) serial implementation.

noise and **independence** (Duc, Dziembowski, Faust 2014)

- Noisy leakage security (Prouff, Rivain 2013)

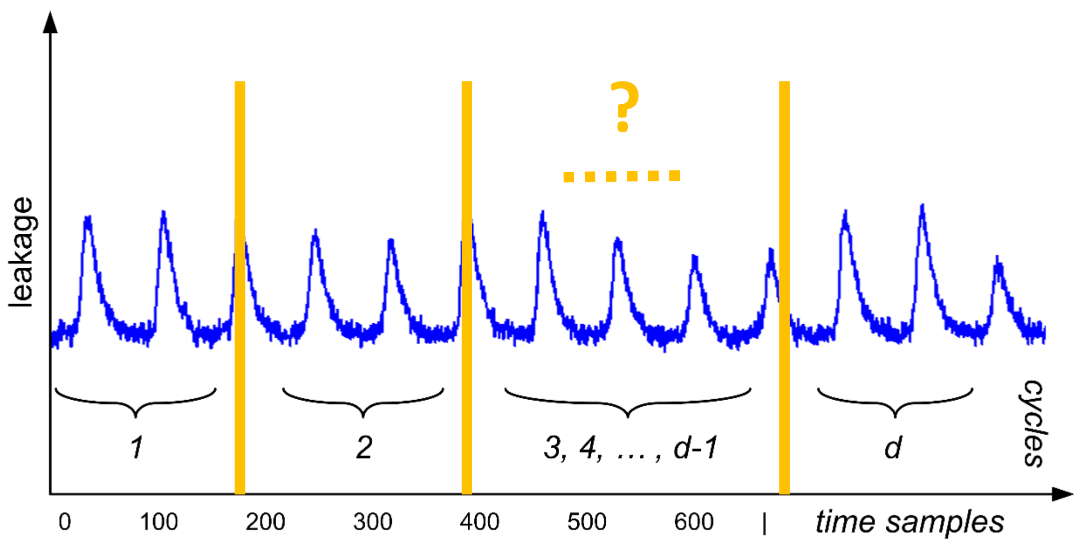
1. What happens with parallel implementations?
   - For example: one probe reveals the shares' sum



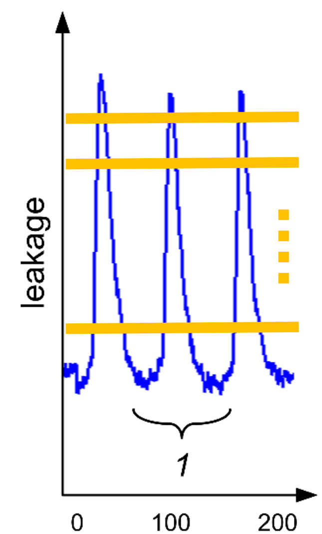
(a) serial implementation.

(b) parallel implementation.

1. What happens with parallel implementations?
   - For example: one probe reveals the shares' sum
2. How to test physical independence? (*consolidating*)

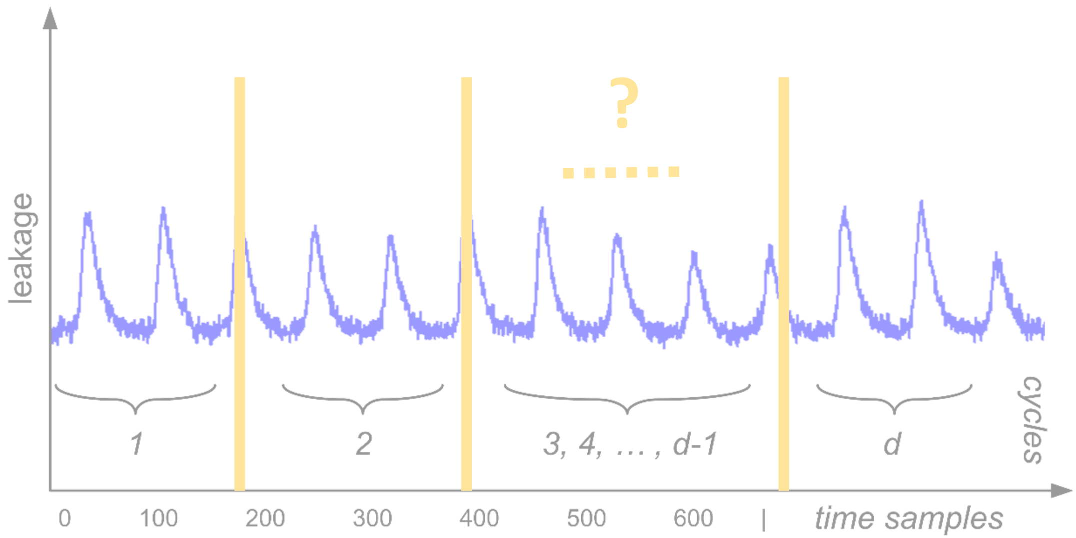


(a) serial implementation.

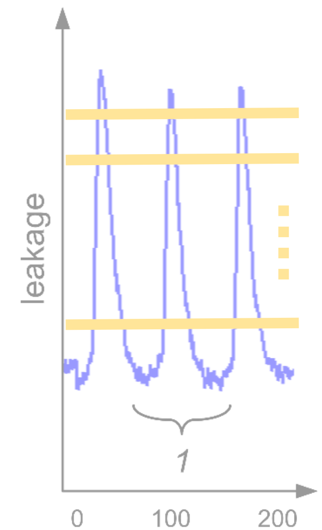


(b) parallel implementation.

1. What happens with parallel implementations?
   - For example: one probe reveals the shares' sum
2. How to test physical independence? (*consolidating*)



(a) serial implementation.



(b) parallel implementation.

- W/O directly working in the noisy leakage model

Outline

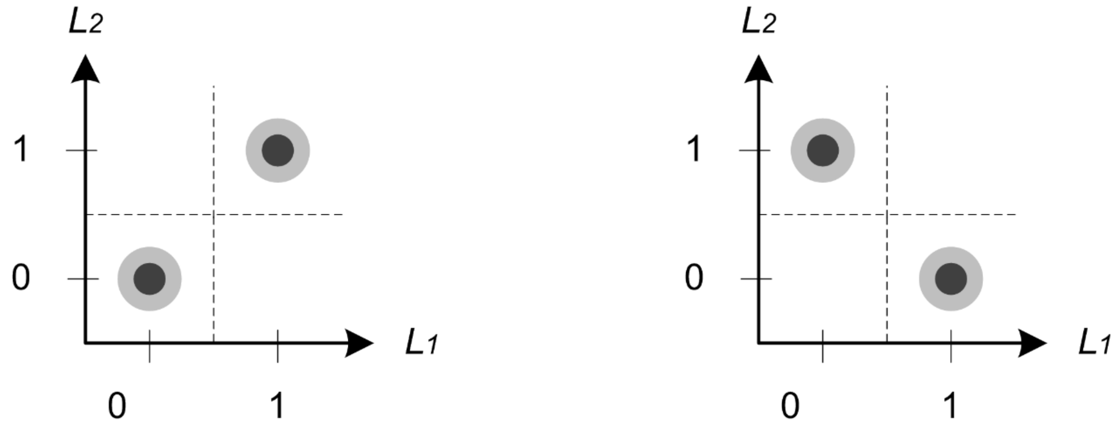
- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security ⇒ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

Masking statistical intuition

- 2-share / 1-bit example, <u>serial</u> implementation

$$L_1 = y_1 + n_1$$

$$L_2 = y_2 + n_2$$



(a) $Y = 0$, serial.

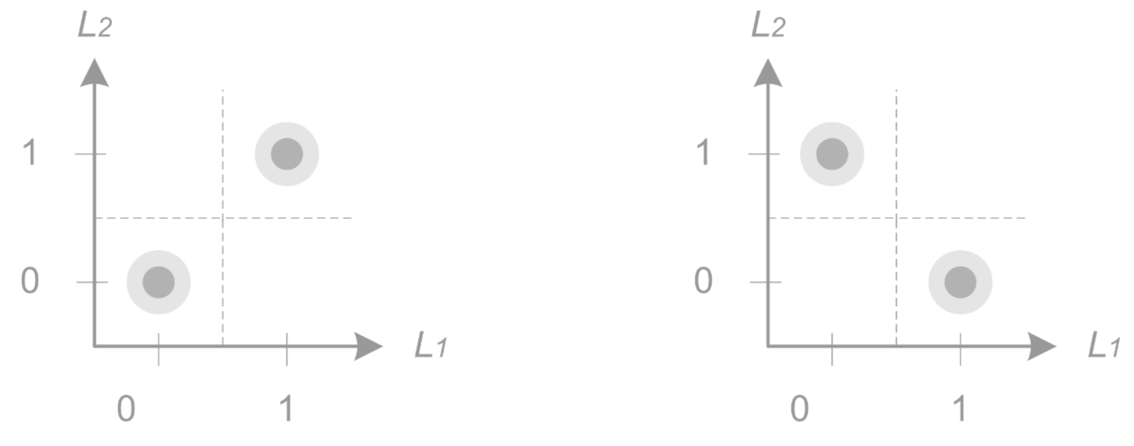
(b) $Y = 1$, serial.

Masking statistical intuition

- 2-share / 1-bit example, <u>parallel</u> implementation

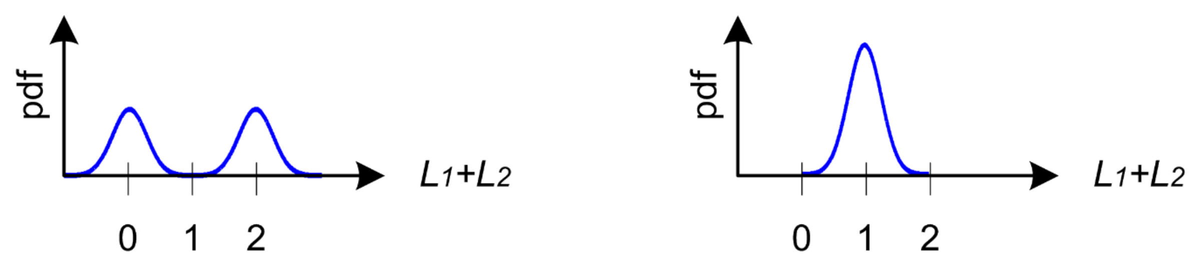
$$L_1 = y_1 + n_1$$

$$L_2 = y_2 + n_2$$



(a) $Y = 0$, serial. (b) $Y = 1$, serial.

$$L = y_1 + y_2 + n$$



(c) $Y = 0$, parallel. (d) $Y = 1$, parallel.

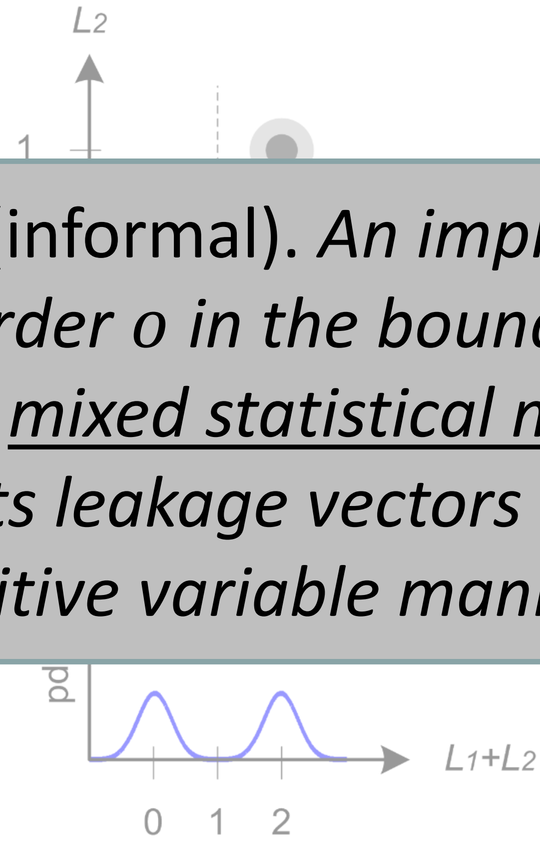
- 2-share / 1-bit example, <u>parallel</u> implementation

$L_1 = y_1 + n_1$

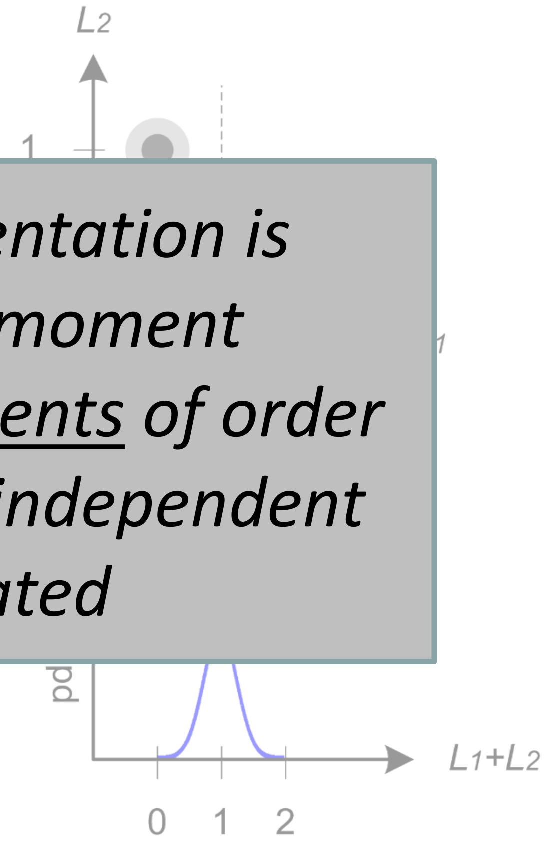
$L_2 =$

Definition (informal). *An implementation is secure at order o in the bounded moment model if all <u>mixed statistical moments</u> of order up to o of its leakage vectors are independent of any sensitive variable manipulated*

$L = y_1 + y_2 + \mathrm{n}$



(c) $Y = 0$, parallel.



(d) $Y = 1$, parallel.

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with "non-mixing" leakages)
  - Continuous security & refreshing gadgets
- Conclusions

Abstract reduction (answer to Q1)

- **Theorem** (informal). *A parallel implementation is secure at order o in the BMM if its serialization is secure at order o in the probing model where*
  - Adv_{pr} can (typically) probe $o = d - 1$ wires
  - Adv_{bm} can observe any $L = \sum_{i=1}^{d} \alpha_i \cdot y_i$

Abstract reduction

- **Theorem** (informal). *A parallel implementation is secure at order o in the BMM if its serialization is secure at order o in the probing model where*
  - Adv_{pr} can (typically) probe $o = d - 1$ wires
  - Adv_{bm} can observe any $L = \sum_{i=1}^{d} \alpha_i \cdot y_i$
- Intuition: summing the shares (in $\mathbb{R}$) does not break the independent leakage assumption

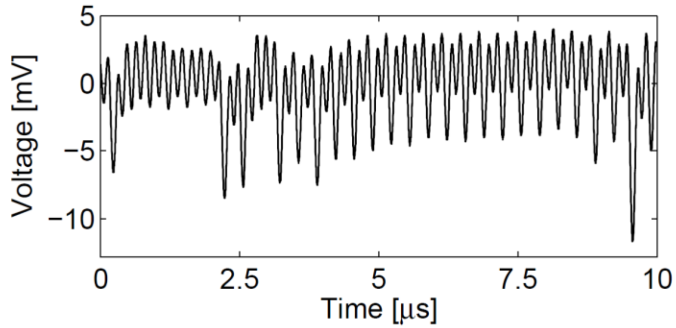
Abstract reduction

- **Theorem** (informal). *A parallel implementation is secure at order o in the BMM if its serialization is secure at order o in the probing model where*
  - Adv_{pr} can (typically) probe $o = d - 1$ wires
  - Adv_{bm} can observe any $L = \sum_{i=1}^{d} \alpha_i \cdot y_i$

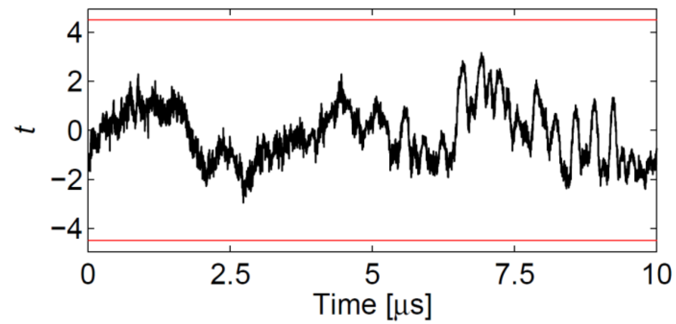
- Intuition: summing the shares (in $\mathbb{R}$) does not break the independent leakage assumption

- Main ≠ between probing and BM security
  - Adv_{bm} can sum over ***all*** the shares!
  - BM security is weaker (moments vs. distributions)

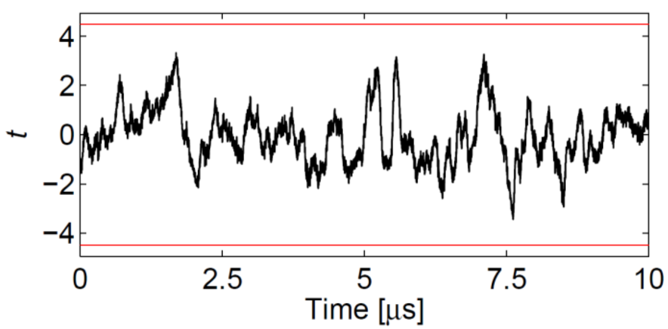
- If physically independent leakages, BM security extends to actual measurements (e.g., $d = 3$)



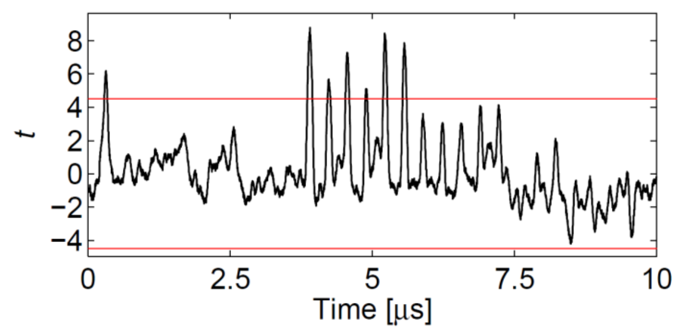
(a) Sample trace



(b) 1st-order

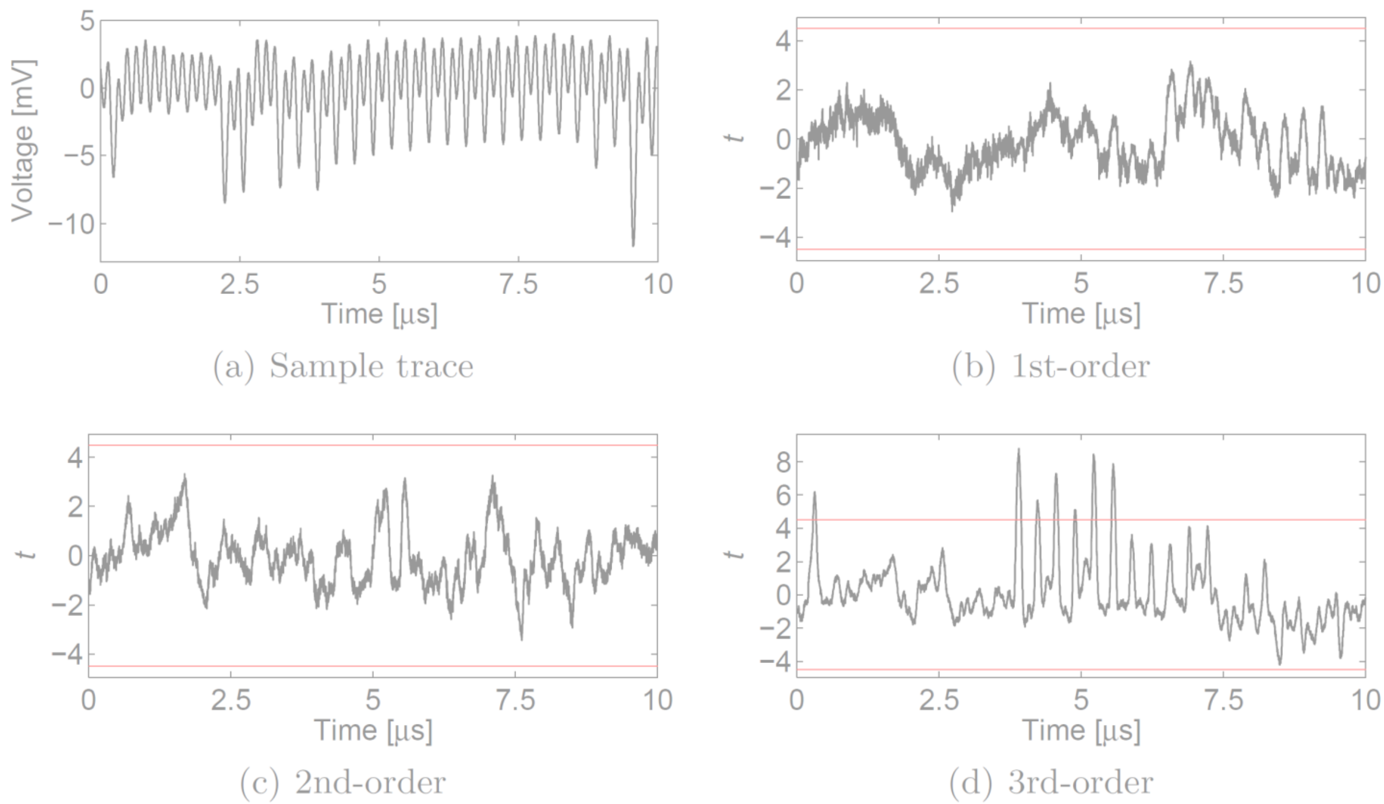


(c) 2nd-order



(d) 3rd-order

- If physically independent leakages, BM security extends to actual measurements (e.g., $d = 3$)



- If not, leakages are not independent

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

- ISW 2003: multiplication $c = a \times b$

$$\underbrace{\begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}}_{\text{partial products}} \oplus \underbrace{\begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix}}_{\text{refresh}} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

compress

- ISW 2003: multiplication $c = a \times b$

$$\underbrace{\begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix}}_{\text{partial products}} \oplus \underbrace{\begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix}}_{\text{refresh}} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \text{compress}$$

- AES S-box ($n = 8$) implementation
  - $a = a_1 \oplus a_2 \oplus \cdots \oplus a_d$ (e.g., $d = 8$)
  - Each register stores an a_i (i.e., a $\mathrm{GF}(2^8)$ element)
  - Memory $\propto n \cdot d$, Time: $\propto \boldsymbol{d^2}$ $\mathrm{GF}(2^8)$ mult.
  - AES S-box $\approx$ 3 multiplications (& 4 squarings)

- Main tweak: interleave & regularize

$$\begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ a_3 b_3 \end{bmatrix} \oplus \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \oplus \begin{bmatrix} a_1 b_3 & a_3 b_1 \\ a_2 b_1 & a_1 b_2 \\ a_3 b_2 & a_2 b_3 \end{bmatrix} \oplus \begin{bmatrix} r_3 \\ r_1 \\ r_2 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

refresh

Parallel multiplication

- Main tweak: interleave & regularize

$$\begin{bmatrix} a_1b_1 \\ a_2b_2 \\ a_3b_3 \end{bmatrix} \oplus \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \oplus \begin{bmatrix} a_1b_3 & a_3b_1 \\ a_2b_1 & a_1b_2 \\ a_3b_2 & a_2b_3 \end{bmatrix} \oplus \begin{bmatrix} r_3 \\ r_1 \\ r_2 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

refresh

- AES S-box ($n = 8$) implementation
  - $a = a_1 \oplus a_2 \oplus \cdots \oplus a_d$ (e.g., $d = 8$)
  - Each register stores n a_i's (i.e., $\mathrm{GF}(2)$ elements)
  - Memory $\propto n \cdot d$, Time: $\propto \boldsymbol{d}$ $\mathrm{GF}(2)$ mult. (i.e., ANDs)
  - AES bitslice S-box $\approx$ 32 AND gates (& 83 XORs)

Parallel multiplication

- Main tweak: interleave & regularize

$$\begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ a_3 b_3 \end{bmatrix} \oplus \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \oplus \begin{bmatrix} a_1 b_3 & a_3 b_1 \\ a_2 b_1 & a_1 b_2 \\ a_3 b_2 & a_2 b_3 \end{bmatrix} \oplus \begin{bmatrix} r_3 \\ r_1 \\ r_2 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

refresh

- AES S-box ($n = 8$) implementation
  - $a = a_1 \oplus a_2 \oplus \cdots \oplus a_d$ (e.g., $d = 8$)
  - Each register stores n a_i's (i.e., $\mathrm{GF}(2)$ elements)
  - Memory $\propto n \cdot d$, Time: $\propto \boldsymbol{d}$ $\mathrm{GF}(2)$ mult. (i.e., ANDs)
  - AES bitslice S-box $\approx$ 32 AND gates (& 83 XORs)

$\Rightarrow$ Performance gains with large d's (8, 16, 32)

Security analysis

- We analyzed the SNI security of the gadgets
  ≈ composable probing security (Barthe et al. 2016)

Security analysis

- We analyzed the SNI security of the gadgets
  $\approx$ composable probing security (Barthe et al. 2016)
- Iterating $\lceil (d-1)/3 \rceil$ refresh is SNI for $d < 12$

- We analyzed the SNI security of the gadgets
  ≈ composable probing security (Barthe et al. 2016)
- Iterating $\lceil (d-1)/3 \rceil$ refresh is SNI for $d < 12$
- Multiplication is more tricky…

Algorithm	d	(d- 1)-SNI	rand (our alg.)	rand (ISW 2003)
multiplication	3	√	3	3
	$d \geq 4$	X	$d(d-1)/4$	$d(d-1)/2$
refresh o multiplication	4	√	8	6
	5	√	10	10
	6	√	18	15
	7	√	21	21
	8	√	**24**	**28**

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

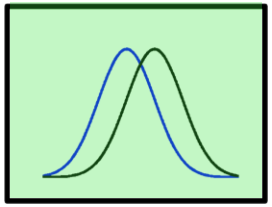
Specialized encodings

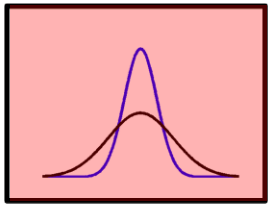
- Probing security is stronger than BM security
  - (And also stronger than noisy leakage security)
- Is it sometimes “too strong”?
  - i.e., breaks designs that are secure against DPA

Specialized encodings

- Probing security is stronger than BM security
  - (And also stronger than noisy leakage security)
- Is it sometimes “too strong”?
  - i.e., breaks designs that are secure against DPA
- Example: Boolean encoding (2 shares)

$$y = y_1 \oplus y_2$$

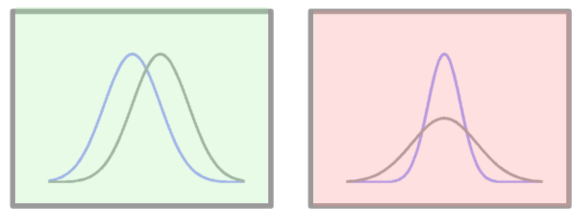




- Probing security is stronger than BM security
  - (And also stronger than noisy leakage security)
- Is it sometimes "too strong"?
  - i.e., breaks designs that are secure against DPA

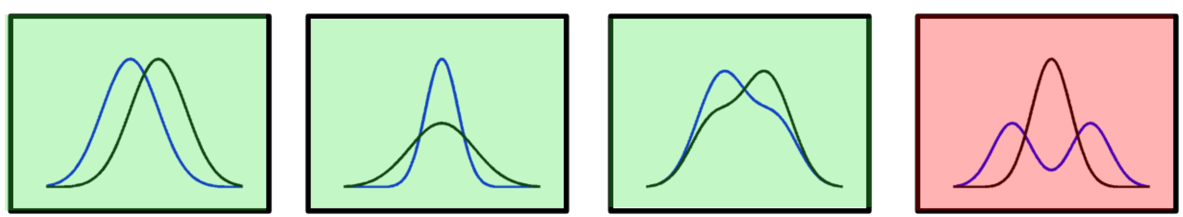
- Example: Boolean encoding (2 shares)

$$y = y_1 \oplus y_2$$



- IP masking *in* GF(2^8) *with "non-mixing" leakages*

$$y = \sum_{i=1}^{2} p_i \times s_i$$



$p_2 = 1$ $p_2 = 5$ $p_2 = 7$

Outline

- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with “non-mixing” leakages)
  - Continuous security & refreshing gadgets
- Conclusions

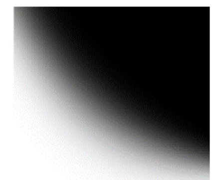
Continuous security

- So far we discussed “one-shot” probing attacks

Continuous security

- So far we discussed "one-shot" probing attacks

- Yet, side-channel attacks are usually continuous
  - i.e, accumulate information from multiple executions



Continuous security

- So far we discussed “one-shot” probing attacks
- Yet, side-channel attacks are usually continuous
  - i.e, accumulate information from multiple executions
- Typical issue: refreshing by add a share of 0
  - Frequently used in practice
  - Yet insecure in the continuous probing model
  - What does it mean concretely?
  - i.e., can we (sometimes) use such a refreshing?

- Target: refresh$(a) = a \oplus r \oplus$ rot(r)

step 1

$a_1^{(1)}$
$a_2^{(1)}$
$a_3^{(1)}$
$a_4^{(1)}$

Accumulated knowledge: $\emptyset$

Continuous probing attack

- Target: refresh$(a) = \mathrm{a} \oplus r \oplus \mathrm{rot}(r)$

step 1

$a_1^{(1)}$

$a_2^{(1)}$

$a_3^{(1)}$

$a_4^{(1)}$

Accumulated knowledge: $\emptyset$

Continuous probing attack

- Target: refresh(a) $=$ a $\oplus$ r $\oplus$ rot(r)

step 1

$\boxed{a_1^{(1)}}$

$a_2^{(1)}$

$a_3^{(1)}$

$a_4^{(1)}$

Accumulated knowledge: $a_1^{(1)}$

Continuous probing attack

- Target: refresh(a) $=$ a $\oplus$ r $\oplus$ rot(r)

step 1	step 2		
$\boxed{a_1^{(1)}}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$

Accumulated knowledge: $a_1^{(1)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\ r \oplus$ rot(r)

step 1		step 2	
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$

Accumulated knowledge: $a_1^{(1)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\ r \oplus$ rot(r)

step 1		step 2		
$a_1^{(1)}$		$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$
$a_2^{(1)}$		$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$
$a_3^{(1)}$		$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$
$a_4^{(1)}$		$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$

Accumulated knowledge: $a_1^{(1)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\ r \oplus$ rot(r)

step 1		step 2	
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$

Accumulated knowledge: $a_1^{(2)} \oplus a_2^{(2)}$

Continuous probing attack

- Target: refresh(a) $=$ a $\oplus$ r $\oplus$ rot(r)

step 1	step 2			step 3		
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$

Accumulated knowledge: $a_1^{(2)} \oplus a_2^{(2)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\, r \,\oplus$ rot(r)

step 1	step 2			step 3		
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$

Accumulated knowledge: $a_1^{(2)} \oplus a_2^{(2)}$

- Target: refresh(a) $=$ a $\oplus$ r $\oplus$ rot(r)

step 1	step 2			step 3		
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$

Accumulated knowledge: $a_1^{(2)} \oplus a_2^{(2)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\, r \oplus$ rot(r)

step 1	step 2			step 3		
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$

Accumulated knowledge: $a_1^{(3)} \oplus a_2^{(3)} \oplus a_3^{(3)}$

Continuous probing attack

- Target: refresh(a) = a $\oplus\, r \,\oplus$ rot(r)

step 1	step 2			step 3			
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$	
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$	$\cdots$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$	
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$	

$\Rightarrow$ After d iterations, a is learned in full by Adv_{pr}

Continuous probing attack

- Target: refresh(a) = a $\oplus$ r $\oplus$ rot(r)

step 1	step 2			step 3			
$\boxed{a_1^{(1)}}$	$\boxed{r_1^{(2)}}$	$\boxed{r_4^{(2)}}$	$\boxed{a_1^{(2)}}$	$r_1^{(3)}$	$\boxed{r_4^{(3)}}$	$\boxed{a_1^{(3)}}$	
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$\boxed{a_2^{(2)}}$	$\boxed{r_2^{(3)}}$	$r_1^{(3)}$	$\boxed{a_2^{(3)}}$	$\cdots$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$\boxed{a_3^{(3)}}$	
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$	

$\Rightarrow$ After d iterations, a is learned in full by Adv_{pr}

- Not possible in the BMM. Intuition: *adaptation does not help* since Adv_{bm} can anyway sum over all shares!

Continuous probing attack

- Target: refresh(a) $=$ a $\oplus\, r \,\oplus$ rot(r)

step 1	step 2			step 3			
$a_1^{(1)}$	$r_1^{(2)}$	$r_4^{(2)}$	$a_1^{(2)}$	$r_1^{(3)}$	$r_4^{(3)}$	$a_1^{(3)}$	
$a_2^{(1)}$	$r_2^{(2)}$	$r_1^{(2)}$	$a_2^{(2)}$	$r_2^{(3)}$	$r_1^{(3)}$	$a_2^{(3)}$	$\cdots$
$a_3^{(1)}$	$r_3^{(2)}$	$r_2^{(2)}$	$a_3^{(2)}$	$r_3^{(3)}$	$r_2^{(3)}$	$a_3^{(3)}$	
$a_4^{(1)}$	$r_4^{(2)}$	$r_3^{(2)}$	$a_4^{(2)}$	$r_4^{(3)}$	$r_3^{(3)}$	$a_4^{(3)}$	

$\Rightarrow$ After d iterations, a is learned in full by Adv_{pr}

- <u>Impact:</u> refresh(.) can be used to refresh the key of a key homomorphic primitive ($\Rightarrow$ fully linear overheads)

Outline

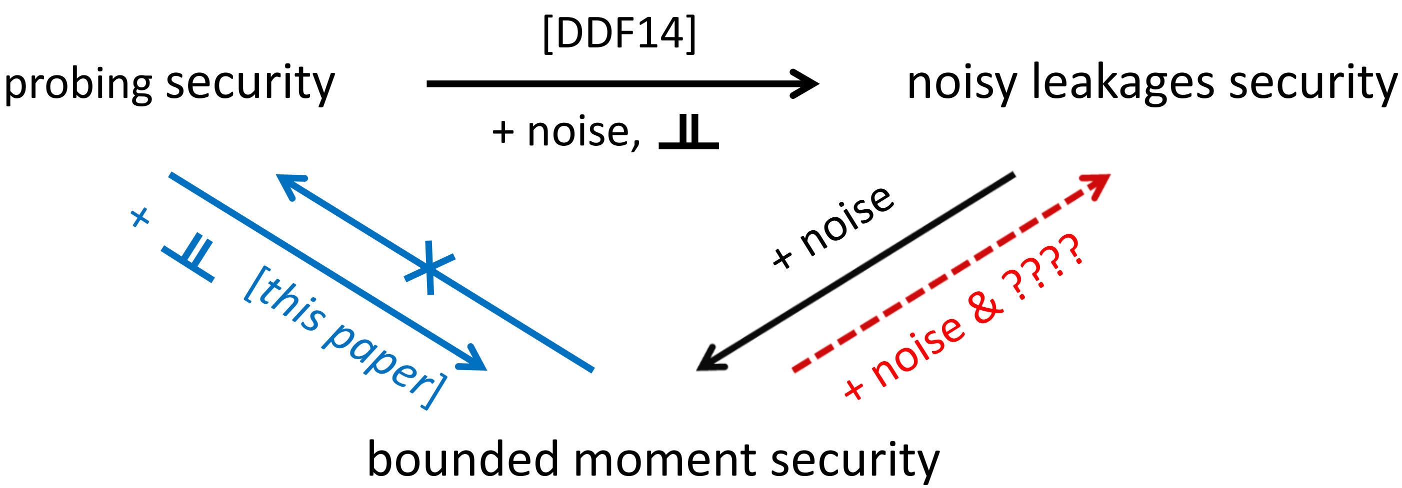
- Introduction / motivation
  - Side-channel attacks and noise
  - Masking and leakage models
- Bounded moment model
  - Masking intuition & BMM definition
  - Probing security $\Rightarrow$ BM security
- Parallel multiplication (& refreshing)
- BM security $\not\Rightarrow$ probing security
  - Inner product masking (with "non-mixing" leakages)
  - Continuous security & refreshing gadgets
- Conclusions

Conclusions

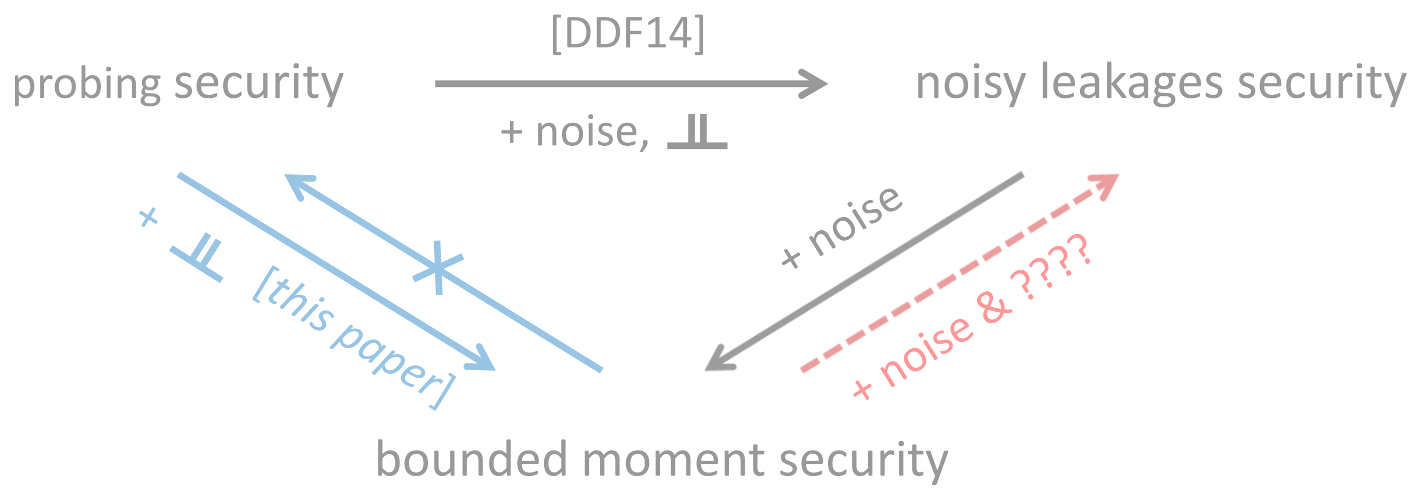
- Probing security is relevant to parallel implem.

Conclusions

- Probing security is relevant to parallel implem.
- BMM suggests a principled path to security eval.



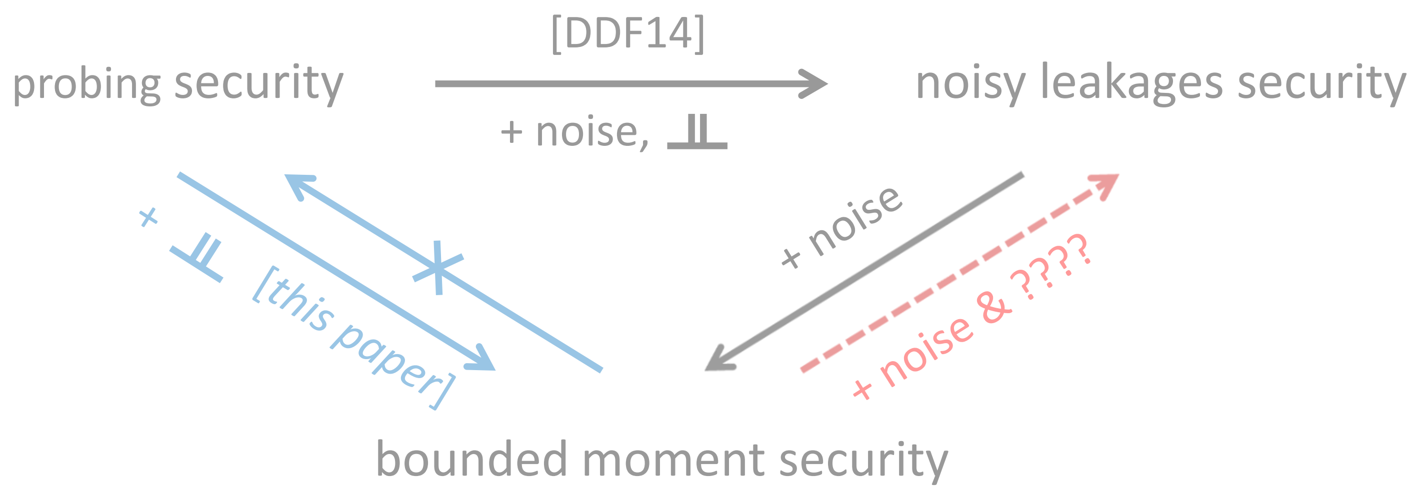
- Probing security is relevant to parallel implem.
- BMM suggests a principled path to security eval.



- Parallel implem. are appealing for masking
  - Leverage the memory needed to store shares

Conclusions

- Probing security is relevant to parallel implem.
- BMM suggests a principled path to security eval.



- Parallel implem. are appealing for masking
  - Leverage the memory needed to store shares
- Cont. probing security sometimes "too strong"

THANKS

http://perso.uclouvain.be/fstandae/