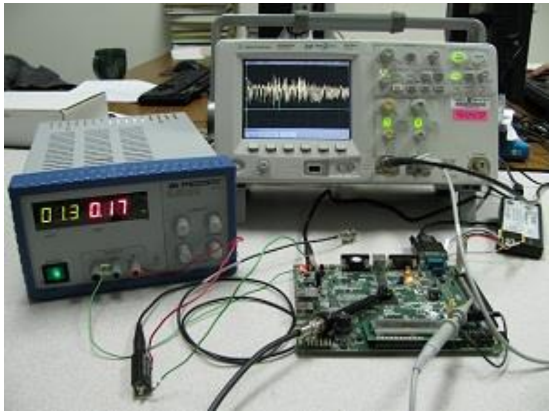
Evaluating and Designing Against Side-Channel Leakage: White Box or Black Box?



François-Xavier Standaert

UCLouvain, ICTEAM, Crypto Group (Belgium)

IHMMSec 2021, Virtual

Outline

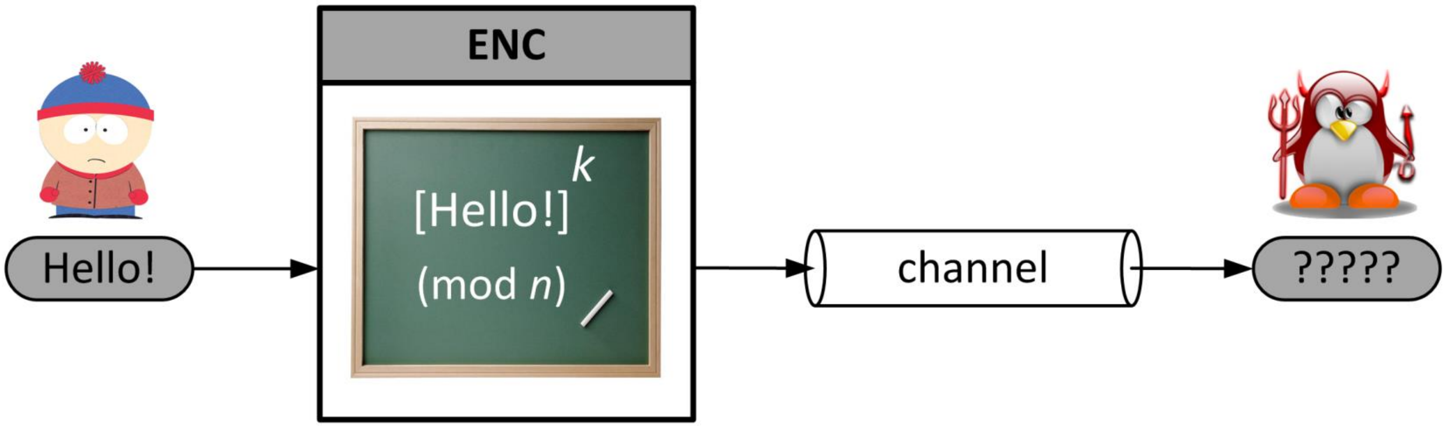
- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

Outline

- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

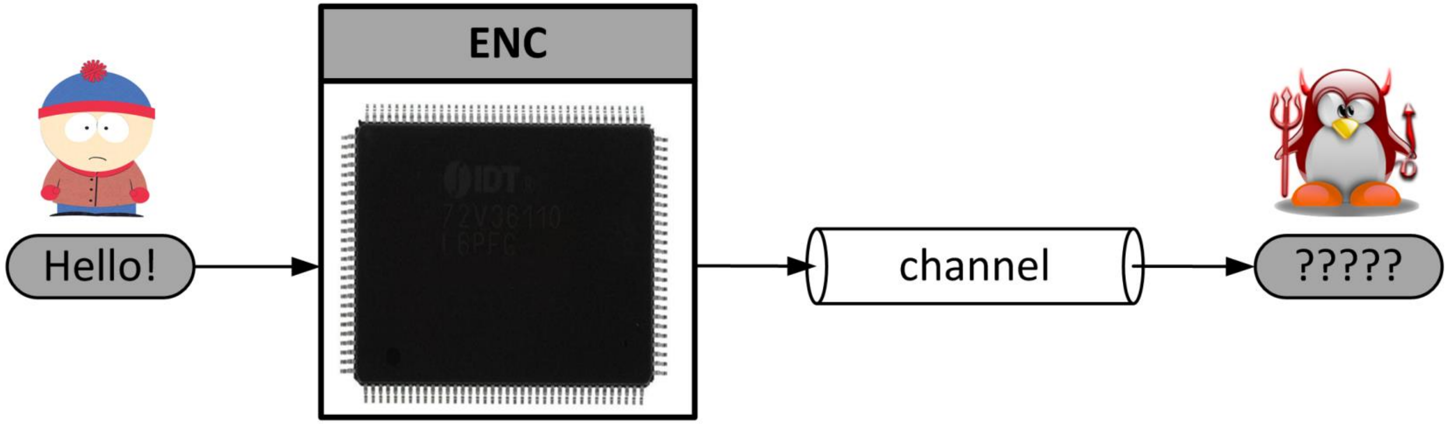
Cryptographic algorithms

- e.g. encryption:



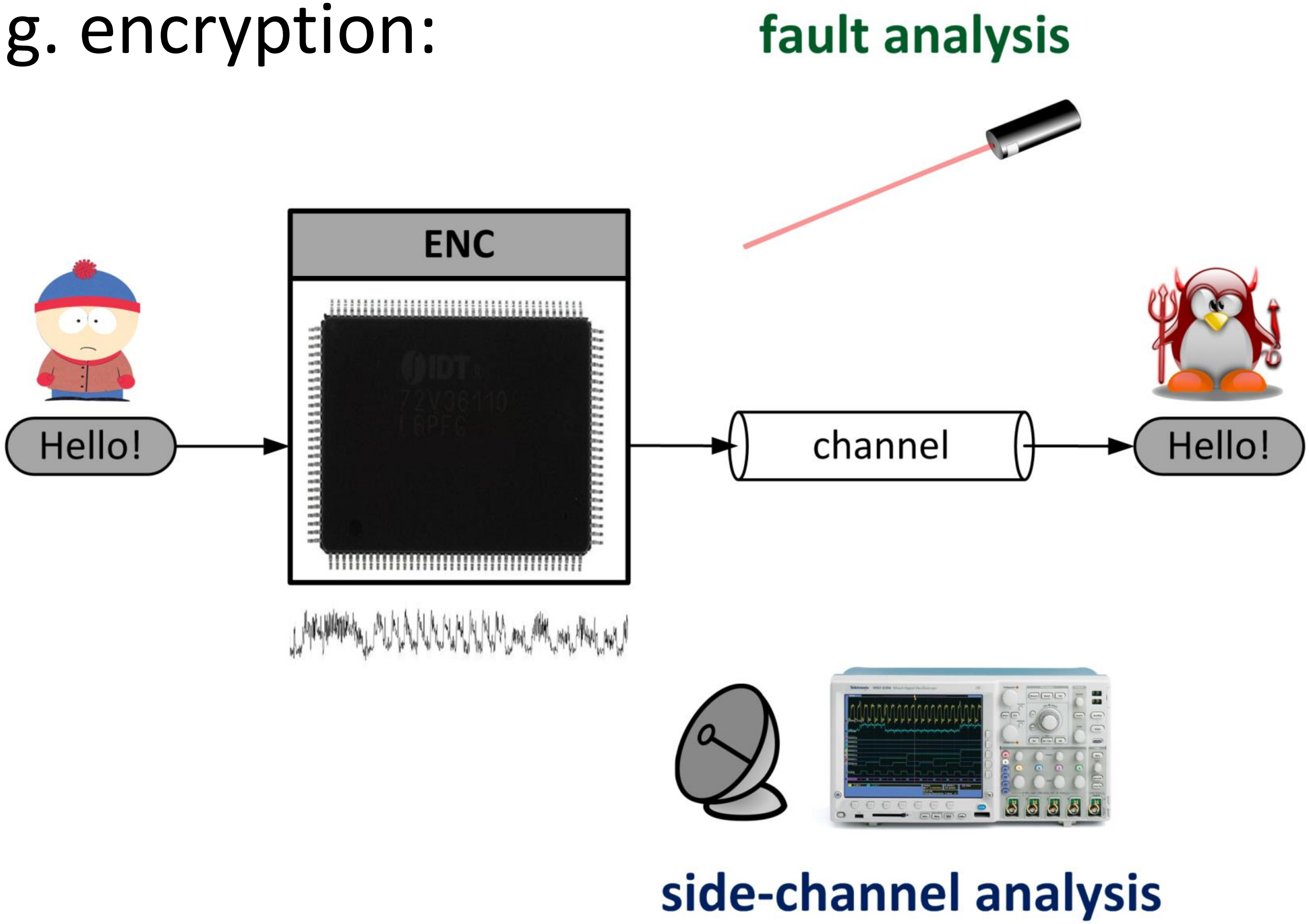
- Public algorithms and secret keys
  - Essential for both security and trust

- e.g. encryption:

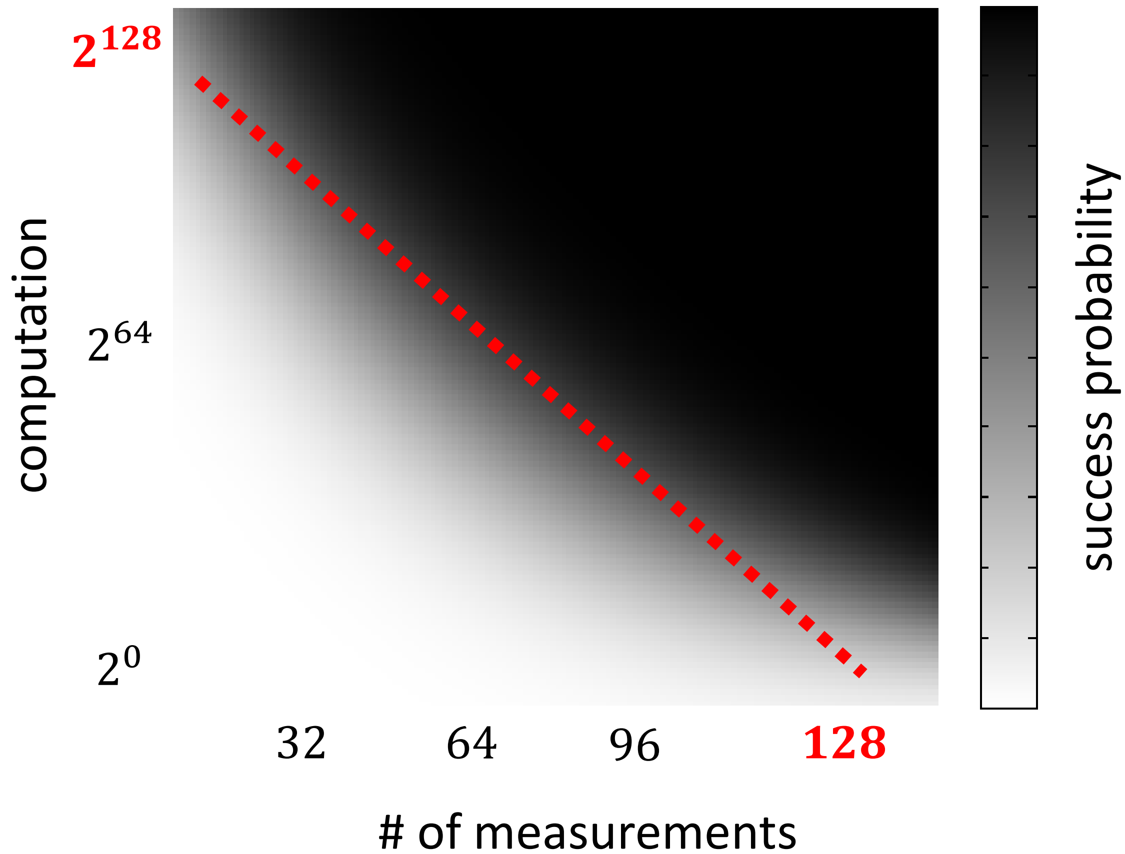


Cryptographic implementations

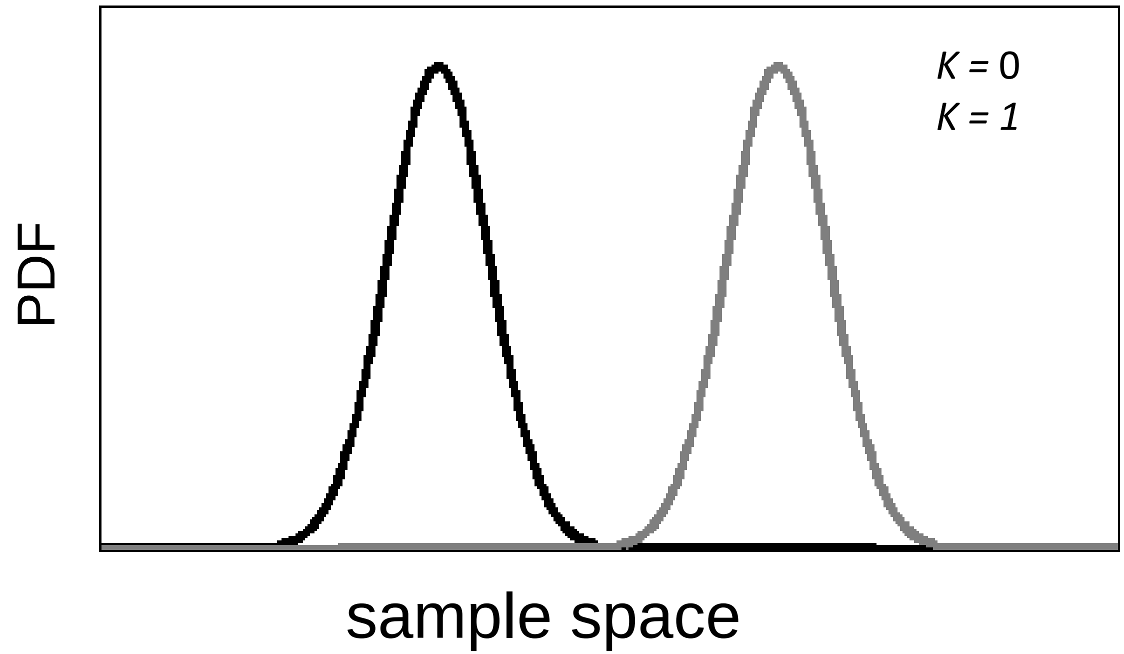
- e.g. encryption:



Side-channel analysis (in two slides)

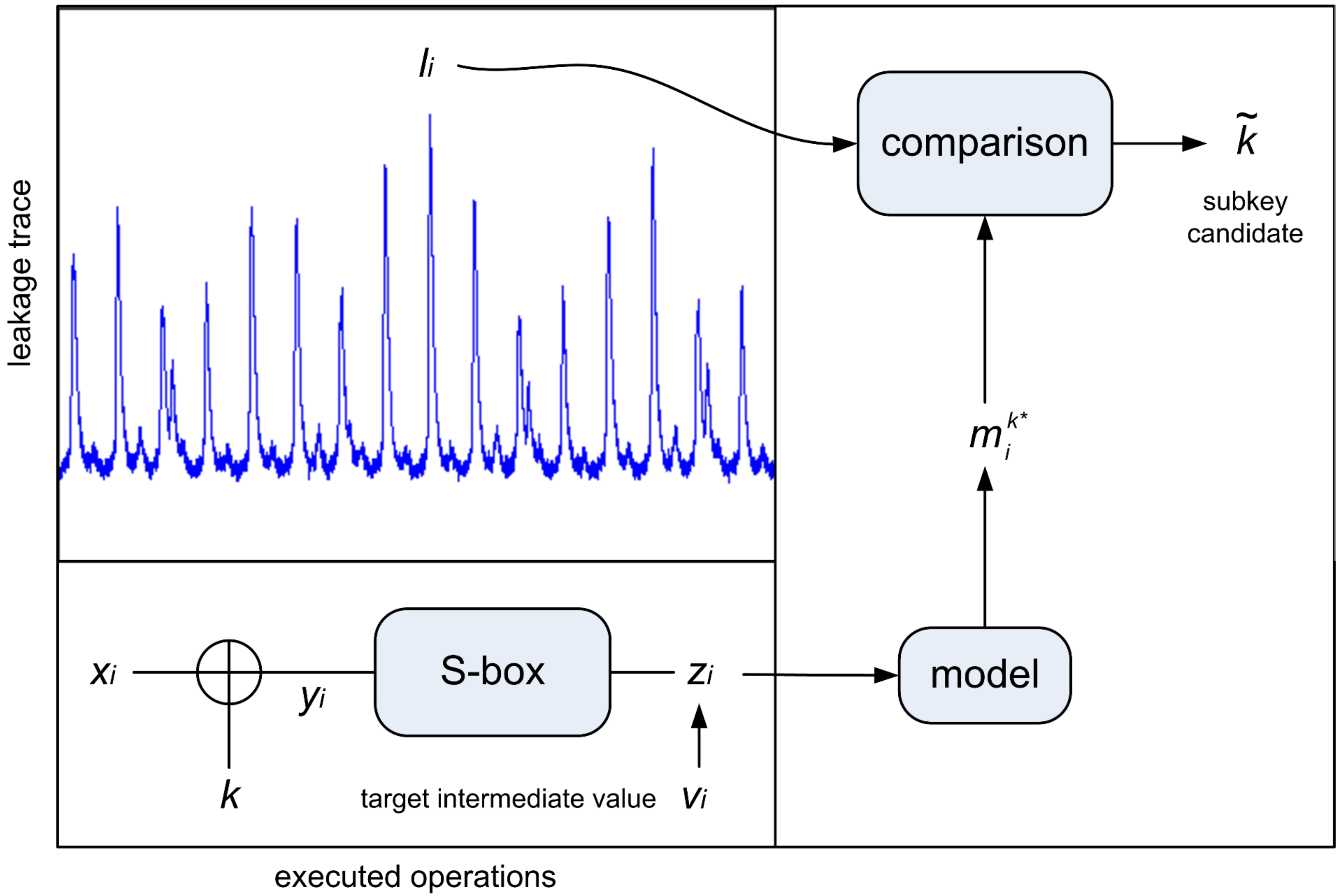


- ≈ physical attacks that decreases security exponentially in the # of measurements

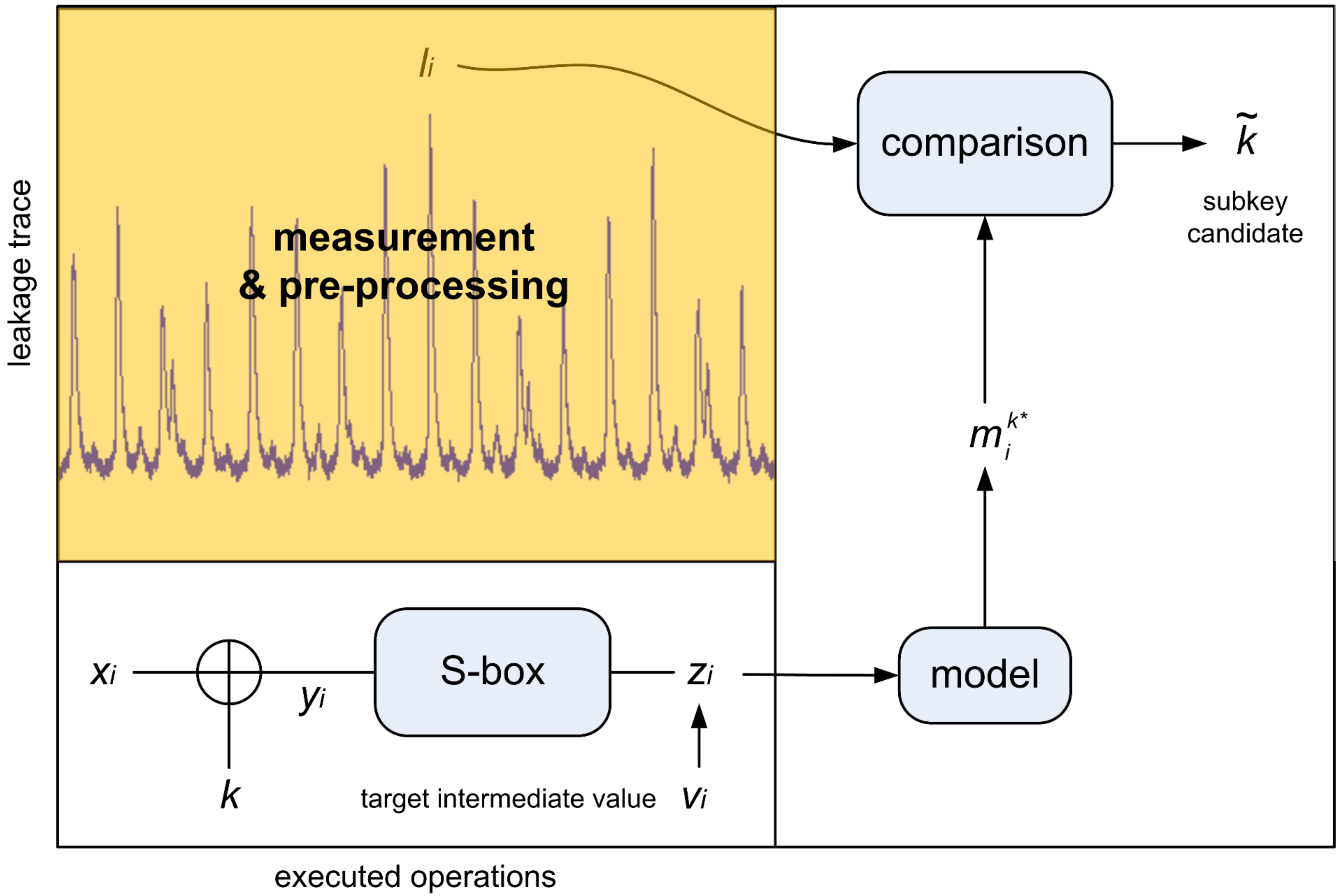


- … & where each bit of secret is learned by distinguishing noisy (leakage) distributions

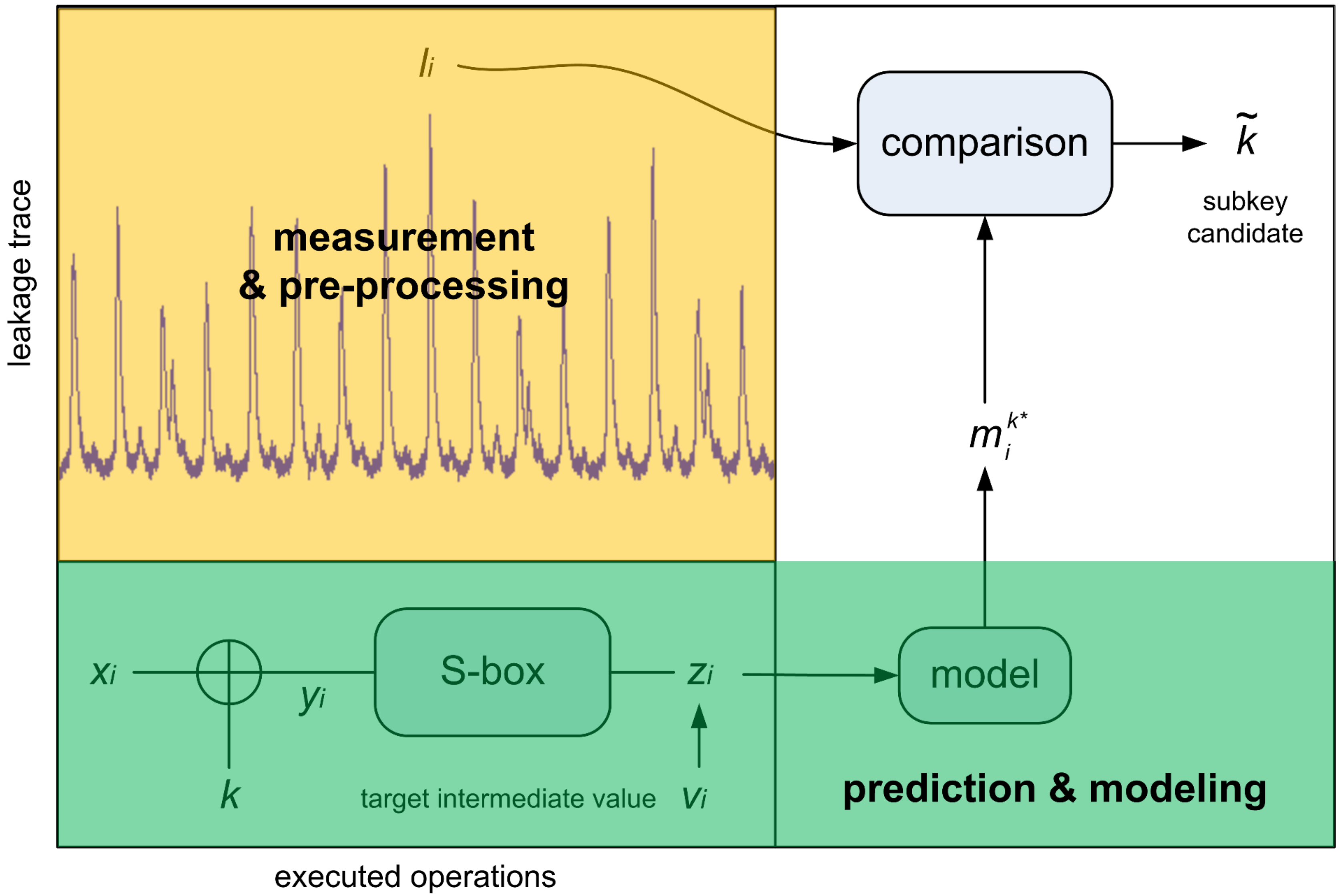
Standard DPA [KJJ99]



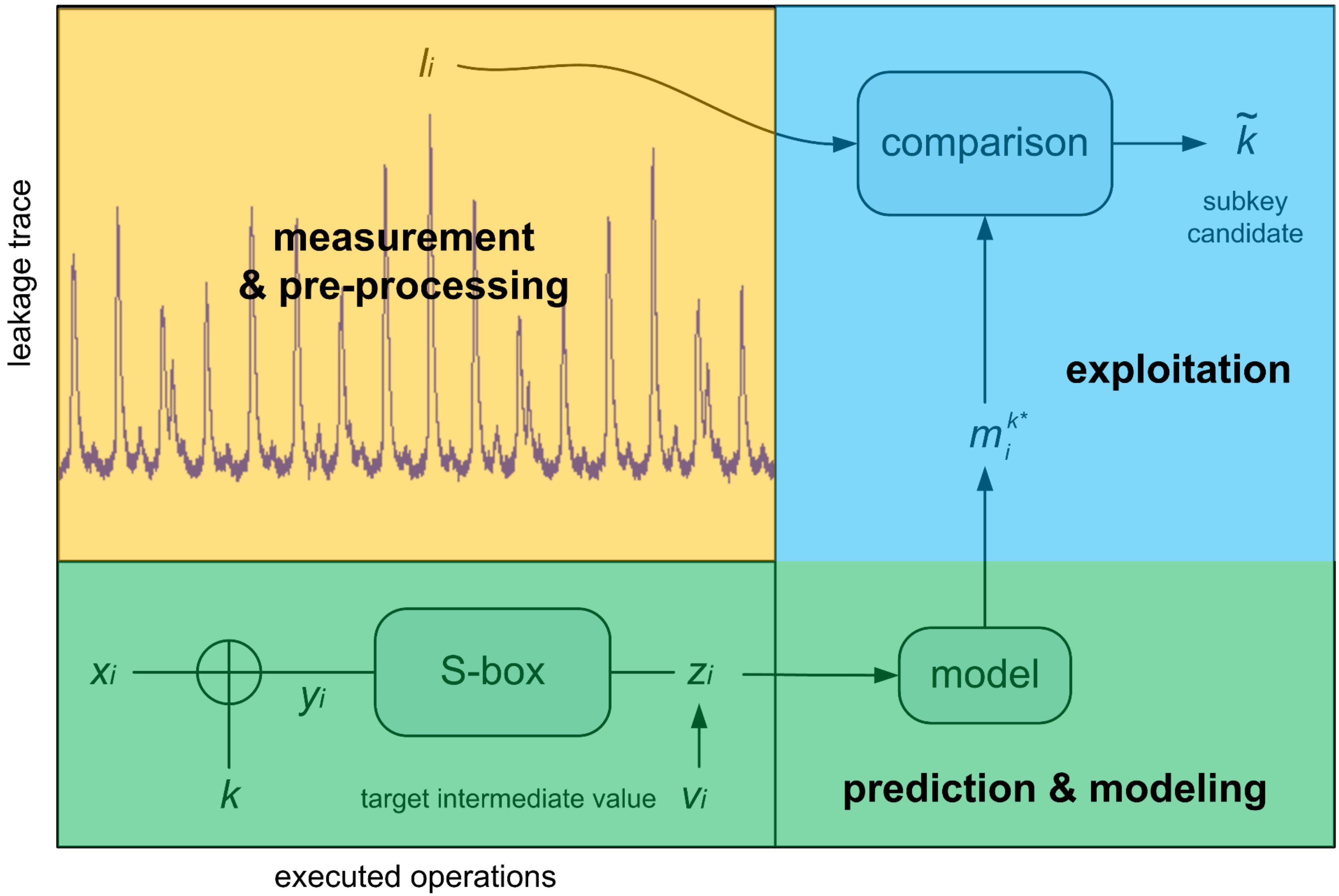
Standard DPA [KJJ99]



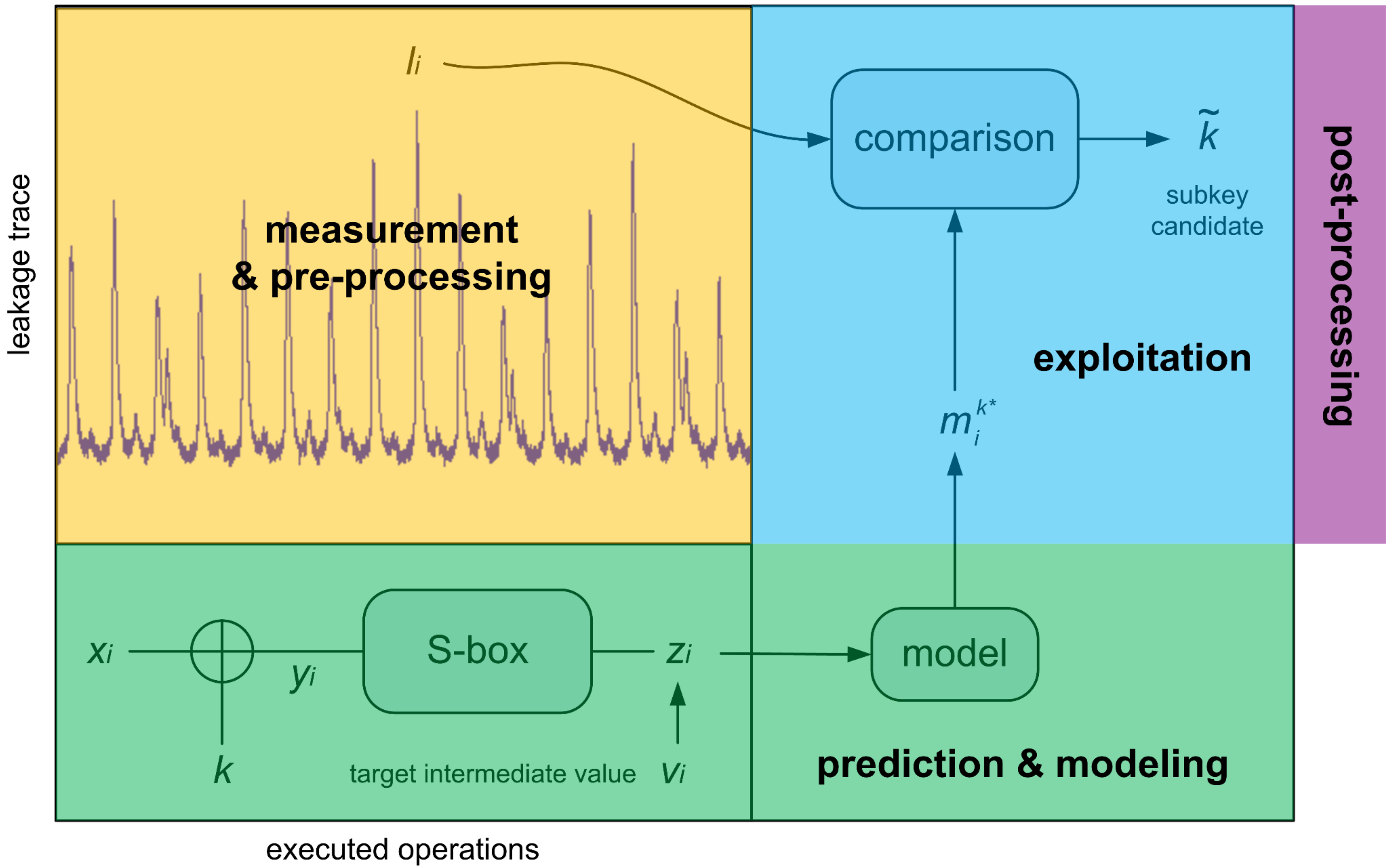
Standard DPA [KJJ99]



Standard DPA [KJJ99]

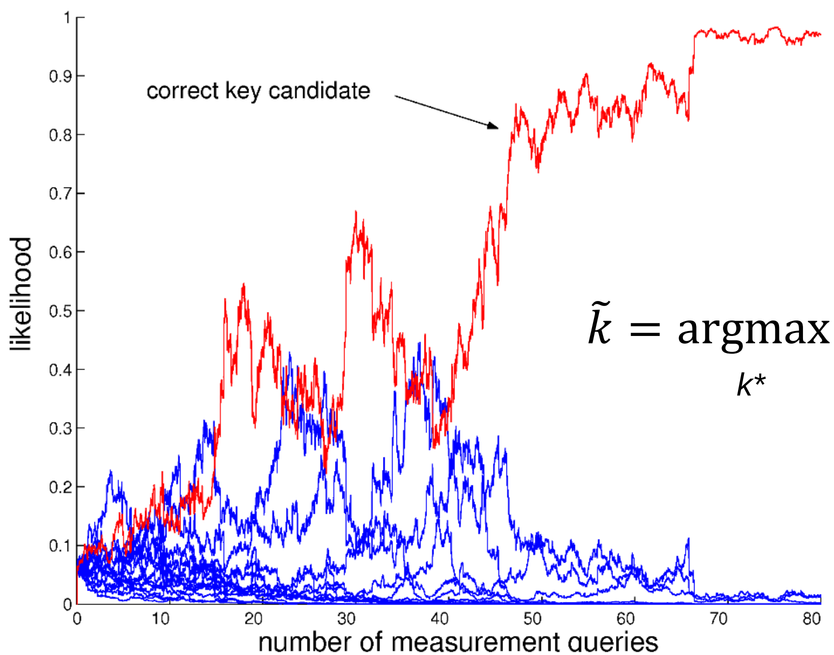


Standard DPA [KJJ99]



- General case: profiled DPA [CRR02]
  - Build "*templates*", i.e. $\hat{f}(l_i|k, x_i)$
    - e.g. Gaussian, regression-based
  - Maximum likelihood attack

- General case: profiled DPA [CRR02]
  - Build "*templates*", i.e. $\hat{f}(l_i|k, x_i)$
    - e.g. Gaussian, regression-based
  - Maximum likelihood attack



$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \prod_{i=1}^{q} \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma(L)} \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{l_i - m_i^{k^*}}{\sigma(L)}\right)^2\right)$$

- Side-channel attacks are continuous
  - Better evaluated with information theoretic metrics that capture the attack data complexity

$$\mathrm{SR} \leq 1 - (1 - \mathrm{MI}(Y; \boldsymbol{L}_Y))^m$$

$\Rightarrow$ # of traces m to reach $\mathrm{SR} \approx 1 \propto \frac{c(n)}{\mathrm{MI}(Y;\boldsymbol{L}_Y)}$

Important attack features

- Side-channel attacks are continuous
  - Better evaluated with information theoretic metrics that capture the attack data complexity

$$\mathrm{SR} \leq 1 - (1 - \mathrm{MI}(Y; \boldsymbol{L}_Y))^m$$

$\Rightarrow$ # of traces m to reach $\mathrm{SR} \approx 1 \propto \frac{c(n)}{\mathrm{MI}(Y; \boldsymbol{L}_Y)}$

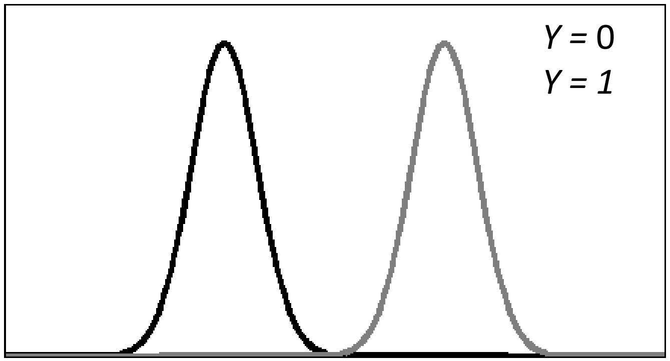
- Attacks target two secrets in parallel
  - The block cipher long-term key
  - The leakage model of the implementation

$\Rightarrow$ An optimal attack requires a perfect model

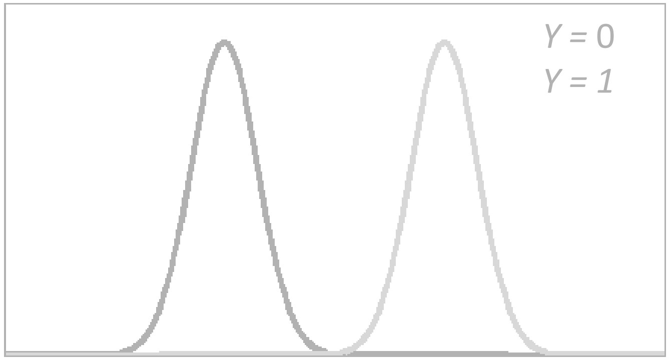
Outline

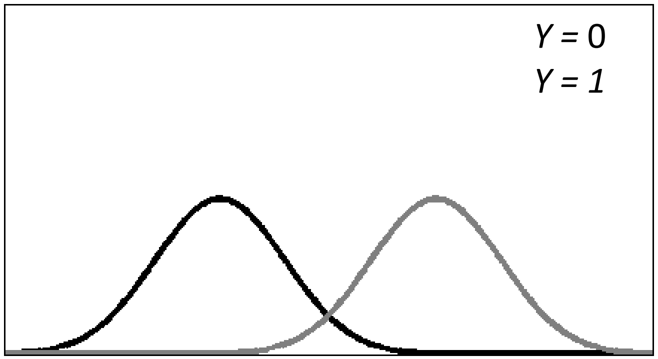
- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

Noise (hardware) is not enough

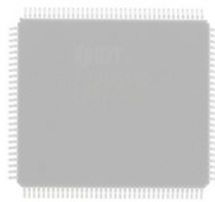


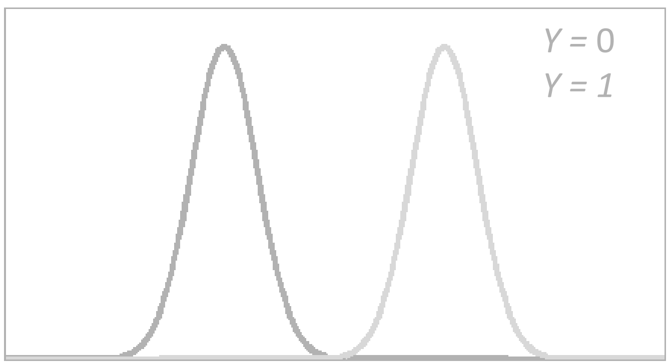
Noise (hardware) is not enough

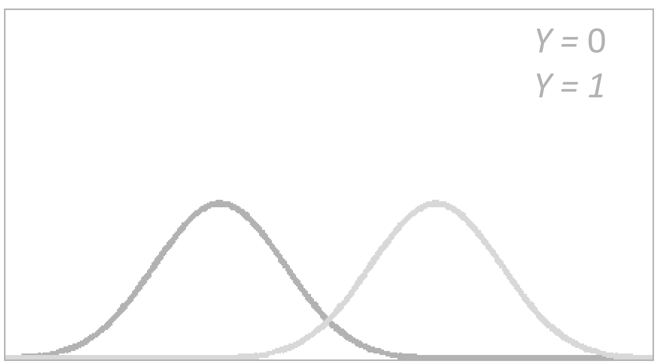




Noise (hardware) is not enough







- Additive noise ≈ cost × 2 ⇒ security × 2
  ⇒ not a good (crypto) security parameter
- ≈ same holds for all hardware countermeasures

Masking ($\approx$ noise amplification)

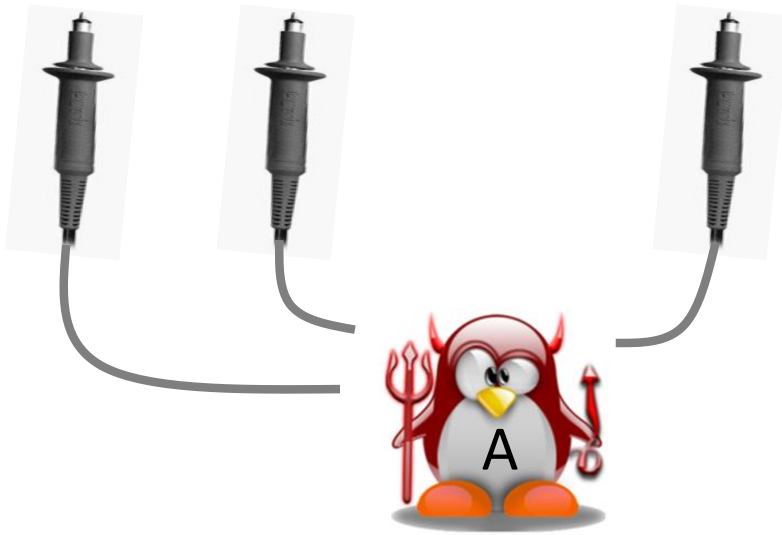
- Example: Boolean encoding

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

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

- Private circuits / probing security [ISW03]

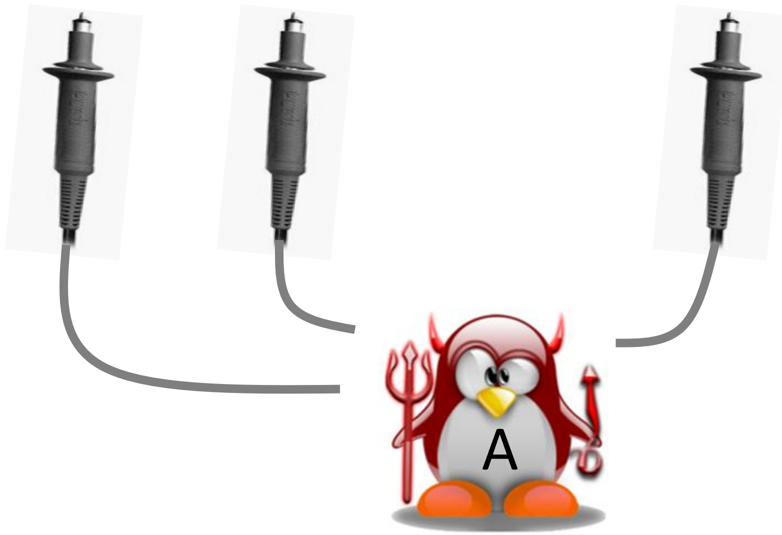
$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$



Masking (abstract view)

- Private circuits / probing security [ISW03]

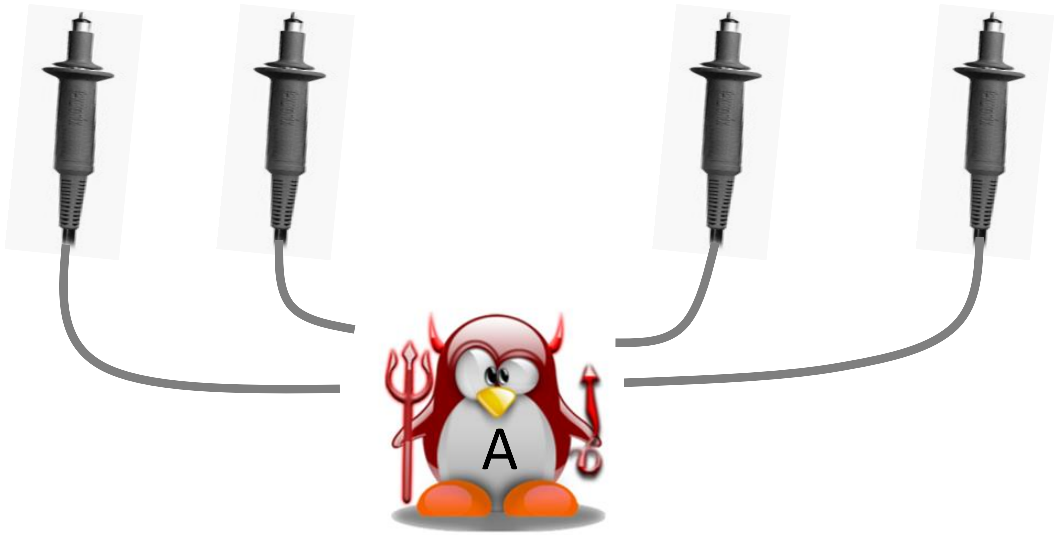
$$y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d$$



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

- Private circuits / probing security [ISW03]

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

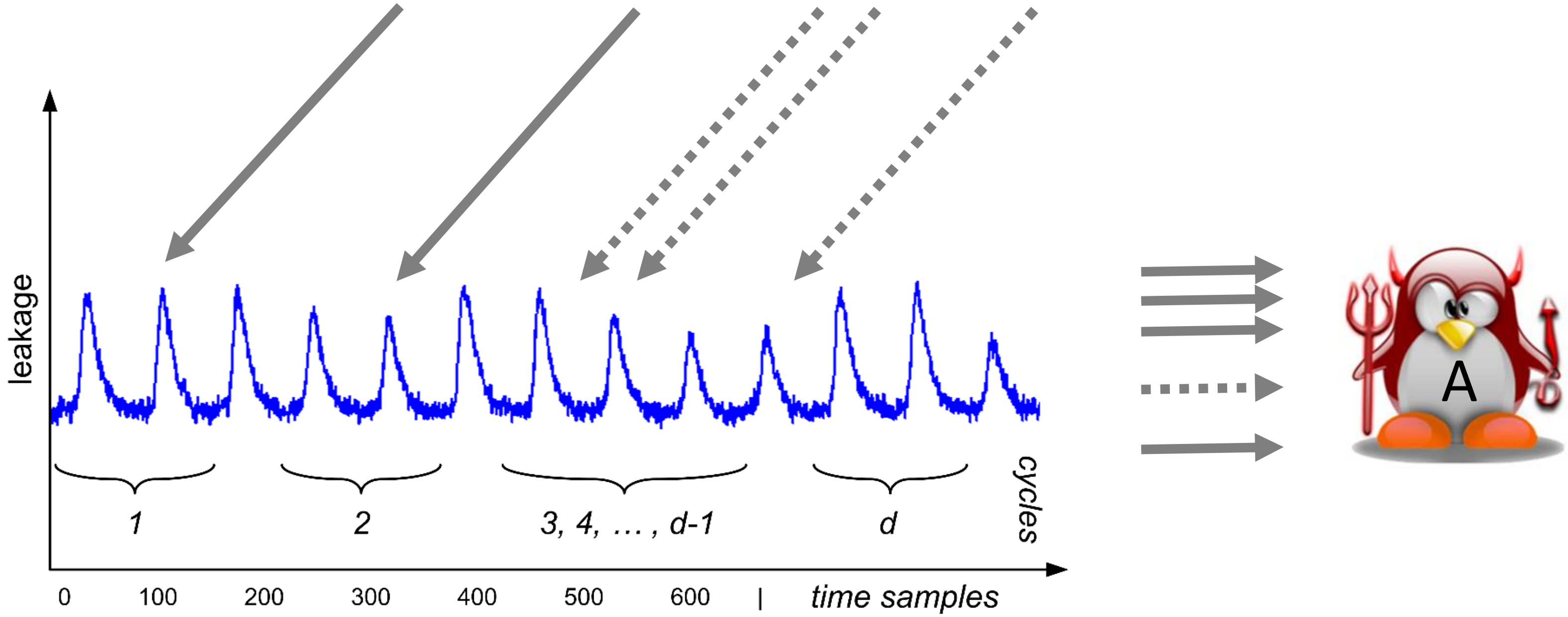


- But d probes completely reveal y

Masking (concrete view)

- Private circuits / probing security [ISW03]

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



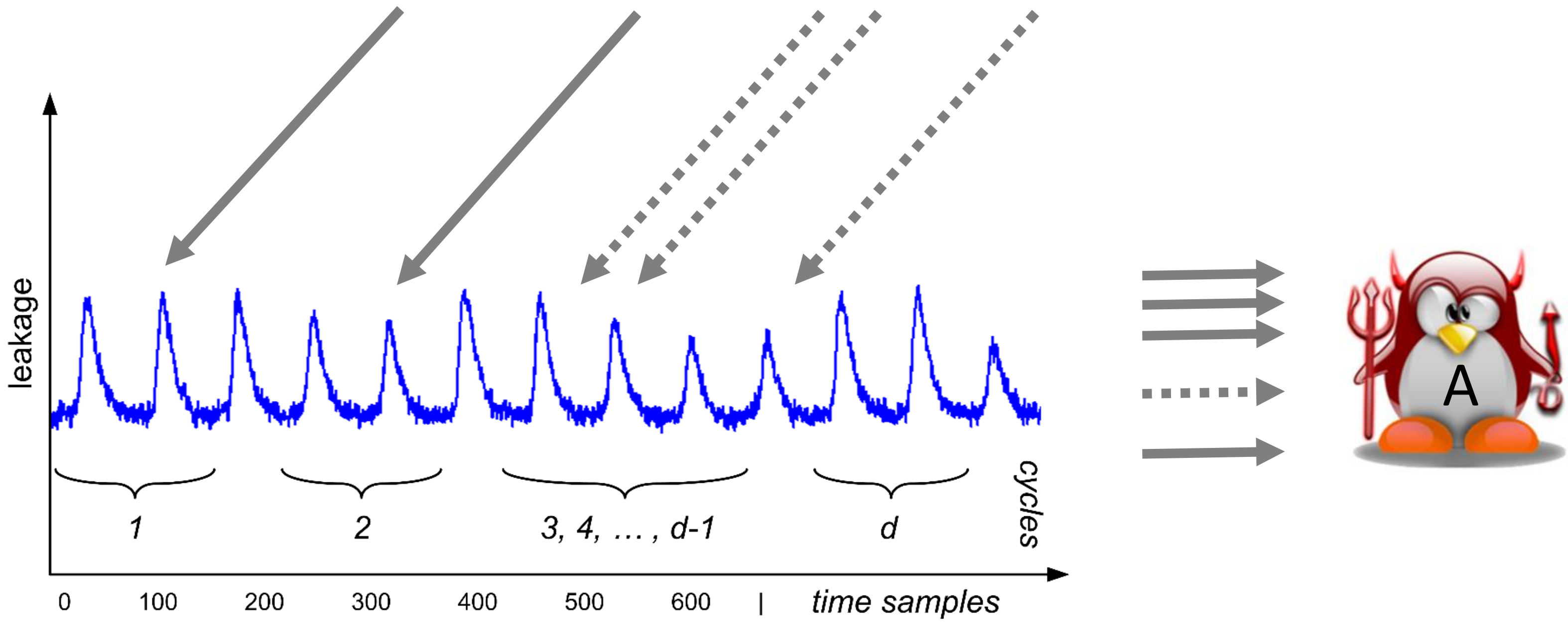
serial implementation.

- Noisy leakage security [PR13]

Masking (concrete view)

- Private circuits / probing security [ISW03]

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



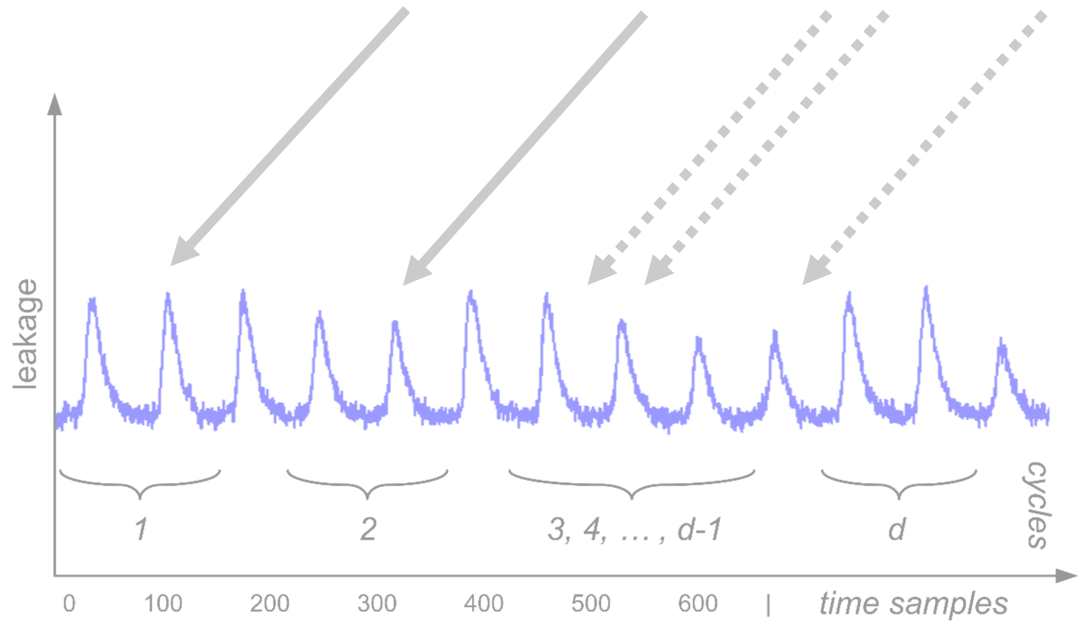
serial implementation.

- Bounded information $\mathrm{MI}(Y;\boldsymbol{L})<\mathrm{MI}(Y_i;\boldsymbol{L}_{Y_i})^d$

Masking (reduction)

- Private circuits / probing security [ISW03]

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



serial implementation.

noise + independence [DDF14]

- Bounded information $\mathrm{MI}(Y; \boldsymbol{L}) < \mathrm{MI}(Y_i; \boldsymbol{L}_{Y_i})^d$

Masked operations [ISW03]

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$

Masked operations [ISW03]

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$
- Multiplications: $\mathrm{c} = a \times b$ in three steps

Masked operations [ISW03]

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$
- Multiplications: $\mathrm{c} = a \times b$ in three steps

$$\begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$

partial products

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$

- Multiplications: $\mathrm{c} = a \times b$ in three steps

$$\begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} + \begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix}$$

partial products refreshing

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$

- Multiplications: $\mathrm{c} = a \times b$ in three steps

$$\begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix} + \begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

partial products refreshing compression

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$

- Multiplications: $\mathrm{c} = a \times b$ in three steps

$$\begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix} + \begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

partial products **refreshing** compression

$a_1 b_1 \oplus a_1 b_2 \oplus a_1 b_3 = \boldsymbol{a_1 b}$ leaks on b

- Linear operations: $\mathrm{f(a)} = \mathrm{f}(a_1) \oplus \mathrm{f}(a_2) \oplus \cdots \oplus \mathrm{f}(a_d)$

- Multiplications: $\mathrm{c} = a \times b$ in three steps

$$\begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} + \begin{bmatrix} 0 & r_1 & r_2 \\ -r_1 & 0 & r_3 \\ -r_2 & -r_3 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

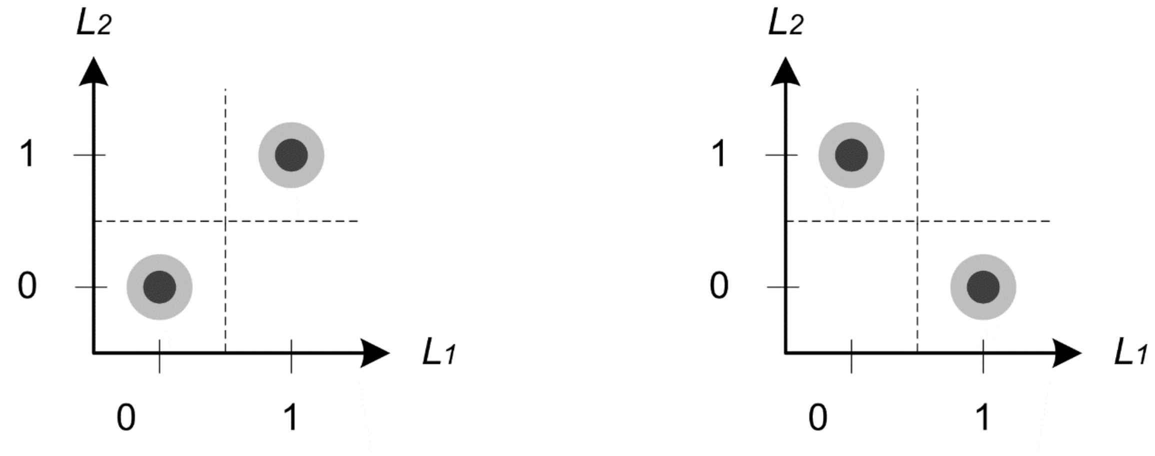
partial products **refreshing** compression

$a_1b_1 \oplus a_1b_2 \oplus a_1b_3 = \boldsymbol{a_1 b}$ leaks on b

$\Rightarrow$ Quadratic overheads & randomness

- (Many published optimizations [R+15,Be+16,GM18])

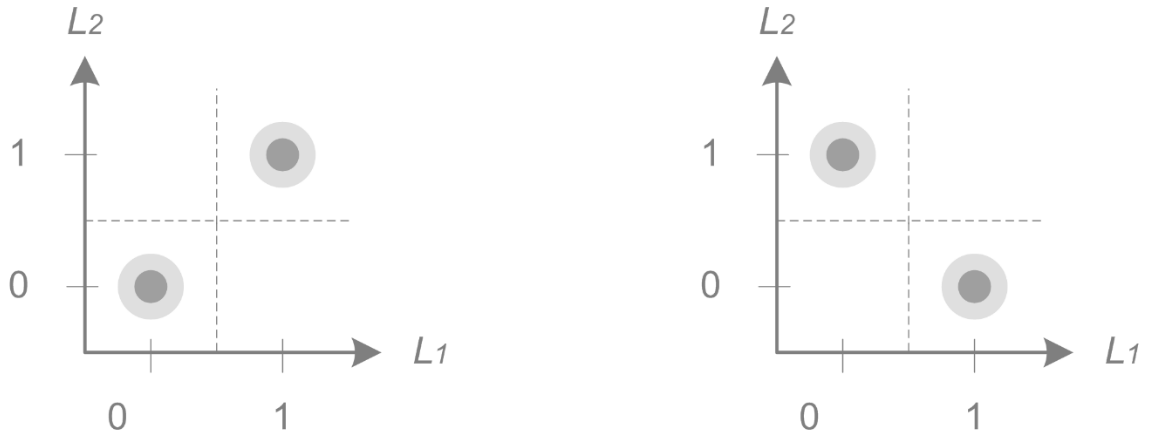
Statistical intuition (2 shares)



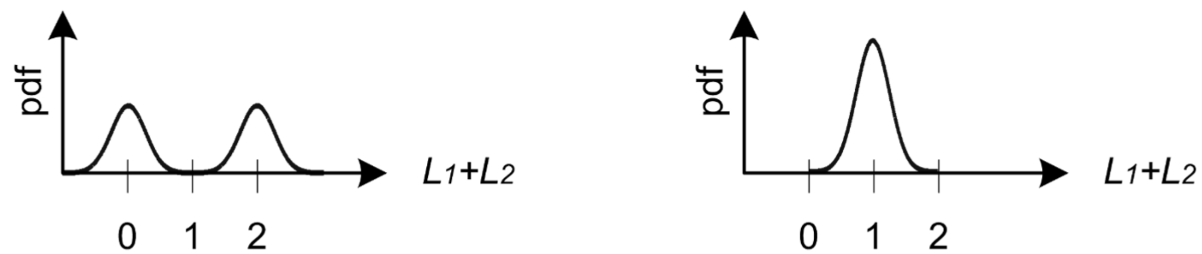
(a) $Y = 0$, serial. (b) $Y = 1$, serial.

- Leakage mean vector for $Y = 0{,}1 = [0.5\ 0.5]$

Statistical intuition (2 shares)

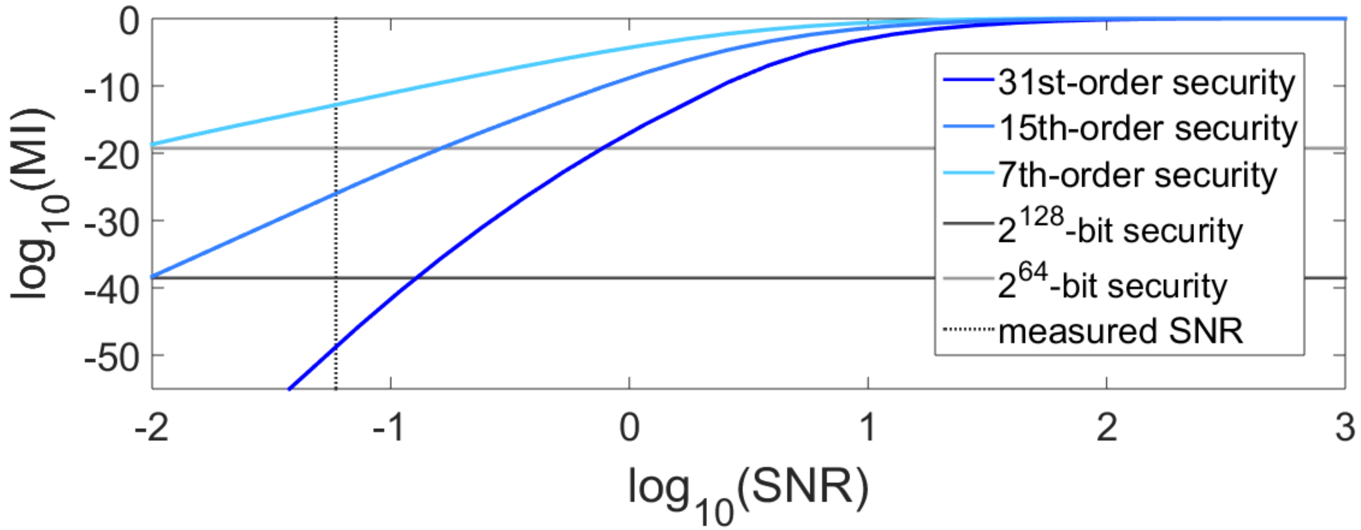


(a) $Y = 0$, serial. (b) $Y = 1$, serial.



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

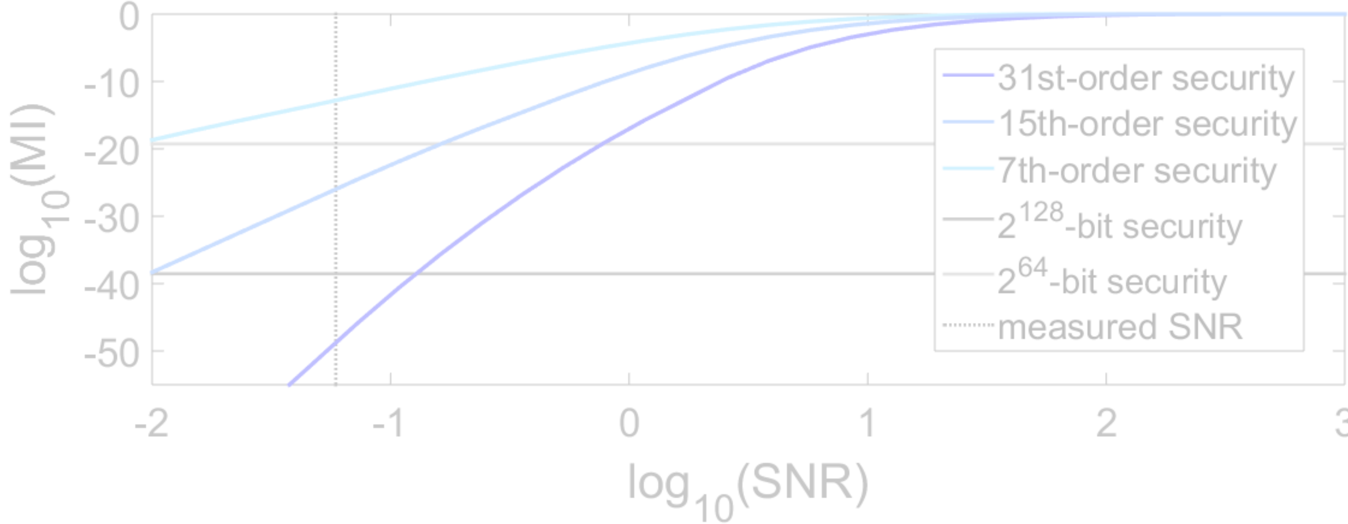
- Leakage mean value for $Y = 0{,}1 = 1$



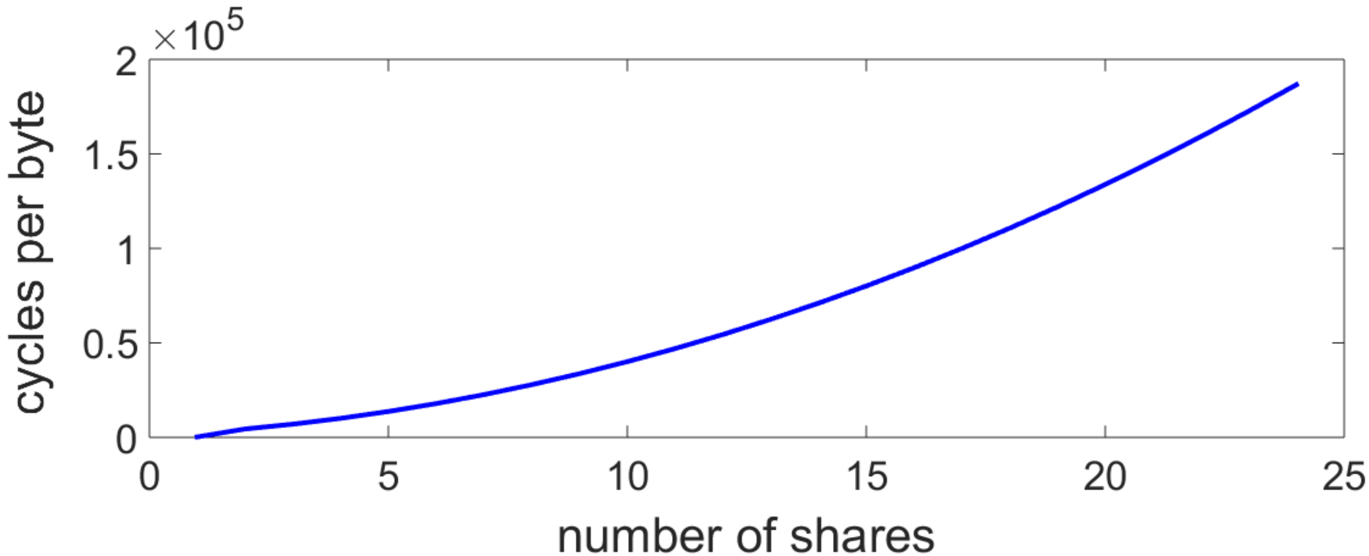
0
-10
-20
-30
-40
-50
$\log_{10}$(MI)
-2
-1
0
1
2
3
$\log_{10}$(SNR)
31st-order security
15th-order security
7th-order security
2^{128}-bit security
2^{64}-bit security
measured SNR

security

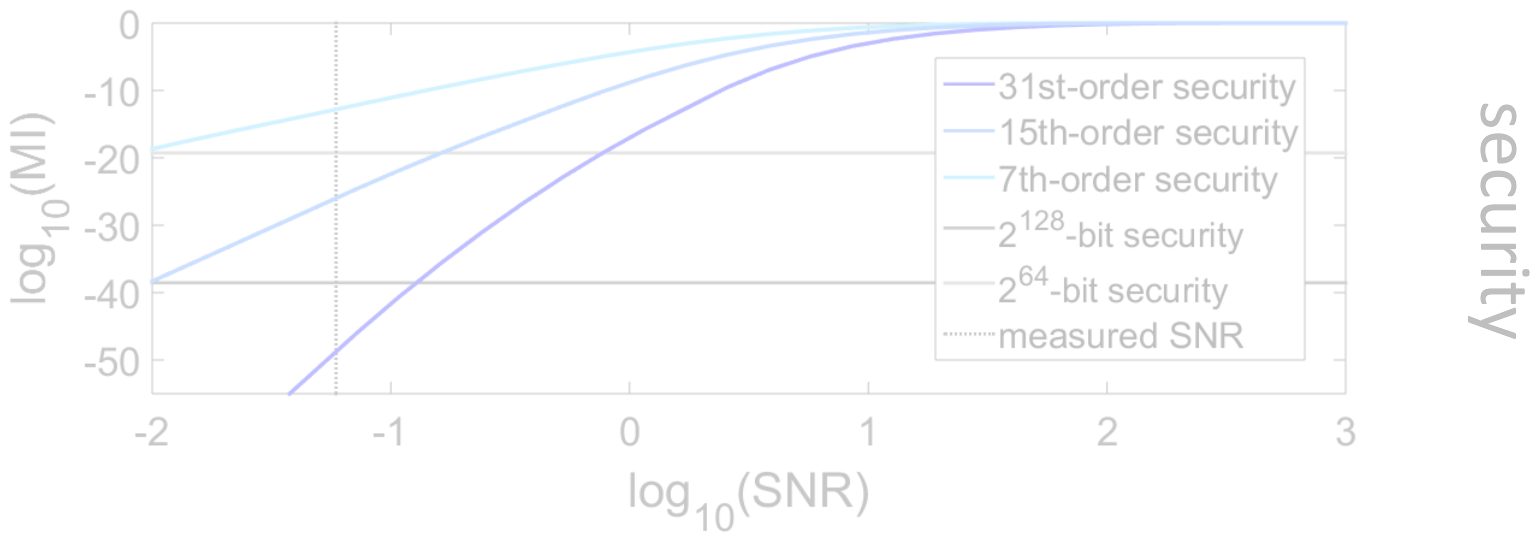
Case study: ARM Cortex M4 [JS17]

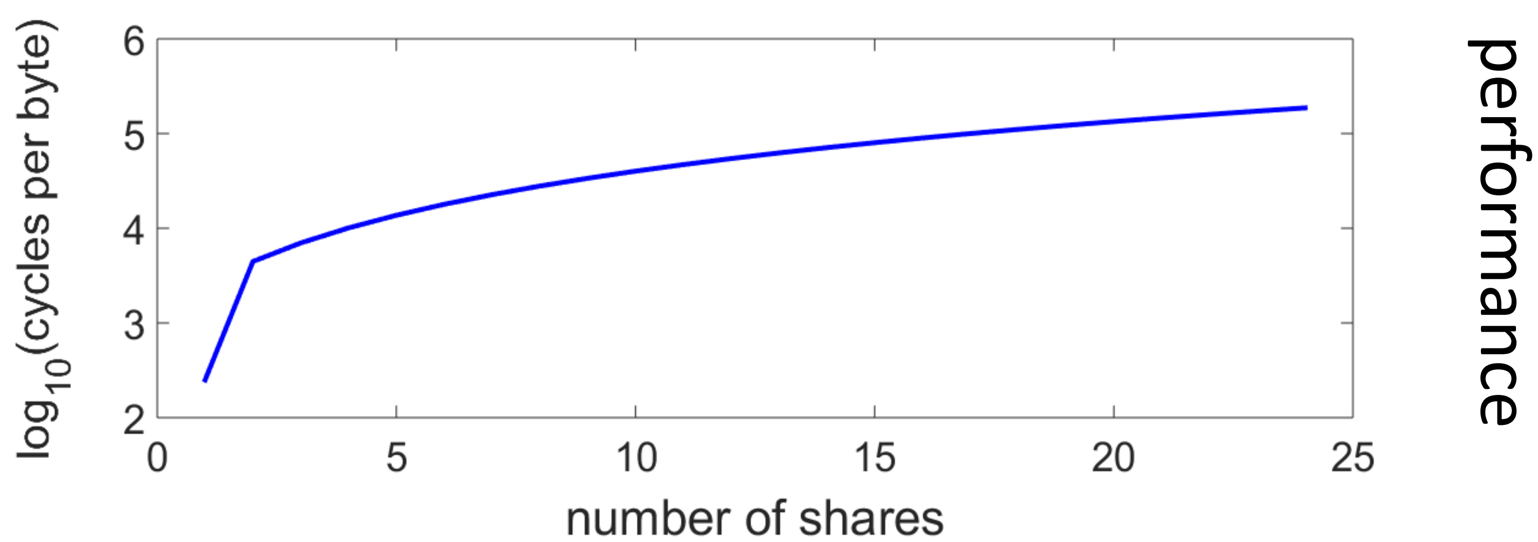


security



performance


0
-10
-20
-30
-40
-50
$\log_{10}(\mathrm{MI})$
31st-order security
15th-order security
7th-order security
2^{128}-bit security
2^{64}-bit security
measured SNR
-2
-1
0
1
2
3
$\log_{10}(\mathrm{SNR})$
security


6
5
4
3
2
$\log_{10}$(cycles per byte)
0
5
10
15
20
25
number of shares
performance

Summarizing

- Sounds easy but implementation is complex

Summarizing

- Sounds easy but implementation is complex
  - ***Independence issue***: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  - Security against horizontal attacks require more ***noise/randomness*** as d increases [BCPZ16,CS19]
  - Scalability/***composition*** are challenging [Ba+15,Ba+16]

- Sounds easy but implementation is complex
  - ***Independence issue***: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  - Security against horizontal attacks require more ***noise/randomness*** as d increases [BCPZ16,CS19]
  - Scalability/***composition*** are challenging [Ba+15,Ba+16]

⇒ High security against DPA can be reached but

- It implies large performance overheads
  - E.g., industry currently uses 2-4 shares (?)
- It « only » protects the key (plaintexts are not shared)

- Sounds easy but implementation is complex
  - ***Independence issue***: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  - Security against horizontal attacks require more ***noise/randomness*** as d increases [BCPZ16,CS19]
  - Scalability/***composition*** are challenging [Ba+15,Ba+16]

⇒ High security against DPA can be reached but

- It implies large performance overheads
  - E.g., industry currently uses 2-4 shares (?)
- It « only » protects the key (plaintexts are not shared)

- SPA security expected to be (much) cheaper

Outline

- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

Leakage evaluation

1. Directly estimate the leakage PDF (or PMF)
2. Try to attack with this estimated model
   - Good if it works (but no guarantees of optimality)
   - Hard to interpret if it does not work:
     - either the leakages are sufficiently noisy, or
     - the model is not accurate ("*false sense of security*")

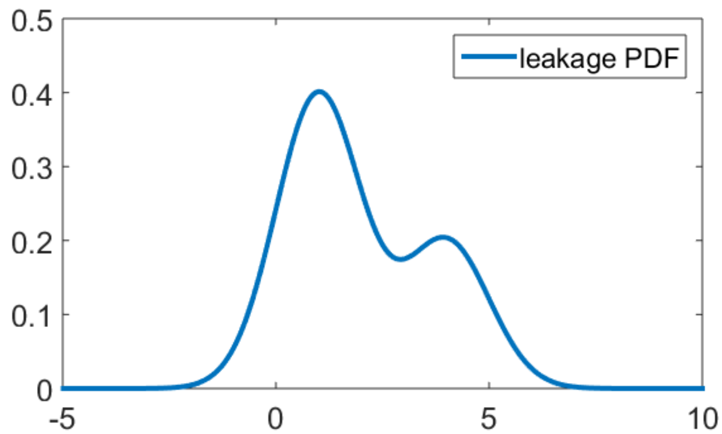
Leakage certification [DSV14]

1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

Leakage certification [DSV14]

1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

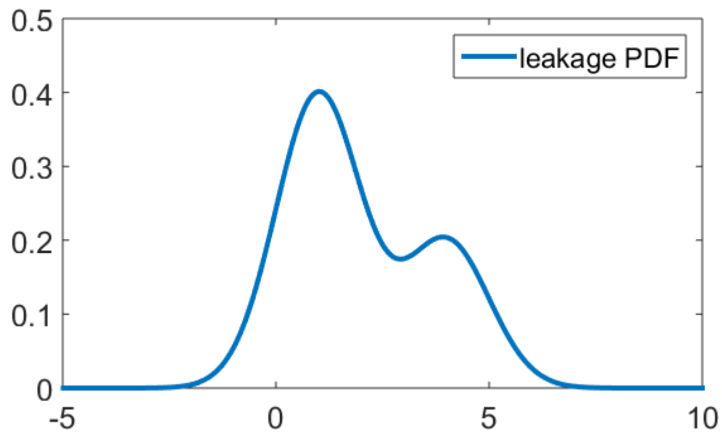
- Example:



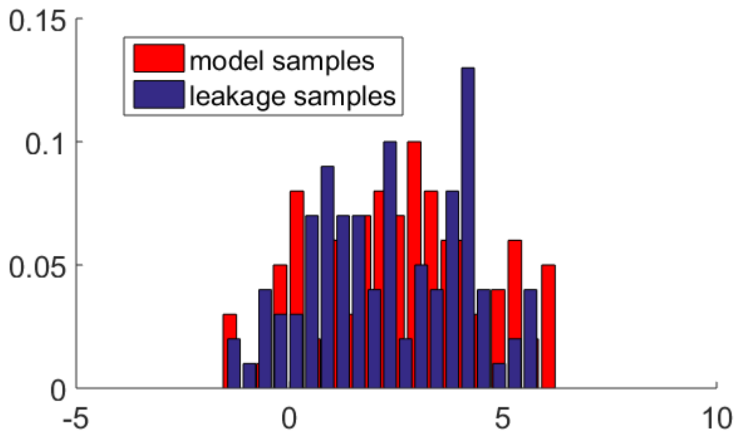
Leakage certification [DSV14]

1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

- Example:



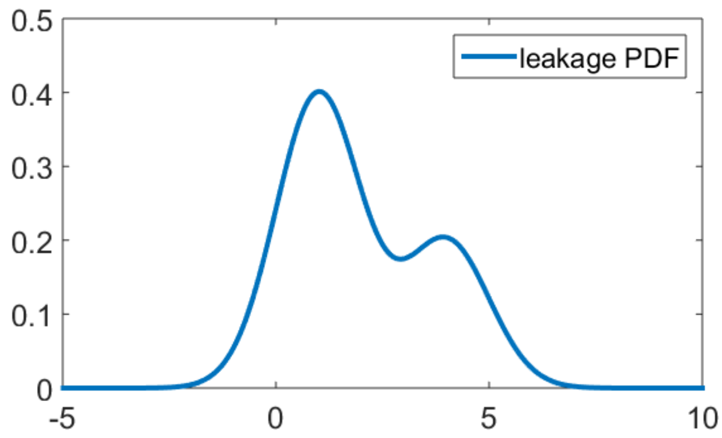
n_0 samples

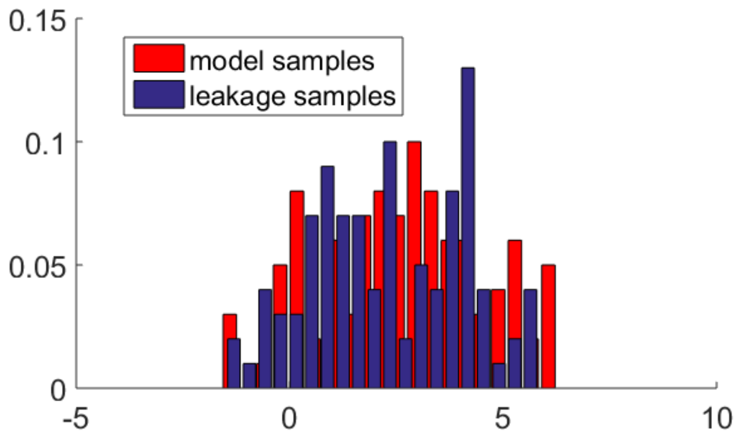


1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

- Example:

estimation errors dominate

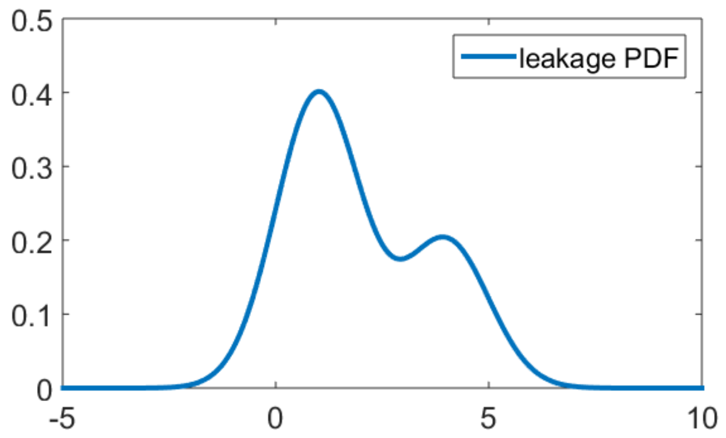




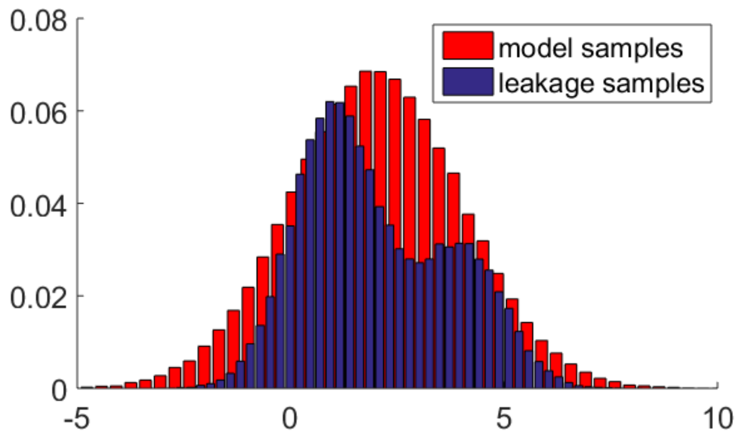
$\Rightarrow$ need to measure more

1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

- Example:

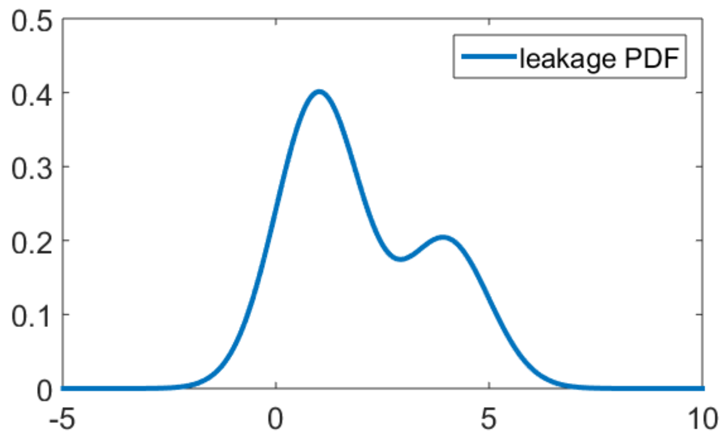


$n_1 > n_0$ samples

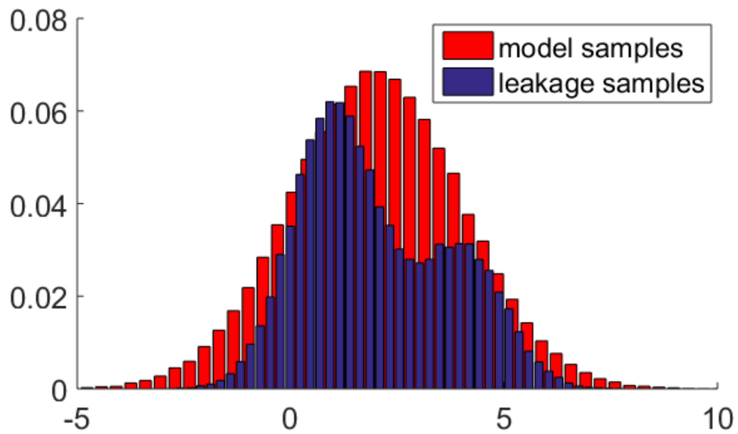


1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

- Example:



assumption errors dominate

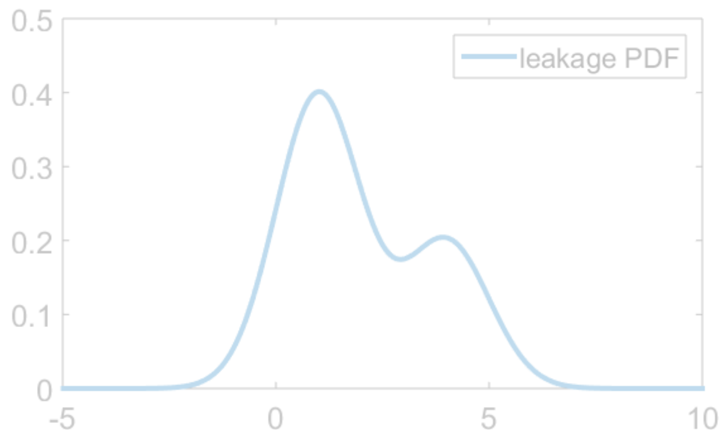


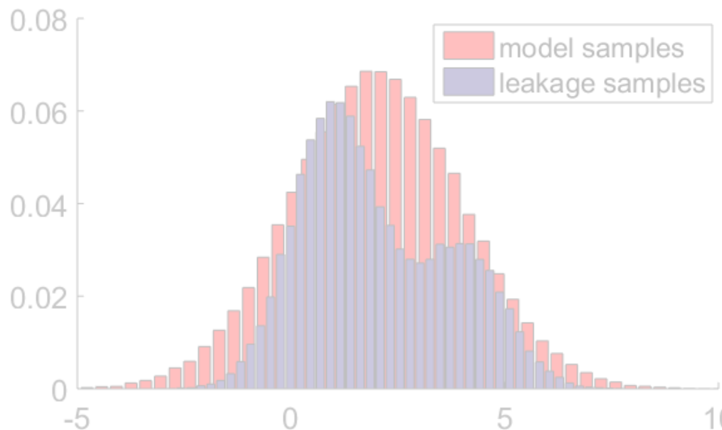
$\Rightarrow$ need another statistical model

1. Directly estimate the leakage PDF (or PMF)
2. Try do distinguish estimation & assumption errors

- Example:

assumption errors dominate





$\Rightarrow$ need another statistical model

$\Rightarrow$ good enough model: *ass. err* << *est. err*. given *n*

Information theoretic view [R+11]

$$\mathrm{PI}\left(Y_i; \boldsymbol{L}_{Y_i}\right) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \int_{l} \mathrm{f}\left(\boldsymbol{l}_{y_i} \middle| y_i\right) \cdot \log_2 \tilde{\mathrm{m}}_n\left(y_i \middle| \boldsymbol{l}_{y_i}\right) dl$$

- Information extracted by a statistical model
  - Possibly biased by estimation & assumption errors

Information theoretic view [R+11]

$$\widehat{\mathrm{PI}}\left(Y_i; \boldsymbol{L}_{Y_i}\right) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i})$$

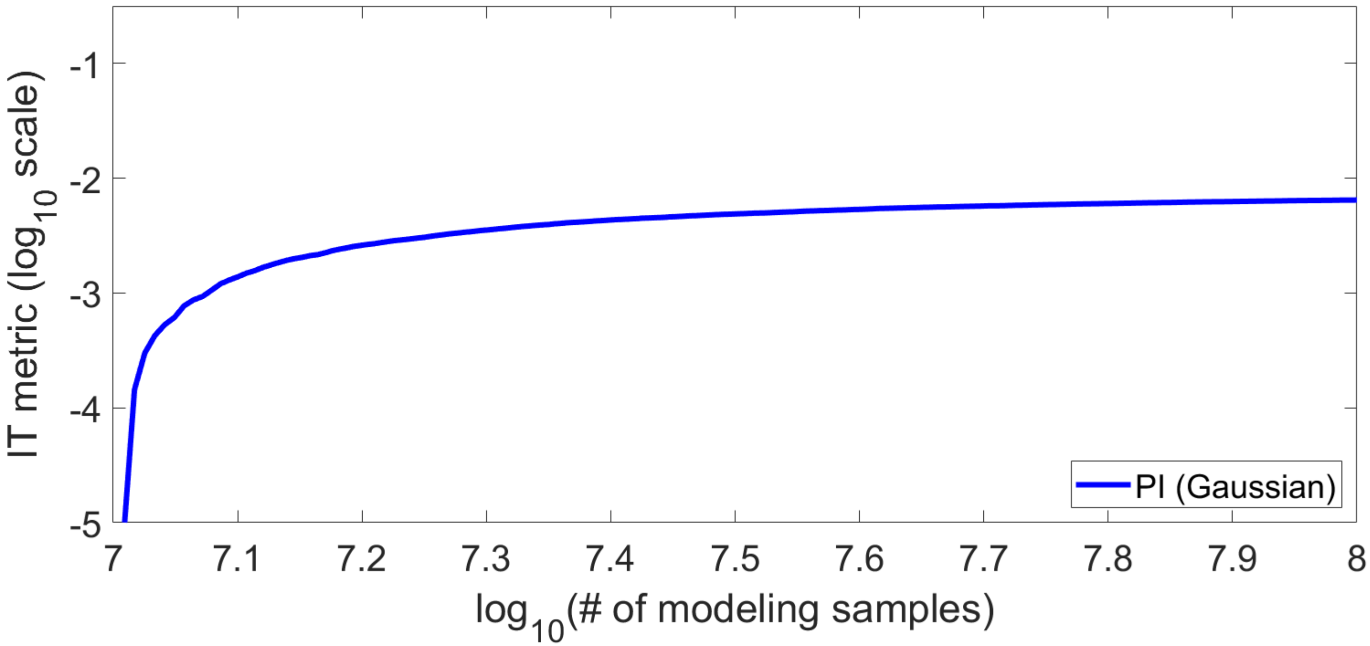
- Information extracted by a statistical model
  - Possibly biased by estimation & assumption errors
- Computed in two 2-steps: (1) model estimation (2) integral by sampling (the true distribution)

$$\widehat{\mathrm{PI}}\left(Y_i;\boldsymbol{L}_{Y_i}\right)=\mathrm{H}(Y_i)+\sum_{y}\mathrm{p}(y_i)\cdot\sum_{j=1}^{n_t(y_i)}\frac{1}{n_t(y_i)}\cdot\log_2\tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i})$$

- Information extracted by a statistical model
  - Possibly biased by estimation & assumption errors
- Computed in two 2-steps: (1) model estimation (2) integral by sampling (the true distribution)
- $\mathrm{PI}=\mathrm{MI}$ if the model is perfect ($\mathrm{PI}\neq\mathrm{MI}$ otherwise)
  - E.g., can be negative if the model is too incorrect

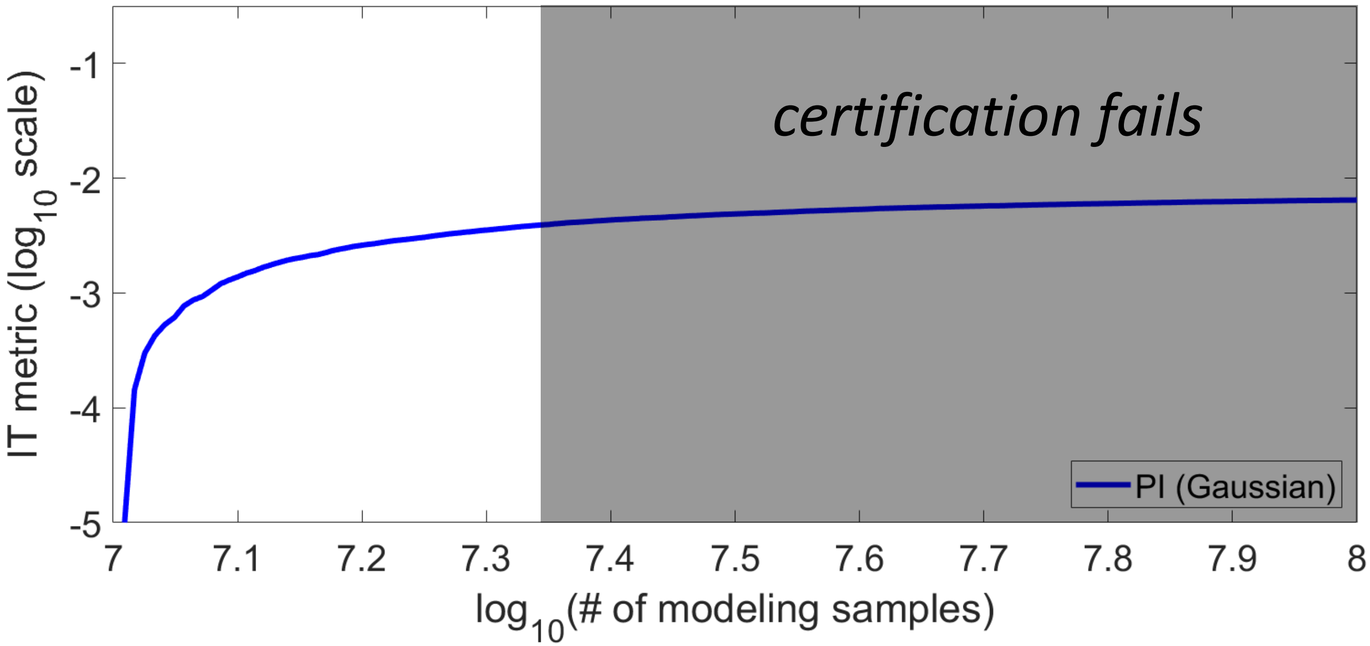
Information theoretic view [R+11]

$$\widehat{\mathrm{PI}}(Y_i; \boldsymbol{L}_{Y_i}) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i})$$



Information theoretic view [R+11]

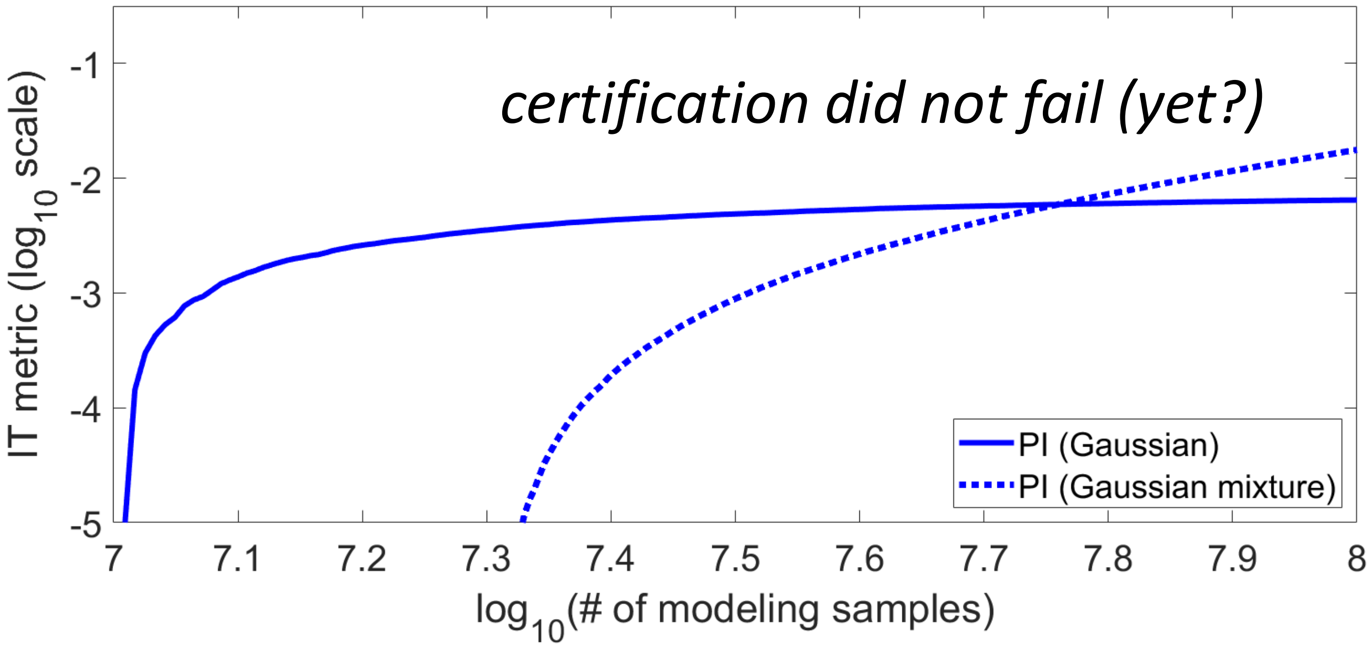
$$\widehat{\mathrm{PI}}(Y_i;\boldsymbol{L}_{Y_i}) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i})$$



- PI curve “saturates” too far from the MI
  - Evaluator has to look for another statistical model

Concrete limitation #1

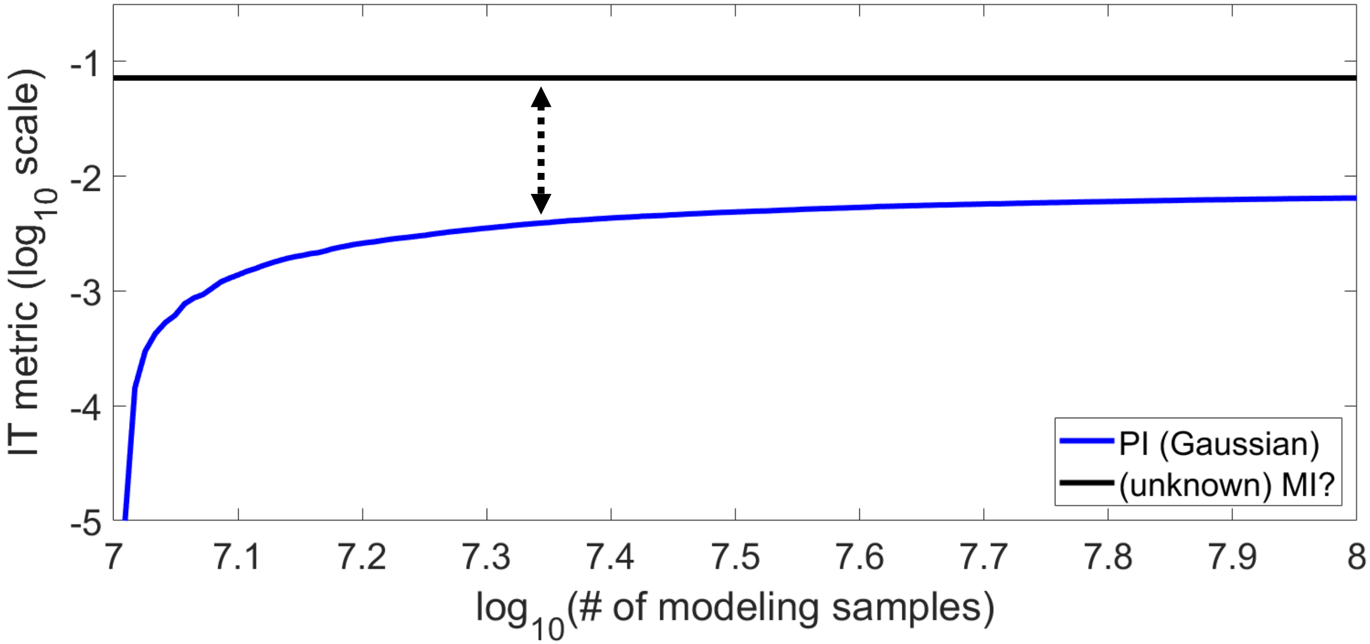
$$\widehat{\mathrm{PI}}(Y_i; \boldsymbol{L}_{Y_i}) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i})$$



- We may lack samples to be conclusive
  - Because estimation errors decrease slowly

Concrete limitation #2

$$\widehat{\mathrm{PI}}(Y_i; \boldsymbol{L}_{Y_i}) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i})$$



- Such certification tests are only qualitative
  - They give no indication about the security loss

Outline

- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

Hypothetical Information [DSM16]

$$\mathrm{HI}\left(Y_i; \boldsymbol{L}_{Y_i}\right) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{l} \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i}) \cdot \log_2 \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i})$$

- Information that would be extractable from the samples ***if*** the true distribution was the model

Hypothetical Information [DSM16]

$$\mathrm{HI}\big(Y_i; \boldsymbol{L}_{Y_i}\big) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{l} \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i}) \cdot \log_2 \tilde{\mathrm{m}}_n(y_i | \boldsymbol{l}_{y_i})$$

- Information that would be extractable from the samples ***if*** the true distribution was the model

+ Easier/faster to compute (known distribution)

Hypothetical Information [DSM16]

$$\mathrm{HI}\left(Y_i; \boldsymbol{L}_{Y_i}\right) = \mathrm{H}(Y_i) + \sum_{y} \mathrm{p}(y_i) \cdot \sum_{l} \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i}) \cdot \log_2 \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i})$$

- Information that would be extractable from the samples ***if*** the true distribution was the model

  \+ Easier/faster to compute (known distribution)

  – Disconnected from the true distribution
  - Remains positive even if model is incorrect

Hypothetical Information [DSM16]

$$\mathrm{HI}\left(Y_i; \boldsymbol{L}_{Y_i}\right) = \mathrm{H}(Y_i) + \sum_y \mathrm{p}(y_i) \cdot \sum_l \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i}) \cdot \log_2 \tilde{\mathrm{m}}_n(y_i|\boldsymbol{l}_{y_i})$$

- Information that would be extractable from the samples ***if*** the true distribution was the model
  - \+ Easier/faster to compute (known distribution)
  - – Disconnected from the true distribution
    - Remains positive even if model is incorrect
    - Unless specific model families are considered
      - Next: empirical distribution $\mathrm{eHI}_n\left(Y_i; \boldsymbol{L}_{Y_i}\right)$

- Upper bound for the MI metric

$$\mathop{\mathrm{E}}_{\mathcal{M}}\left(\mathrm{eHI}_n\left(Y_i;\boldsymbol{L}_{Y_i}\right)\right) \geq \mathrm{MI}\left(Y_i;\boldsymbol{L}_{Y_i}\right)$$

$$\lim_{n\to\infty} \mathrm{eHI}_n\left(Y_i;\boldsymbol{L}_{Y_i}\right) = \mathrm{MI}\left(Y_i;\boldsymbol{L}_{Y_i}\right)$$

Bounds for the Mutual Information

- Upper bound for the MI metric

$$\mathop{\mathrm{E}}_{\mathcal{M}}\left(\mathrm{eHI}_n\big(Y_i;\boldsymbol{L}_{Y_i}\big)\right) \geq \mathrm{MI}\big(Y_i;\boldsymbol{L}_{Y_i}\big)$$

$$\lim_{n\to\infty} \mathrm{eHI}_n\big(Y_i;\boldsymbol{L}_{Y_i}\big) = \mathrm{MI}\big(Y_i;\boldsymbol{L}_{Y_i}\big)$$

  - Uniform (constant) distribution for the secret Y_i
    ⇒ MI biased upwards everywhere (like the entropy)
  - Monotonic convergence of the empirical distrib.

Bounds for the Mutual Information

- Upper bound for the MI metric

$$\mathop{\mathrm{E}}_{\mathcal{M}}\left(\mathrm{eHI}_n\left(Y_i;\boldsymbol{L}_{Y_i}\right)\right) \geq \mathrm{MI}\left(Y_i;\boldsymbol{L}_{Y_i}\right)$$

$$\lim_{n\to\infty} \mathrm{eHI}_n\left(Y_i;\boldsymbol{L}_{Y_i}\right) = \mathrm{MI}\left(Y_i;\boldsymbol{L}_{Y_i}\right)$$

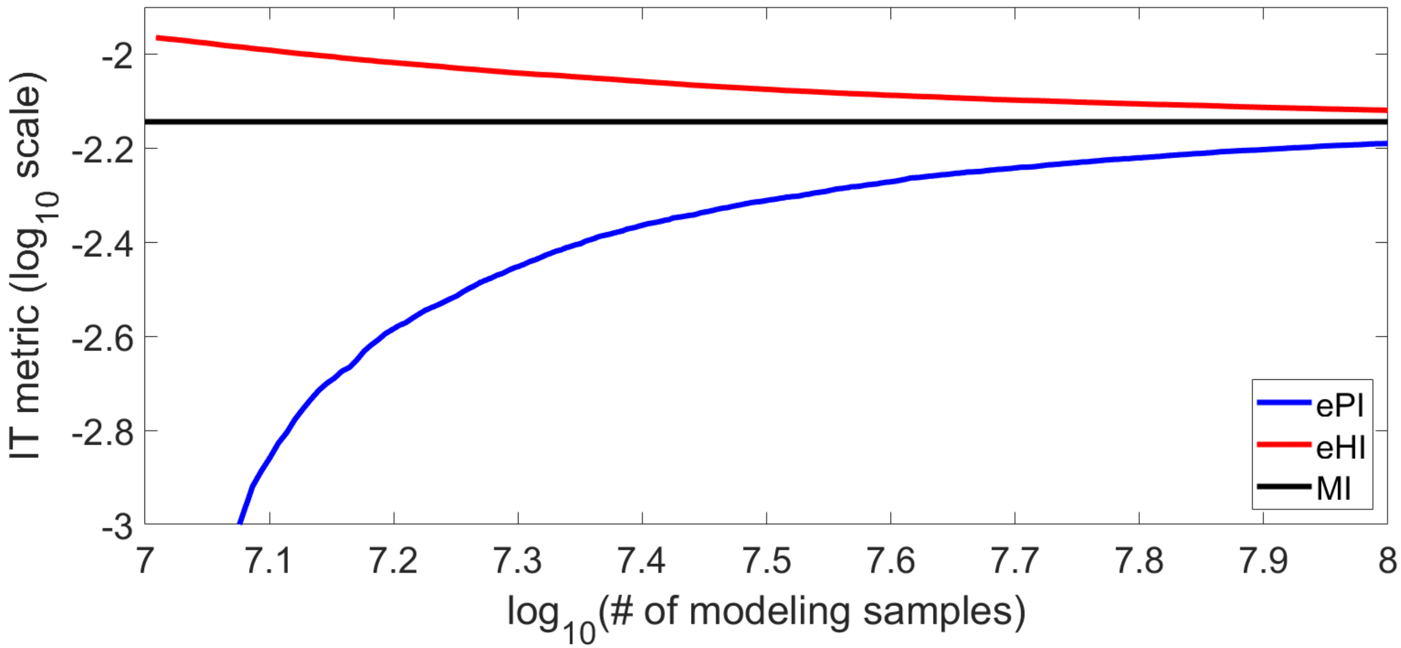
  - Uniform (constant) distribution for the secret Y_i
    $\Rightarrow$ MI biased upwards everywhere (like the entropy)
  - Monotonic convergence of the empirical distrib.

- Lower bound for the MI metric

$$\mathrm{PI}_n\left(Y_i;\boldsymbol{L}_{Y_i}\right) \leq \mathrm{MI}\left(Y_i;\boldsymbol{L}_{Y_i}\right)$$

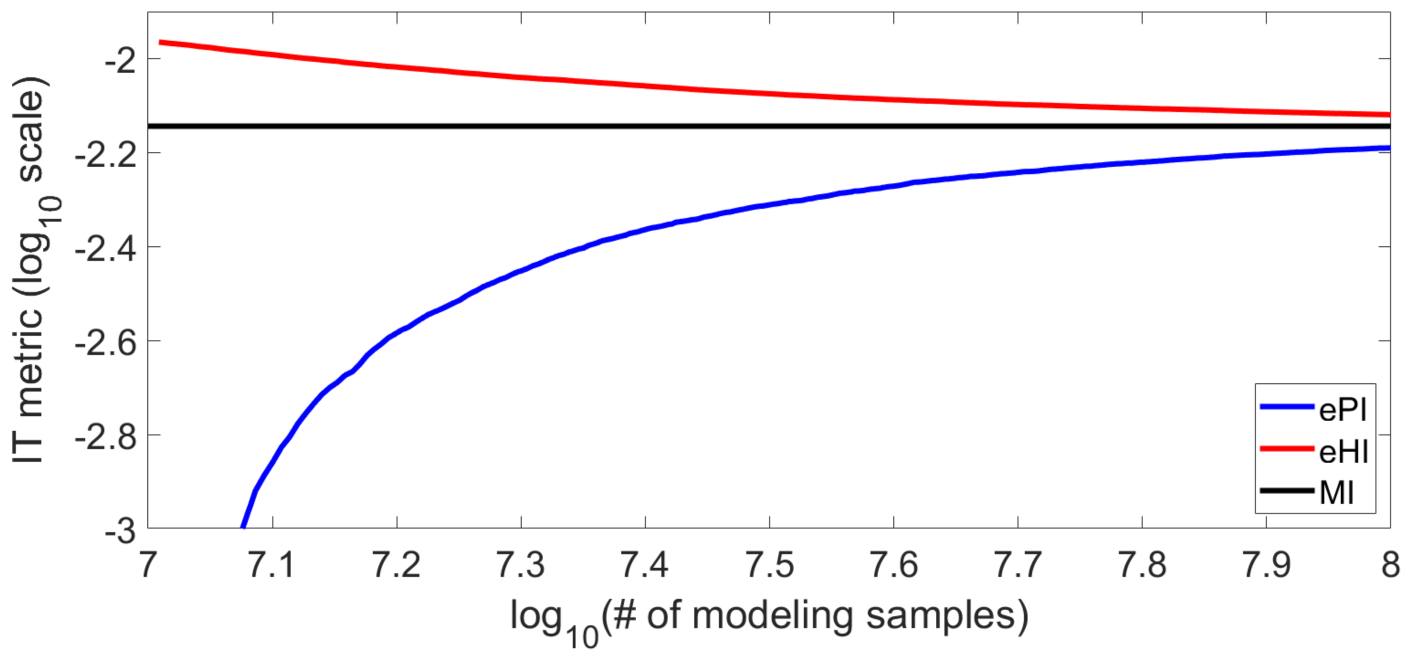
- We can only loose information if $\tilde{\mathrm{m}}_n\left(y_i\middle|\boldsymbol{l}_{y_i}\right) \neq \mathrm{p}\left(y_i|\boldsymbol{l}_{y_i}\right)$

Experimental results (simulations)



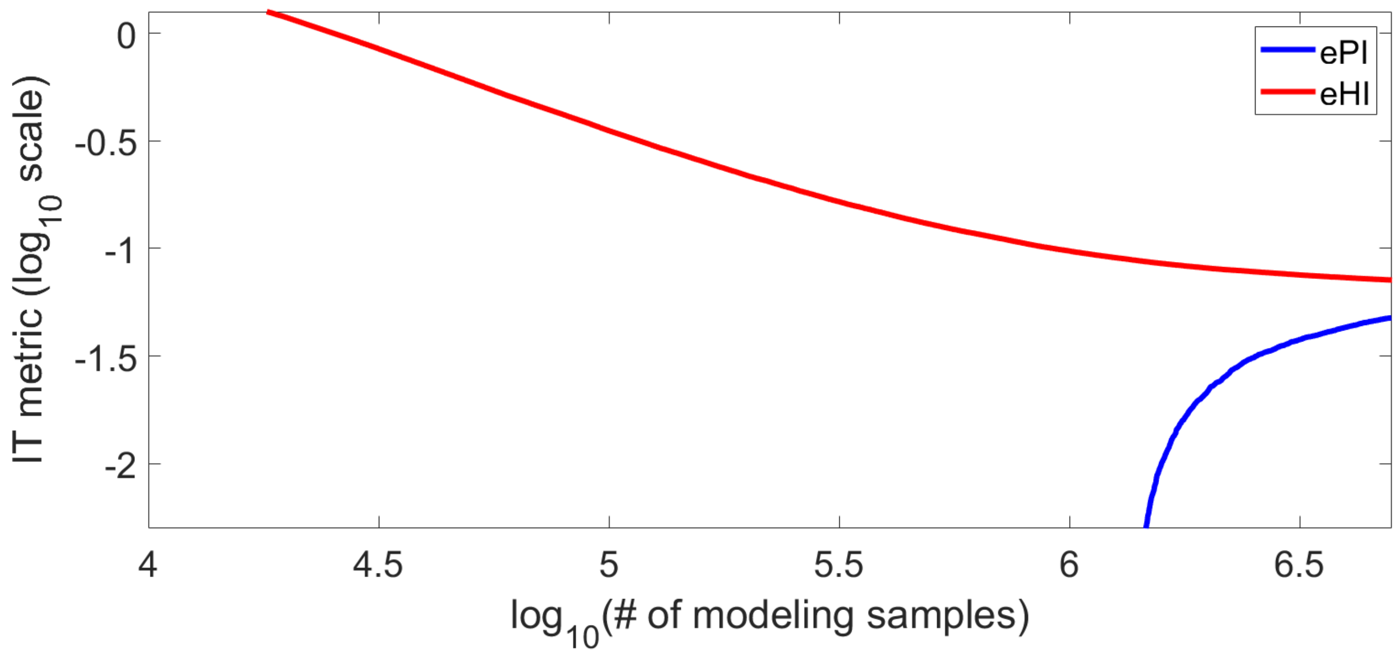
- eHI converges faster than ePI (no cross-validation)

Experimental results (simulations)



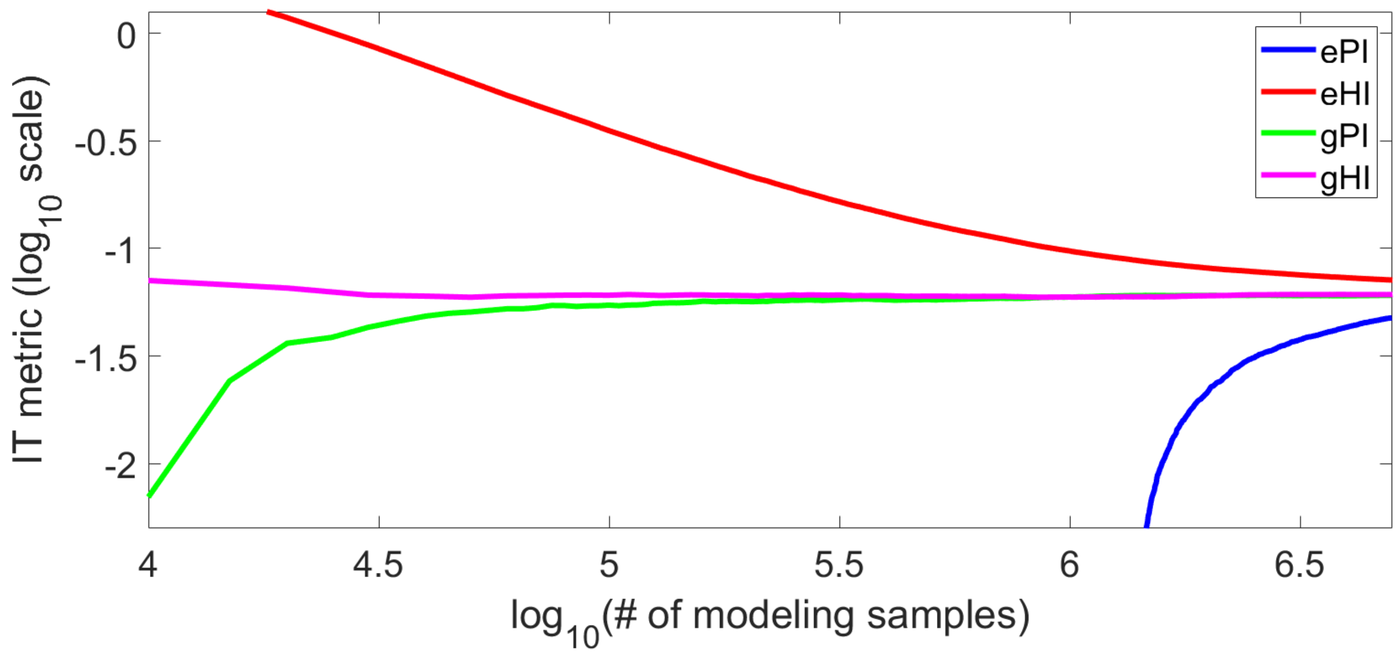
- eHI converges faster than ePI (no cross-validation)
- Bound becomes tighter as *n* increases
  - More eval. efforts lead to better sec. guarantees

Experimental results (real device)



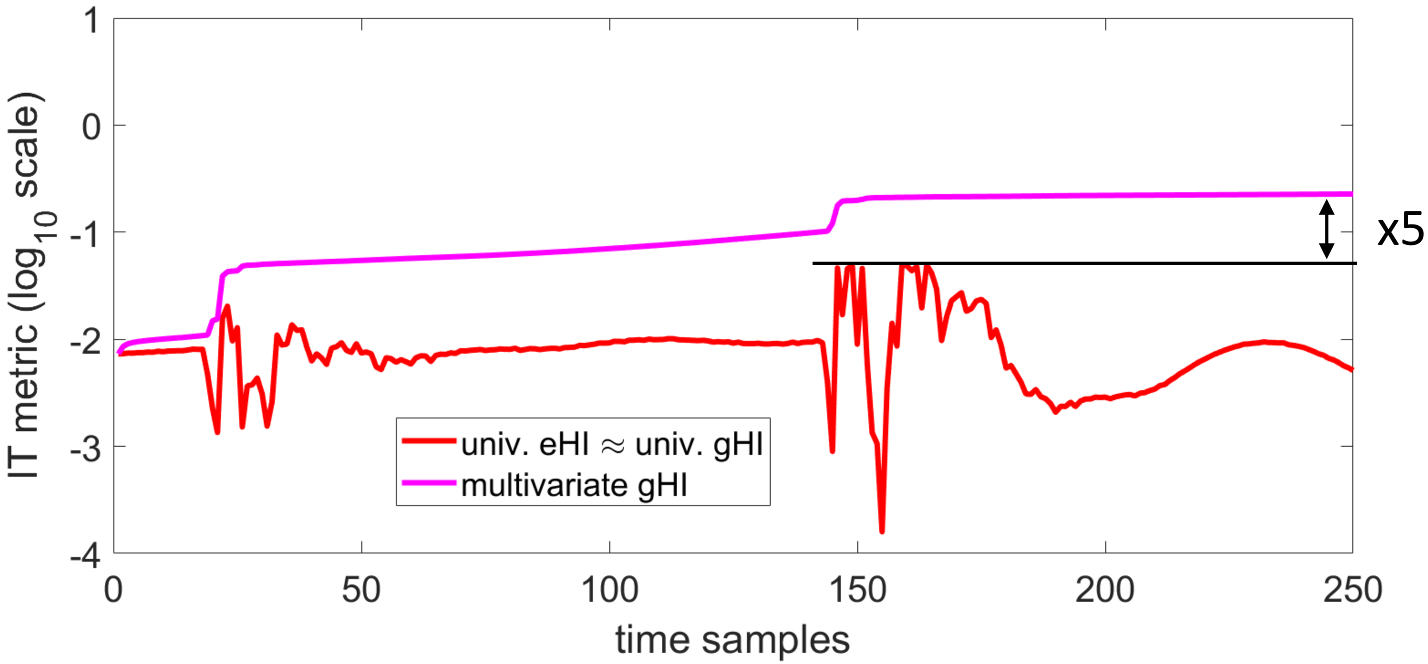
- Quite similar results (but unknown MI & lower *n*)

Experimental results (real device)



- Quite similar results (but unknown MI & lower *n*)
- Gaussian HI/PI converge (much) faster
  - And are close to the eHI/ePI (*in our case study*!)

Multivariate analyzes



- Curse of dimensionality ⇒ need assumptions?
  - (But then the connection with the MI is lost)
- Nice learning problem: multivariate & higher-order
  - Link with statistical learning theory (Vapnik)

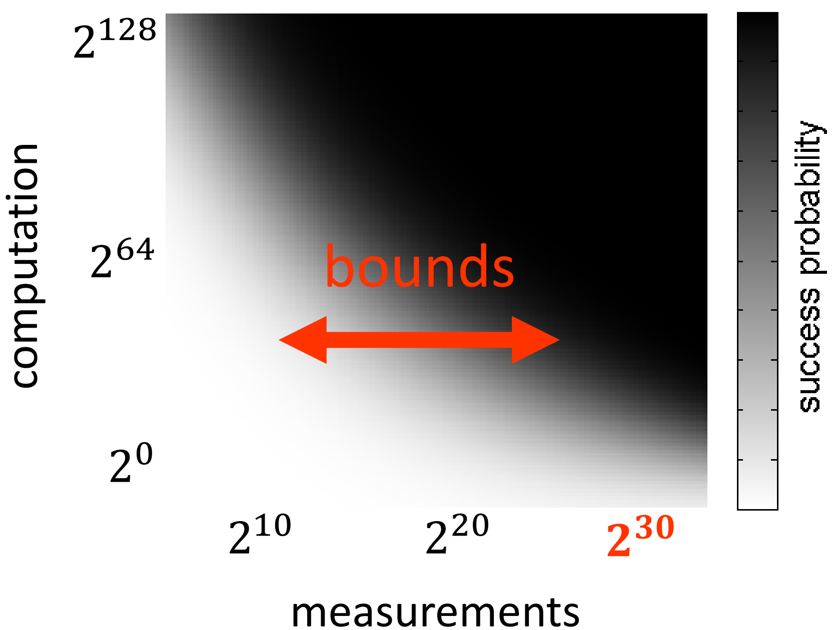
Outline

- Introduction to side-channel analysis
- Masking (aka secret sharing) countermeasure
- Leakage evaluation and certification
  - Problem statement & first approach
  - Bounding the Perceived Information
- Conclusions: white box design & evaluation

Evaluation challenge

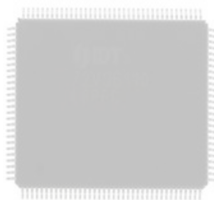
standard practice

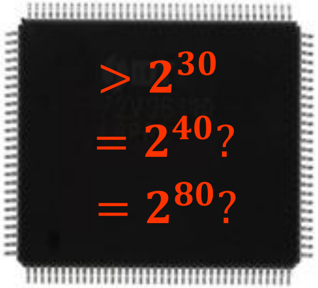
evidence-based evaluations
(*assumptions tested per device!*)



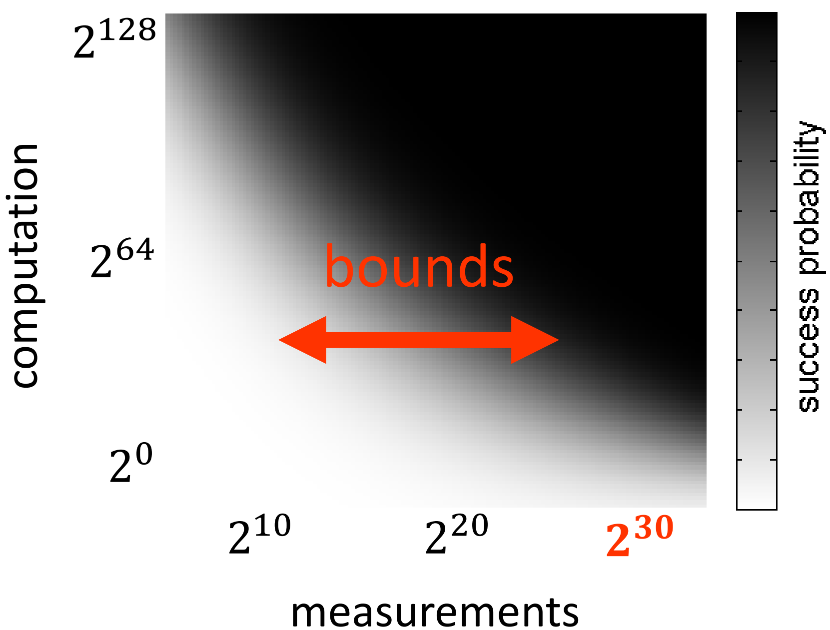
Evaluation challenge

standard practice





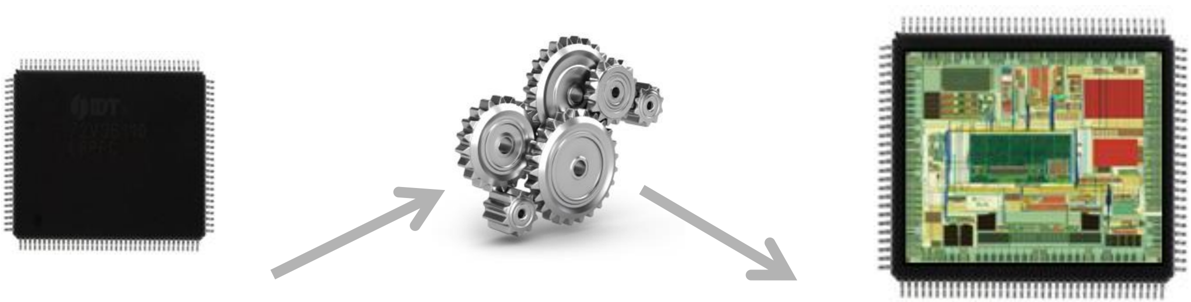
evidence-based evaluations
(*assumptions tested per device!*)



Evaluation challenge

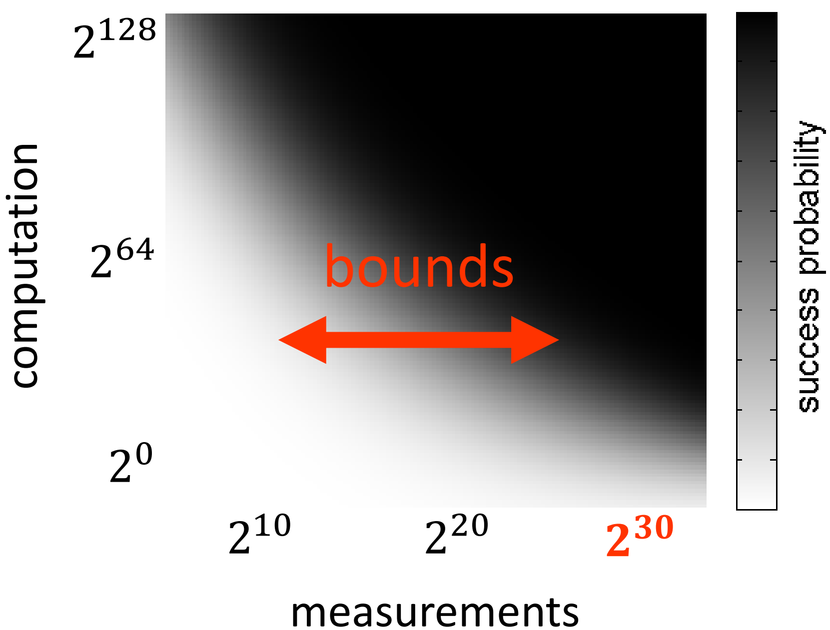
standard practice

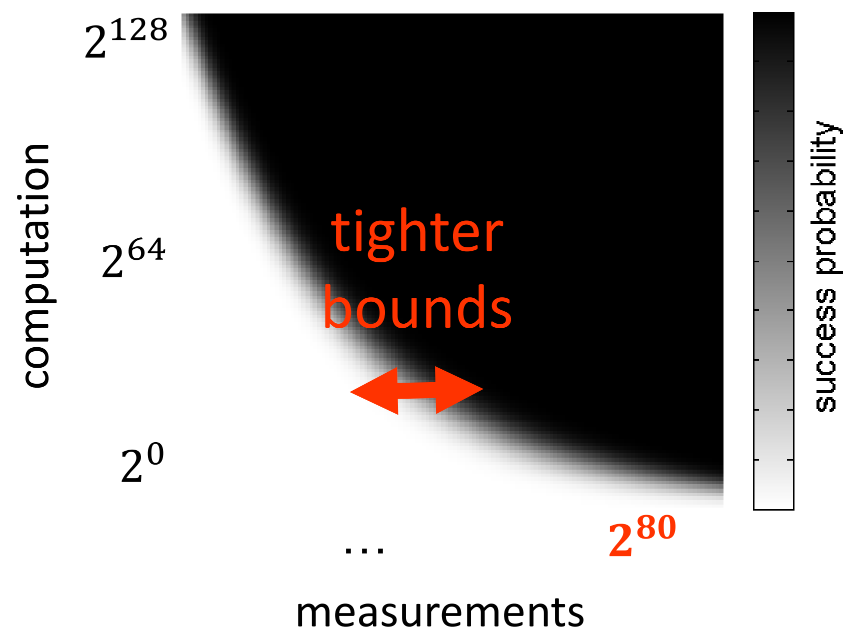
open design & evaluation

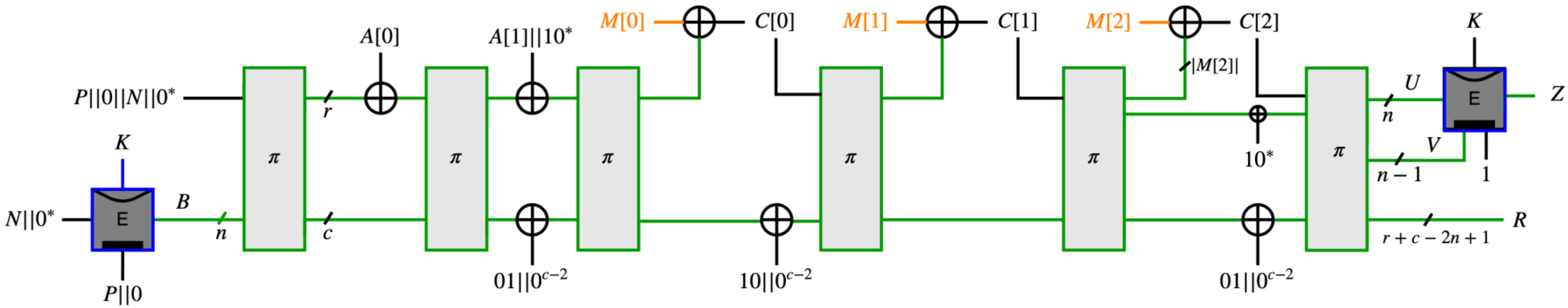


evidence-based evaluations
on reduced versions

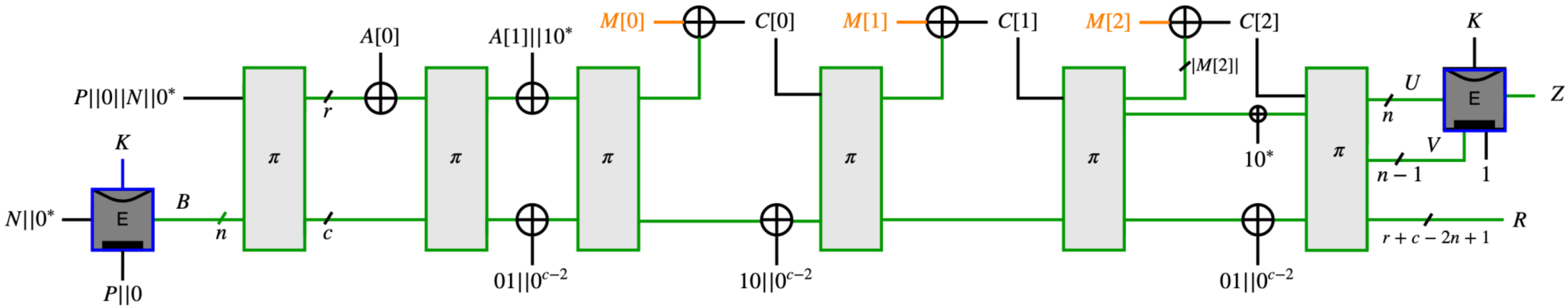
proof-based evaluations [DFS15,GS18]



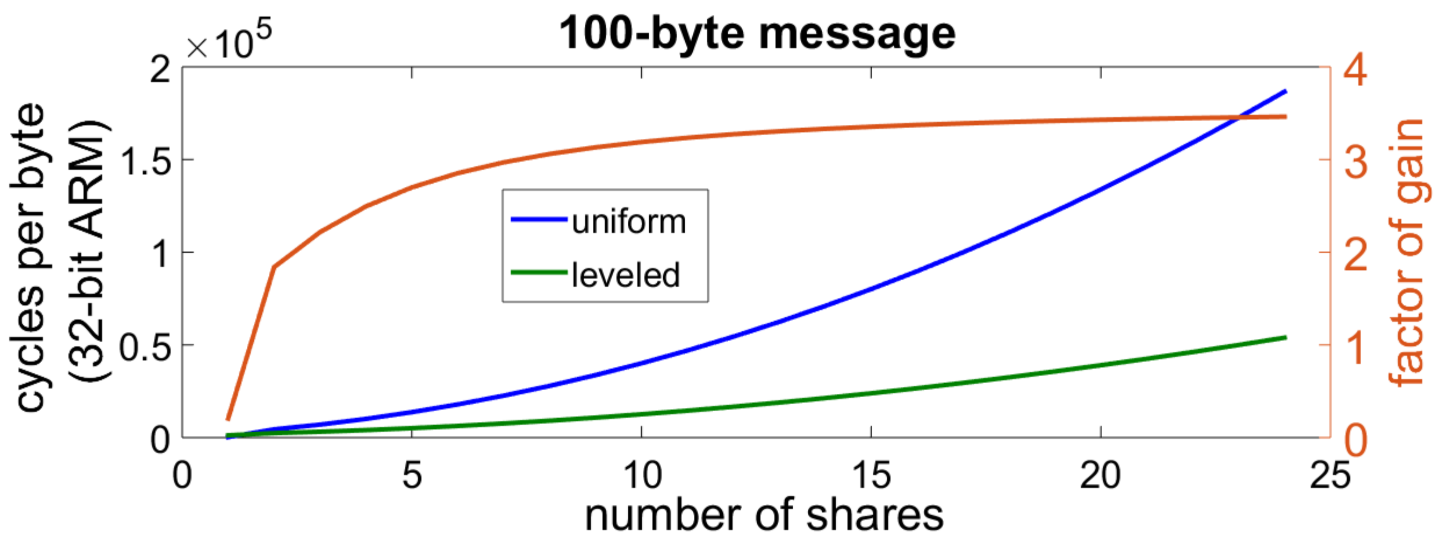


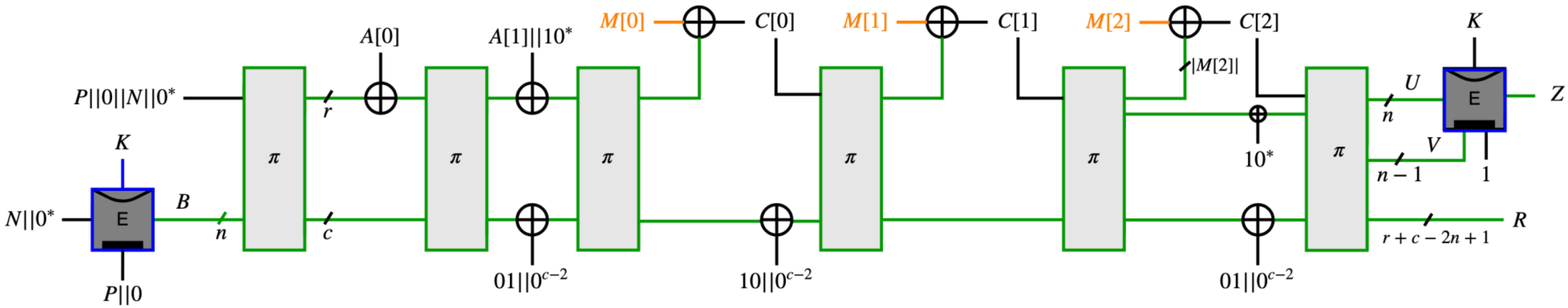


- Try leveraging « leveled implementations »
  - Strongly protected BC: high-order masking
  - Weakly protected permutation: low-latency
- Raises many definitional challenges (leakage-resilience)
- For such implementations, two different primitives are not an issue (since implementations are different)

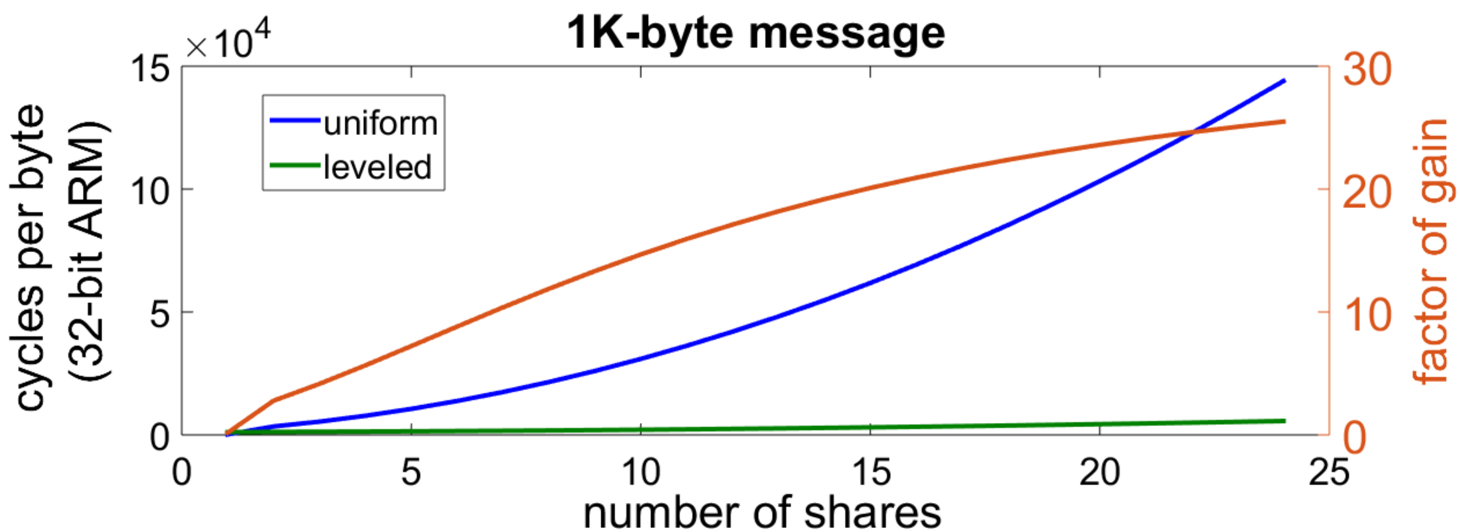


- Performance gains of leveled implementations

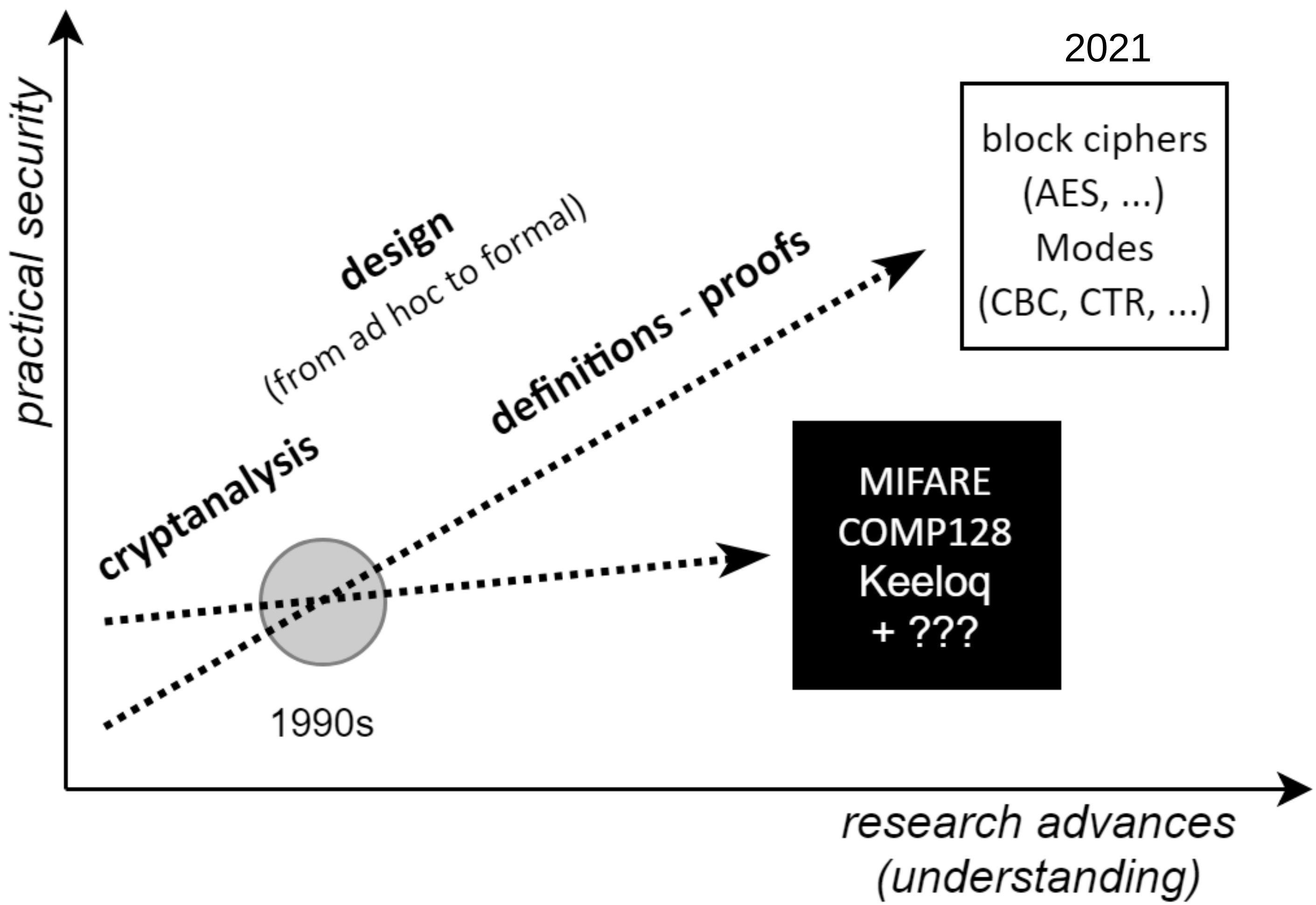




- Performance gains of leveled implementations

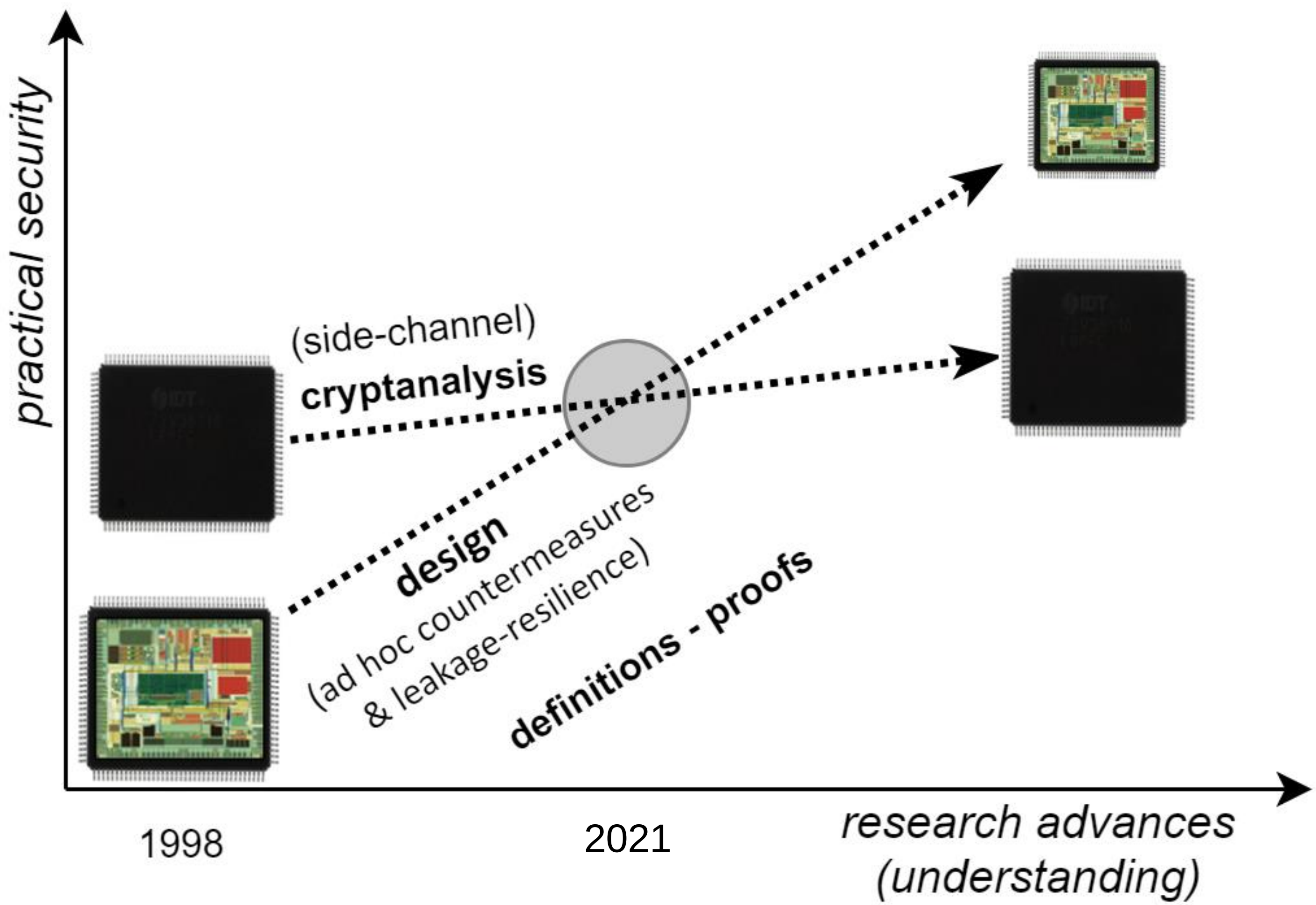


- Block ciphers & symmetric encryption



Transparency (as a measure of maturity)

- Secure cryptographic implementations



THANKS

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