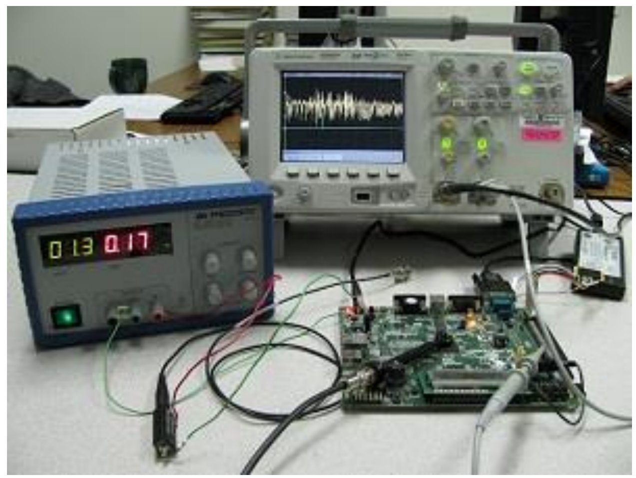
Towards Fair and Efficient Evaluations of Leaking Cryptographic Devices
(Overview of the ERC Project CRASH, part I)



François-Xavier Standaert

UCL Crypto Group, Belgium

SPACE, December 2016

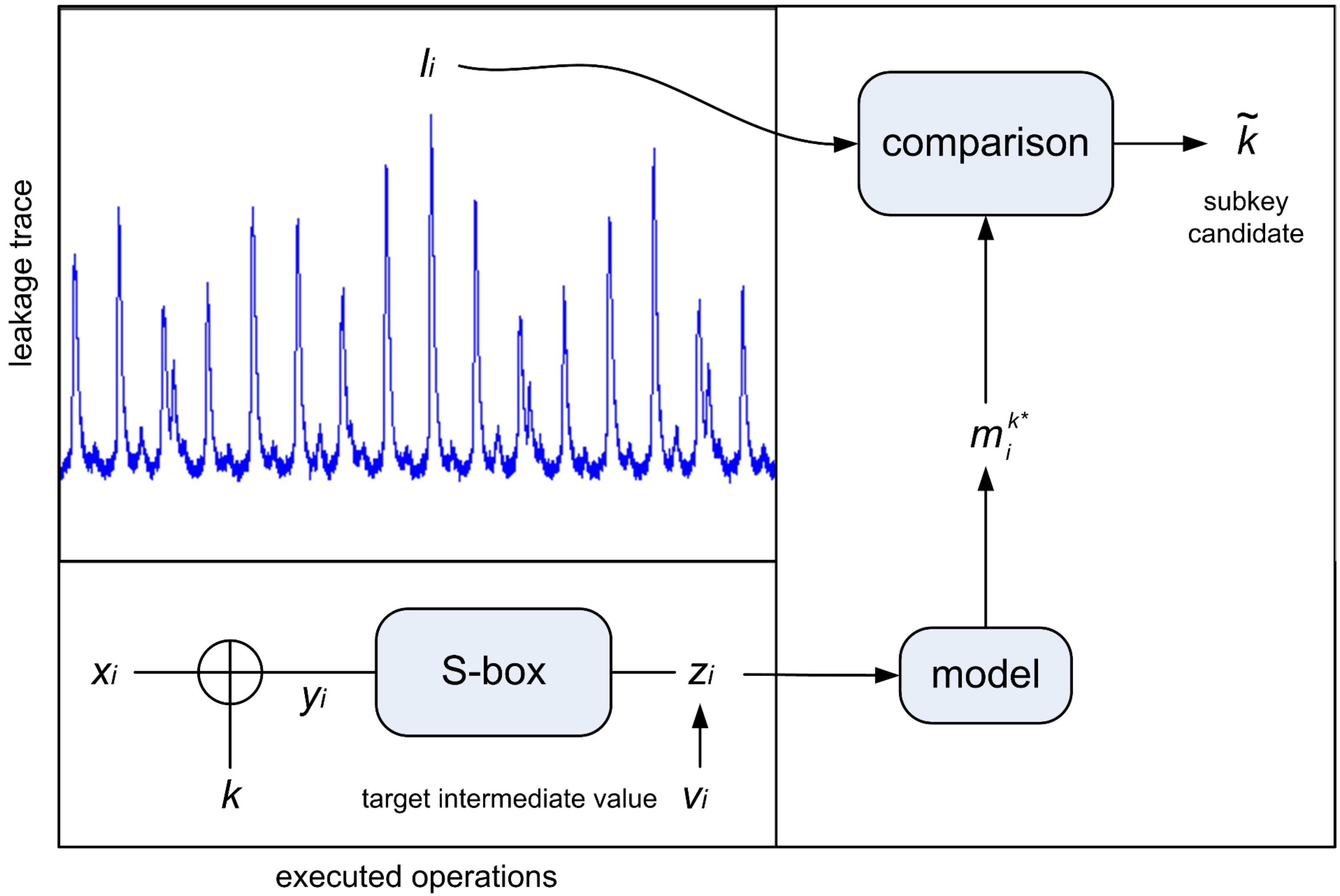
Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

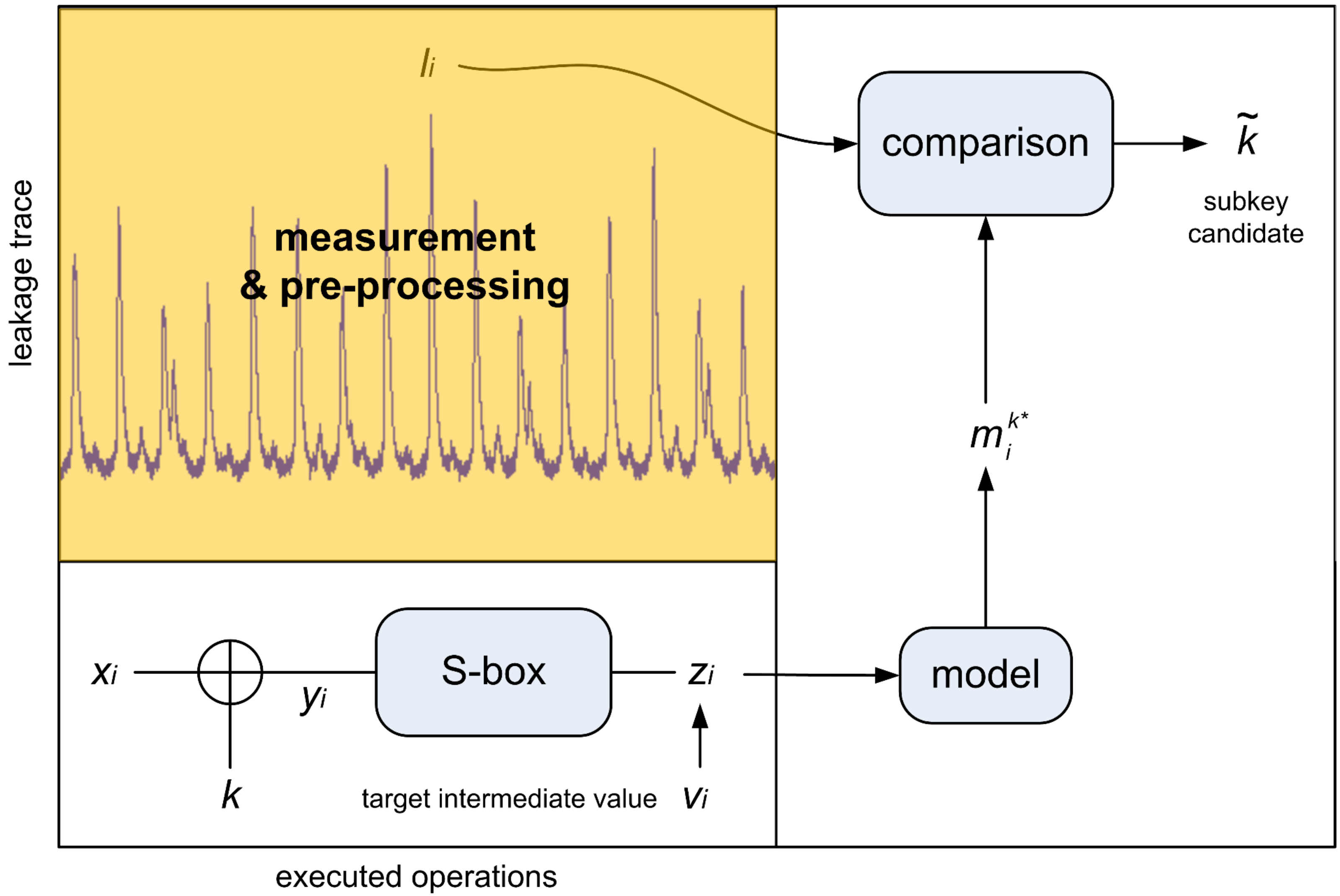
Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

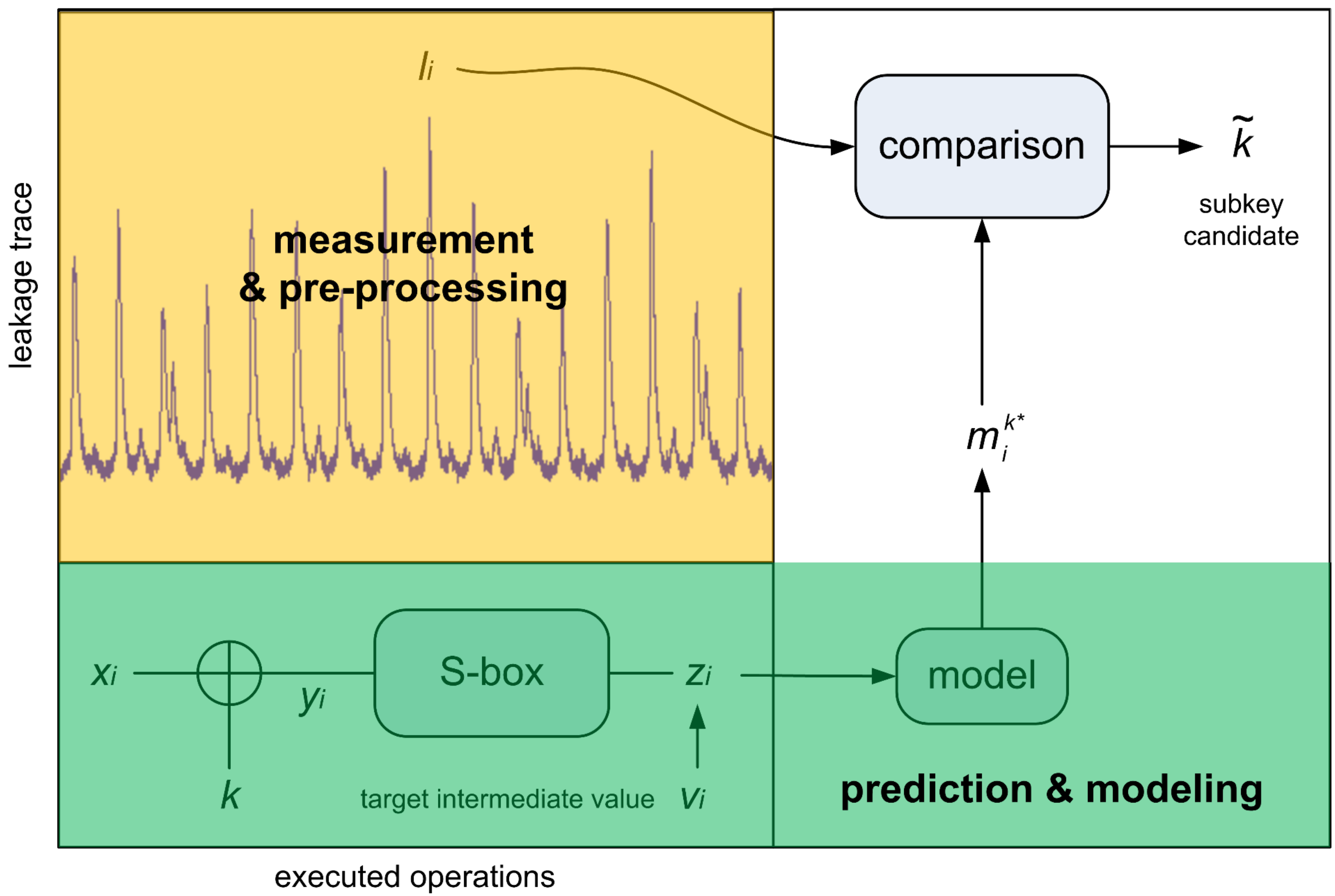
Standard DPA



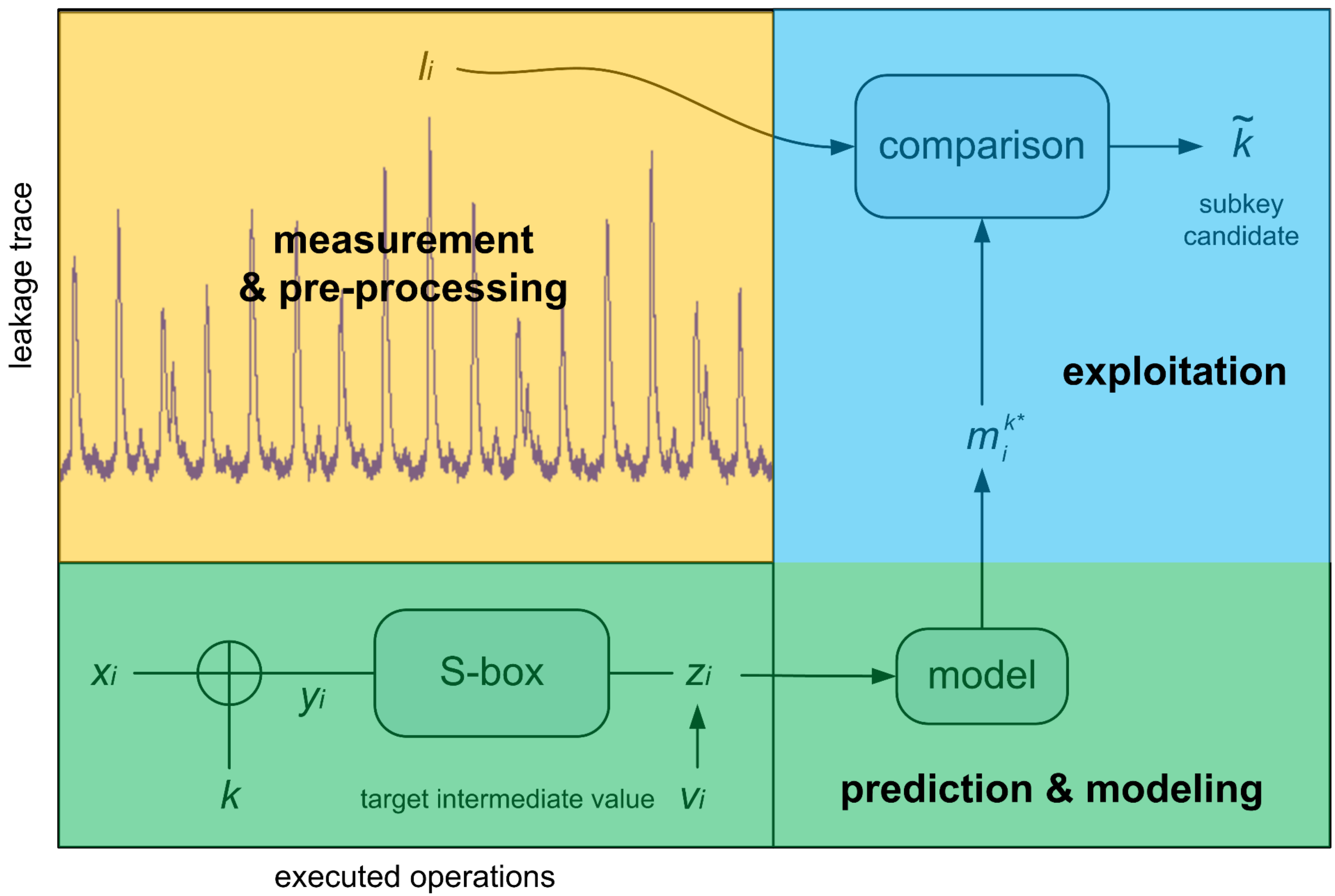
Standard DPA



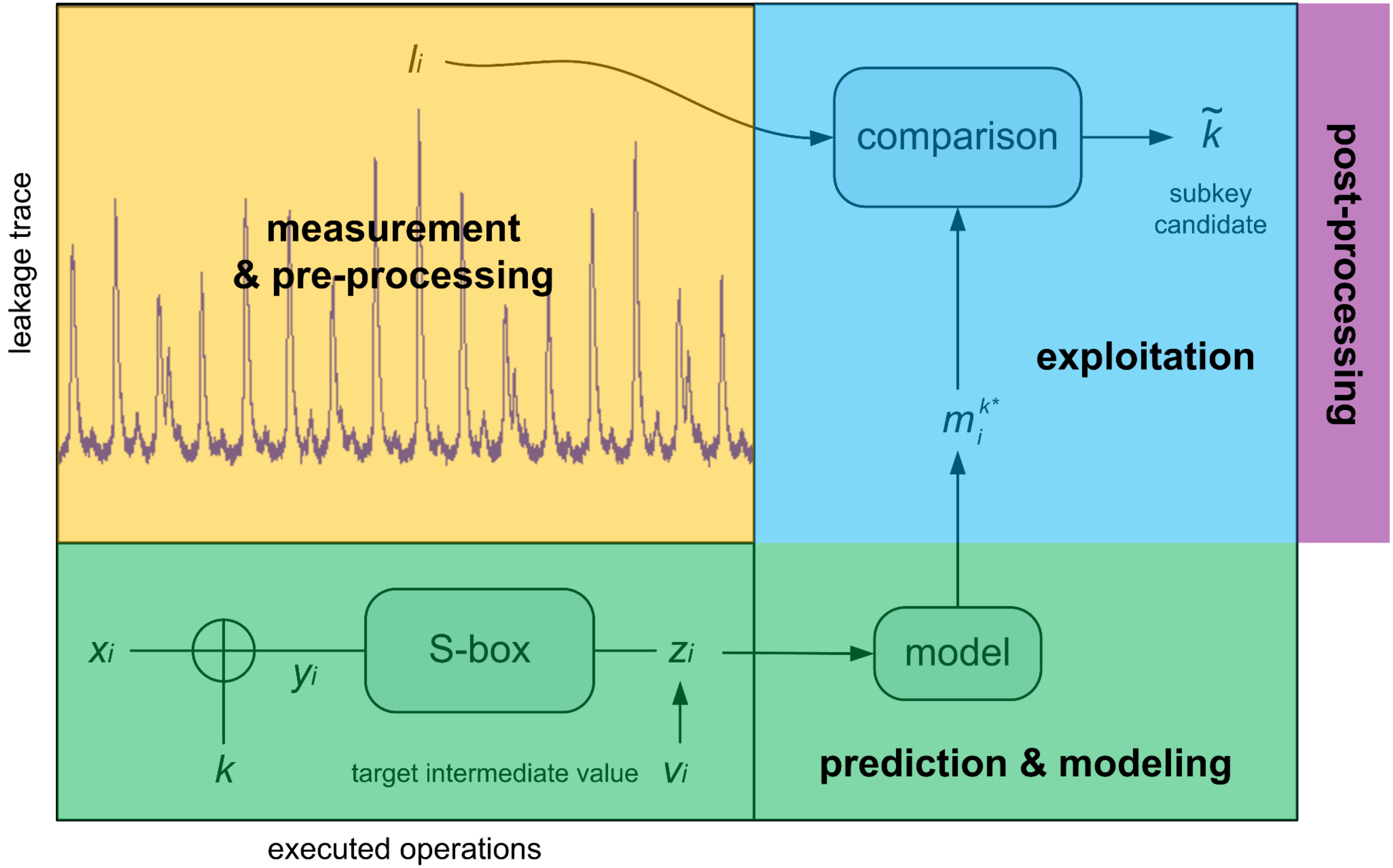
Standard DPA



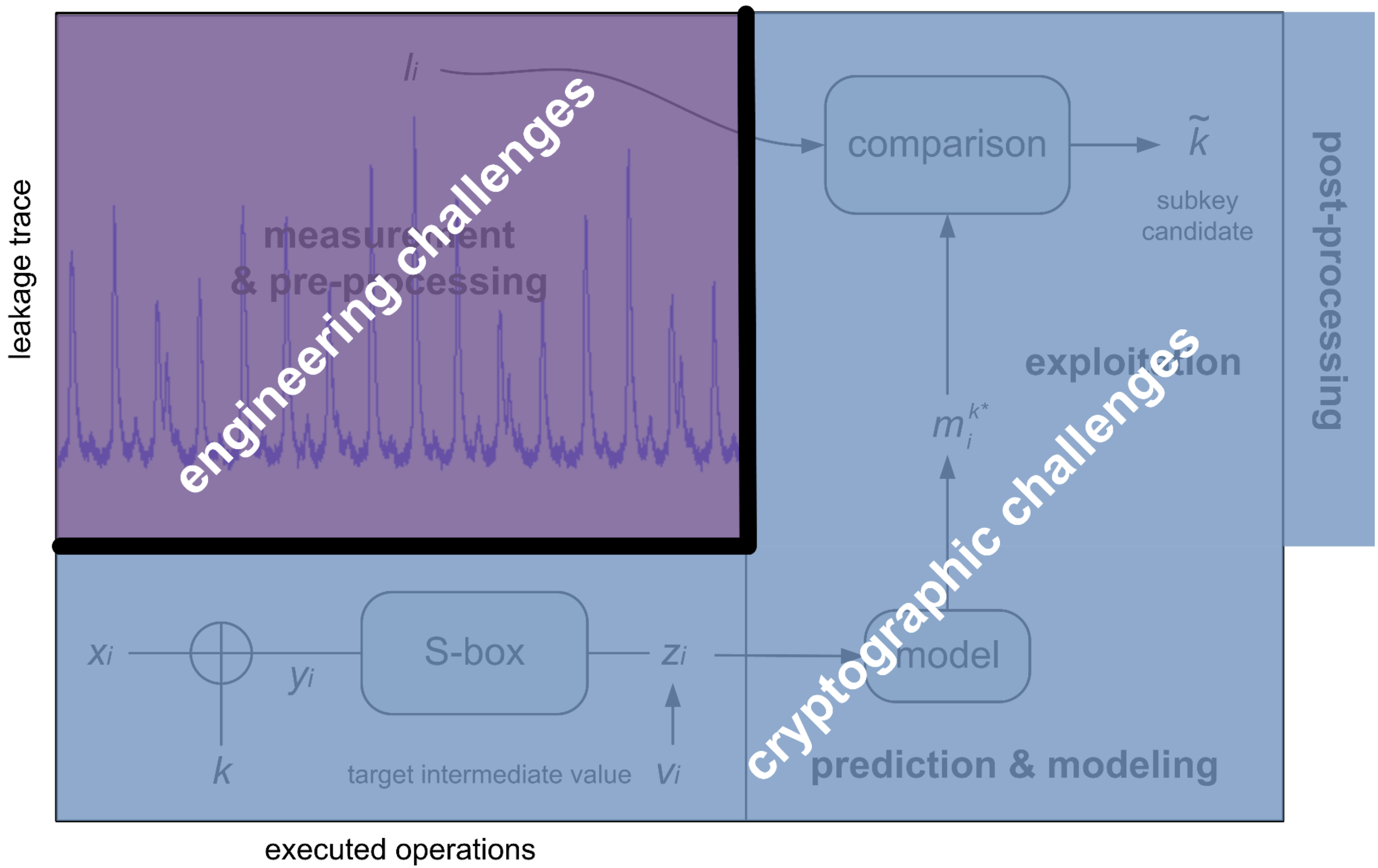
Standard DPA



Standard DPA



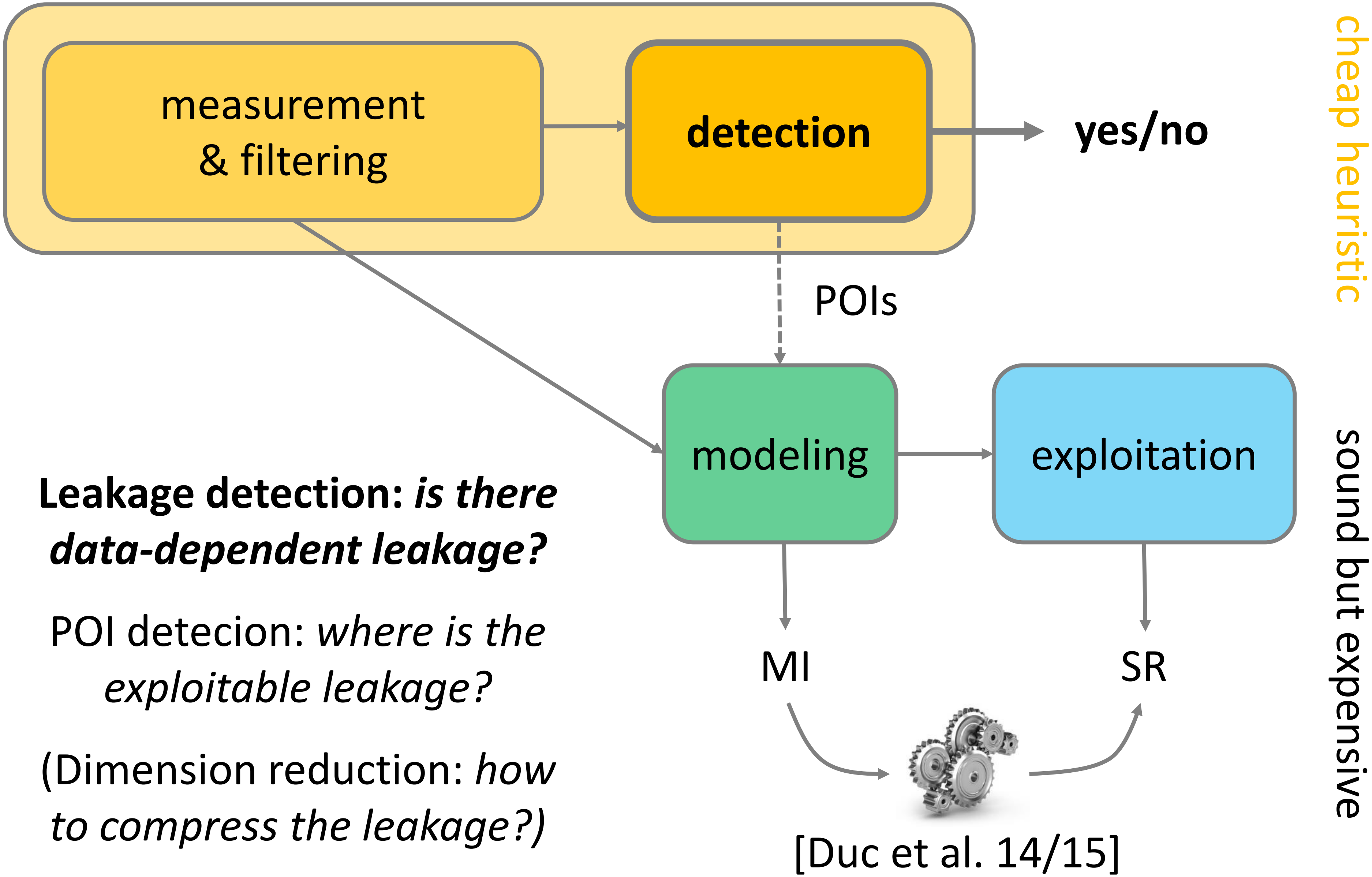
Standard DPA



Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

Side-channel attacks workflow



Leakage detection: *is there data-dependent leakage?*

POI detecion: *where is the exploitable leakage?*

(Dimension reduction: *how to compress the leakage?)*

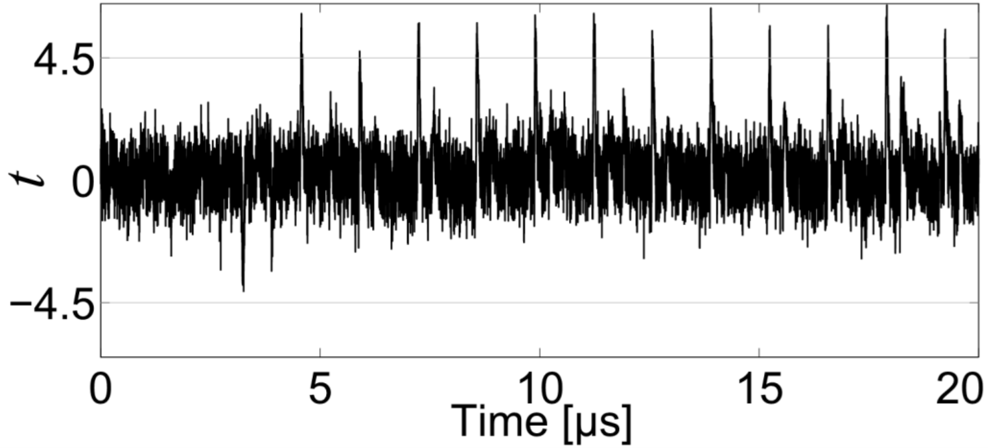
Tool #1: CRI's "fixed vs. random" t-test

- AES Rijndael example
  - 128-bit key fixed
  - N_r traces with <u>random</u> plaintexts
  - N_f traces with a <u>fixed</u> plaintexts
  - Apply Student's t-test to the f&r classes:
  - $\Delta(t) = \left[\hat{\mu}_f(t) - \hat{\mu}_r(t)\right] / \left[\left(\hat{\sigma}_f^2\ (t)/N_f\right) + (\hat{\sigma}_r^2\ (t)/N_r)\right]$

Tool #1: CRI's "fixed vs. random" t-test

- AES Rijndael example
  - 128-bit key fixed
  - N_r traces with <u>random</u> plaintexts
  - N_f traces with a <u>fixed</u> plaintexts
  - Apply Student's t-test to the f&r classes:
  - $\Delta(t) = \left[\hat{\mu}_f(t) - \hat{\mu}_r(t)\right] / \left[\left(\hat{\sigma}_f^2\,(t)/N_f\right) + \left(\hat{\sigma}_r^2\,(t)/N_r\right)\right]$

- Large t statistic: « some data dependency detected »

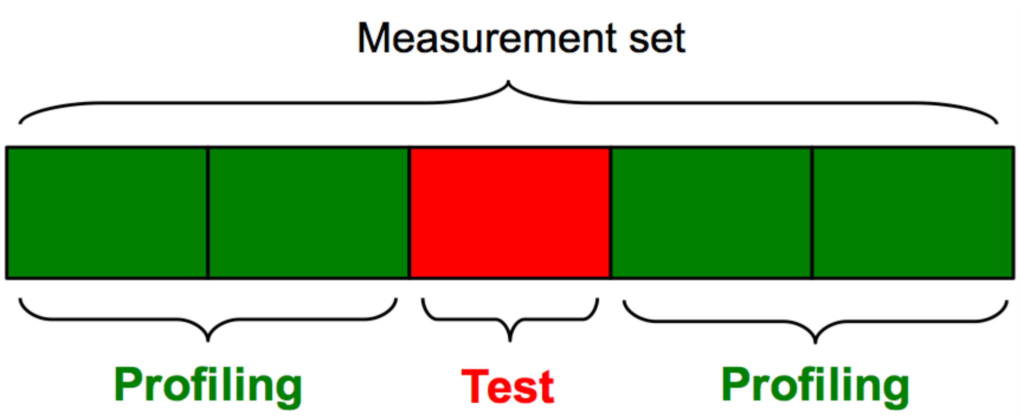


Tool #2: correlation-based detection

- ρ-test
  - 128-bit key fixed
  - *N* traces with random plaintexts
  - Targets an enumerable intermediate value *X*
  - Estimate Pearson's coefficient: $\hat{r}(t) = \hat{\rho}(L_X(t), \hat{model}_t(X))$

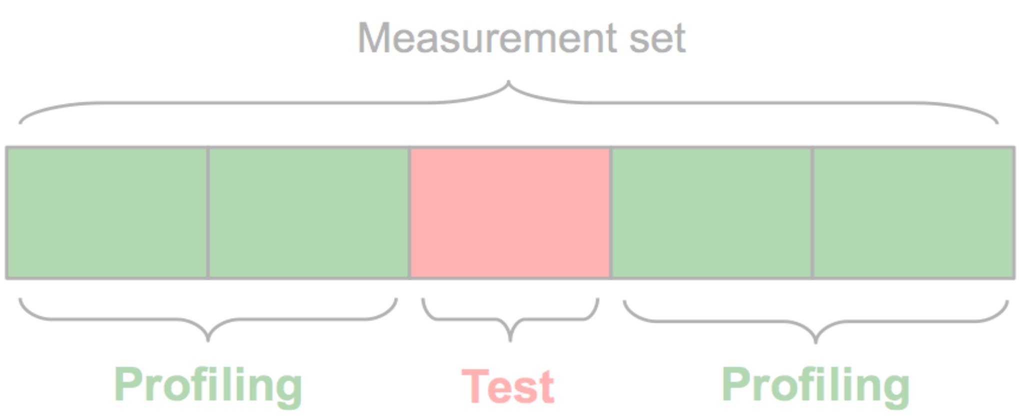
Tool #2: correlation-based detection

- ρ-test
  - 128-bit key fixed
  - *N* traces with random plaintexts
  - Targets an enumerable intermediate value *X*
  - Estimate Pearson's coefficient: $\hat{r}(t) = \hat{\rho}(L_X(t), \widehat{model}_t(X))$
- Model estimated with cross-validation



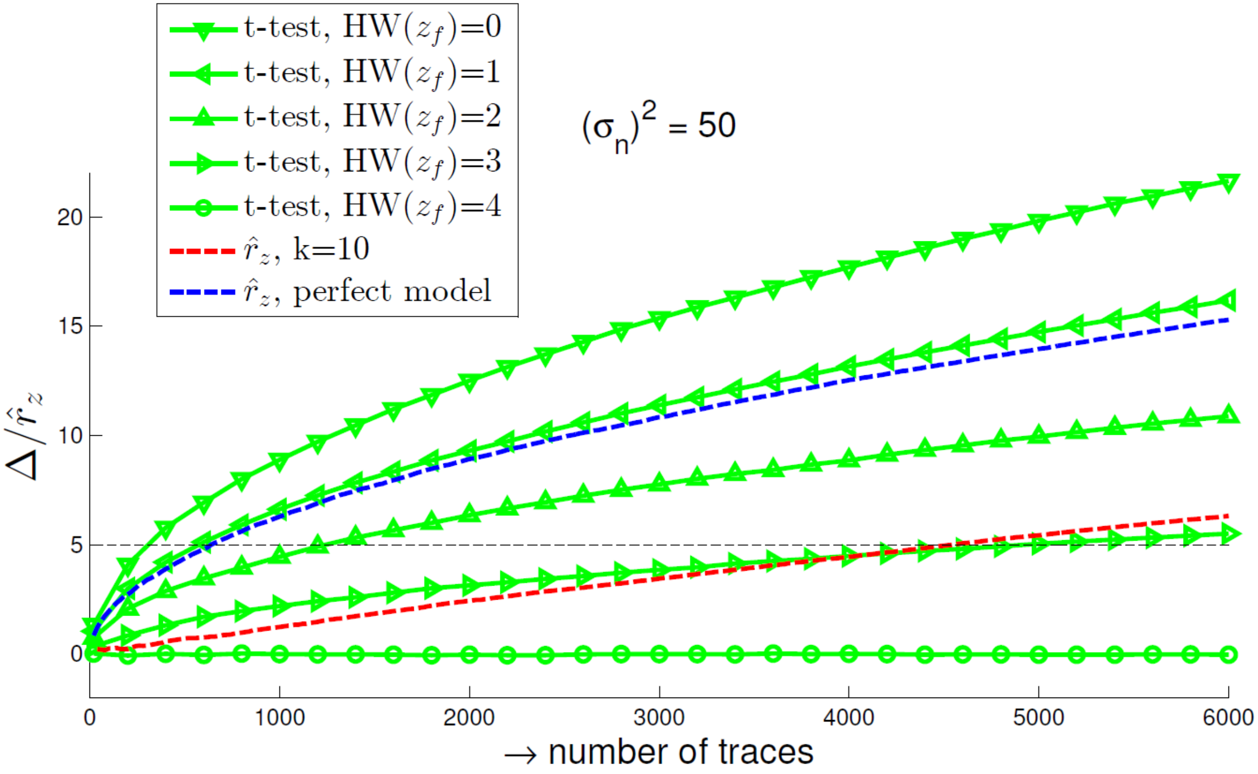
Tool #2: correlation-based detection

- ρ-test
  - 128-bit key fixed
  - *N* traces with random plaintexts
  - Targets an enumerable intermediate value *X*
  - Estimate Pearson's coefficient: $\hat{r}(t) = \hat{\rho}(L_X(t), \hat{model}_t(X))$
- Model estimated with cross-validation

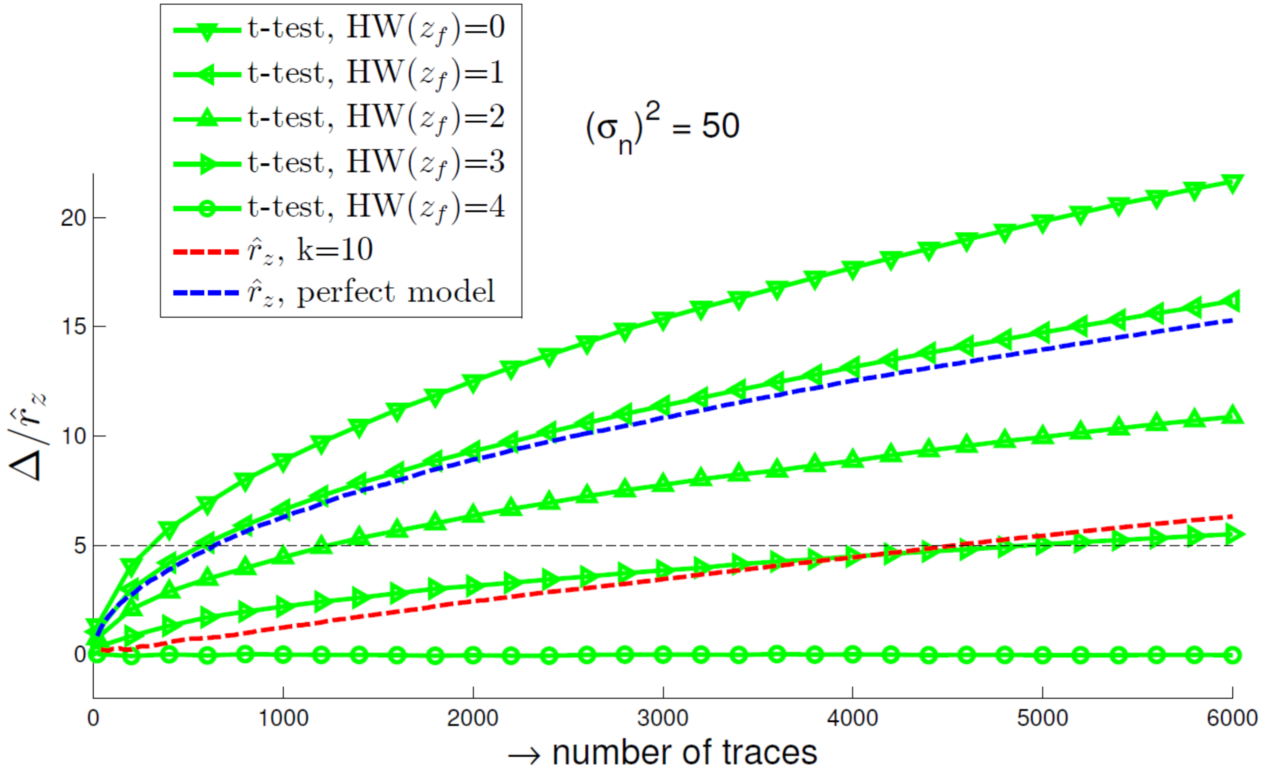


- Hypothesis test (Z transform): $\frac{1}{2} \times ln\left(\frac{1+\hat{r}(t)}{1-\hat{r}(t)}\right) \sim N\left(0, \frac{1}{\sqrt{N-3}}\right)$

Simulated experiments (8-bit noisy HW's)

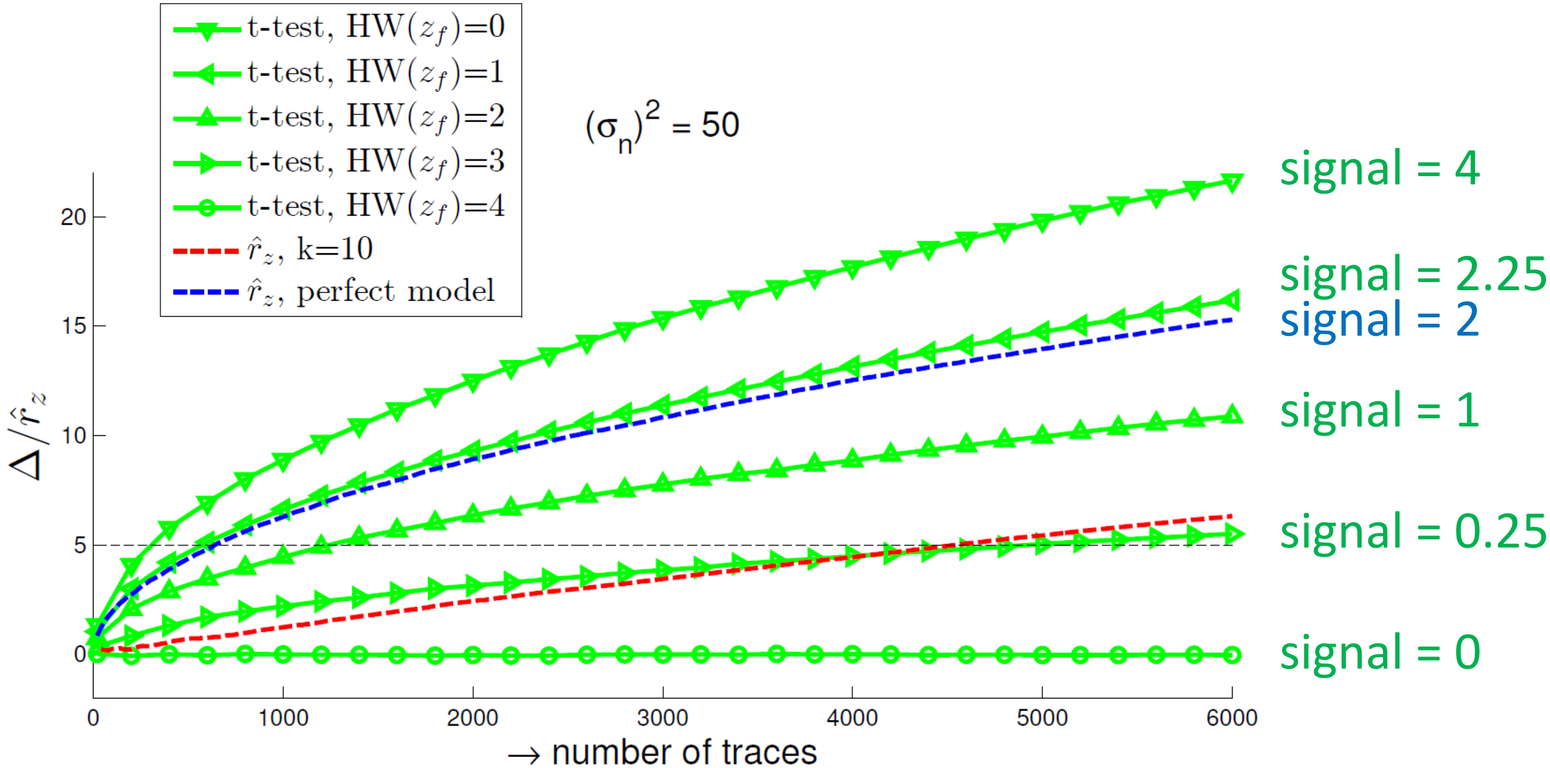


Simulated experiments (8-bit noisy HW's)

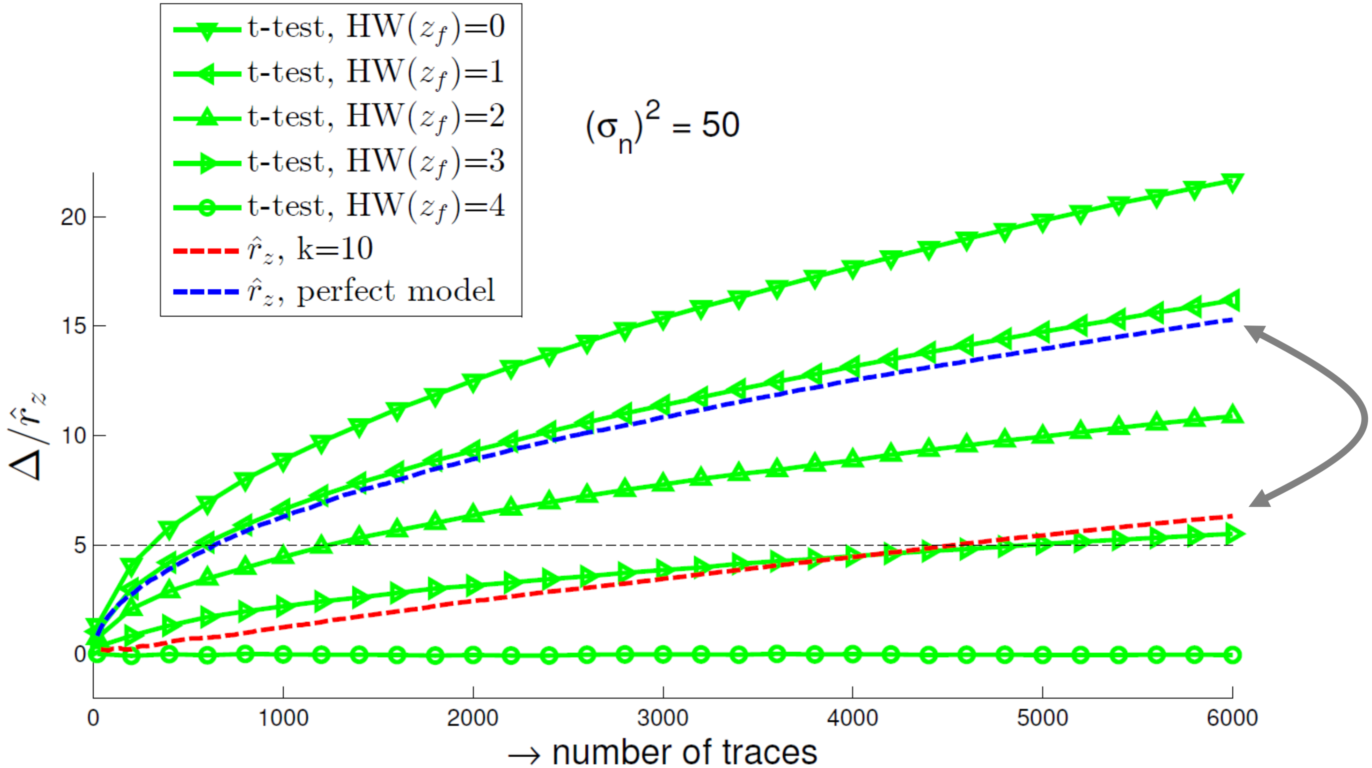


- CRI's t-test pro: sampling complexity!

Simulated experiments (8-bit noisy HW's)

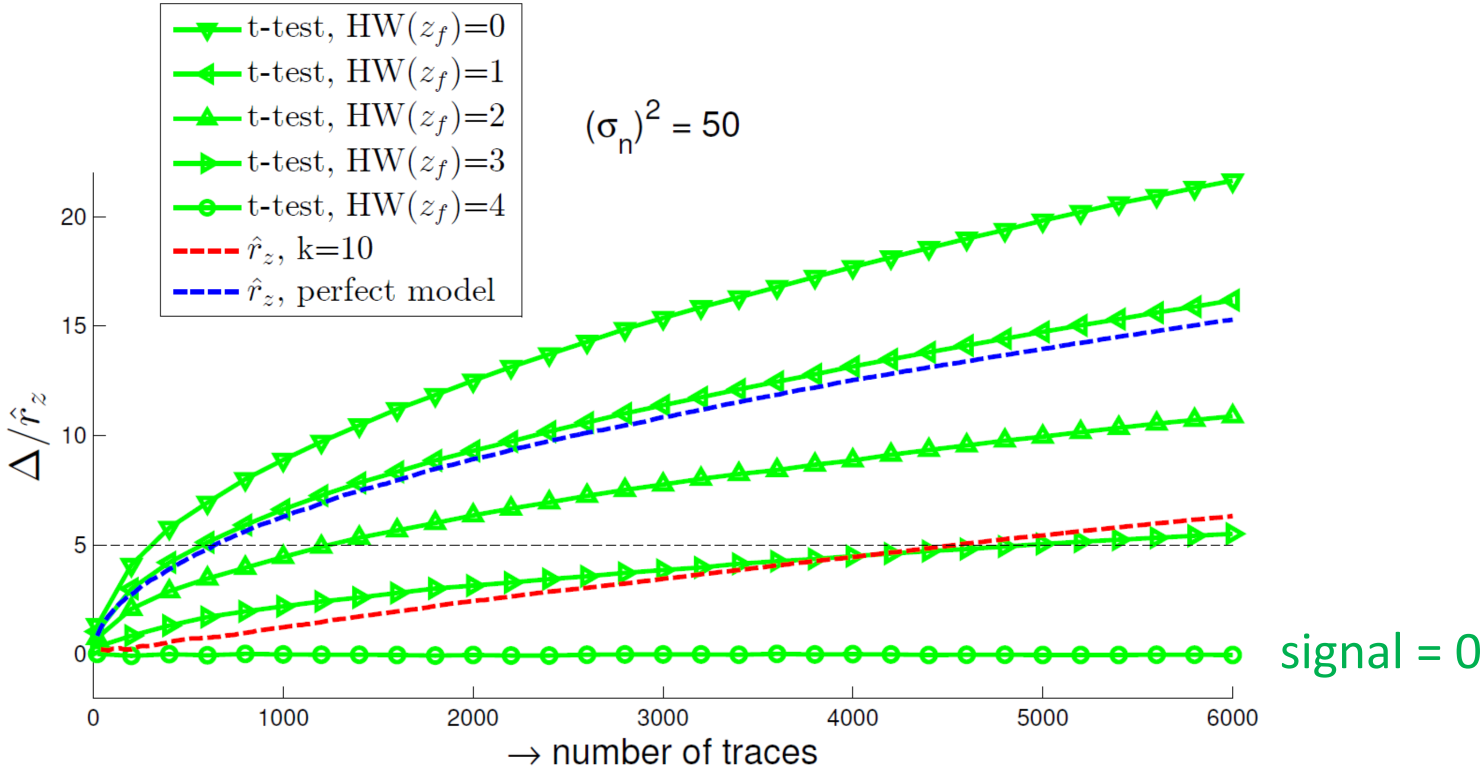


- CRI's t-test pro: sampling complexity!
  - Better **signal** for well-chosen fixed classes



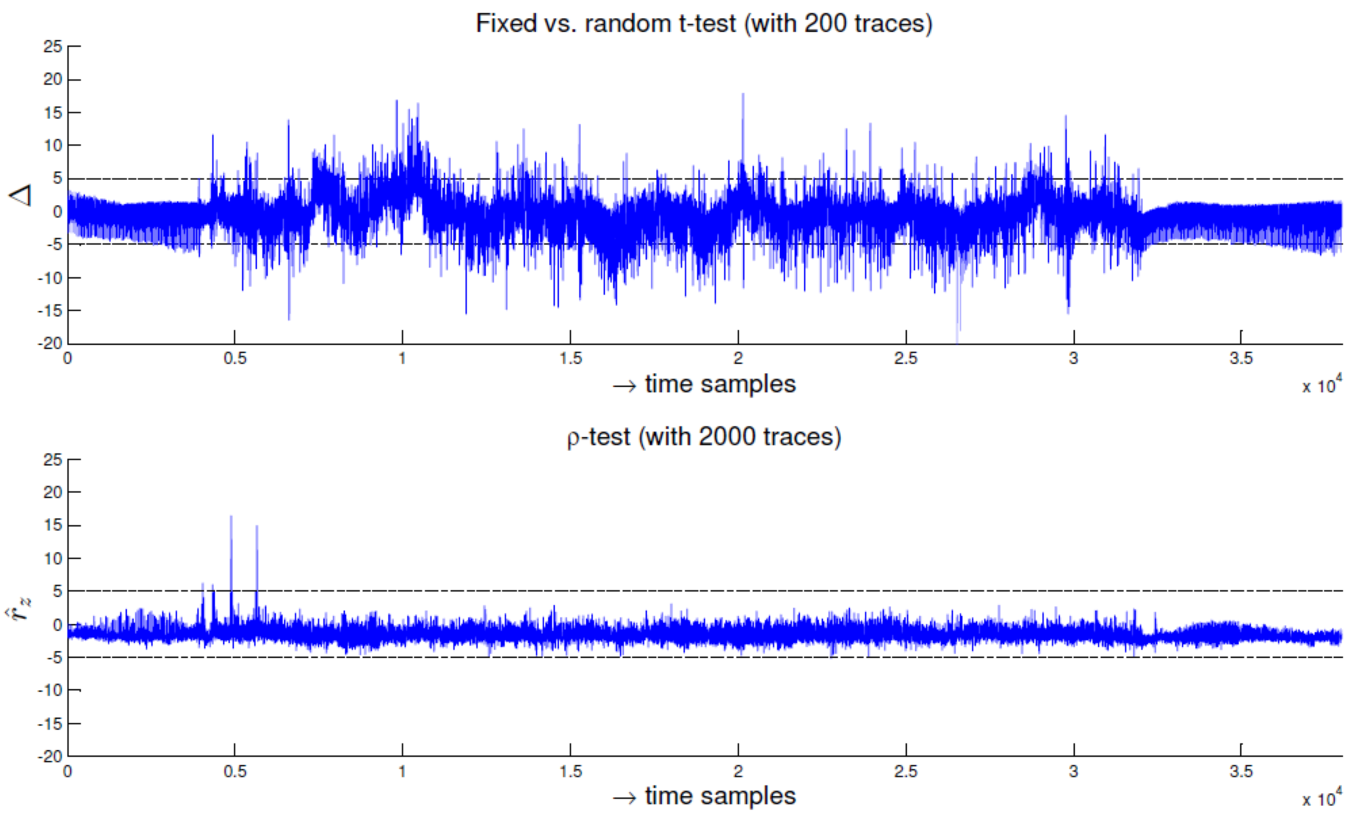
- CRI's t-test pro: sampling complexity!
  - Better **signal** for well-chosen fixed classes
  - Easier estimation (**2** classes vs. **256** classes)

Simulated experiments (8-bit noisy HW’s)

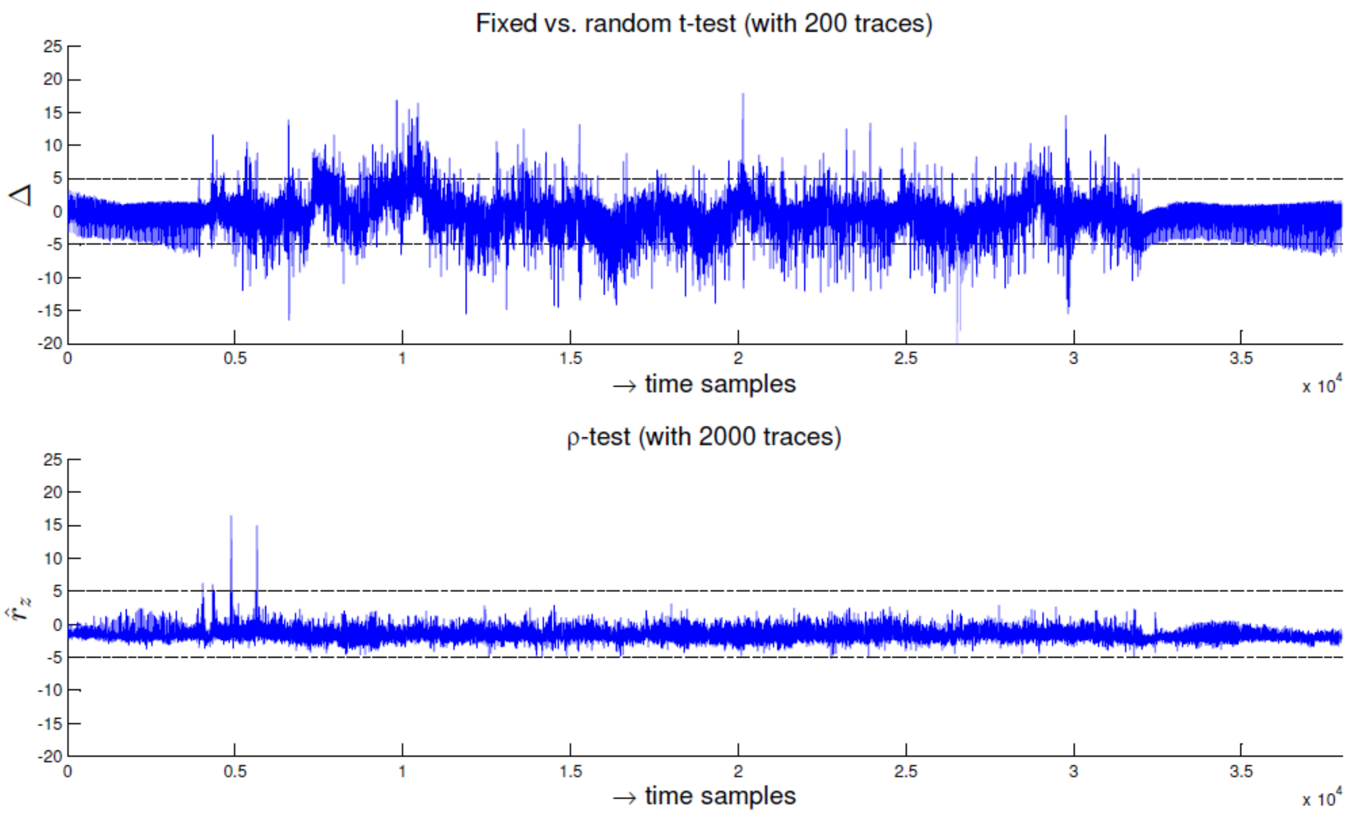


- CRI’s t-test con: possible false negative!
  - Possibly **no signal** for badly-chosen fixed classes

Real measurements (Atmel Atmega)

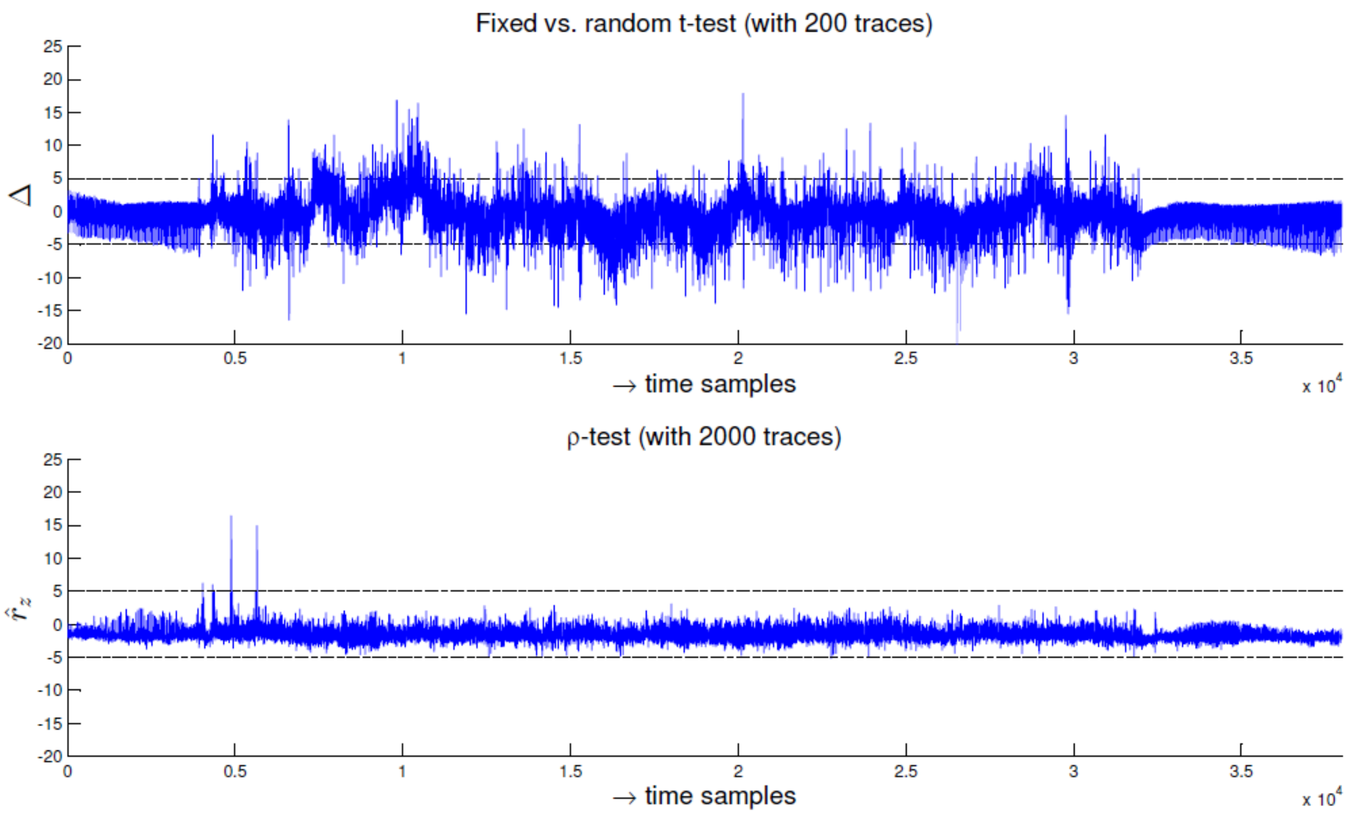


Real measurements (Atmel Atmega)



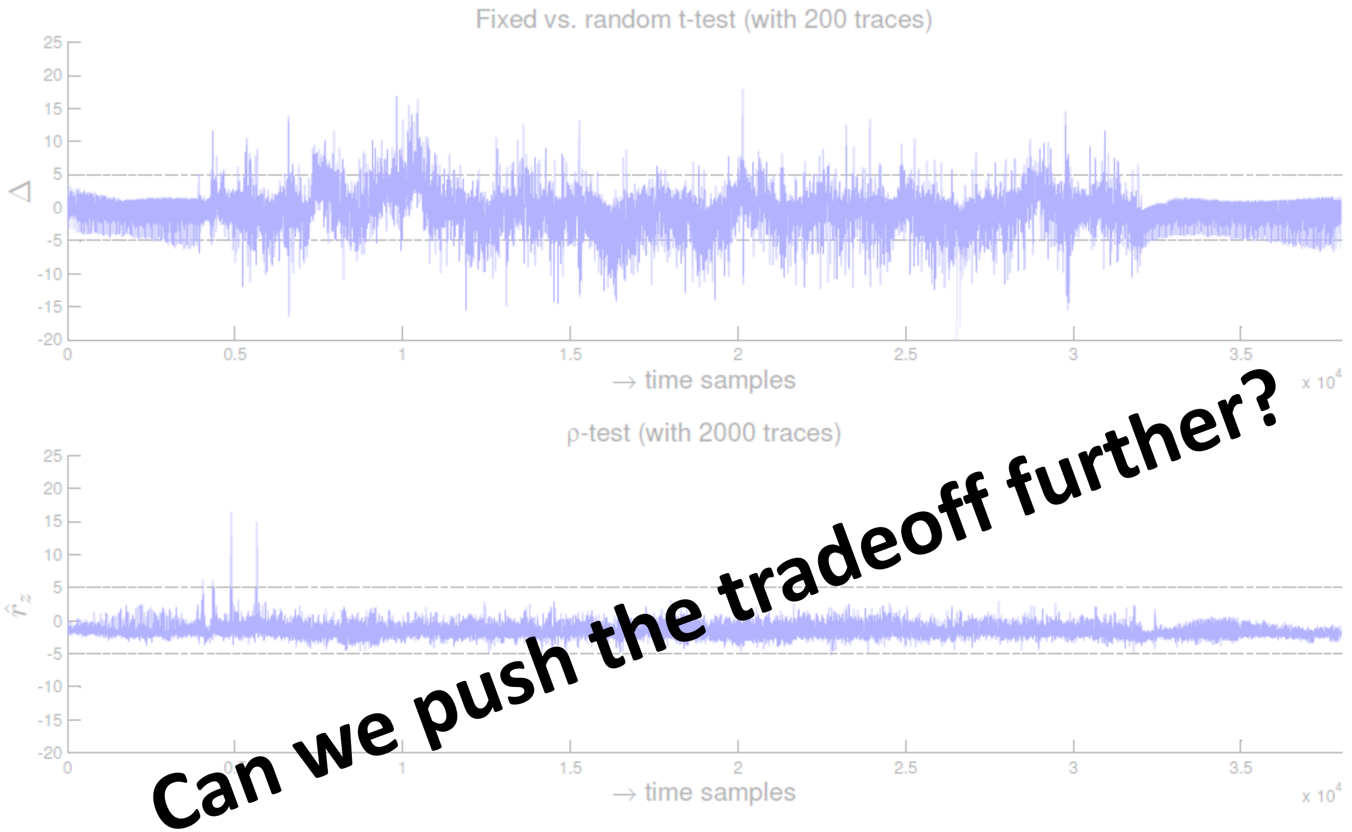
- t-test false negatives do not hurt (integrated over time)
- But there are many false positives (w.r.t. DPA)
- Only the value of ρ is connected with key recovery SR

Real measurements (Atmel Atmega)



sampling complexity vs. informativeness tradeoff

- t-test false negatives do not hurt (integrated over time)
- But there are many false positives (w.r.t. DPA)
- Only the value of ρ is connected with key recovery SR

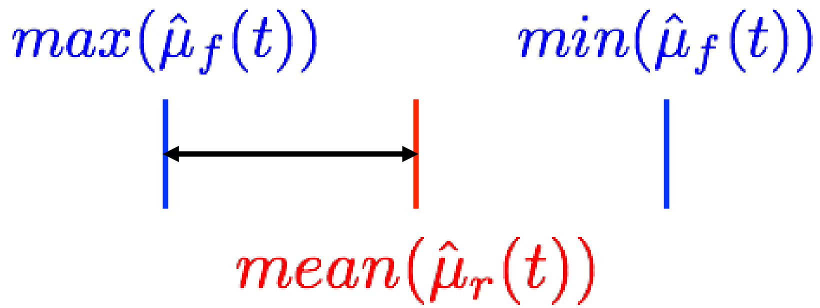


sampling complexity vs. informativeness tradeoff

- t-test false negatives do not hurt (integrated over time)
- But there are many false positives (w.r.t. DPA)
- Only the value of ρ is connected with key recovery SR

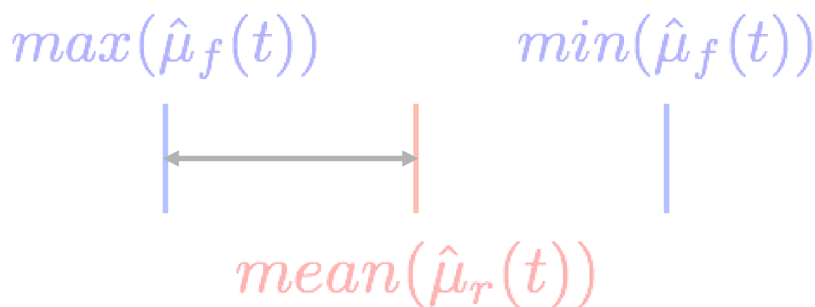
Fixed vs. fixed leakage detection test

- CRI's fixed vs. random test (e.g. HW leakages)
  - Maximum HW difference observed = 4
  - "Algorithmic noise" due to the random class

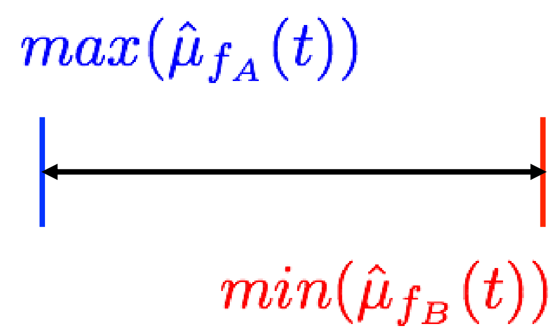


Fixed vs. fixed leakage detection test

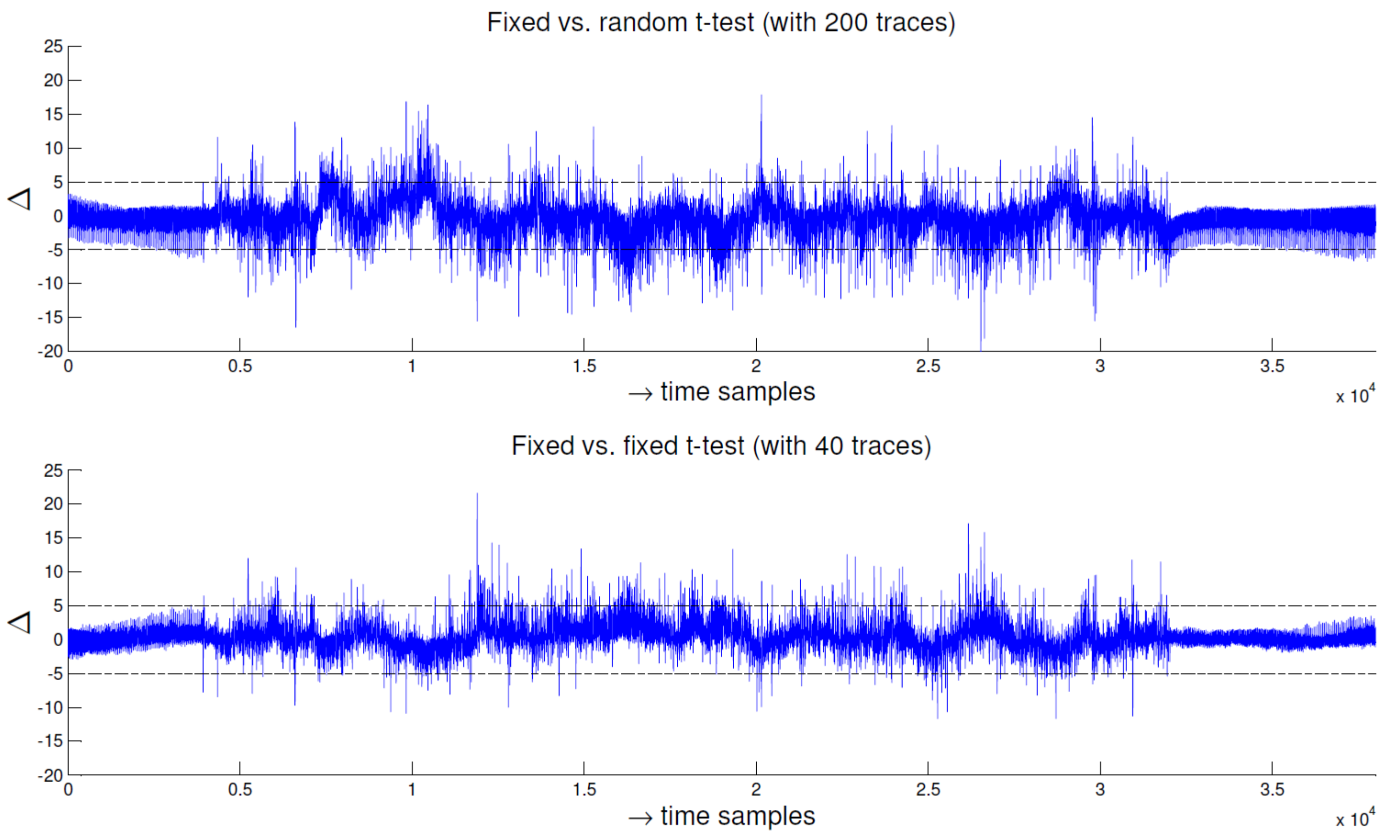
- CRI’s fixed vs. random test (e.g. HW leakages)
  - Maximum HW difference observed = 4
  - “Algorithmic noise” due to the random class



- Natural extension: *fixed vs. fixed test*
  - Maximum HW difference = 8
    - Average signal multiplied 2
  - No algorithmic noise!

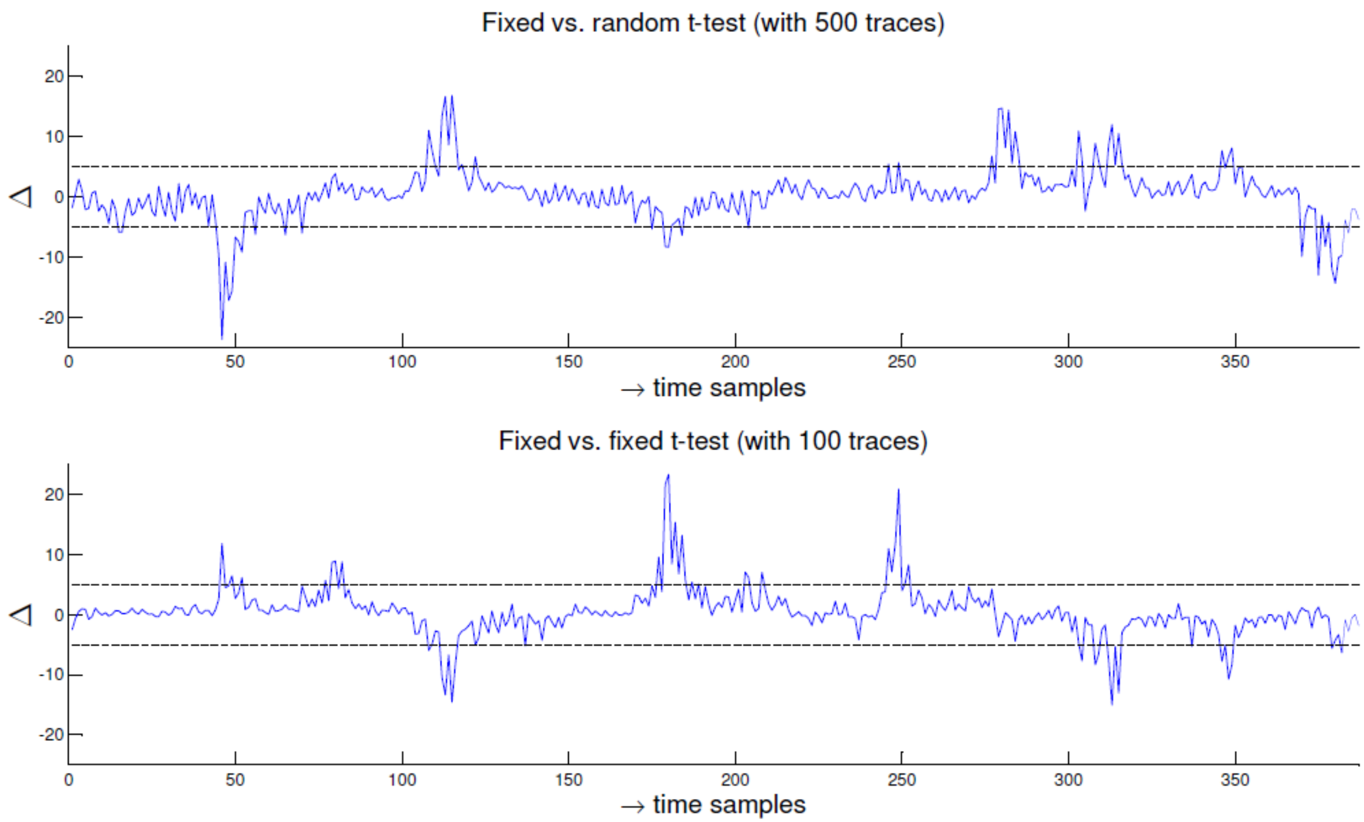


Real measurements (Atmel Atmega)



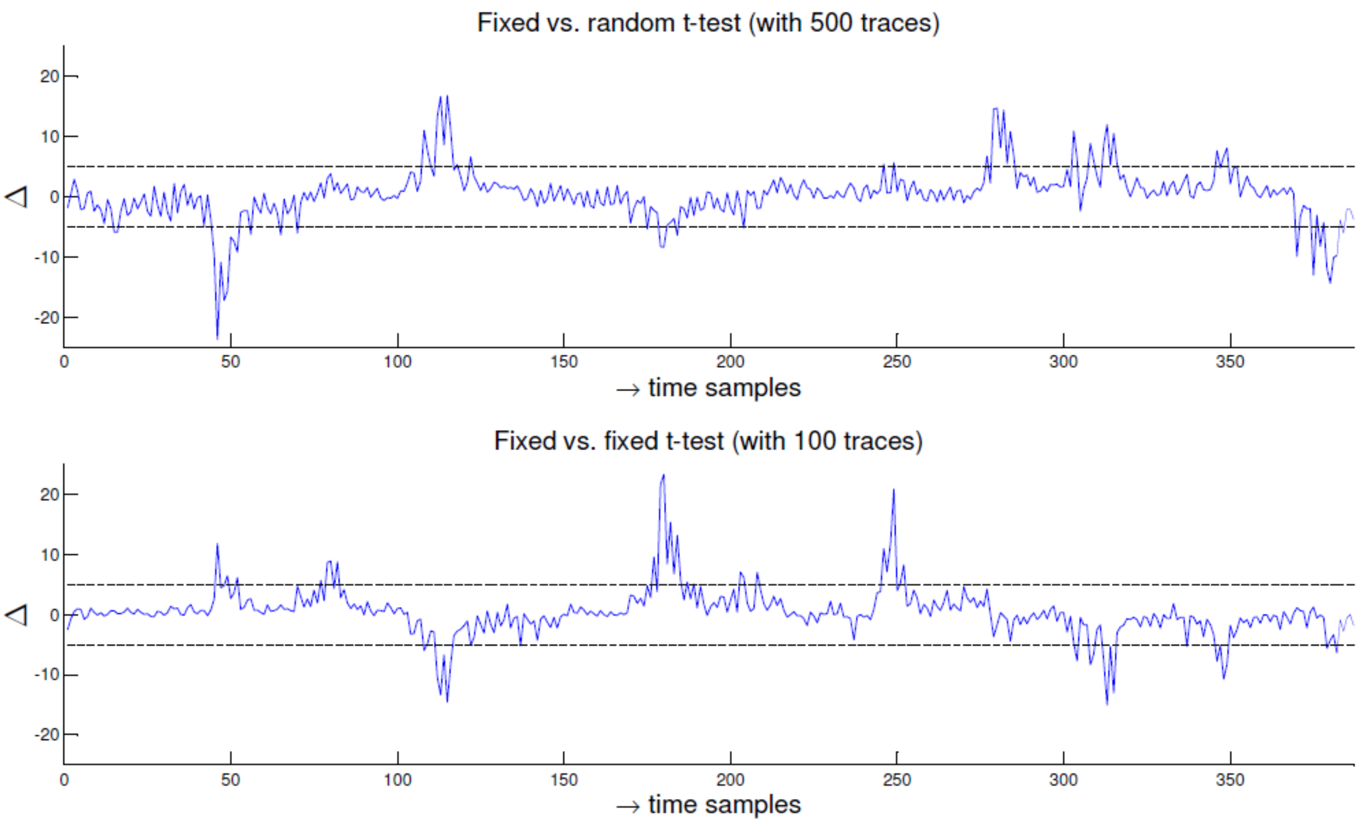
- Gain dominated by the increased signal
- Reduction of the sampling complexity by a factor ≈ 5

Real measurements (parallel AES in an FPGA)



- Gain dominated by the reduced noise
- Reduction of the sampling complexity by a factor ≈ 5

Real measurements (parallel AES in an FPGA)



F vs. F better deal on average; similar risks of fasle positives & negatives

- Gain dominated by the reduced noise
- Reduction of the sampling complexity by a factor ≈ 5

Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

Prediction and modeling

- General case: profiled DPA
  - Build "*templates*", i.e. $\hat{f}(l_i|k, x_i)$
    - e.g., Gaussian, regression-based
  - Which directly leads to $\widehat{\Pr}[k|l_i, x_i]$

Prediction and modeling

- General case: profiled DPA
  - Build "*templates*", i.e. $\hat{f}(l_i|k, x_i)$
    - e.g., Gaussian, regression-based
  - Which directly leads to $\widehat{\Pr}[k|l_i, x_i]$

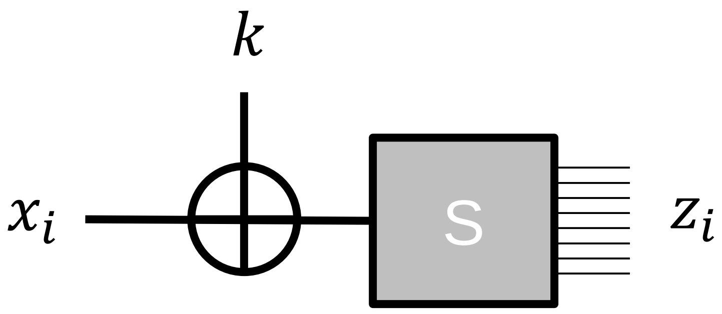
- "Simplified" case: non-profiled DPA
  - Just assumes some model
  - e.g., CPA with $m_i^{k^*} = \mathrm{HW}(z_i)$
  - e.g., DPA with $m_i^{k^*} = z_i[1]$
  - …

Separation result (I)

- Only profiled DPA is *guaranteed* to succeed!

Separation result (I)

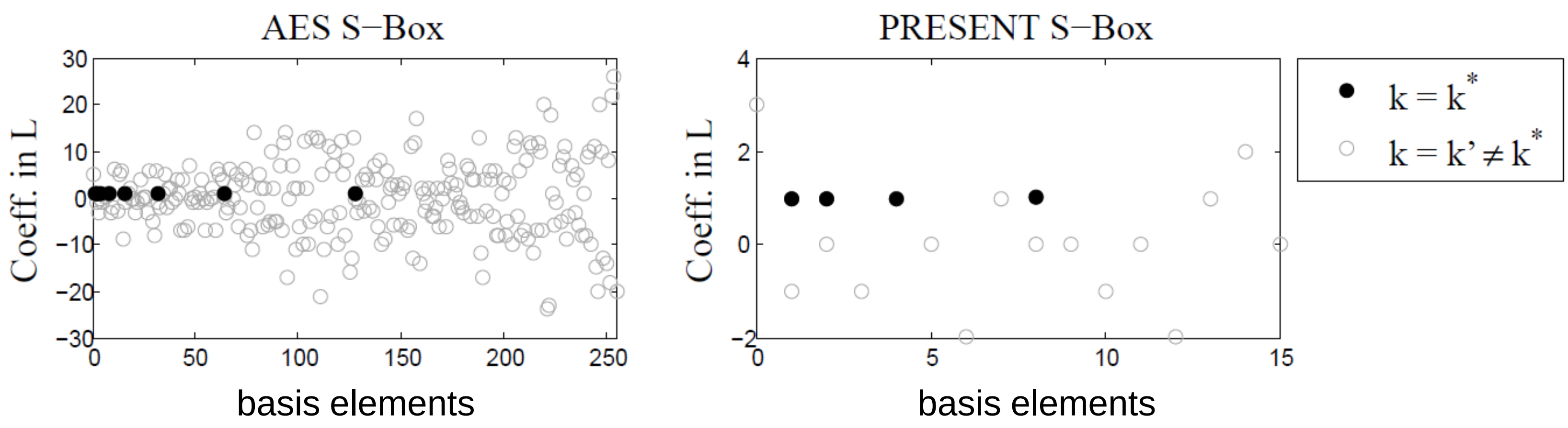
- Only profiled DPA is *guaranteed* to succeed!



- Regression: $\mathrm{L}(z_i) \approx \mathrm{F}(z_i) + N_i$ with:
  - Linear basis (i.e., 8 bits)
  - Quadratic basis (i.e., add 28 products)
  - …
  - Full basis (i.e., 256 elements)

Separation result (II)

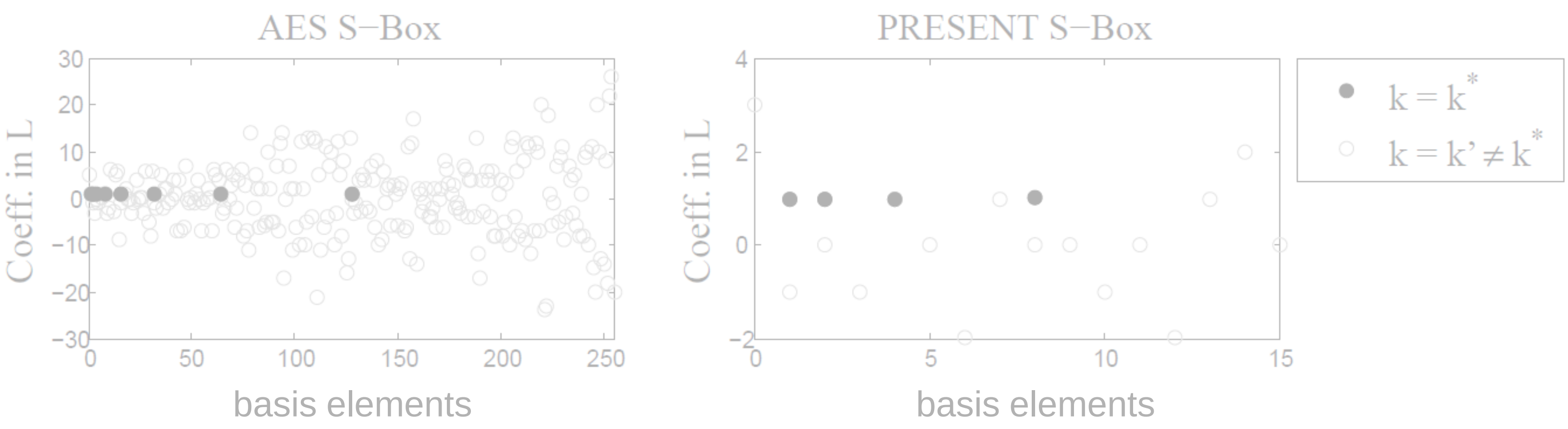
- e.g., if $\mathrm{L}(z_i) = \mathrm{HW}(z_i)$



- Full basis perfectly explains any L by overfitting
  - Even for incorrect key candidates!

Separation result (II)

- e.g., if $\mathrm{L}(z_i) = \mathrm{HW}(z_i)$



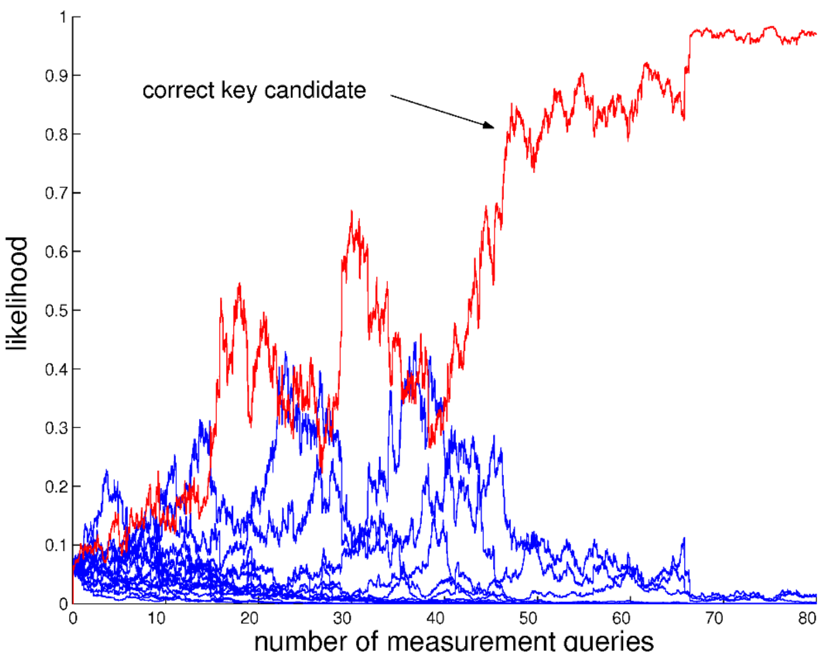
- Full basis perfectly explains any L by overfitting
  - Even for incorrect key candidates!

⇒ Non-profiled DPA needs a good assumption

- e.g., the model is linear, simple, …
- This, *in general*, is only provided by profiling

Illustration

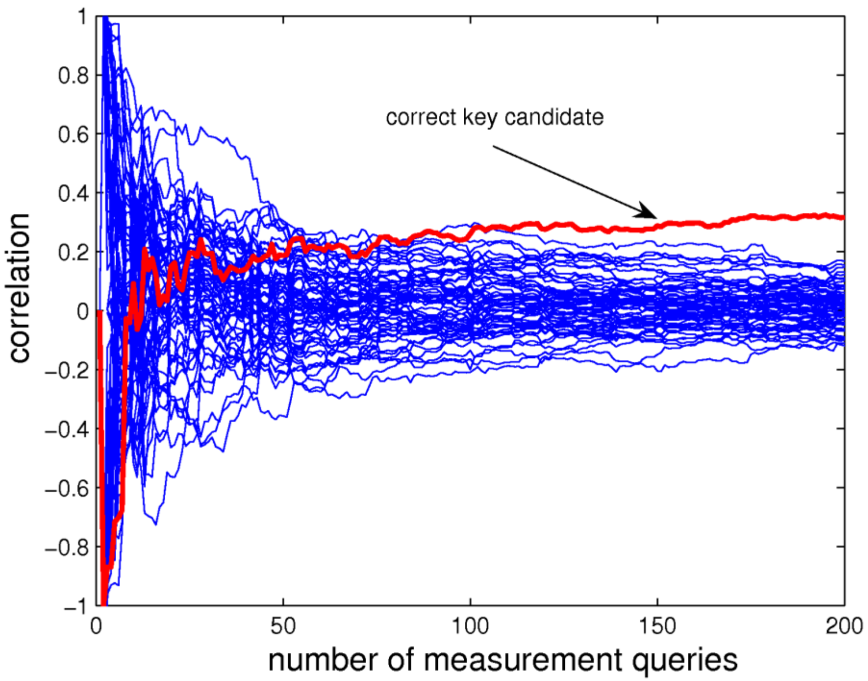
Gaussian templates



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

- More efficient **(better model)**
- Outputs probabilities

CPA

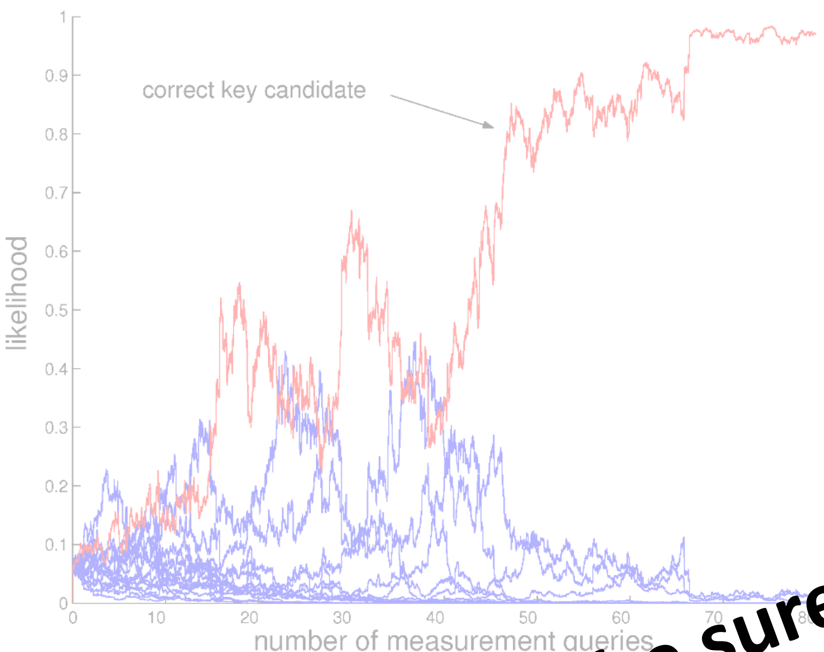


$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \frac{\mathrm{E}(L \cdot M^{k^*}) - \mathrm{E}(L) \cdot \mathrm{E}(M^{k^*})}{\sigma(L) \cdot \sigma(M^{k^*})}$$

- Less efficient **(worse model)**
- Outputs scores

Illustration

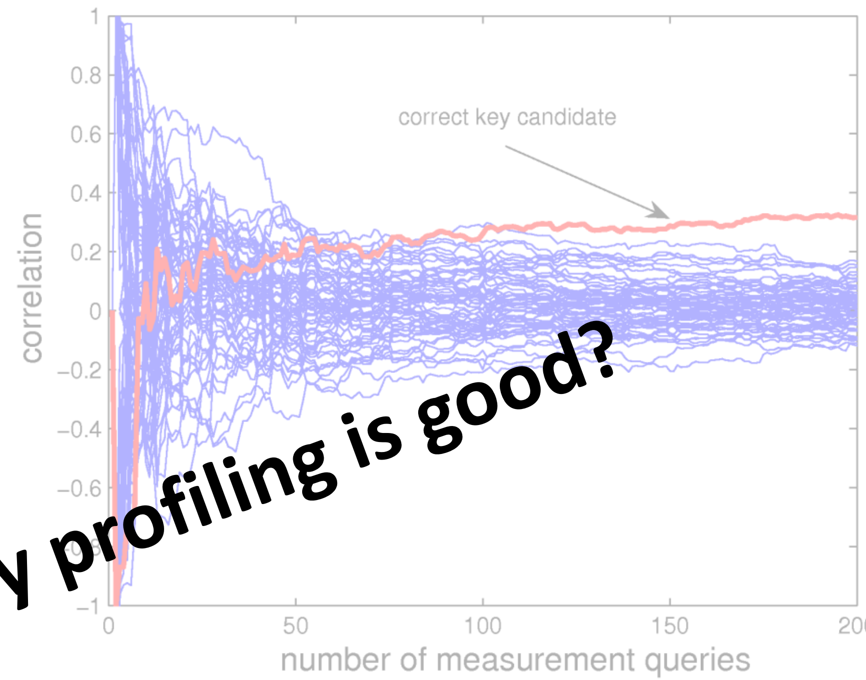
Gaussian templates



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

- More efficient **(better model)**
- Outputs probabilities

CPA



$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \frac{\mathrm{E}\left(L\cdot M^{k^*}\right) - \mathrm{E}(L)\cdot\mathrm{E}\left(M^{k^*}\right)}{\sigma(L)\cdot\sigma\left(M^{k^*}\right)}$$

- Less efficient **(worse model)**
- Outputs scores

How can I be sure my profiling is good?

Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

Model optimality caveats

- A model is optimal if $\widehat{\Pr}_{model}\,[l|k] = \Pr_{chip}\,[l|k]$

Model optimality caveats

- A model is optimal if $\widehat{\Pr}_{model}[l|k] = \Pr_{chip}[l|k]$

$\Rightarrow$ Theory would say it is ε-close to optimal if

$$\mathrm{SD}(\widehat{\Pr}_{model}[l|k], \Pr_{chip}[l|k]) < \varepsilon$$

- (with SD a statistical distance)

Model optimality caveats

- A model is optimal if $\widehat{\Pr}_{model}\,[l|k] = \Pr_{chip}\,[l|k]$

$\Rightarrow$ Theory would say it is ε-close to optimal if

$$\mathrm{SD}(\widehat{\Pr}_{model}\,[l|k], \Pr_{chip}\,[l|k]) < \varepsilon$$

- (with SD a statistical distance)

- Convenient since ε would quantify the loss
  - That could be reported in SR bounds [DFS15]

[DFS15] A Duc, S. Faust, F.-X. Standaert, *Making Masking Security Proofs Concrete […]*, EUROCRYPT 2015.

- A model is optimal if $\widehat{\Pr}_{model}[l|k] = \Pr_{chip}[l|k]$

$\Rightarrow$ Theory would say it is ε-close to optimal if

$$\mathrm{SD}(\widehat{\Pr}_{model}[l|k], \Pr_{chip}[l|k]) < \varepsilon$$

- (with SD a statistical distance)

- Convenient since ε would quantify the loss
  - That could be reported in SR bounds [DFS15]

- Problem: $\Pr_{chip}[l|k]$ is unknown

[DFS15] A Duc, S. Faust, F.-X. Standaert, *Making Masking Security Proofs Concrete [...]*, EUROCRYPT 2015.

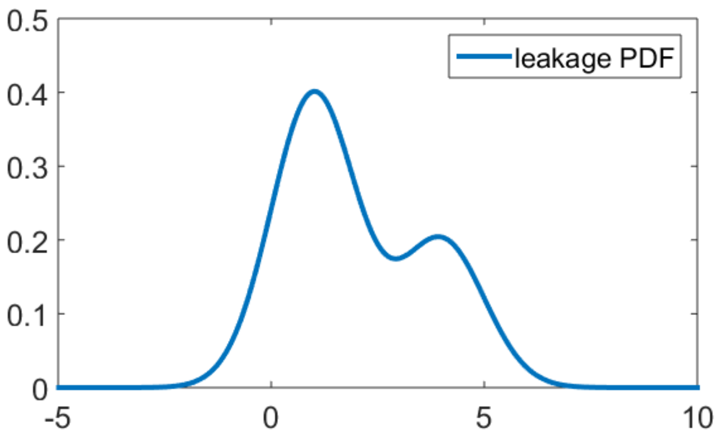
Leakage certification [DSV14]

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

Leakage certification [DSV14]

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.
- Example:

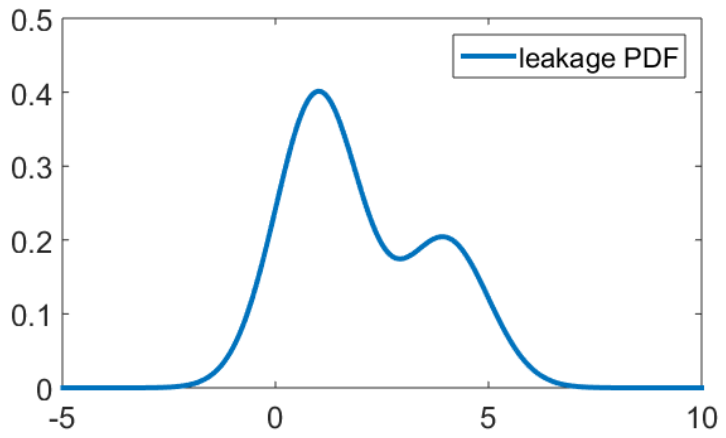


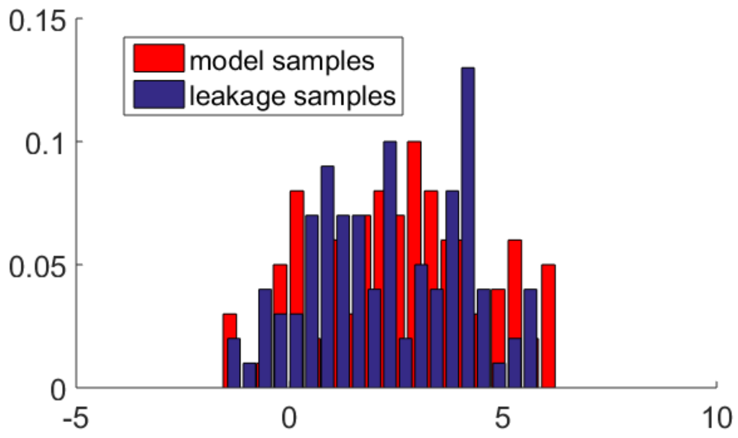
[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

- Example:

No samples





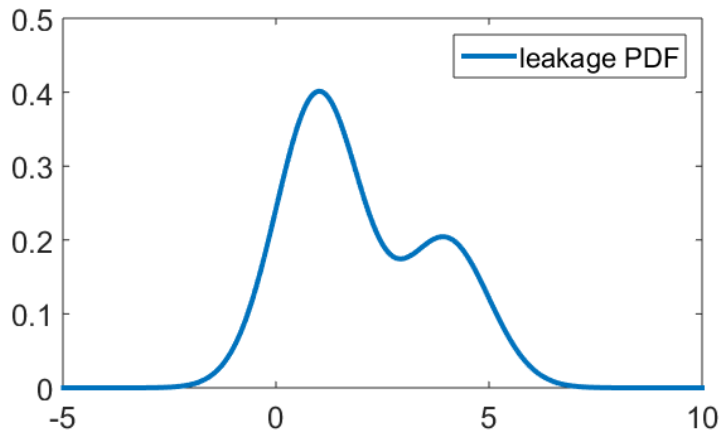
[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

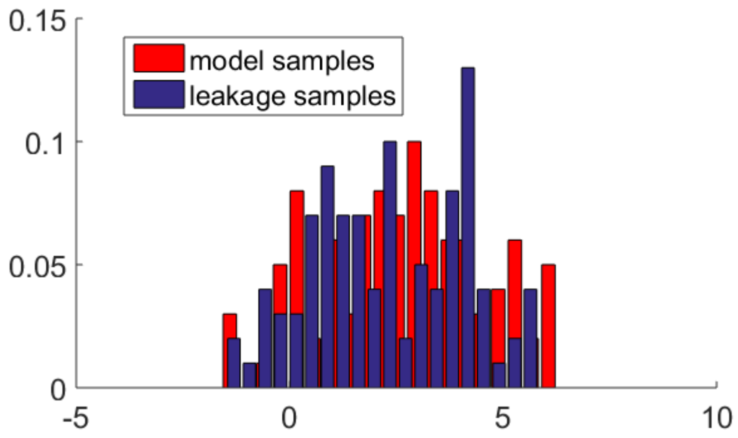
Leakage certification [DSV14]

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

- Example:

estimation errors dominate



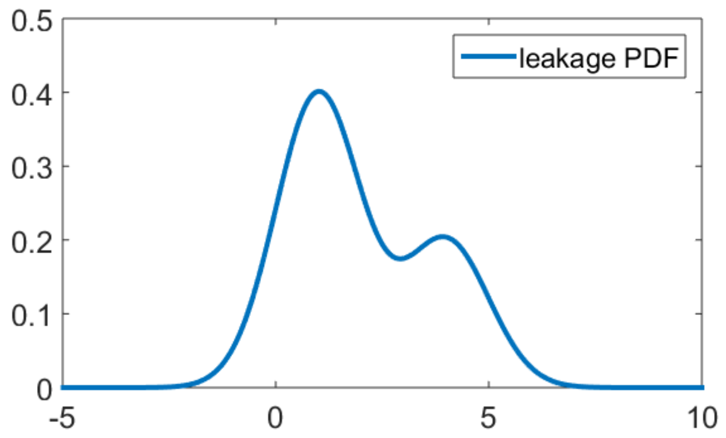


$\Rightarrow$ need to measure more

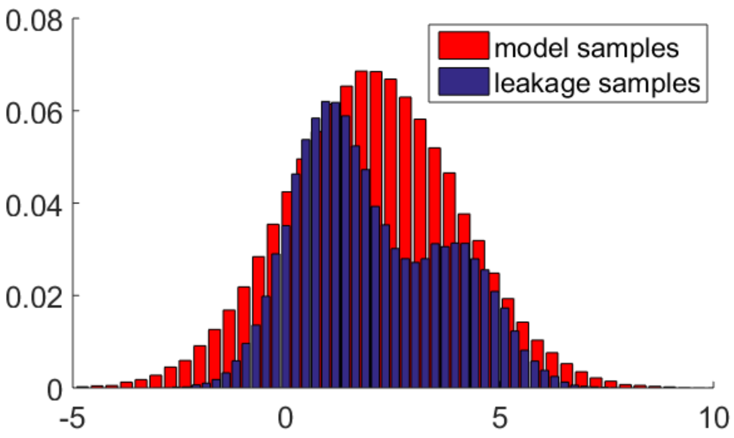
[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

- Example:



$N_1 > N_0$ samples



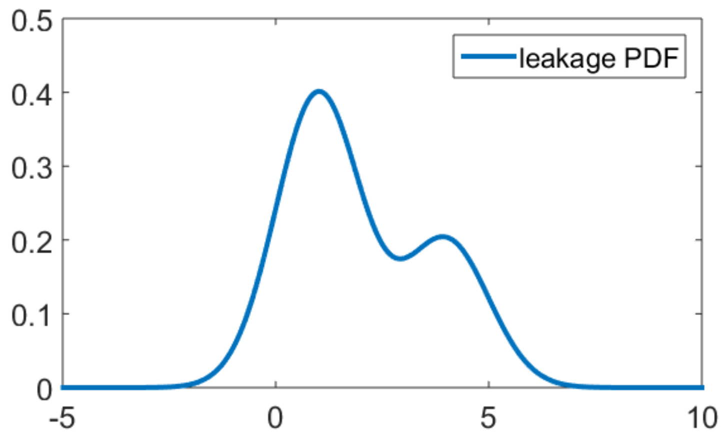
[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

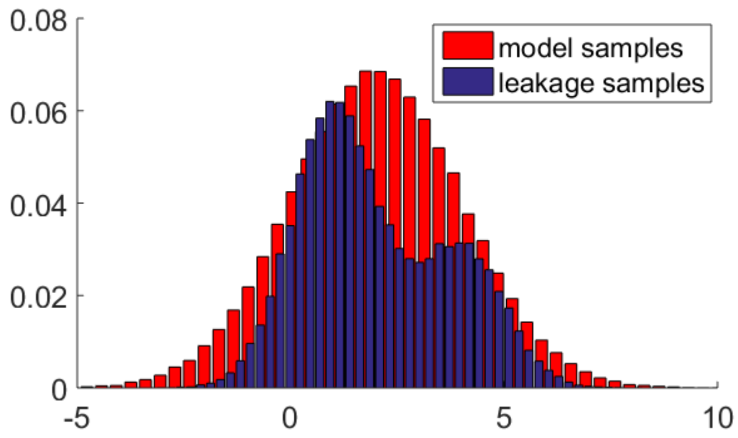
Leakage certification [DSV14]

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

- Example:

assumption errors dominate





$\Rightarrow$ **need another model**

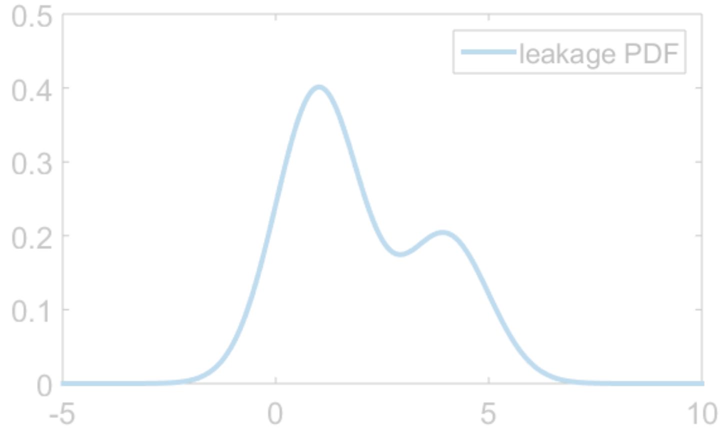
[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

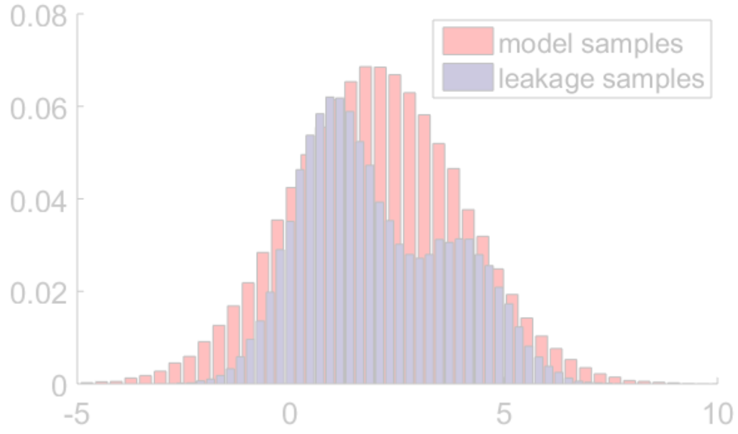
Leakage certification [DSV14]

- Distinguish estimation & assumption errors
  - Recall estimation errors decrease with # meas.

- Example:

assumption errors dominate





$\Rightarrow$ **need another model**

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

[DSV14] F. Durvaux, F.-X. Standaert, N. Veyrat-Charvillon, *How to Certify the Leakage of a Chip*, EUROCRYPT 2014.

Leakage certification (II)

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \overset{N}{=} \Pr_{chip}[l|k]$$

Leakage certification (II)

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \stackrel{N}{=} \Pr_{chip}[l|k]$$

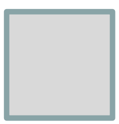
- Taking advantage of cross-validation

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \stackrel{N}{=} \Pr_{chip}[l|k]$$

- Taking advantage of cross-validation


modeling samples


test samples

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \overset{N}{=} \Pr_{chip}[l|k]$$

- Taking advantage of cross-validation

Leakage certification (II)

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \overset{N}{=} \Pr_{chip}[l|k]$$

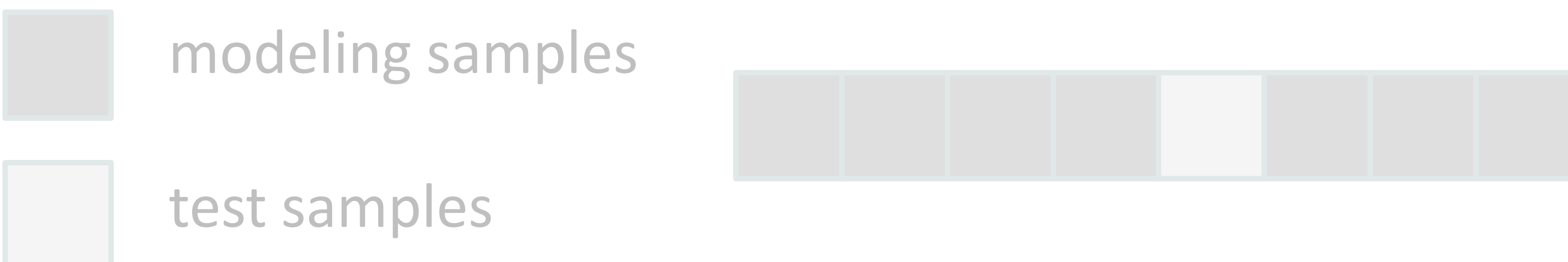
- Taking advantage of cross-validation

Leakage certification (II)

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \stackrel{N}{=} \Pr_{chip}[l|k]$$

- Taking advantage of cross-validation

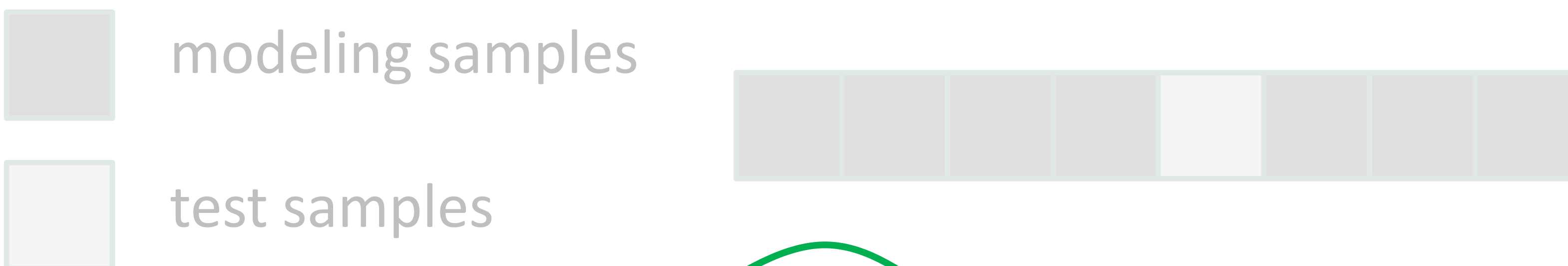


- Output a p-value p(*N*)
  - Small p's indicate hyp. is likely incorrect

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \overset{N}{=} \Pr_{chip}[l|k]$$

- Taking advantage of cross-validation



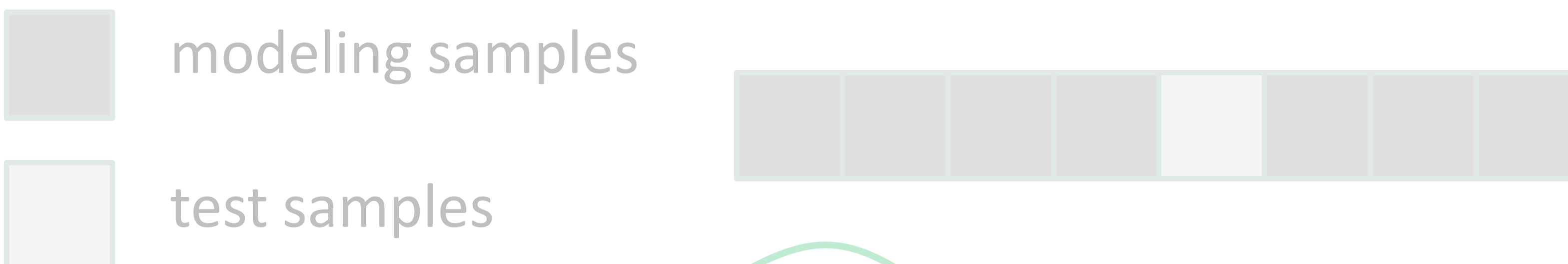
- Output a p-value p(N) Eval. lab. limit
  - Small p's indicate hyp. is likely incorrect

Leakage certification (II)

- Test the hypothesis that

$$\widehat{\Pr}_{model}[l|k] \stackrel{N}{=} \Pr_{chip}[l|k]$$

- Taking advantage of cross-validation



- Output a p-value p(N) Eval. lab. limit
  - Small p's indicate hyp. is likely incorrect

- Main drawback: cost (of sampling distributions)

Towards easy certification

Towards easy certification

- Compare moments (rather than distributions)

1. $\widehat{M}_d \overset{N}{\leftarrow} \widehat{\Pr}_{model}\,[l|k]$
2. $\widetilde{M}_d \overset{N}{\leftarrow} \Pr_{chip}\,[l|k]$
3. Test equality

$$\widehat{M}_d = \widetilde{M}_d$$

Towards easy certification

- Compare moments (rather than distributions)

1. $\widehat{M}_d \xleftarrow{N} \widehat{\Pr}_{model}\,[l|k]$
2. $\widetilde{M}_d \xleftarrow{N} \Pr_{chip}\,[l|k]$
3. Test equality $\widehat{M}_d = \widetilde{M}_d$

\+ Can be done with simple univariate tests

- e.g., T-test (assuming $\widehat{M}_d, \widetilde{M}_d$ are Gaussian)

Towards easy certification

- Compare moments (rather than distributions)

1. $\widehat{M}_d \overset{N}{\leftarrow} \widehat{\Pr}_{model}\,[l|k]$
2. $\widetilde{M}_d \overset{N}{\leftarrow} \Pr_{chip}\,[l|k]$
3. Test equality $\widehat{M}_d = \widetilde{M}_d$

\+ Can be done with simple univariate tests

- e.g., T-test (assuming $\widehat{M}_d, \widetilde{M}_d$ are Gaussian)

– Is it theoretically sound? No!

Towards easy certification

- Compare moments (rather than distributions)

1. $\widehat{M}_d \xleftarrow{N} \widehat{\Pr}_{model}\,[l|k]$
2. $\widetilde{M}_d \xleftarrow{N} \Pr_{chip}\,[l|k]$

3. Test equality
$$\widehat{M}_d = \widetilde{M}_d$$

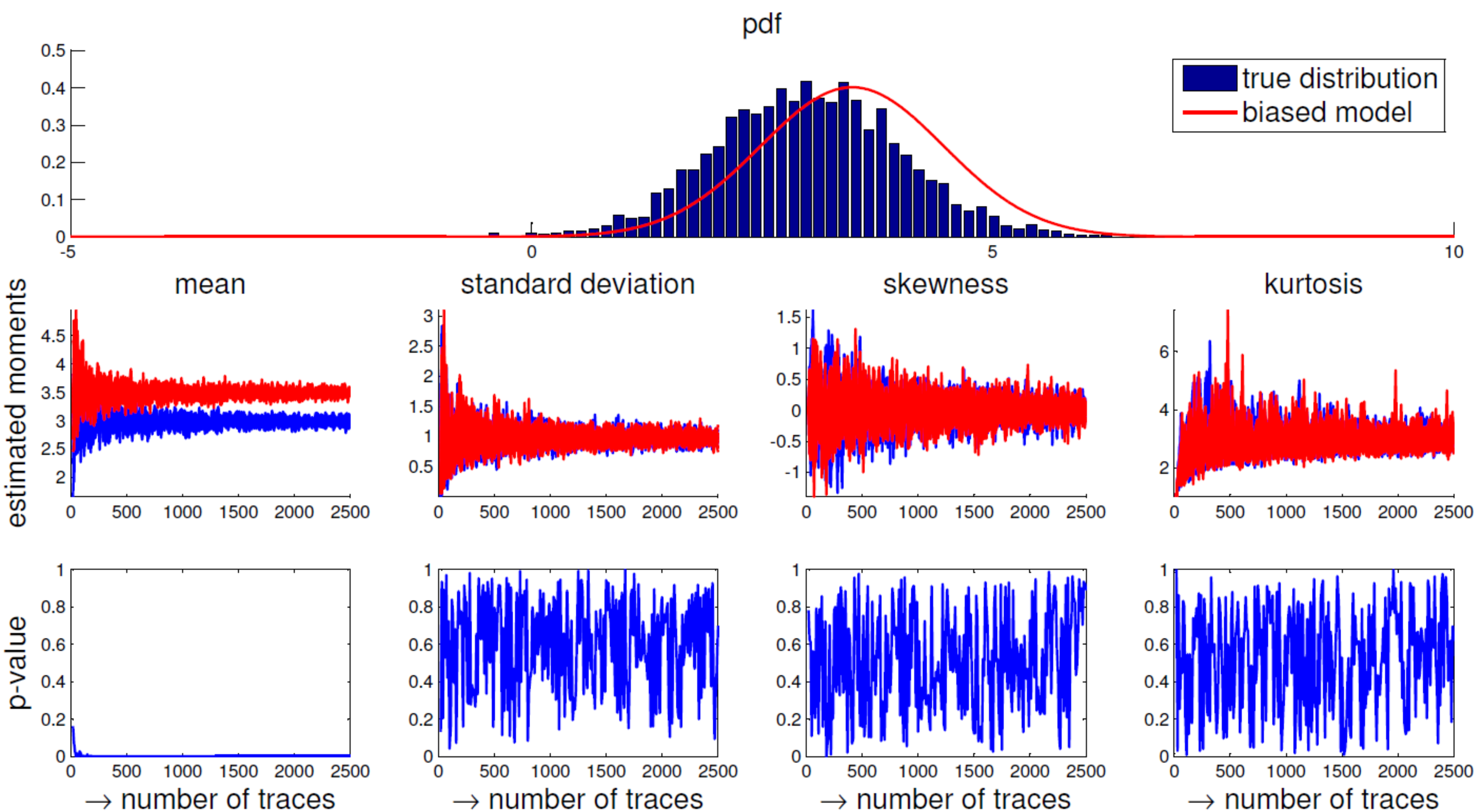
\+ Can be done with simple univariate tests

- e.g., T-test (assuming $\widehat{M}_d, \widetilde{M}_d$ are Gaussian)

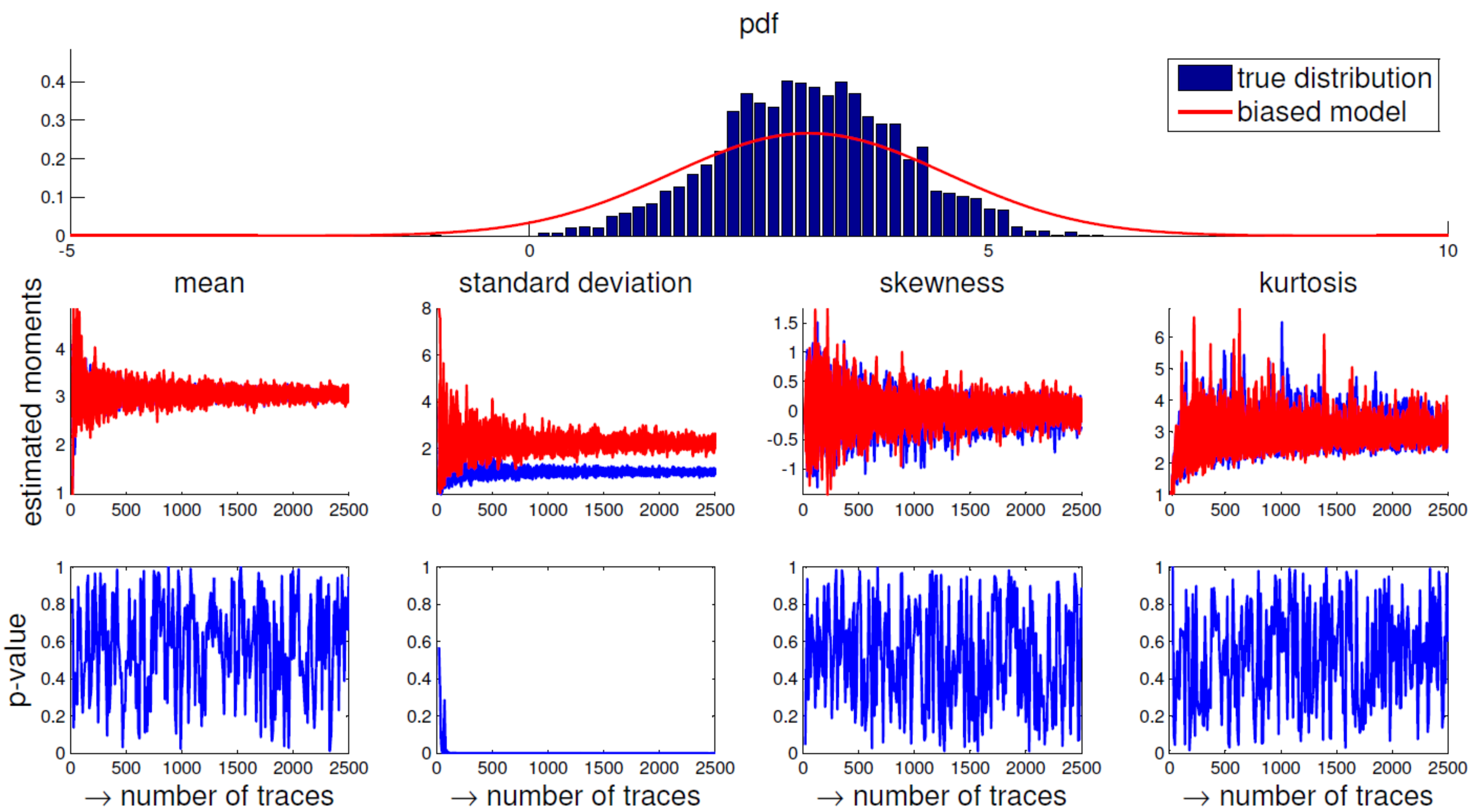
– Is it theoretically sound? No!

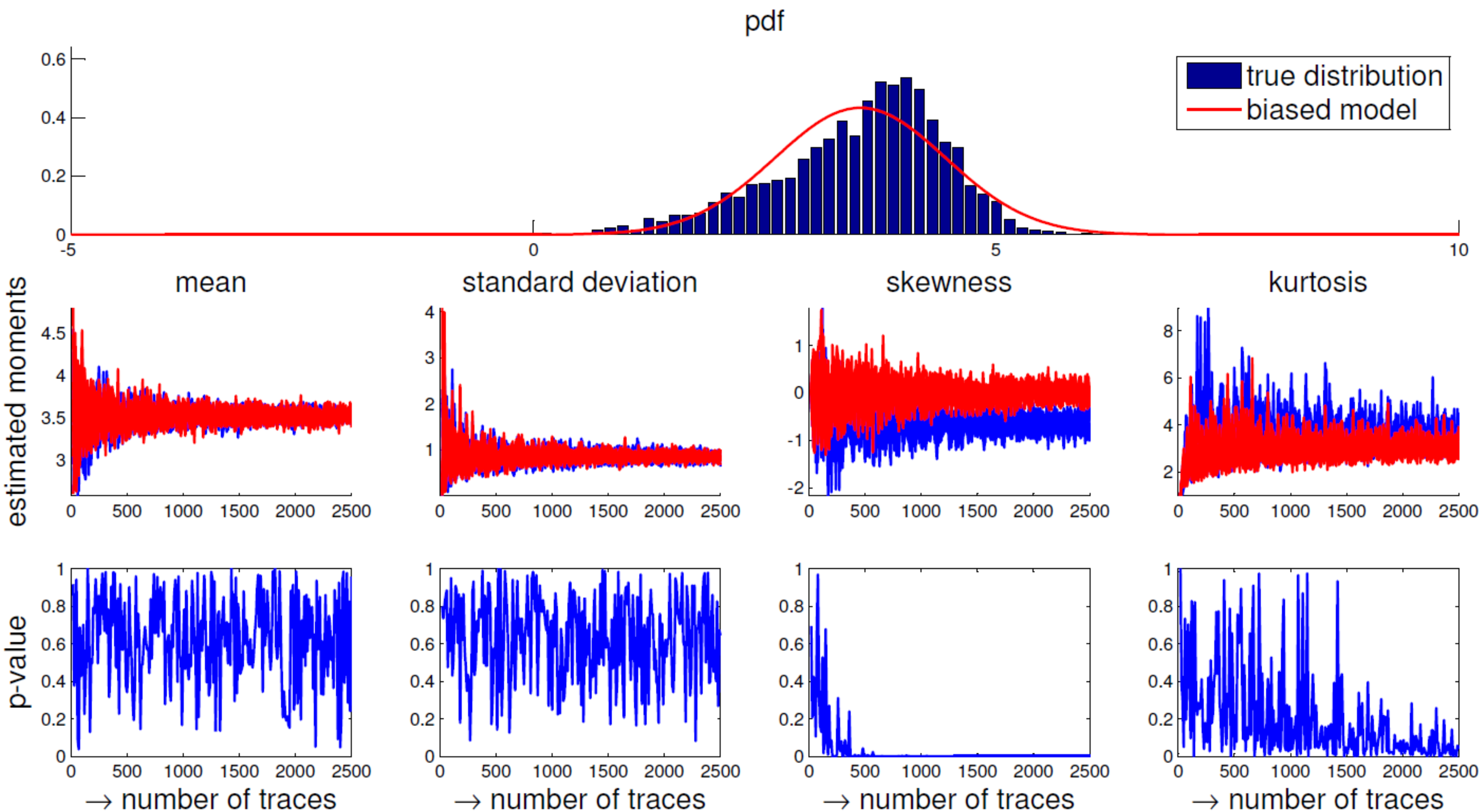
- But counterexamples are involved
- & SCA literature frequently does it
  - Leakage detection, HO attacks, …

Simulated examples

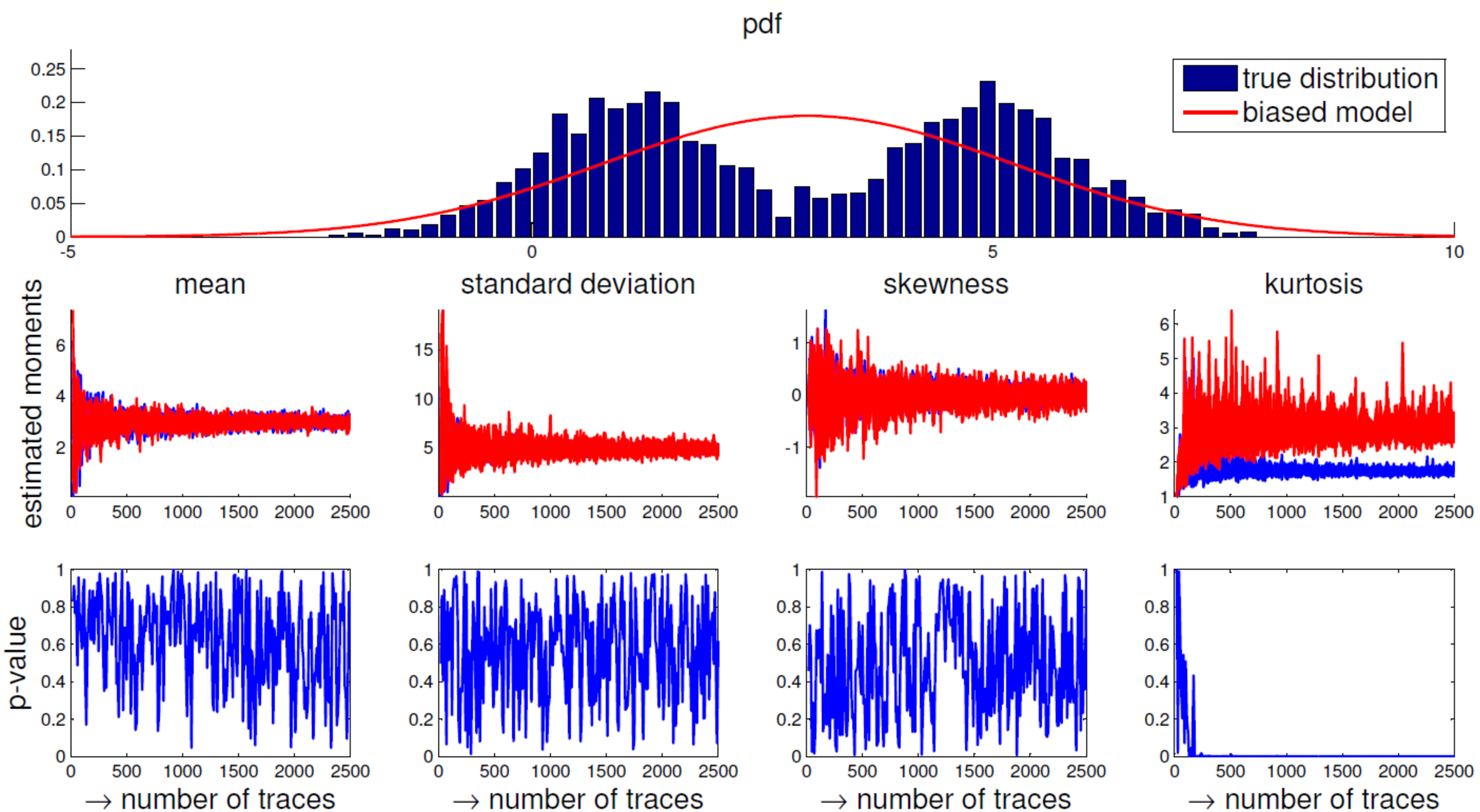


Simulated examples




pdf
true distribution
biased model
mean
standard deviation
skewness
kurtosis
estimated moments
p-value
→ number of traces
→ number of traces
→ number of traces
→ number of traces

Simulated examples



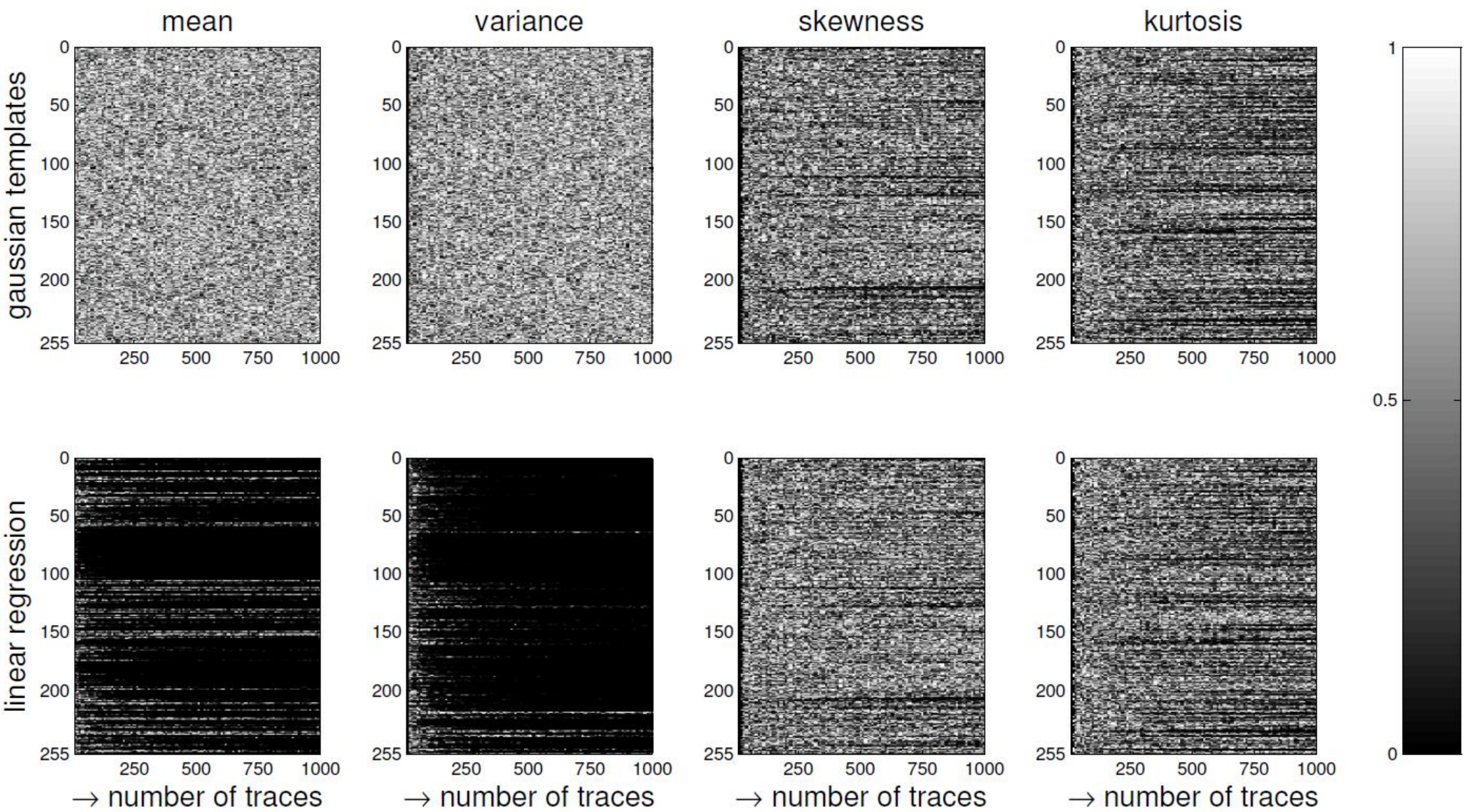
Software experiments

- Repeating the Eurocrypt 2014 case study

Software experiments

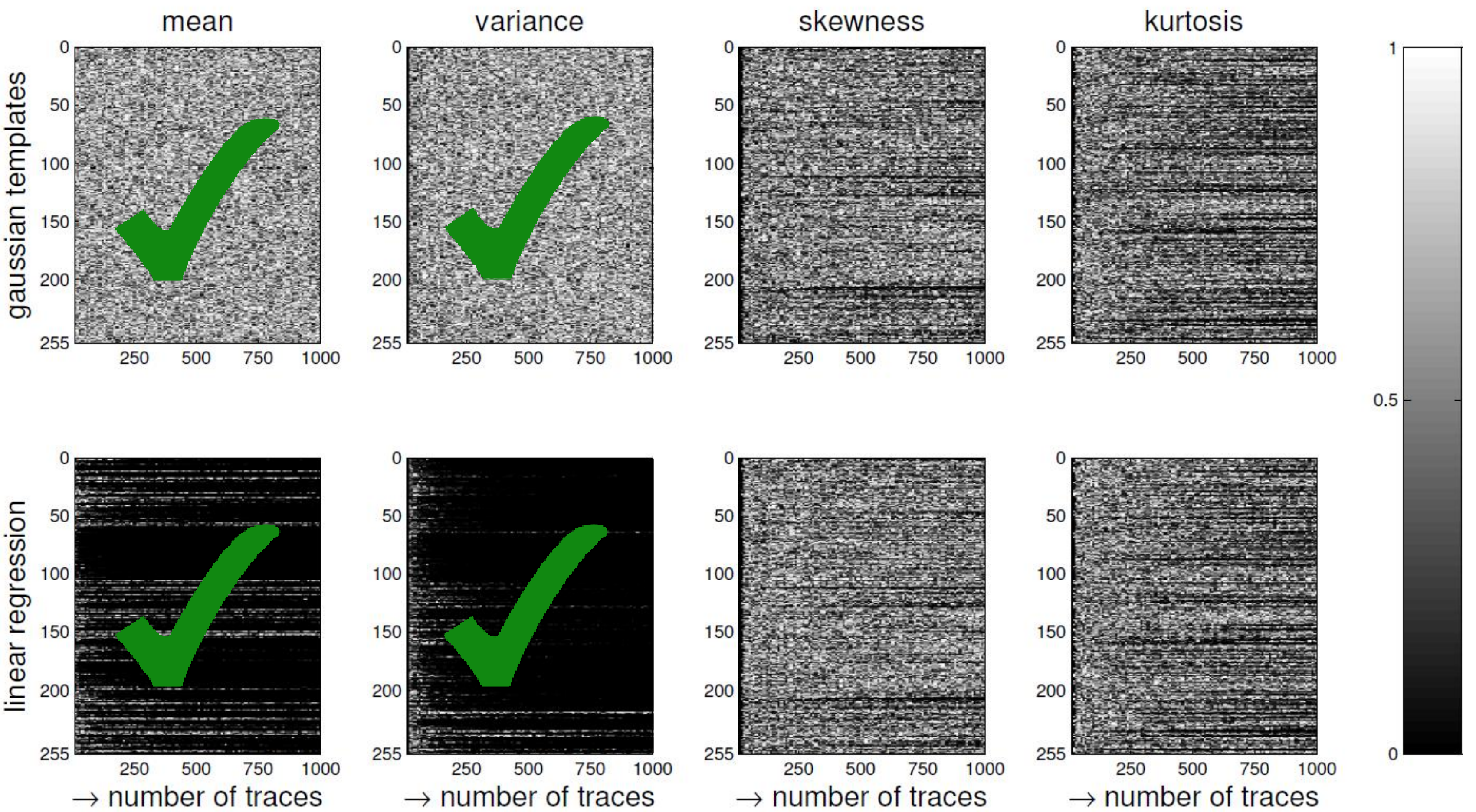
- Unprotected AES implementation, Atmel AVR

- Unprotected AES implementation, Atmel AVR



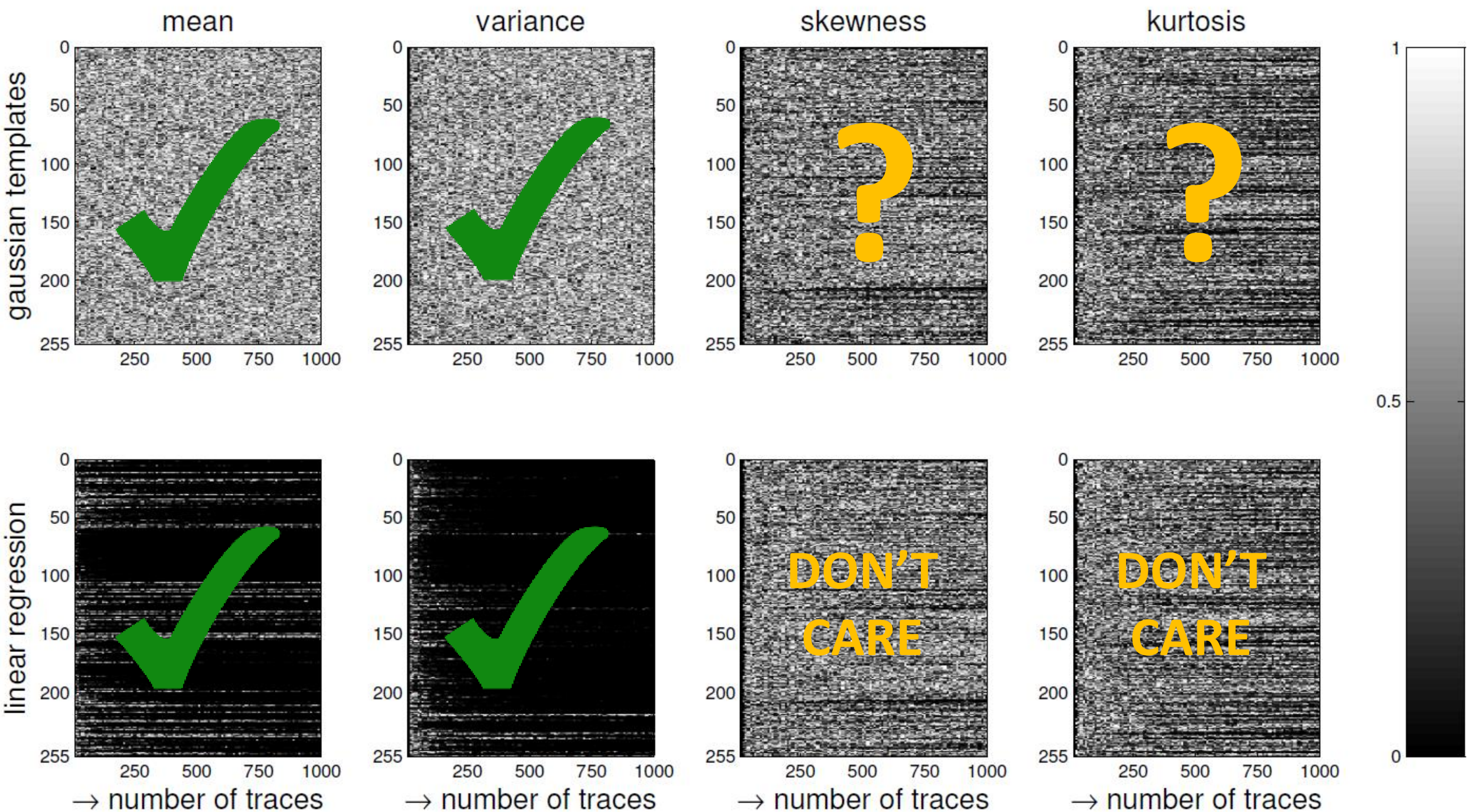
Software experiments

- Unprotected AES implementation, Atmel AVR



Software experiments

- Unprotected AES implementation, Atmel AVR



Software experiments (II)

- Eurocrypt 2014: no errors detected with up to 256x1000 measurements & Gaussian template
- CHES 2016: small errors in $\widetilde{M}_3$ and $\widetilde{M}_4$

$\Rightarrow$ *Is there an inconsistency in our results?*

$\Rightarrow$ Do these errors lead to significant information loss

Software experiments (II)

- Eurocrypt 2014: no errors detected with up to 256x1000 measurements & Gaussian template
- CHES 2016: small errors in $\widetilde{M}_3$ and $\widetilde{M}_4$

$\Rightarrow$ *Is there an inconsistency in our results?*

$\Rightarrow$ Do these errors lead to significant information loss

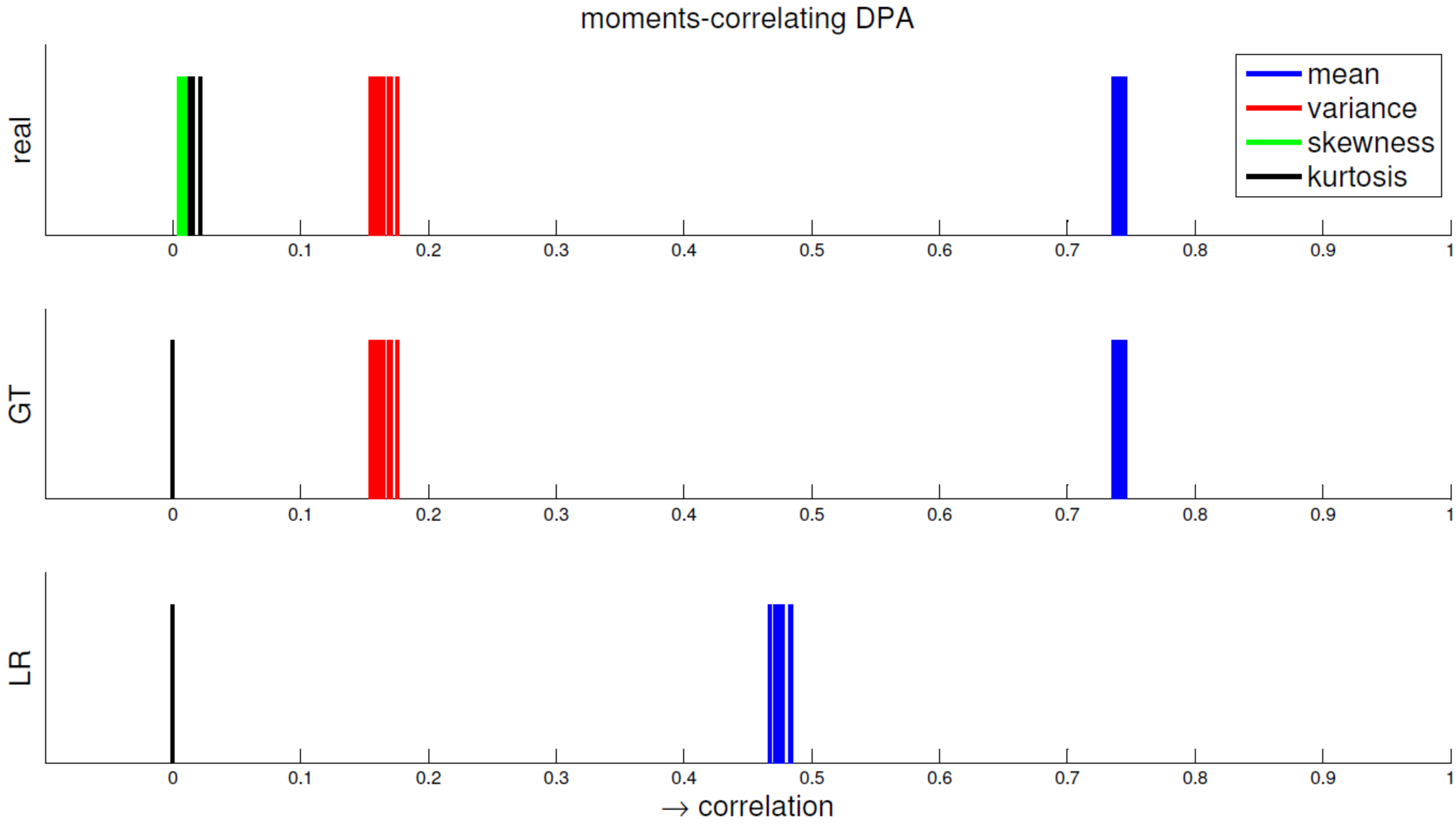
- Additional test: Moments-Correlating DPA [MS14]

$$\text{MPC-DPA}(d) = \hat{\rho}(\hat{M}_d, l^d)$$

- Metric intuition: $N_s = \frac{c}{\hat{\rho}(\hat{M}_d, l^d)^2}$

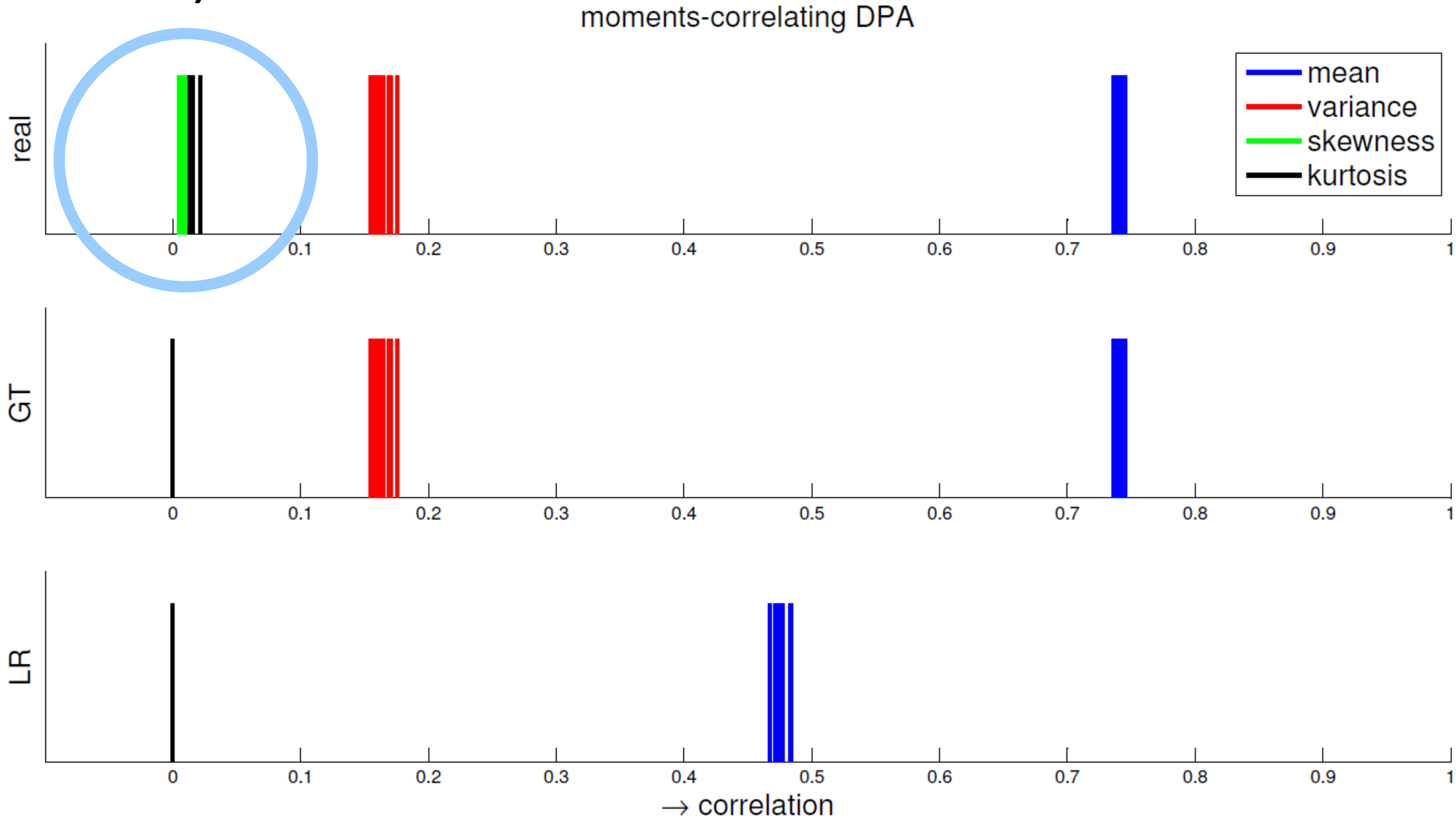
[MS14] A. Moradi, F.-X. Standaert, Moments-Correlating DPA, Theory of Implementations workshop, 2014.

Software experiments (III)

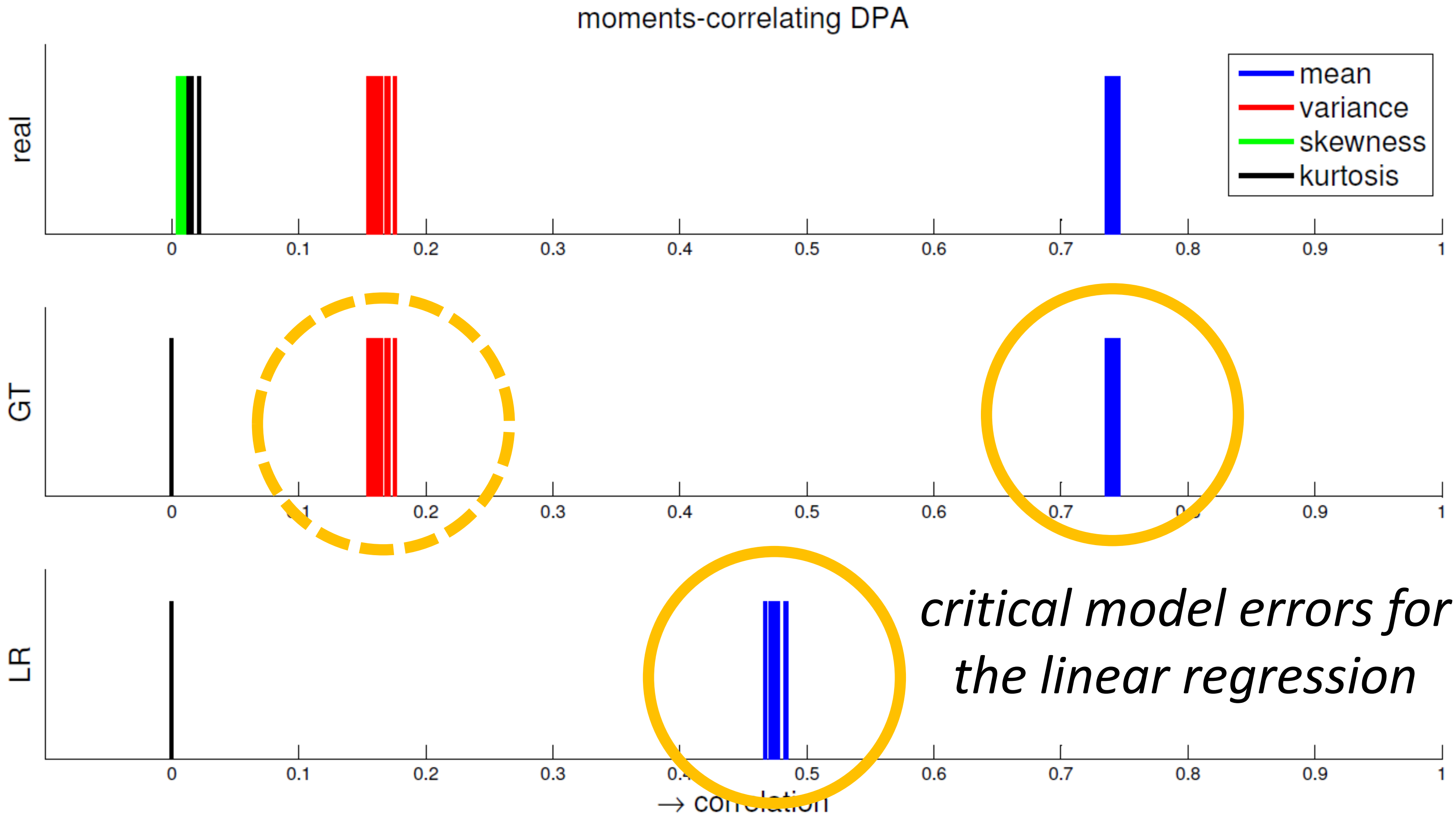


Software experiments (III)

little information in skewness/kurtosis



Software experiments (III)

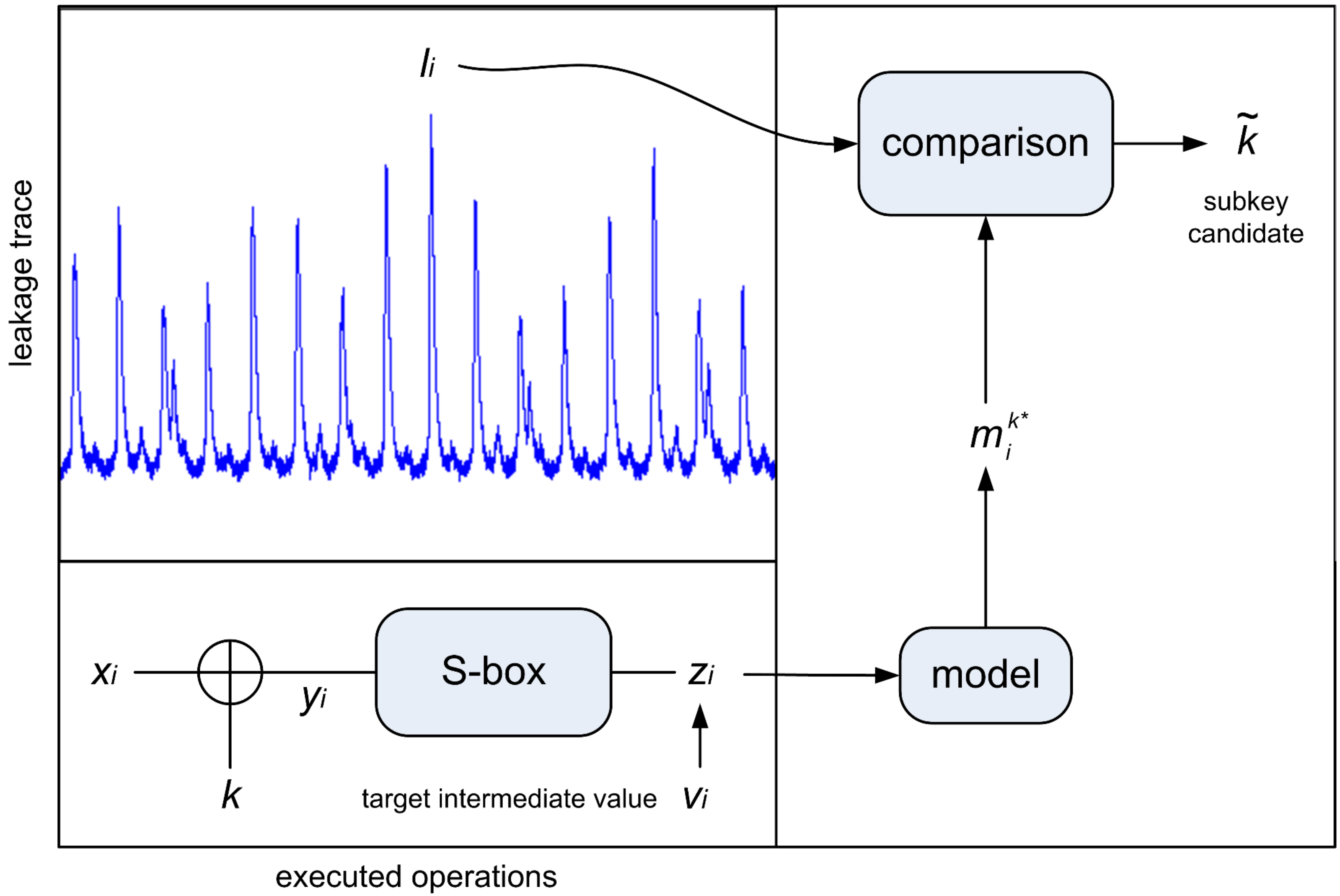


critical model errors for the linear regression

Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

Standard DPA

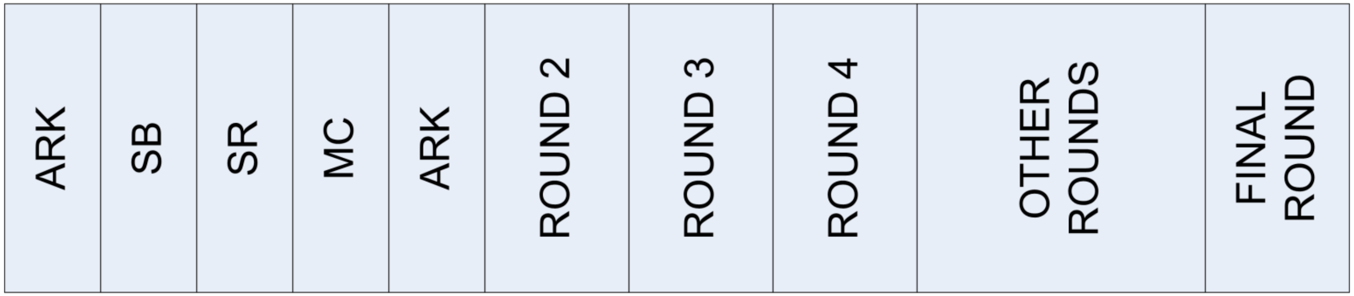


How to extract more information?

- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …

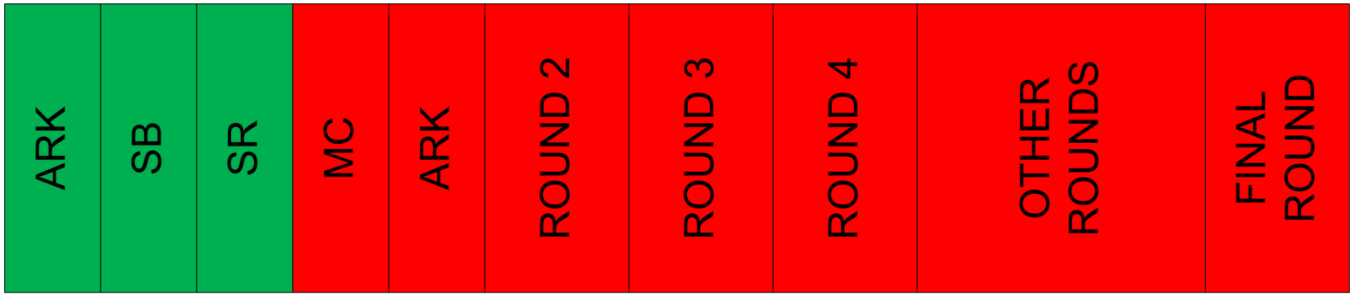
How to extract more information?

- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …
- More intermediate values



How to extract more information?

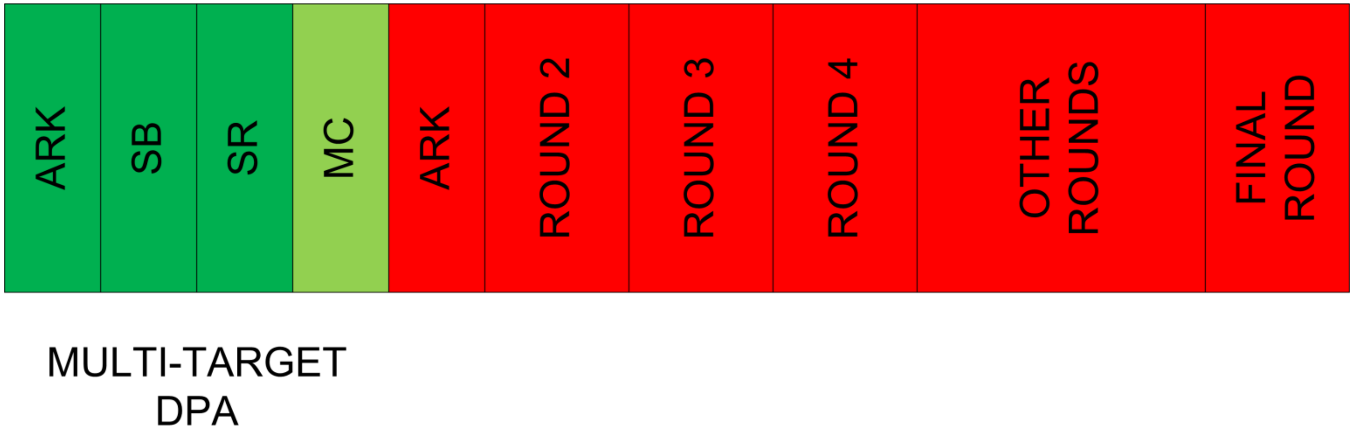
- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …
- More intermediate values (CRYPTO 1998)



STANDARD
DPA

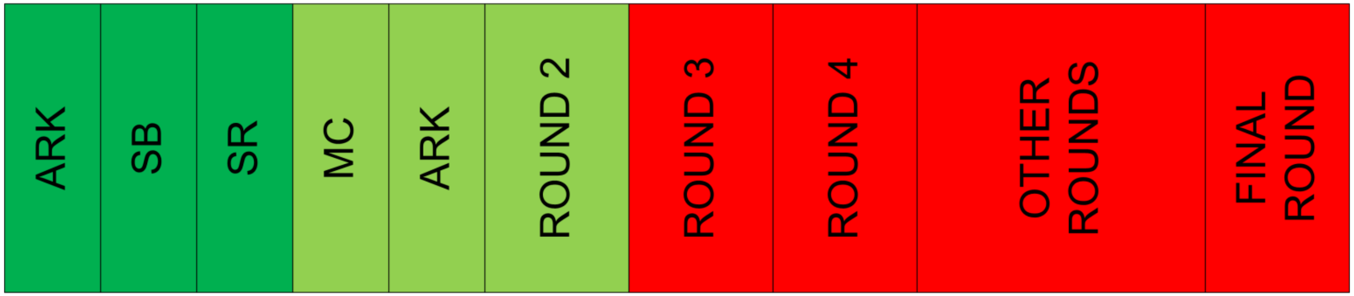
How to extract more information?

- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …
- More intermediate values (ASIACRYPT 2014)



How to extract more information?

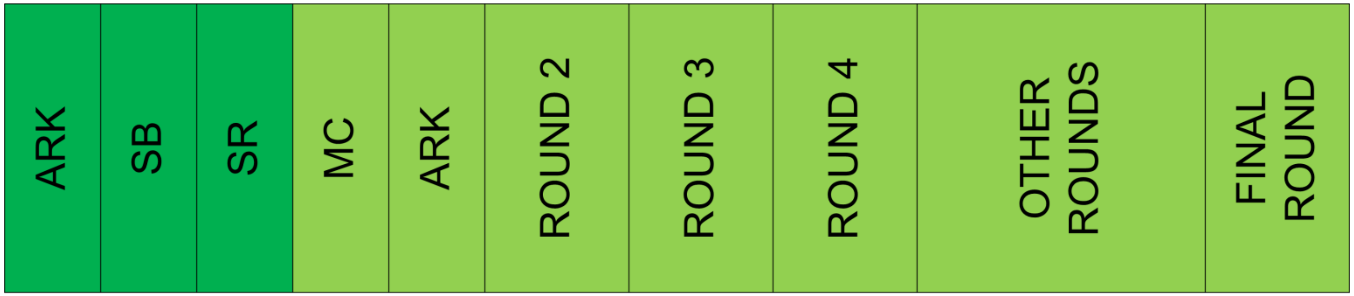
- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …
- More intermediate values (FSE 2003)



COLLISION ATTACKS

How to extract more information?

- More samples per intermediate value
  - Multivariate DPA
  - e.g., dimensionality reduction
    - Principal Component Analysis (PCA)
    - Linear Discriminant Analysis (LDA)
    - …
- More intermediate values (CHES 2009)



ALGEBRAIC SIDE-CHANNEL ATTACKS (ASCA)

ASCA limitations

- Cannot deal with measurement noise
  - Despite progresses, e.g., Tolerant ASCA
- Large (time and) memory complexities
  - Limited to a single plaintext, typically
    - Sometimes even less

ASCA limitations

- Cannot deal with measurement noise
  - Despite progresses, e.g., Tolerant ASCA
- Large (time and) memory complexities
  - Limited to a single plaintext, typically
    - Sometimes even less

$\Rightarrow$ Emerging intuition: ASCA require “hard” information and only standard DPA can efficiently exploit “soft” (probabilistic) information obtained from the measurements of multiple plaintexts

ASCA limitations

- Cannot deal with measurement noise
  - Despite progresses, e.g., Tolerant ASCA
- Large (time and) memory complexities
  - Limited to a single plaintext, typically
    - Sometimes even less

$\Rightarrow$ Emerging intuition: ASCA require “hard” information and only standard DPA can efficiently exploit “soft” (probabilistic) information obtained from the measurements of multiple plaintexts

- ***Our contribution***: show this intuition is incorrect!

SASCA (I): the factor graph

- Representation of the algorithm/implementation

SASCA (I): the factor graph

- Representation of the algorithm/implementation
- Based on two types of nodes

SASCA (I): the factor graph

- Representation of the algorithm/implementation
- Based on two types of nodes
  - Variable nodes x_i (i.e., intermediate values)

SASCA (I): the factor graph

- Representation of the algorithm/implementation
- Based on two types of nodes
  - Variable nodes x_i (i.e., intermediate values)
  - Factor nodes of two types
    - A priori knowledge $f_i(x_i)=\Pr[x_i|L]$
      - ***Exactly the output of standard DPA!***
    - Operations: $f(x_1, x_2, x_3) = \begin{cases} 1 & \text{if } OP(x_1, x_2)=x_3 \\ 0 & \text{otherwise} \end{cases}$

- Representation of the algorithm/implementation
- Based on two types of nodes
  - Variable nodes x_i (i.e., intermediate values)
  - Factor nodes of two types
    - A priori knowledge $f_i(x_i)=\Pr[x_i|L]$
      - ***Exactly the output of standard DPA!***
    - Operations: $f(x_1, x_2, x_3) = \begin{cases} 1 \text{ if } OP(x_1, x_2)=x_3 \\ 0 \text{ otherwise} \end{cases}$

- Edges carry two types of messages
  - Type *q* messages: from variables to factors
  - Type *r* messages: from factors to variables

SASCA (II): belief propagation

- Propagates the information (probabilities) through the factor graph via message passing

- Propagates the information (probabilities) through the factor graph via message passing

- From variables to factors

$$q_{v_n \to f_m}(x_n) = \prod_{m' \in M \backslash m} r_{f_{m'} \to v_n}(x_n)$$

- ≈ product over all incoming messages excluding the one of the target factor node

SASCA (II): belief propagation

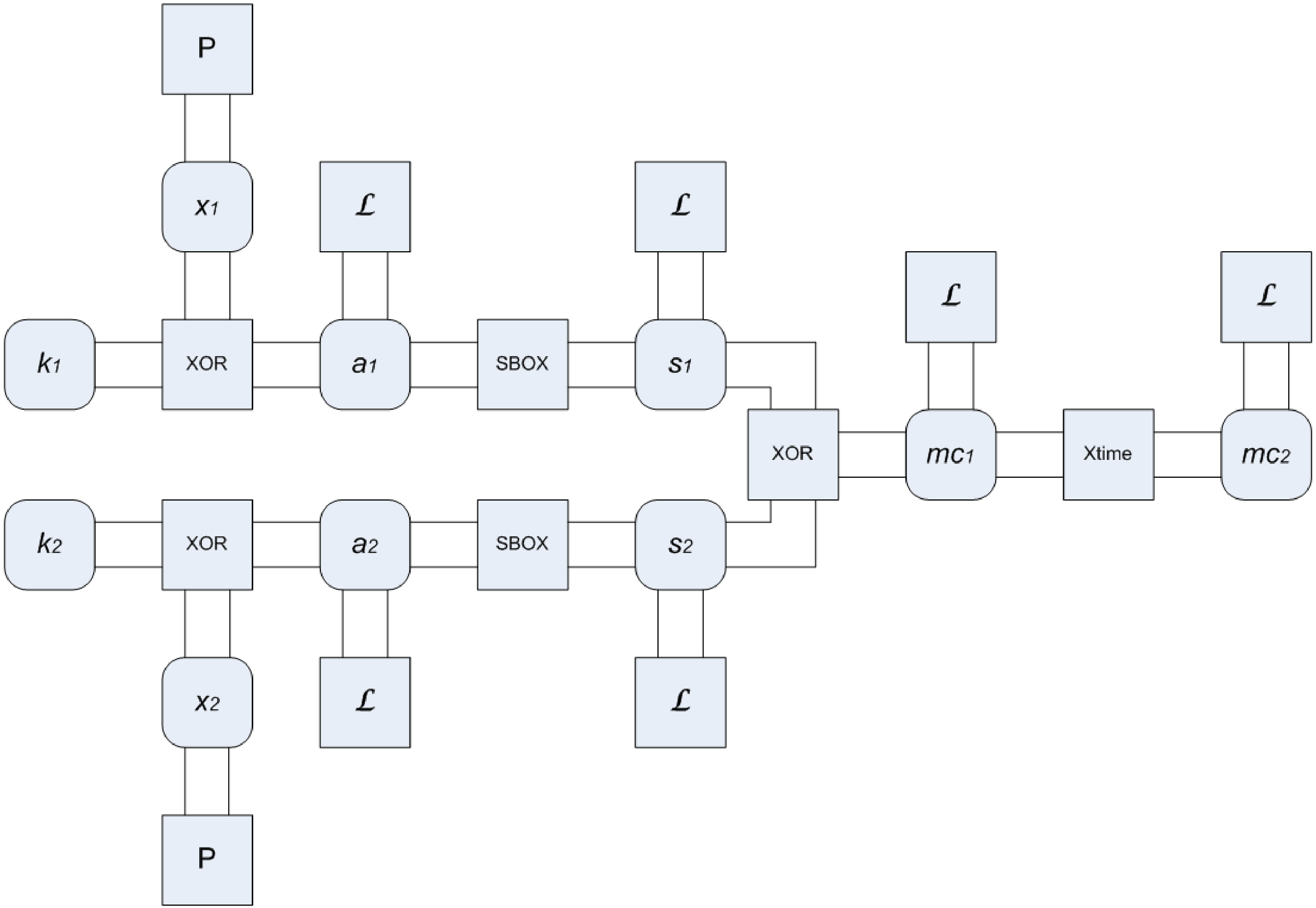
- Propagates the information (probabilities) through the factor graph via message passing

- From factors to variables

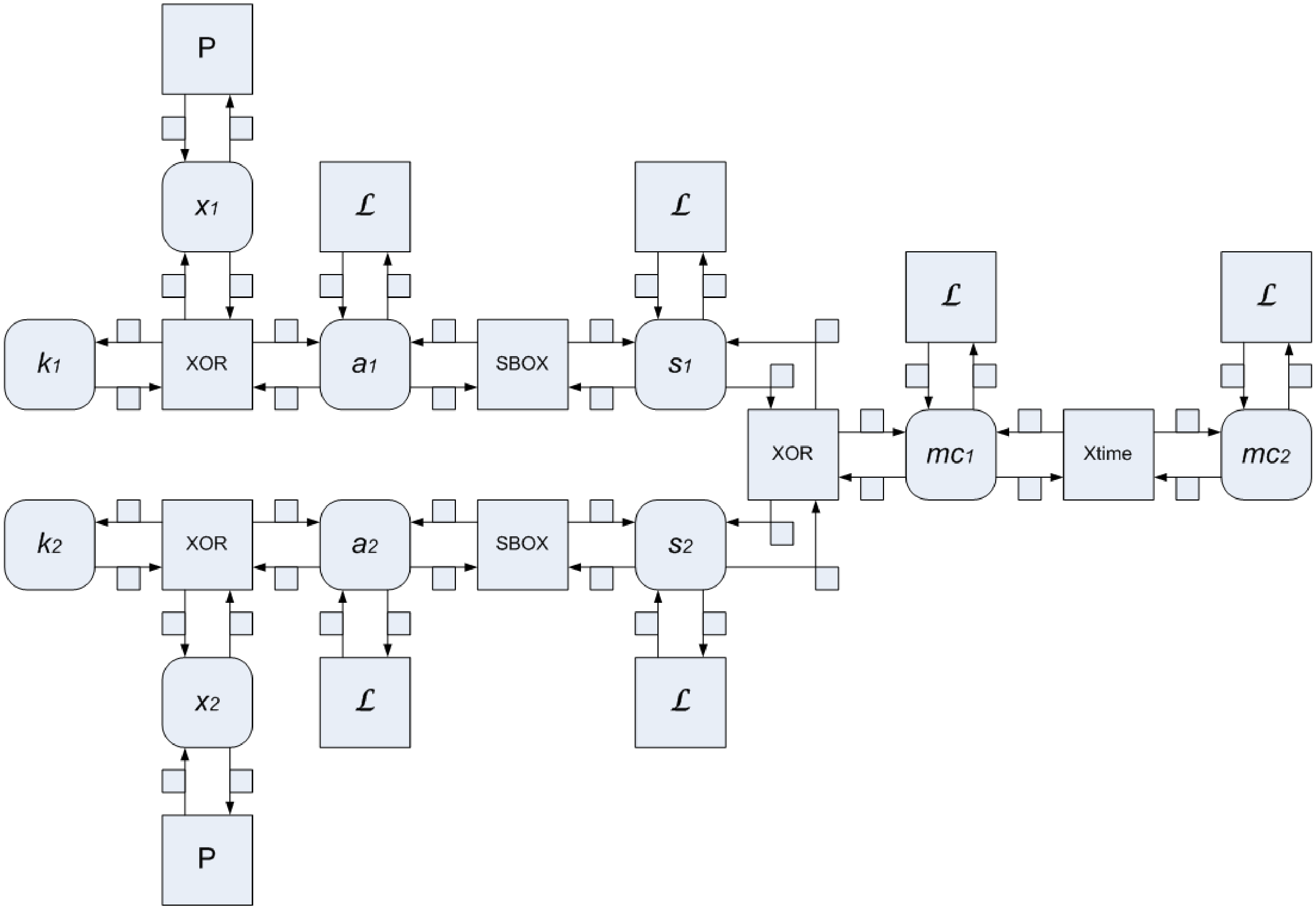
$$r_{f_m \to v_n}(x_n) = \sum_{x_m \neq x_n} f_m(x_m) \cdot \prod_{n' \in N \setminus n} q_{v_{n'} \to f_m}(x_{n'})$$

- ≈ weighted sum of products over all incoming messages excluding the target variable node

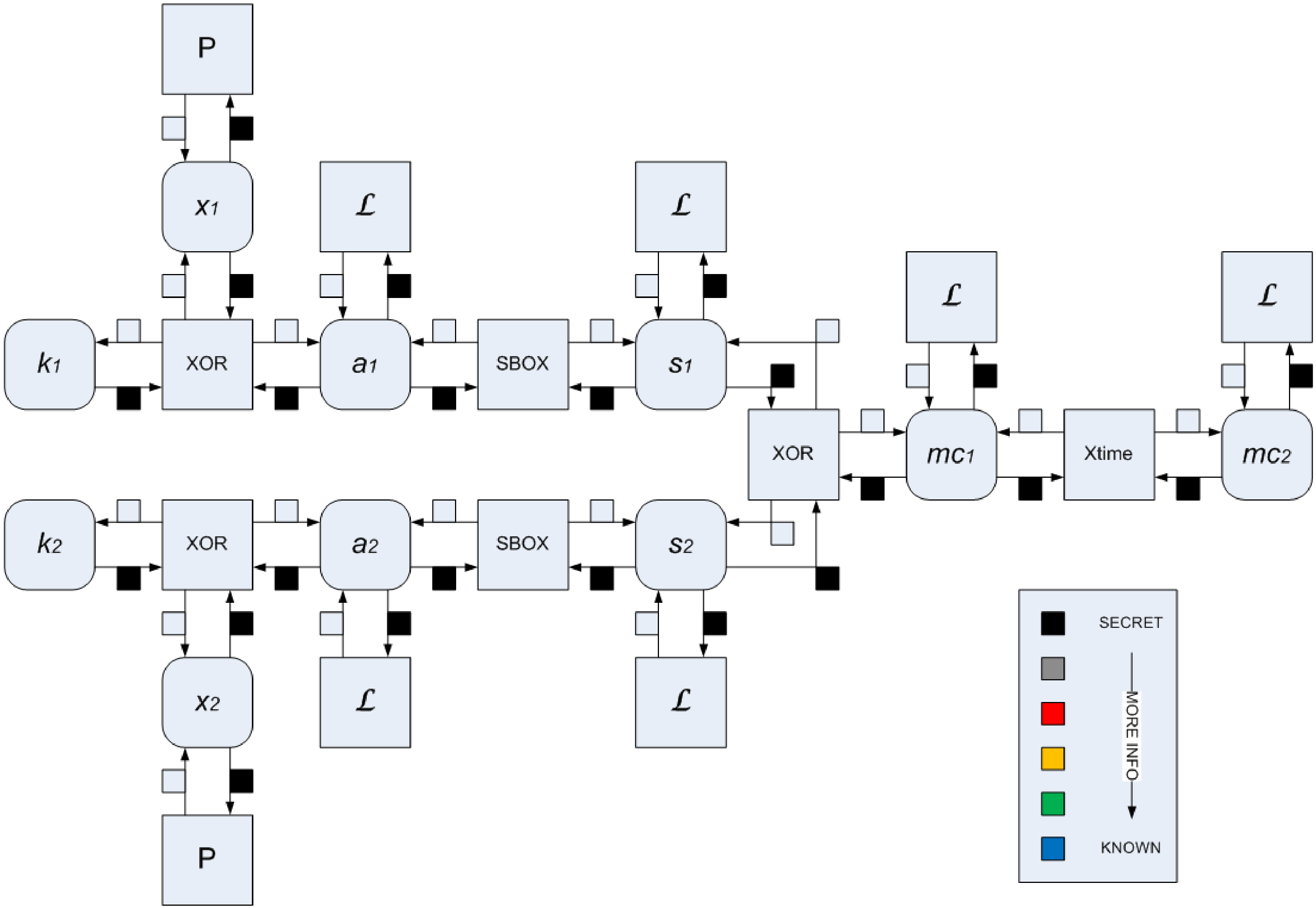
Example (I): factor graph



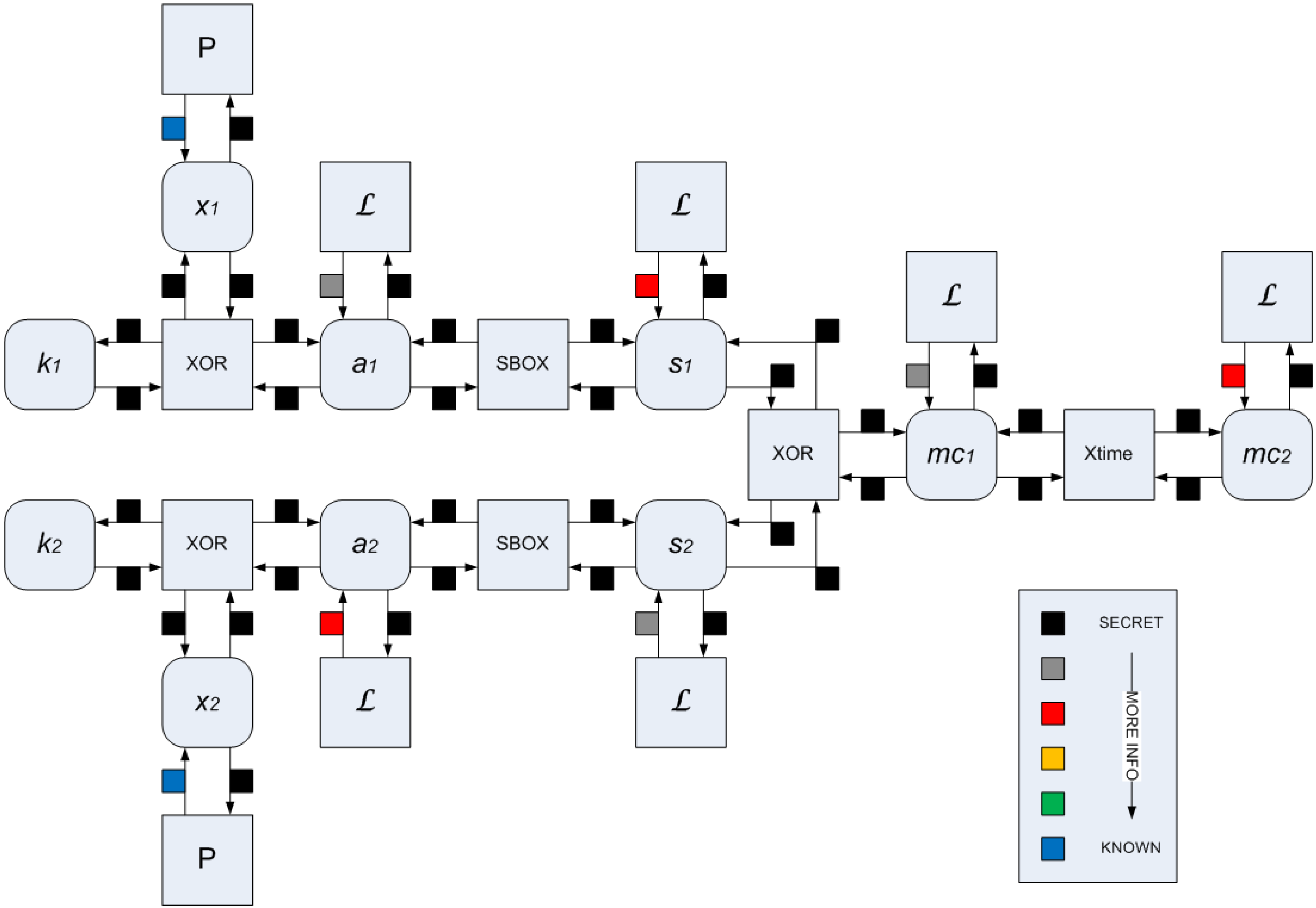
Example (II): adding the messages



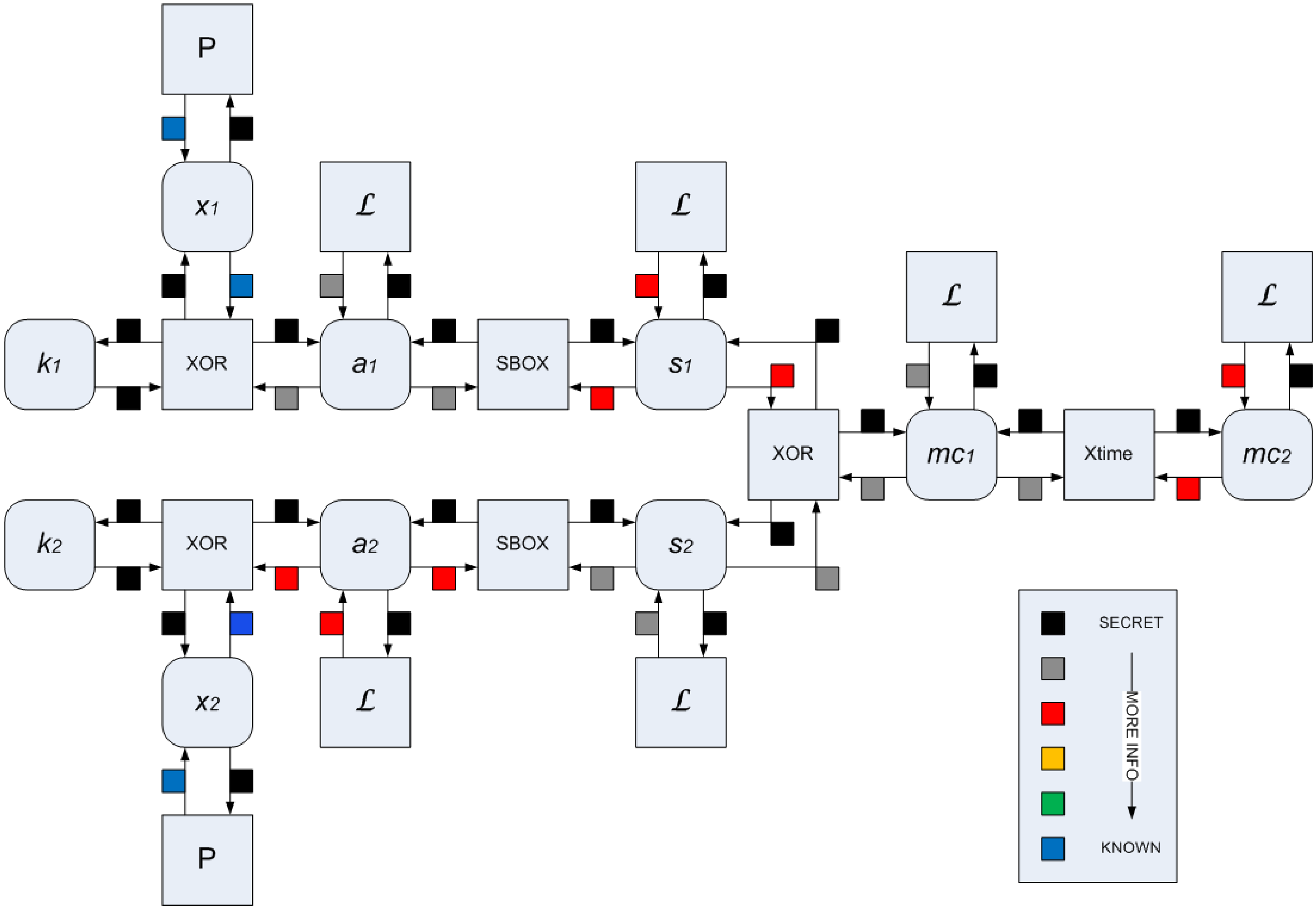
Example (III): initialize the *q*'s (v to f)



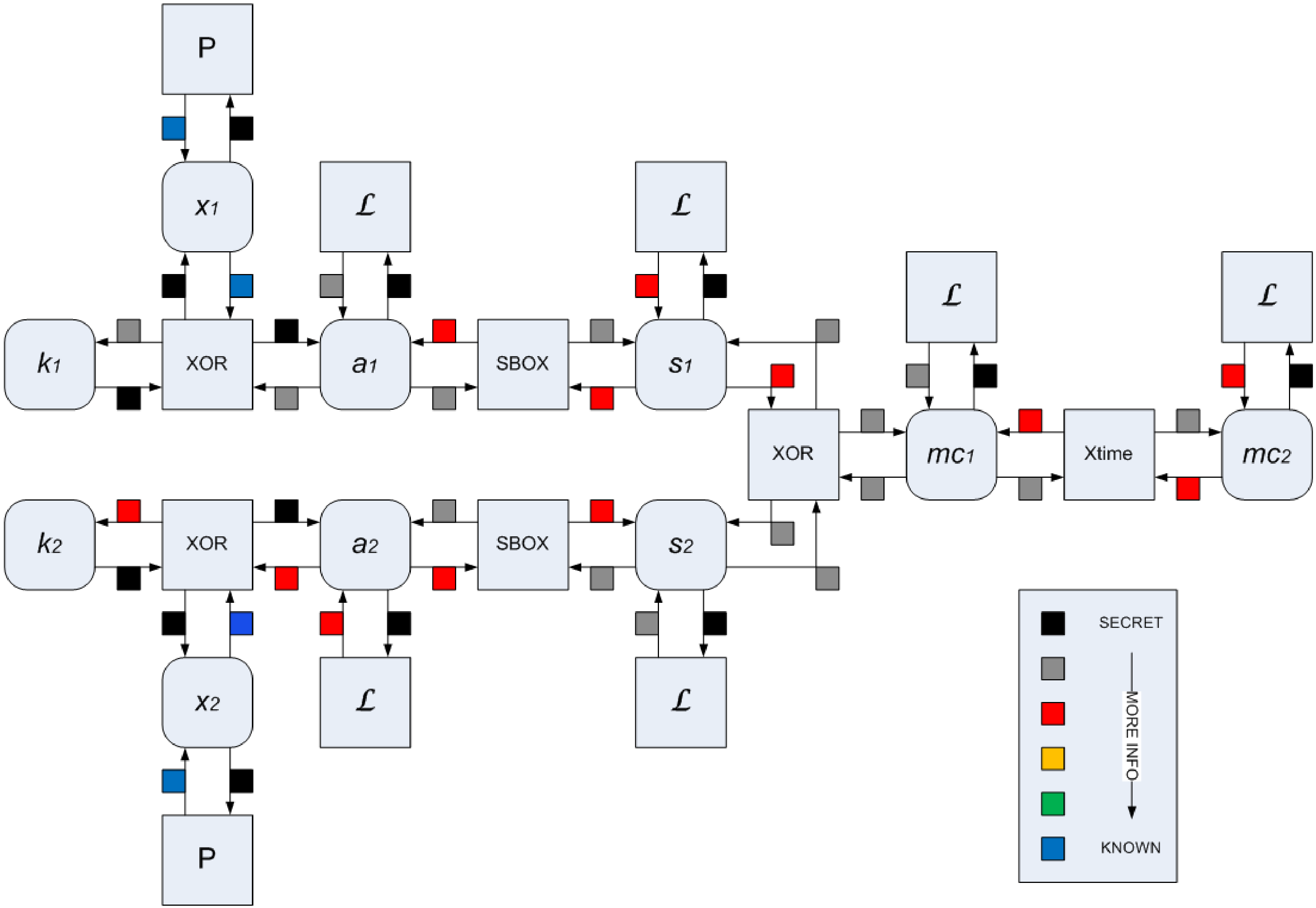
Example (IV): initialize the *r*’s (f to v)



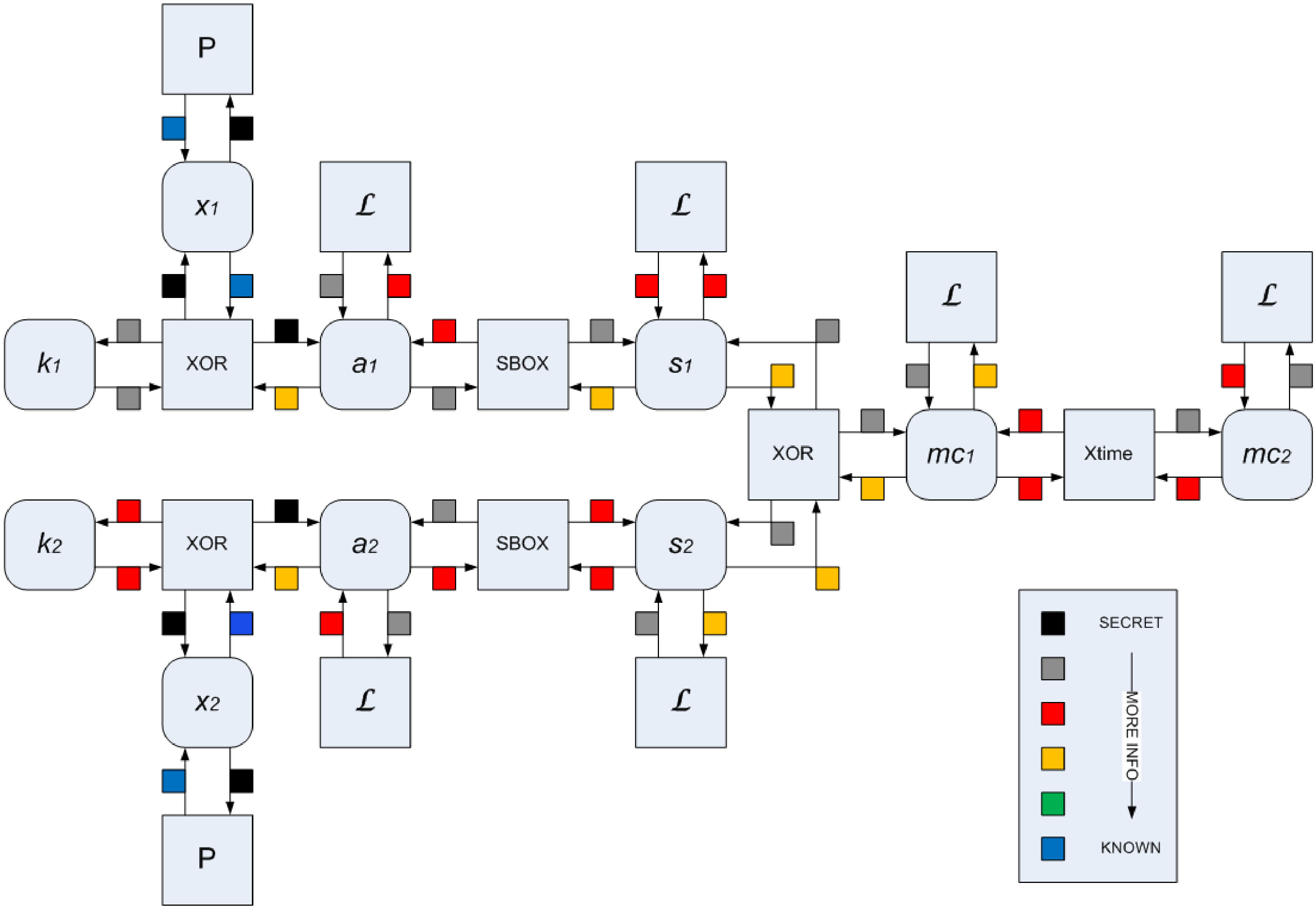
Example (V): update the *q*'s (v to f)



Example (VI): update the *r*'s (f to v)



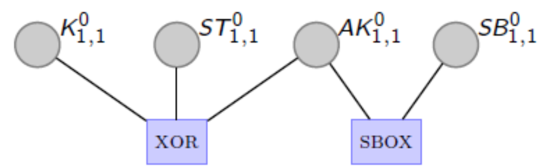
Example (VII): update the q's (v to f)

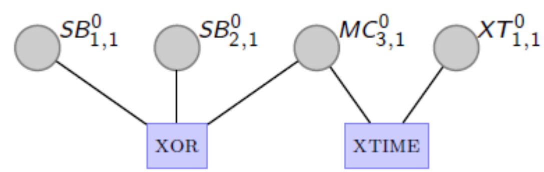


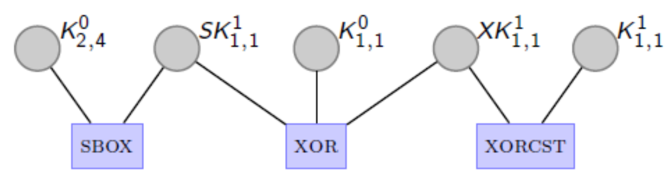
Experimental setting

- Good news: any knowledge can be exploited
- e.g. (open source) AES furious assembly code

ASM code	Graph description	Factor graph

`ld H1, Y+`
`eor ST11, H1`
`mov ZL, ST11`
`lpm ST11, Z`
	`*`
	`_Xor AK[1,1]_0 ST[1,1]_0 K[1,1]_0`
	`*`
	`_Sbox SB[1,1]_0 AK[1,1]_0`
		

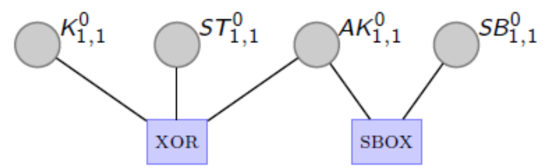
`mov H3, ST11`
`eor H3, ST21`
`mov ZL, H3`
`lpm H3, Z`
	`*`
	`_Xor MC[3,1]_0 SB[1,1]_0 SB[2,1]_0`
	`*`
	`_Xtime XT[1,1]_0 MC[3,1]_0`
		

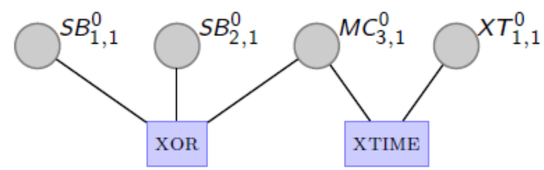
`mov ZL, ST24`
`lpm H3, Z`
`eor ST11, H3`
`eor ST11, H1`
	`*`
	`_Sbox SK[1,1]_1 K[2,4]_0`
	`_Xor XK[1,1]_1 SK[1,1]_1 K[1,1]_0`
	`_XorCst K[1,1]_1 XK[1,1]_1 0x1`
		

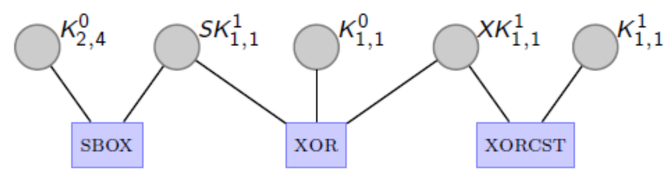
Experimental setting

- Good news: any knowledge can be exploited
- e.g. (open source) AES furious assembly code

ASM code	Graph description	Factor graph

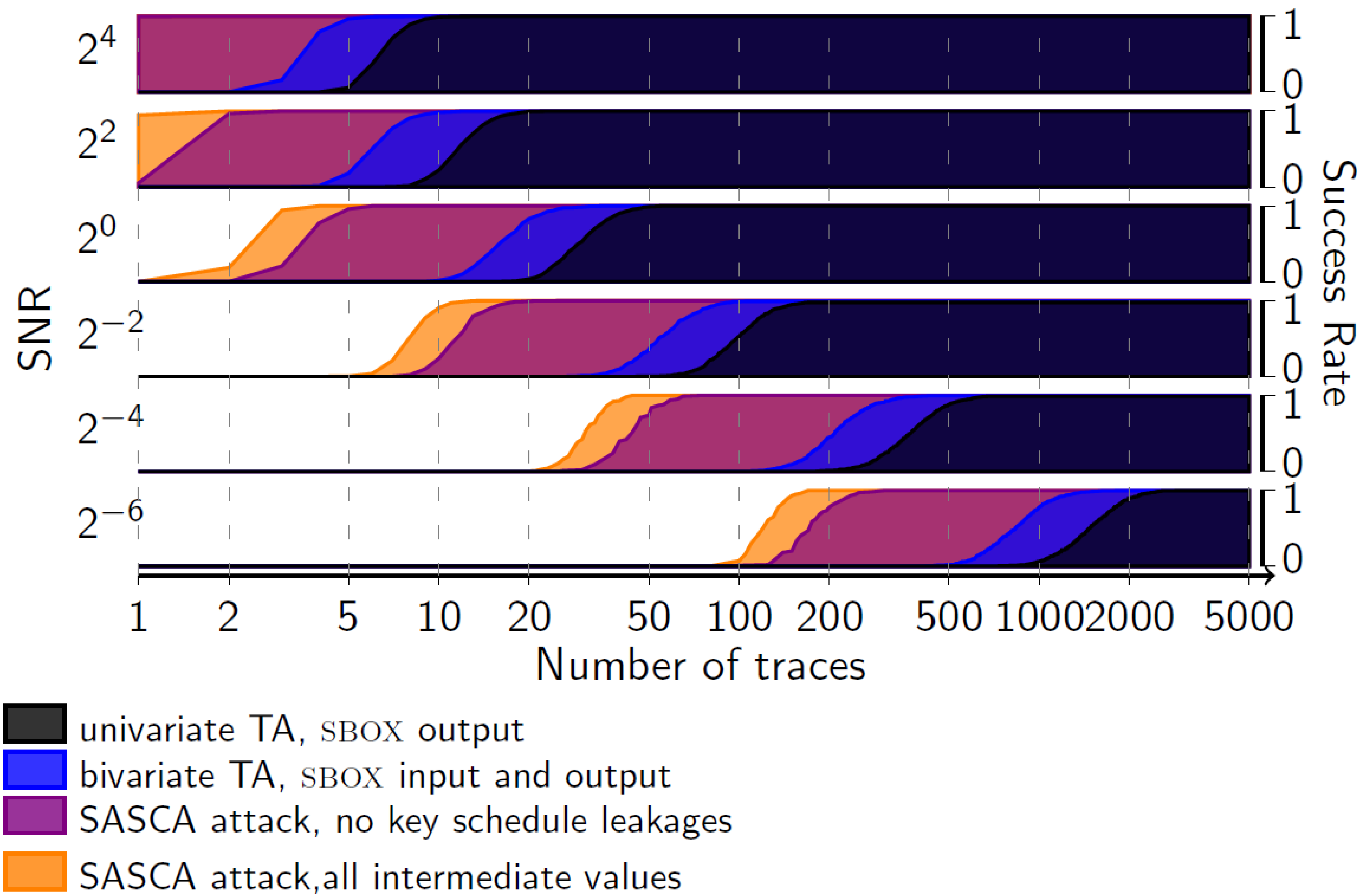
`ld H1, Y+`
`eor ST11, H1`
`mov ZL, ST11`
`lpm ST11, Z`
	`*`
	`_Xor AK[1,1]_0 ST[1,1]_0 K[1,1]_0`
	`*`
	`_Sbox SB[1,1]_0 AK[1,1]_0`
		

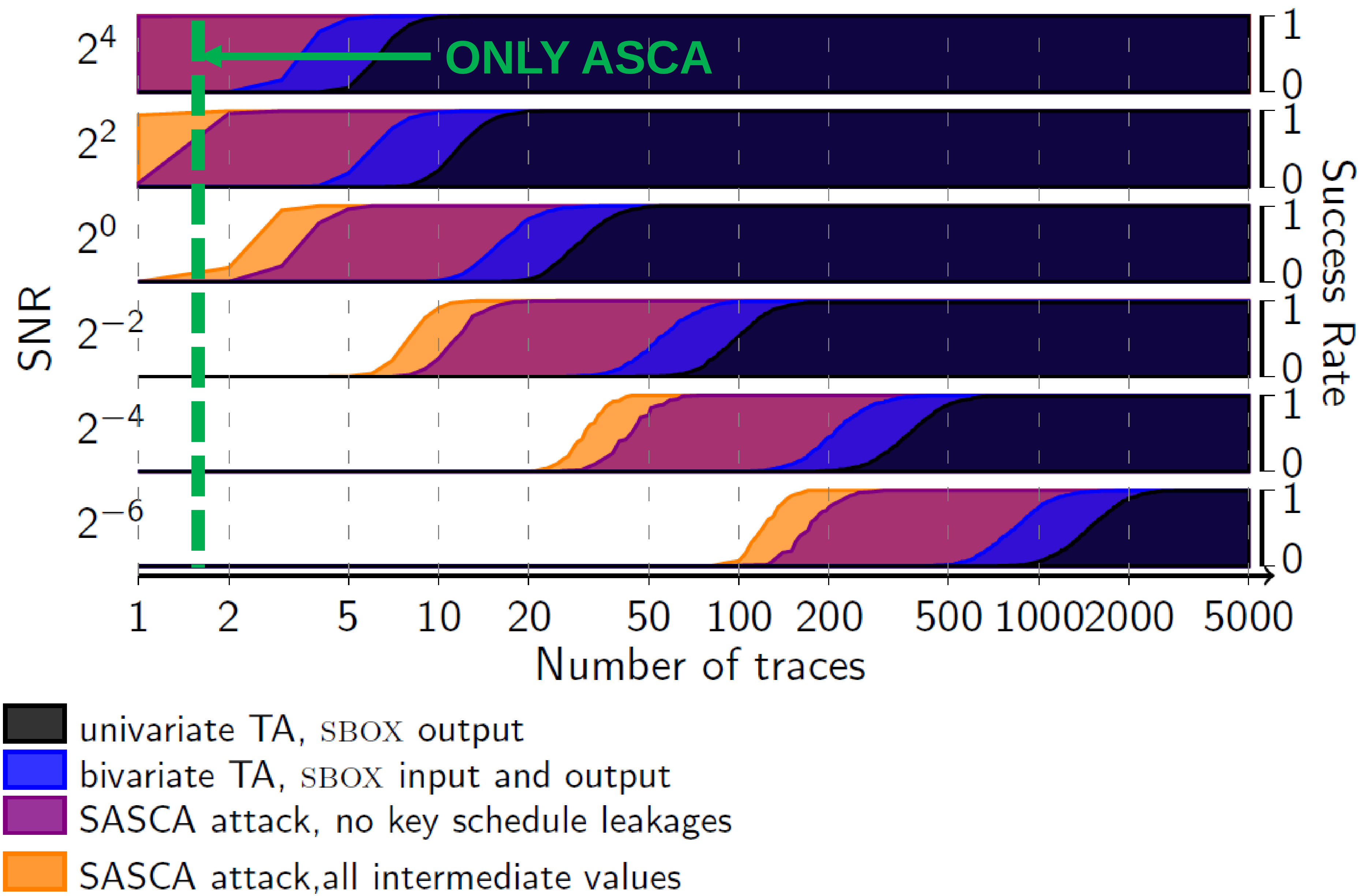
`mov H3, ST11`
`eor H3, ST21`
`mov ZL, H3`
`lpm H3, Z`
	`*`
	`_Xor MC[3,1]_0 SB[1,1]_0 SB[2,1]_0`
	`*`
	`_Xtime XT[1,1]_0 MC[3,1]_0`
		

`mov ZL, ST24`
`lpm H3, Z`
`eor ST11, H3`
`eor ST11, H1`
	`*`
	`_Sbox SK[1,1]_1 K[2,4]_0`
	`_Xor XK[1,1]_1 SK[1,1]_1 K[1,1]_0`
	`_XorCst K[1,1]_1 XK[1,1]_1 0x1`
		

- Simulated HW leakages with variable noise

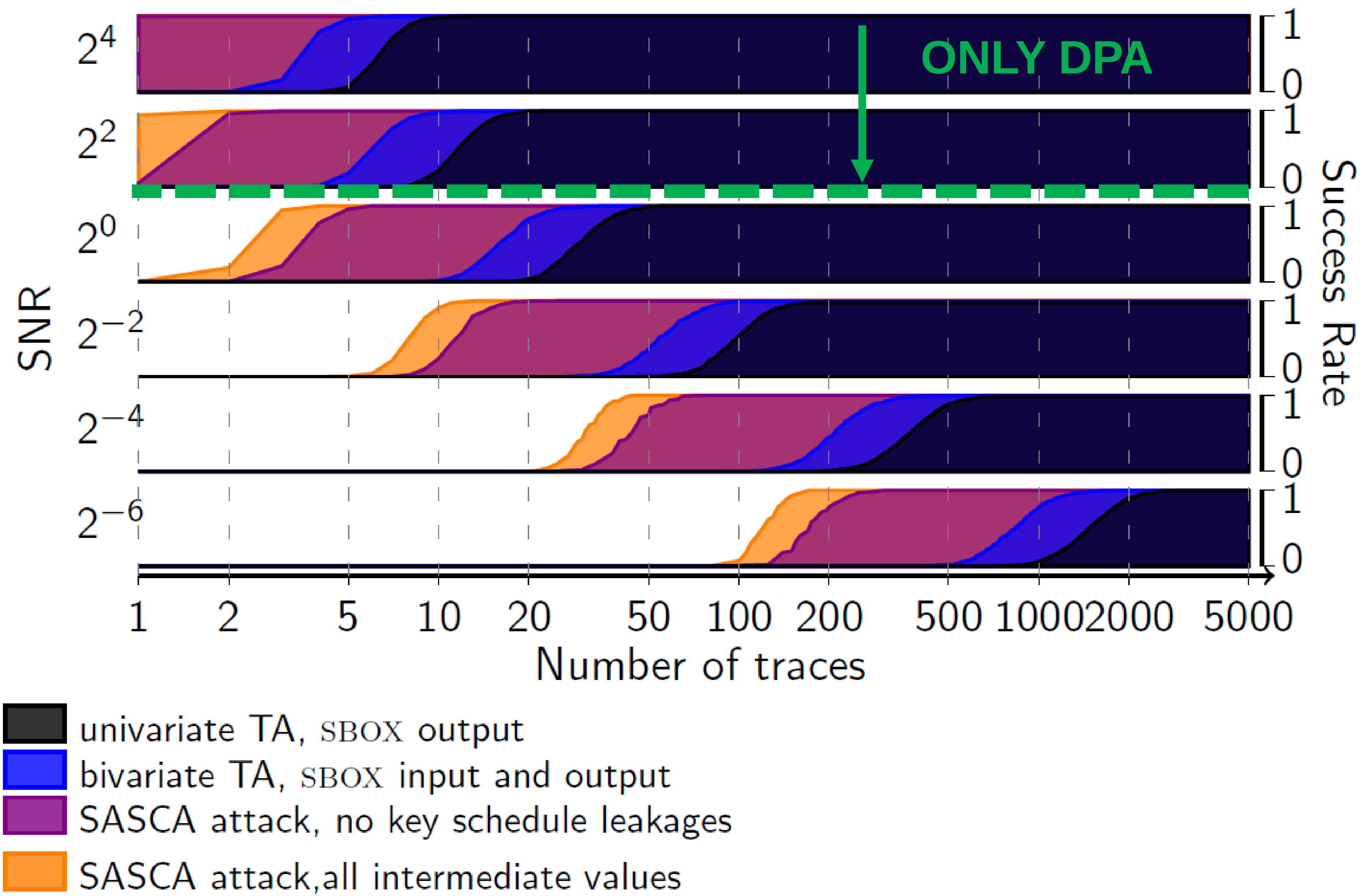
Experimental results



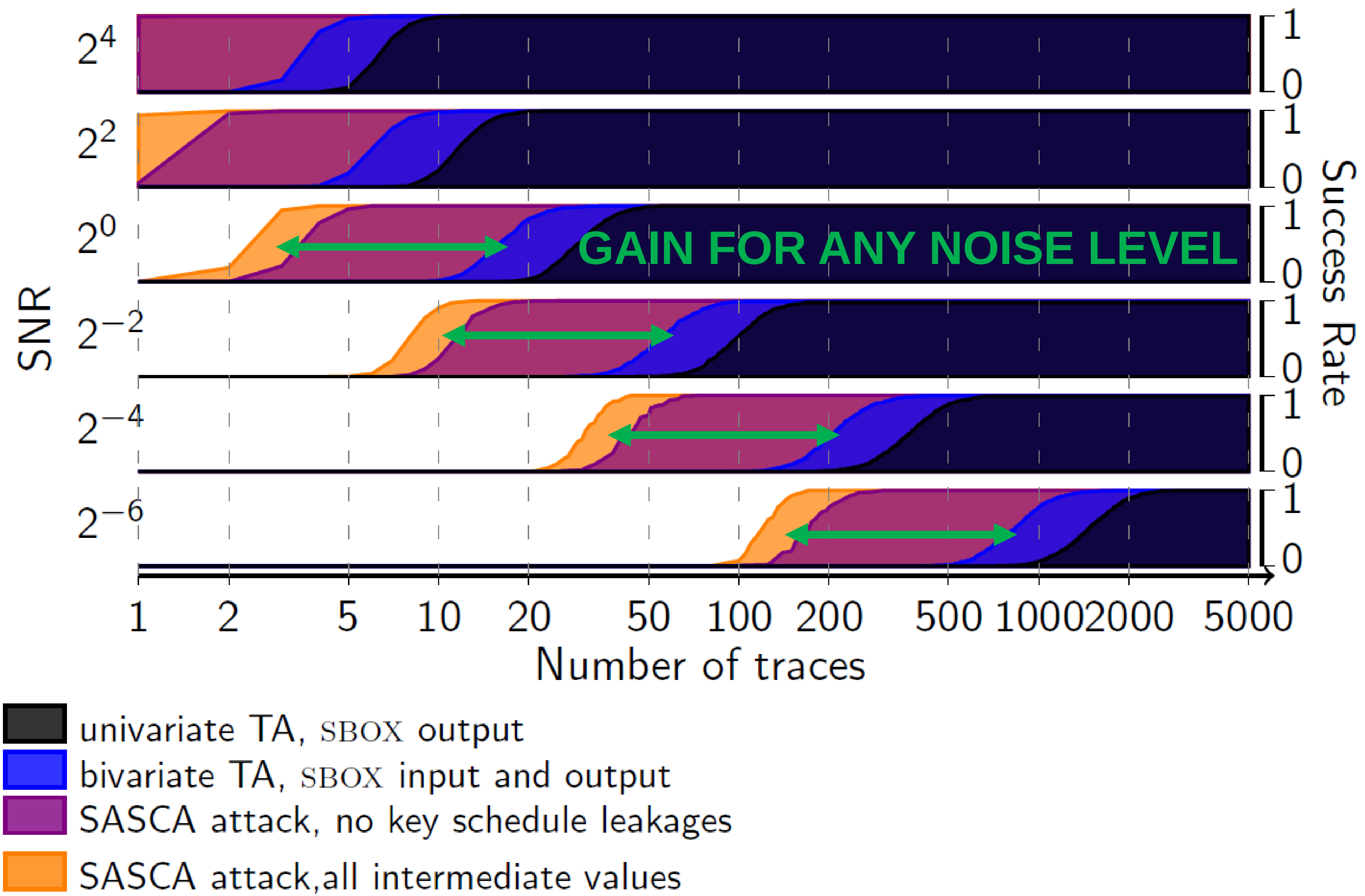


ONLY ASCA
SNR
2^4
2^2
2^0
2^{-2}
2^{-4}
2^{-6}
Success Rate
1
0
1 2 5 10 20 50 100 200 500 1000 2000 5000
Number of traces
univariate TA, SBOX output
bivariate TA, SBOX input and output
SASCA attack, no key schedule leakages
SASCA attack,all intermediate values

Experimental results



Experimental results

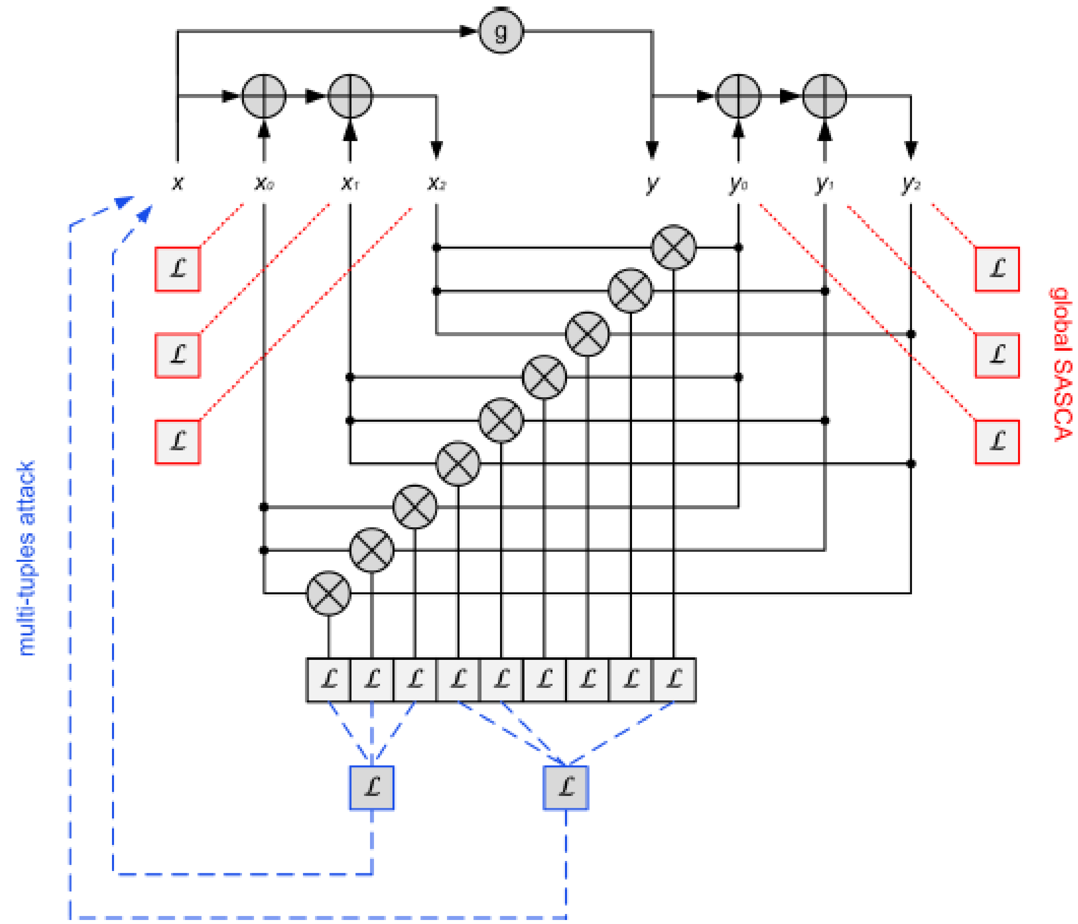


More results

- Also works in practice (Asiacrypt 2015)

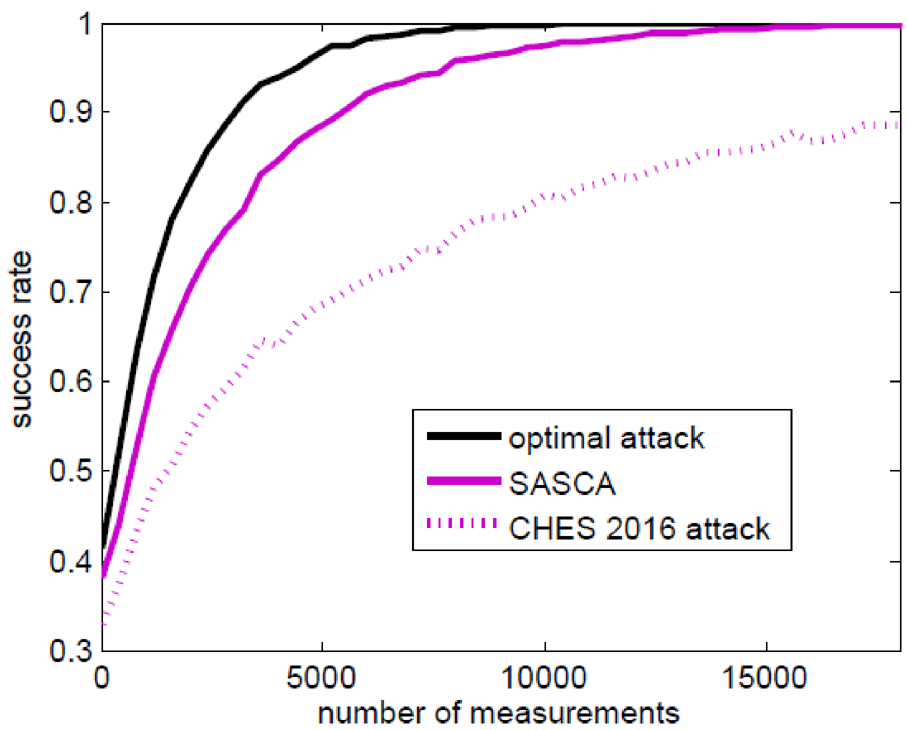
More results

- Also works in practice (Asiacrypt 2015)
- And against masked implementations



More results

- Also works in practice (Asiacrypt 2015)
- And against masked implementations
  - Improvement of the CHES 2016 horizontal attacks



Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

SCA possible outcomes

- Enough measurements ⇒ direct key recovery

SCA possible outcomes

- Enough measurements ⇒ direct key recovery
- Not enough measurements but enough computational power ⇒ key enumeration

SCA possible outcomes

- Enough measurements ⇒ direct key recovery
- Not enough measurements but enough computational power ⇒ key enumeration

- But what to do if it is not enough?
  - Key rank estimation (requires key knowledge)
    ⇒ Only possible for evaluators

SCA possible outcomes

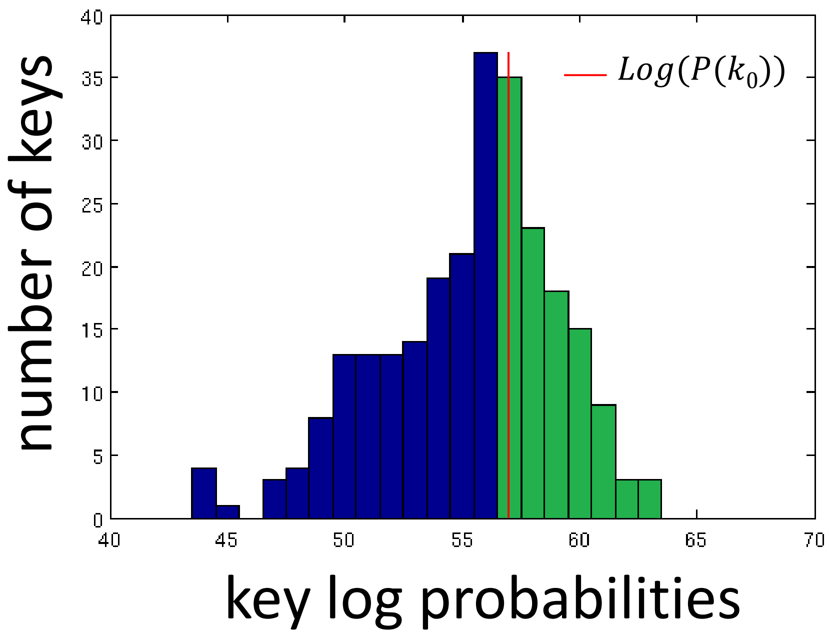
- Enough measurements ⇒ direct key recovery
- Not enough measurements but enough computational power ⇒ key enumeration
- But what to do if it is not enough?
  - Key rank estimation (requires key knowledge)
    ⇒ Only possible for evaluators
- Note: only optimal with probabilities (to combine the information of ≠ S-boxes)

FSE 2015 rank estimation (I)

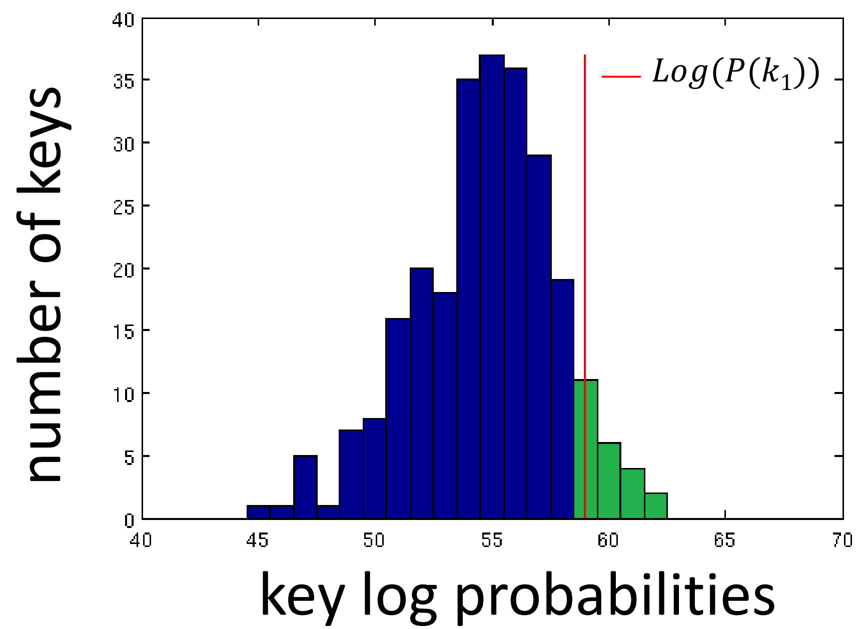
- Thanks to T-systems people (Glowacz, Schueth)

- Thanks to T-systems people (Glowacz, Schueth)
- Representation with histograms

S-box 0



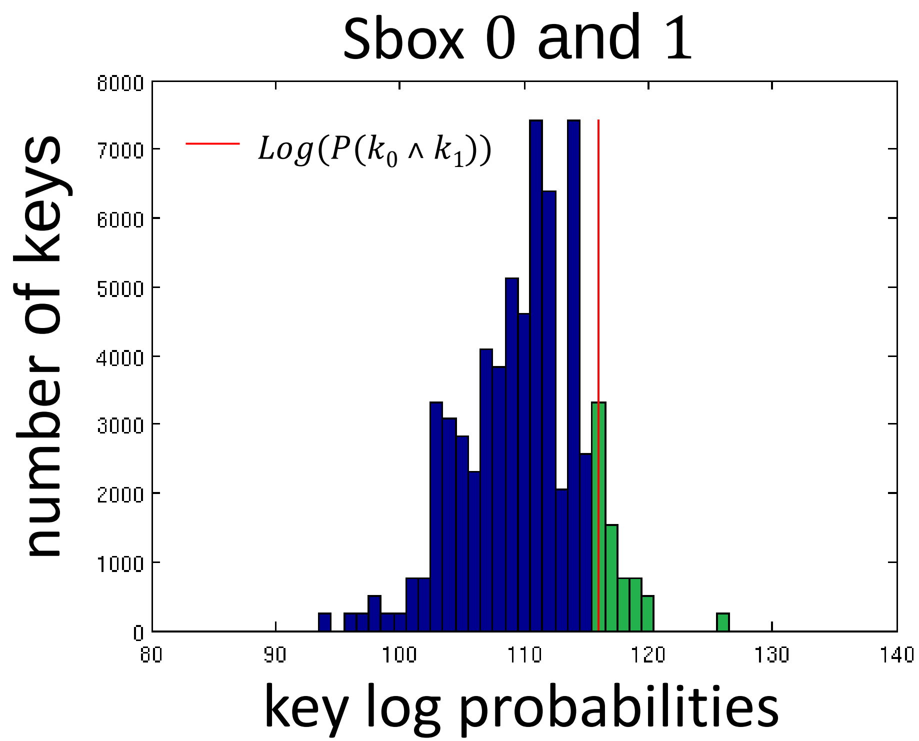
S-box 1



- The rank is the number of key in the green zone

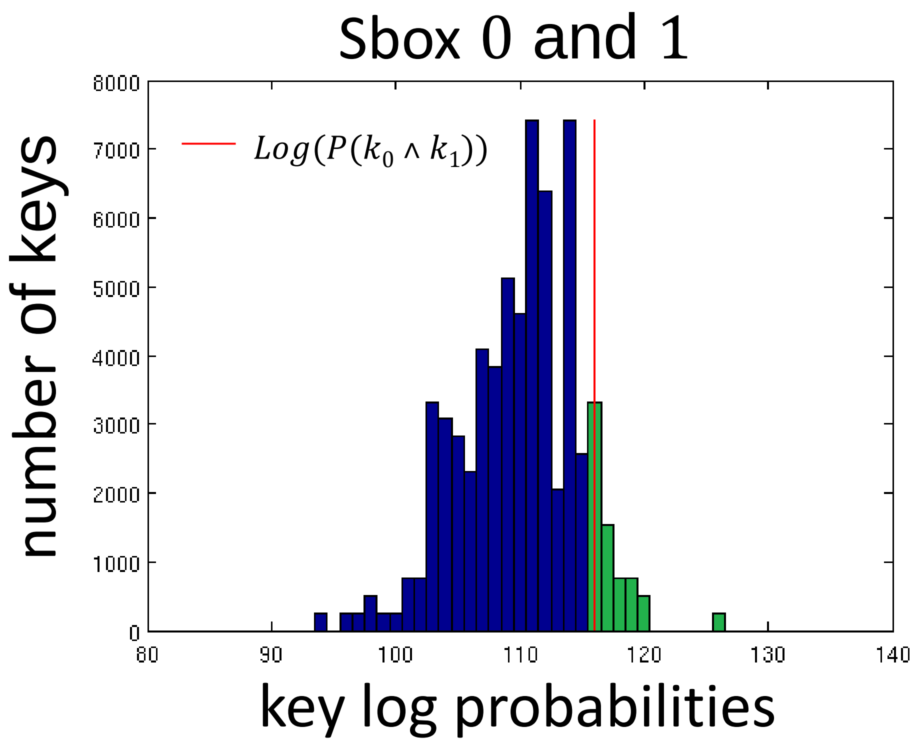
FSE 2015 rank estimation (II)

- Combination with convolution



FSE 2015 rank estimation (II)

- Combination with convolution

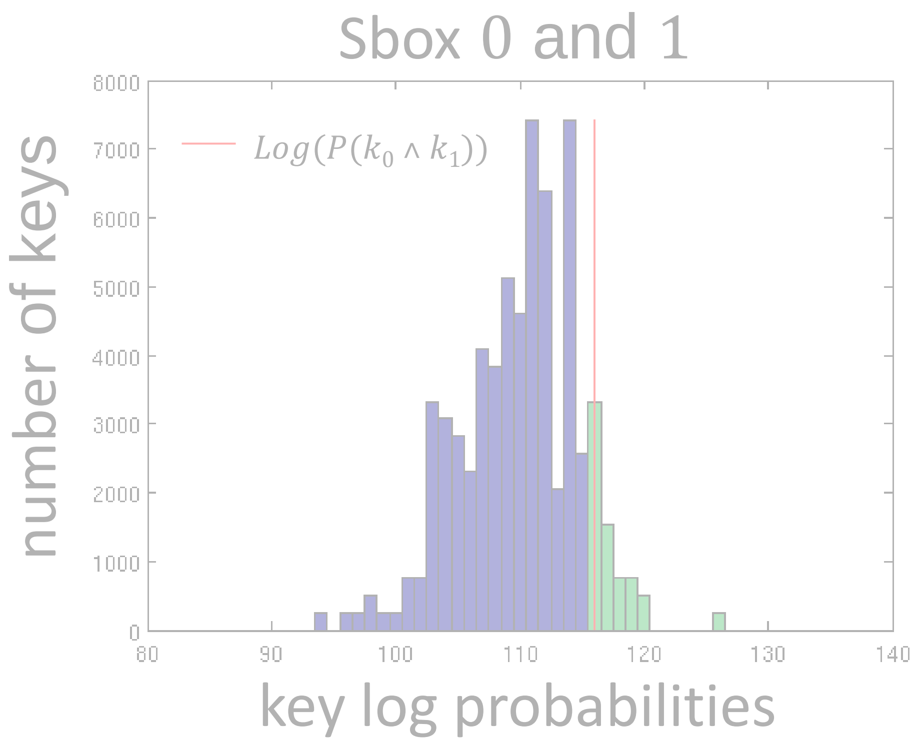


… and keep track of the error that depends on:

- The number of bins
- The number of conv.

FSE 2015 rank estimation (II)

- Combination with convolution



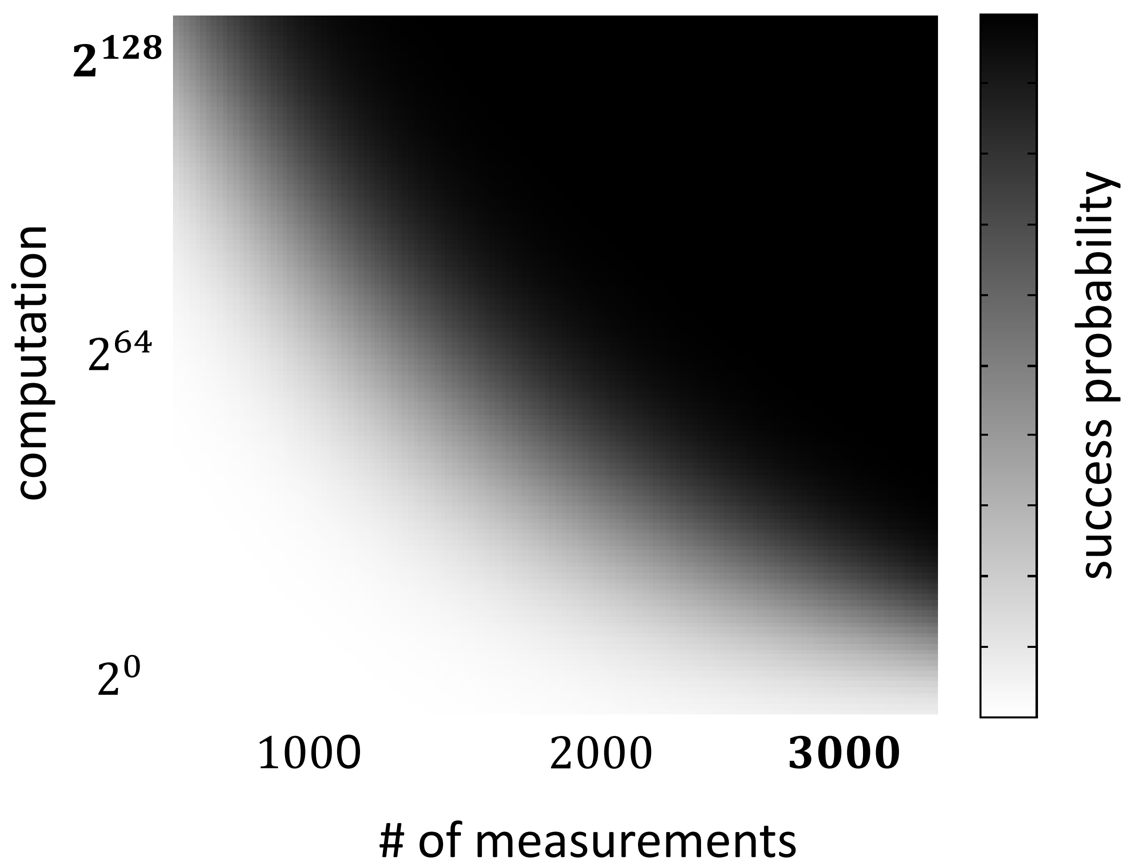
... and keep track of the error that depends on:

- The number of bins
- The number of conv.

- Just iterating this gives the key rank accurately
    - e.g., < 1 bits in < 1 sec. for a 128-bit key

Rank estimation result

- Security graph (IMO the sound outcome of an evaluation)

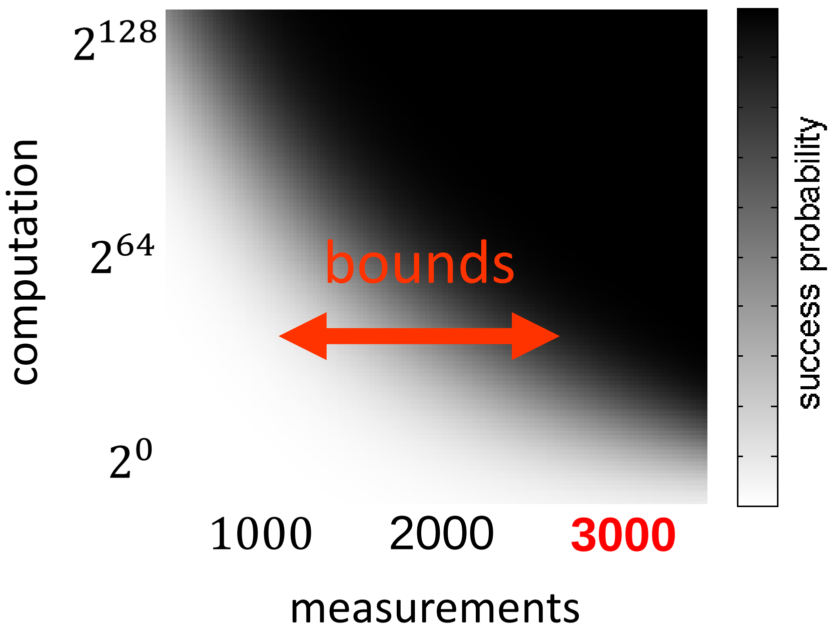


Outline

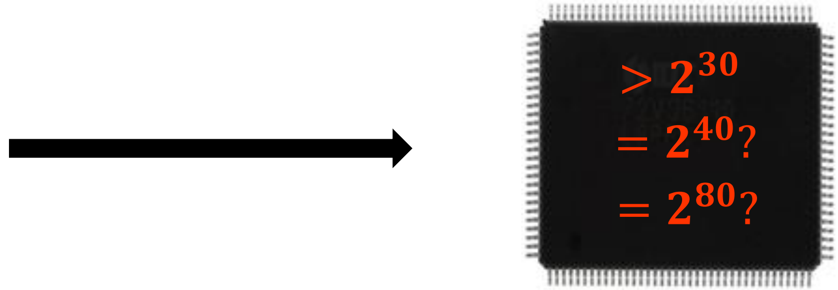
- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

standard practice

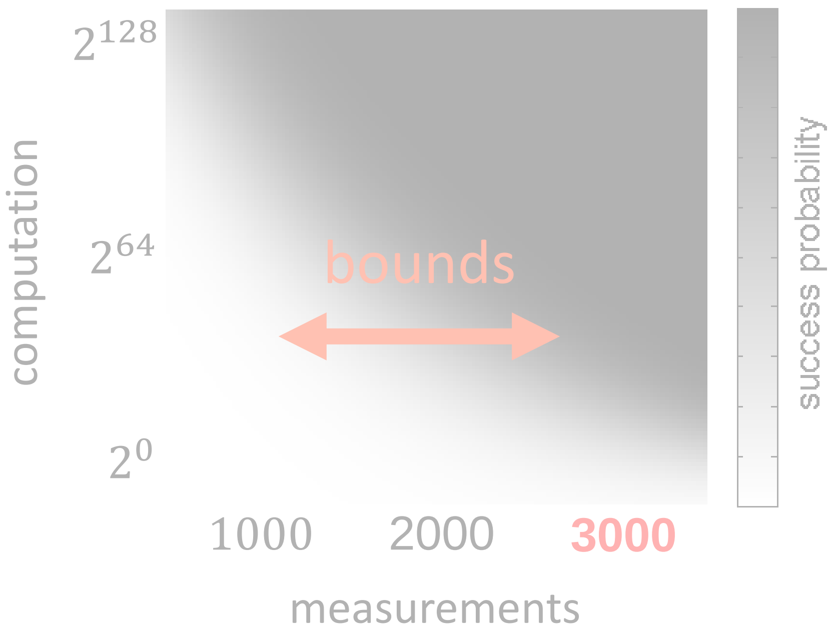
attack-based evaluations



standard practice



attack-based evaluations

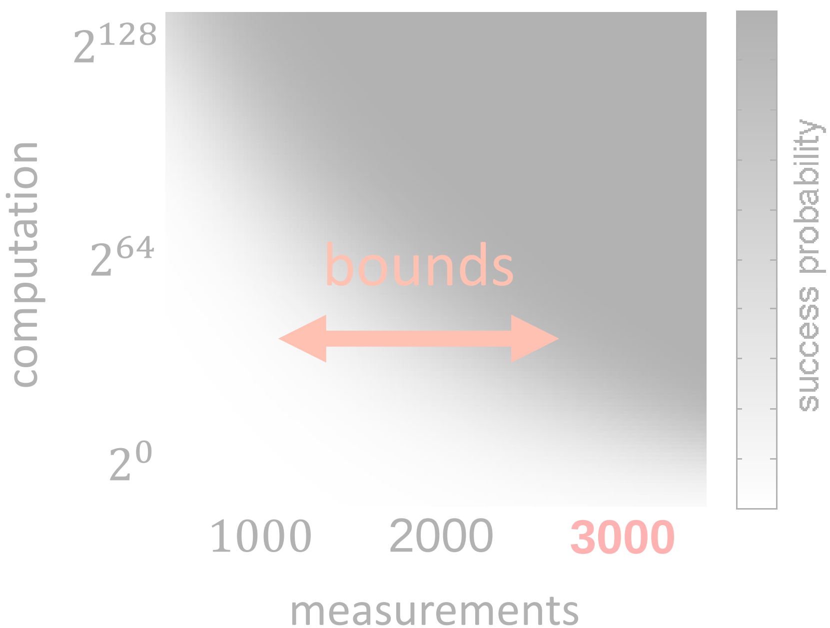


standard practice

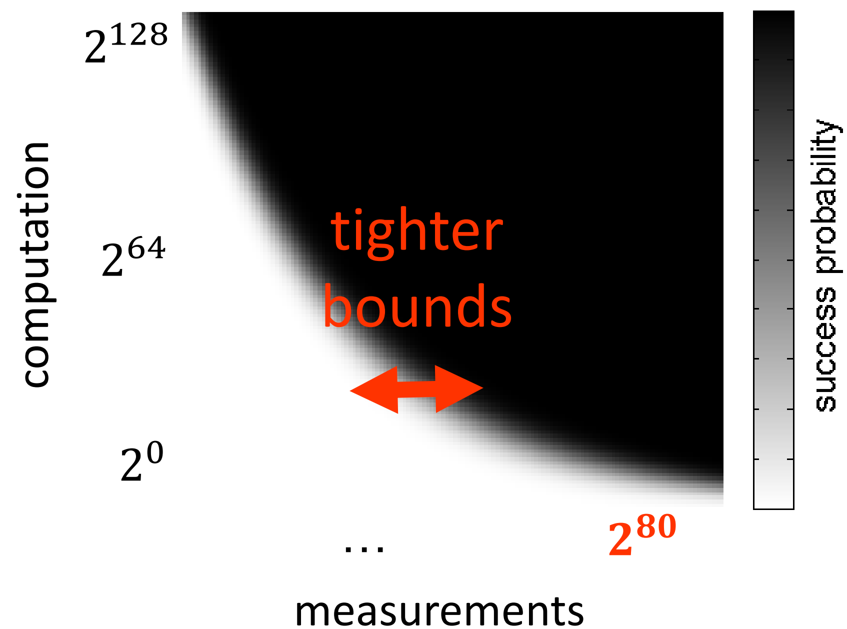
transparency

helps evaluations

attack-based evaluations



proof-based evaluations



Outline

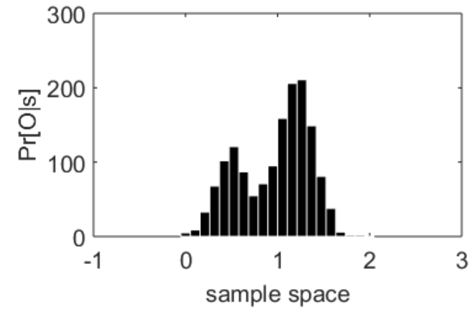
- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - IT metrics & (tight) proofs

Mutual information

- Discrete observations:

$$\mathrm{MI}(S;O) = \mathrm{H}[S] + \sum_{s \in S} \Pr[s] \cdot \sum_{o \in O} \Pr[o|s] \cdot \log_2(\Pr[s|o])$$

- With: $\Pr[s|o] = \dfrac{\Pr[o|s]}{\sum_{s^*} \Pr[o|s^*]}$

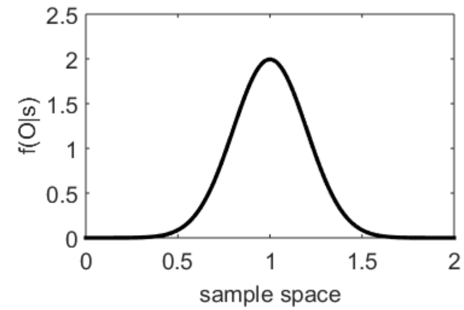


Mutual information

- Continuous observations:

$$\mathrm{MI}(S;O) = \mathrm{H}[S] + \sum_{s \in S} \Pr[s] \cdot \int_{O} \mathrm{f}(o|s) \cdot \log_2(\Pr[s|o]) \, do$$

- With: $\Pr[s|o] = \dfrac{f(o|s)}{\sum_{s^*} \mathrm{f}(o|s^*)}$



Mutual information

- Continuous observations:

$$\mathrm{MI}(S;O) = \mathrm{H}[S] + \sum_{s \in S} \Pr[s] \cdot \int_{O} \mathrm{f}(o|s) \cdot \log_2(\Pr[s|o]) \, do$$

- With: $\Pr[s|o] = \dfrac{f(o|s)}{\sum_{s^*} \mathrm{f}(o|s^*)}$

- Intuition: average amount of information on S that is gained by observing a sample o

Main theorem (informal)

- Higher $\mathrm{MI}(S;O)$ asymptotically implies higher Bayesian classification success rate:

$$\mathrm{SR}(n) = \Pr_{O}\left[\ \operatorname*{argmax}_{s^*} \Pr[s^*|o_1]\cdot\Pr[s^*|o_2]\cdots\Pr[s^*|o_n] = s\ \right]$$

- With n observations used for classification

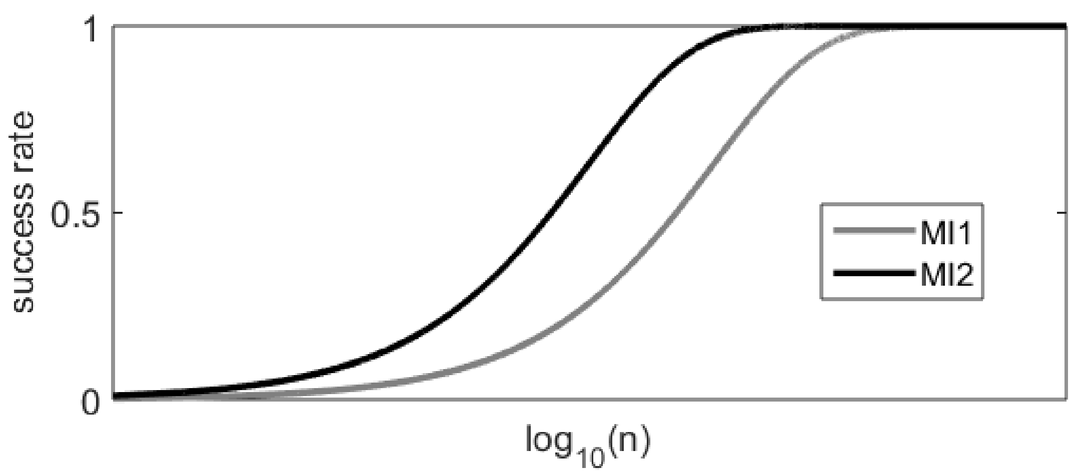
Main theorem (informal)

- Higher $\mathrm{MI}(S;O)$ asymptotically implies higher Bayesian classification success rate:

$$\mathrm{SR}(n) = \Pr_{O}\left[\ \operatorname*{argmax}_{s^*} \Pr[s^*|o_1]\cdot\Pr[s^*|o_2]\cdots\Pr[s^*|o_n] = s\ \right]$$

- With n observations used for classification

Example, with MI2 > MI1



- In practice, the true distributions $\mathrm{Pr}_{\mathrm{real}}[o|s]$ and $\mathrm{f}_{\mathrm{real}}(o|s)$ are generally unknown
  - So they can be sampled (i.e., measured)
  - But they cannot be computed exactly

- In practice, the true distributions $\Pr_{\mathrm{real}}[o|s]$ and $\mathrm{f}_{\mathrm{real}}(o|s)$ are generally unknown
  - So they can be sampled (i.e., measured)
  - But they cannot be computed exactly

- Hence, for the classification we use statistical models $\Pr_{\mathrm{model}}[o|s]$ and $\mathrm{f}_{\mathrm{model}}(o|s)$
  - But these models can suffer from both estimation errors and assumption errors

Concrete challenge

- In practice, the true distributions $\Pr_{\text{real}}[o|s]$ and $\mathrm{f}_{\text{real}}(o|s)$ are generally unknown
  - So they can be sampled (i.e., measured)
  - But they cannot be computed exactly
- Hence, for the classification we use statistical models $\Pr_{\text{model}}[o|s]$ and $\mathrm{f}_{\text{model}}(o|s)$
  - But these models can suffer from both estimation errors and assumption errors

$\Rightarrow$ *How to be sure that the model is good?*

- Exactly reflects this practical challenge
- For example in the continuous case:

$$\mathrm{PI}(S;O) = \mathrm{H}[S] + \sum_{s\in S} \mathrm{Pr}[s] \cdot \int_{O} \mathrm{f}_{\mathrm{real}}(o|s) \cdot \log_2\left(\mathrm{Pr}_{\mathrm{model}}[s|o]\right) do$$

The Perceived Information

- Exactly reflects this practical challenge
- For example in the continuous case:

$$\mathrm{PI}(S;O) = \mathrm{H}[S] + \sum_{s\in S} \mathrm{Pr}[s] \cdot \int_{o} \mathrm{f}_{\mathrm{real}}(o|s) \cdot \log_2\big(\mathrm{Pr}_{\mathrm{model}}[s|o]\big)\, do$$

- Intuition: average amount of information on S that is gained by observing a sample o, biased by the model estimation/assumption errors

- Exactly reflects this practical challenge
- For example in the continuous case:

$$\mathrm{PI}(S;O) = \mathrm{H}[S] + \sum_{s \in S} \Pr[s] \cdot \int_{o} \mathrm{f}_{\mathrm{real}}(o|s) \cdot \log_2\left(\Pr_{\mathrm{model}}[s|o]\right) do$$

- Intuition: average amount of information on S that is gained by observing a sample o, biased by the model estimation/assumption errors
- $\mathrm{PI}(S;O)$ is a statistical distance between the real and modeled distributions (i.e., a measure of how well a model "explains" real observations)

How to estimate the PI?

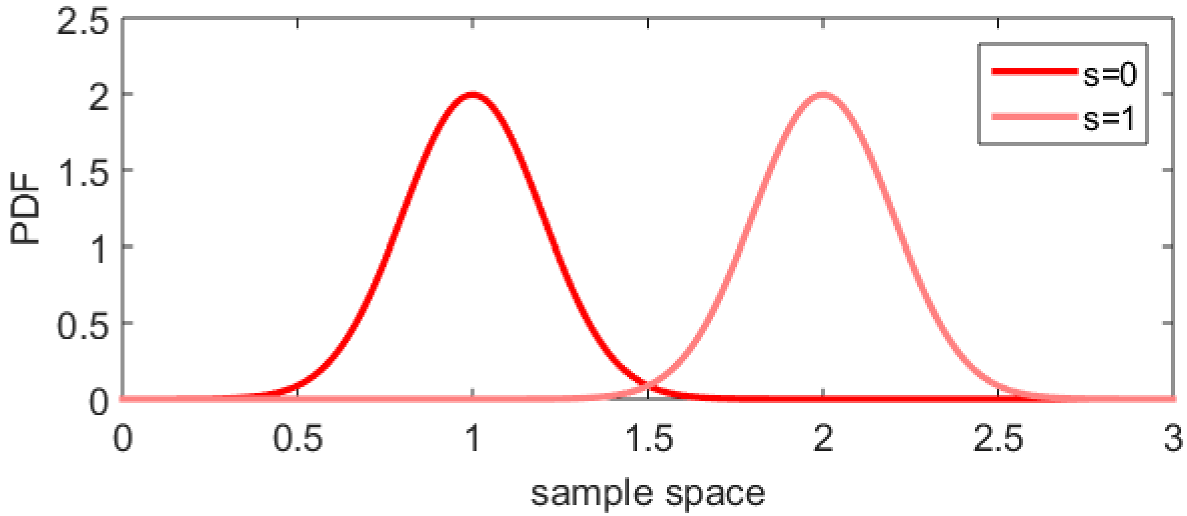
- Say you have a set of $2m$ observations per s

How to estimate the PI?

- Say you have a set of $2m$ observations per s

1. Use m observations to build a model for each s:

$$\hat{\mathrm{f}}_{\mathrm{model}}(o|s) \overset{m}{\leftarrow} \mathrm{f}_{\mathrm{real}}(o|s)$$



How to estimate the PI?

- Say you have a set of $2m$ observations per s

1. Use m observations to build a model for each s:

$$\hat{\mathrm{f}}_{\mathrm{model}}(o|s) \overset{m}{\leftarrow} \mathrm{f}_{\mathrm{real}}(o|s)$$

2. Use the other m observations to test the model:

$$\widehat{\mathrm{PI}}(S;O) = \mathrm{H}[S] + \sum_{s\in S} \Pr[s] \cdot \sum_{o' \overset{m}{\leftarrow} \mathrm{f}_{\mathrm{real}}(o|s)} \frac{1}{m} \cdot \log_2(\widehat{\Pr}_{\mathrm{model}}[s|o'])$$

- Say you have a set of $2m$ observations per s

1. Use m observations to build a model for each s:

$$\hat{\mathrm{f}}_{\mathrm{model}}(o|s) \overset{m}{\leftarrow} \mathrm{f}_{\mathrm{real}}(o|s)$$

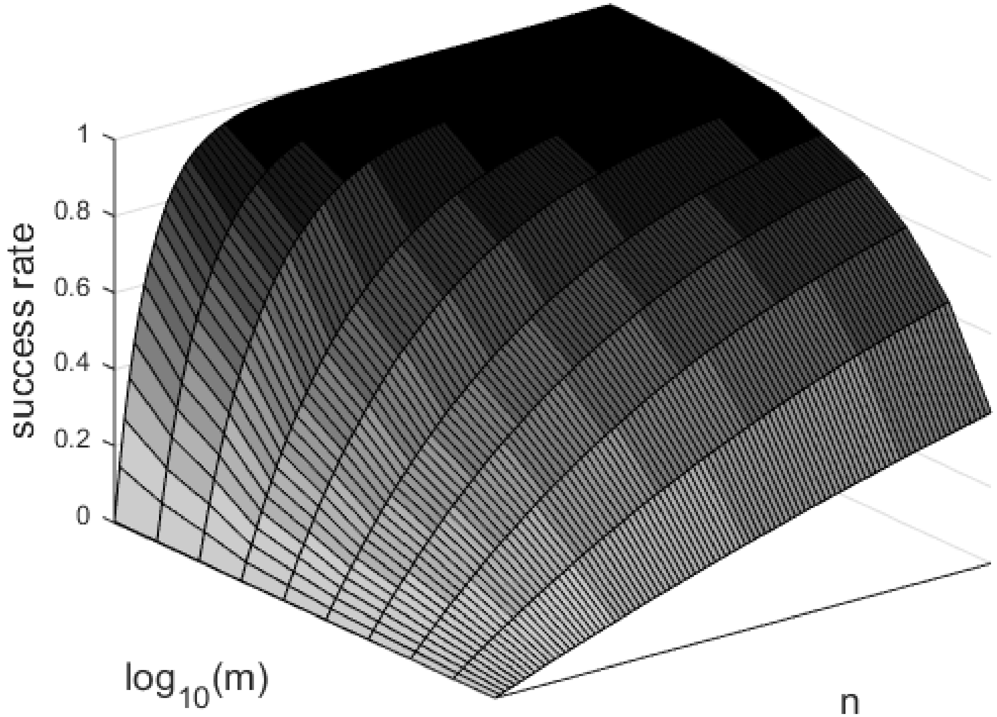
2. Use the other m observations to test the model:

$$\widehat{\mathrm{PI}}(S;O) = \mathrm{H}[S] + \sum_{s\in S} \mathrm{Pr}[s] \cdot \sum_{o' \overset{m}{\leftarrow} \mathrm{f}_{\mathrm{real}}(o|s)} \frac{1}{m} \cdot \log_2(\widehat{\mathrm{Pr}}_{\mathrm{model}}[s|o'])$$

(*More efficient use of observations with cross-validation*)

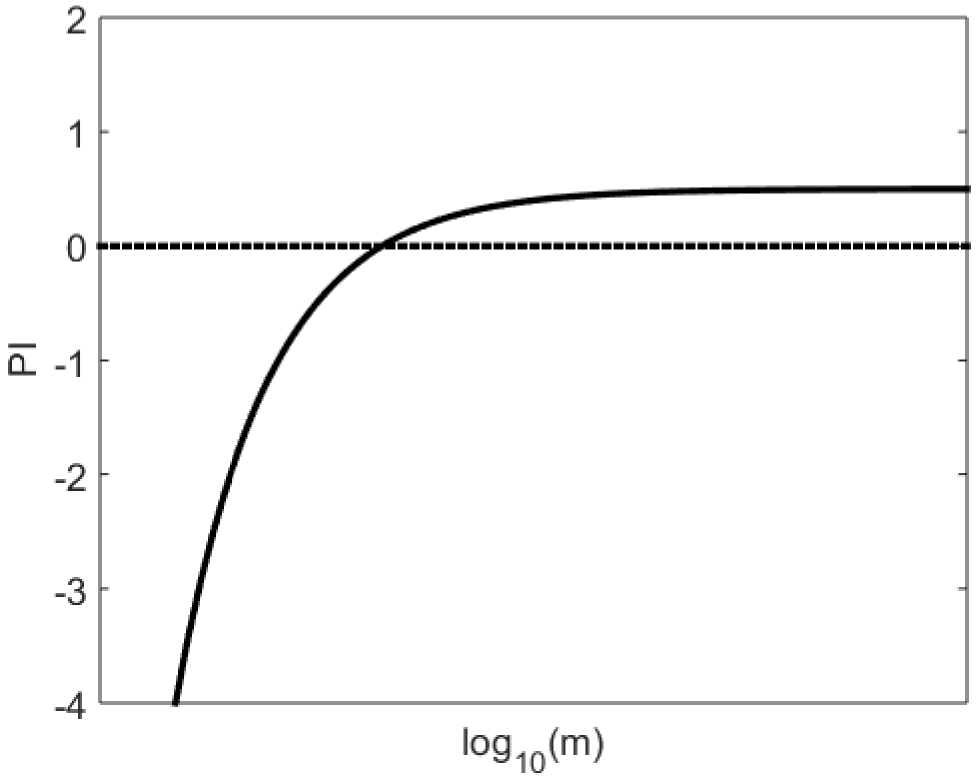
Efficiency gain (I)

- Directly estimating model *convergence* and *informativeness* based on the SR is expensive



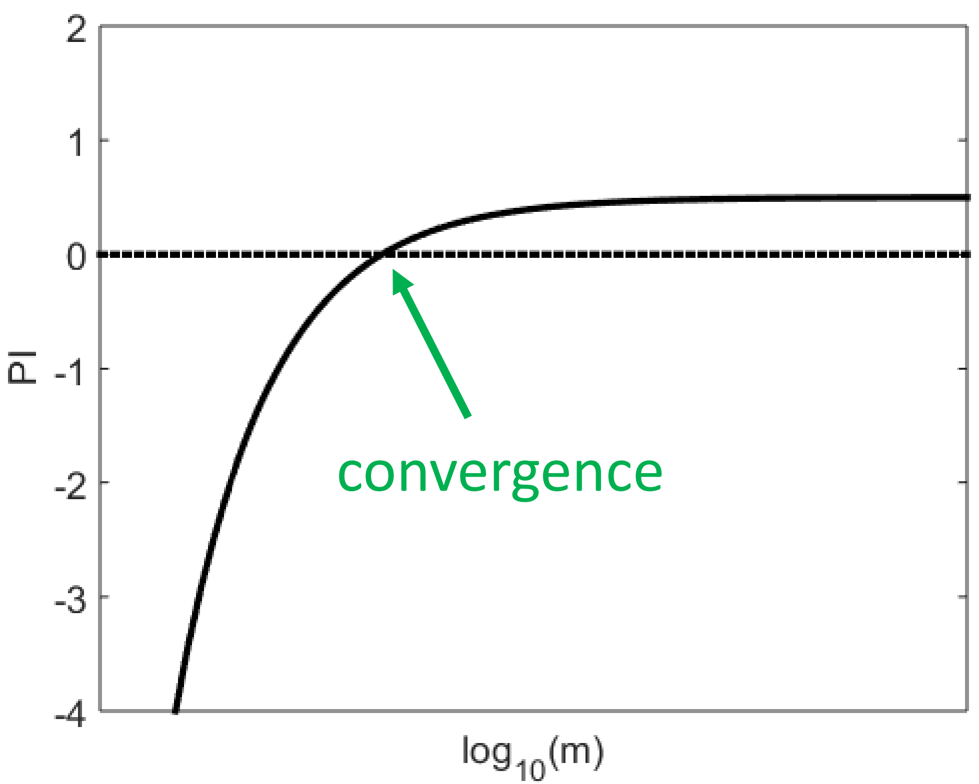
Efficiency gain (II)

- The PI curve allows "getting rid of" the *n* axis
  ⇒ it is faster to estimate than the SR surface

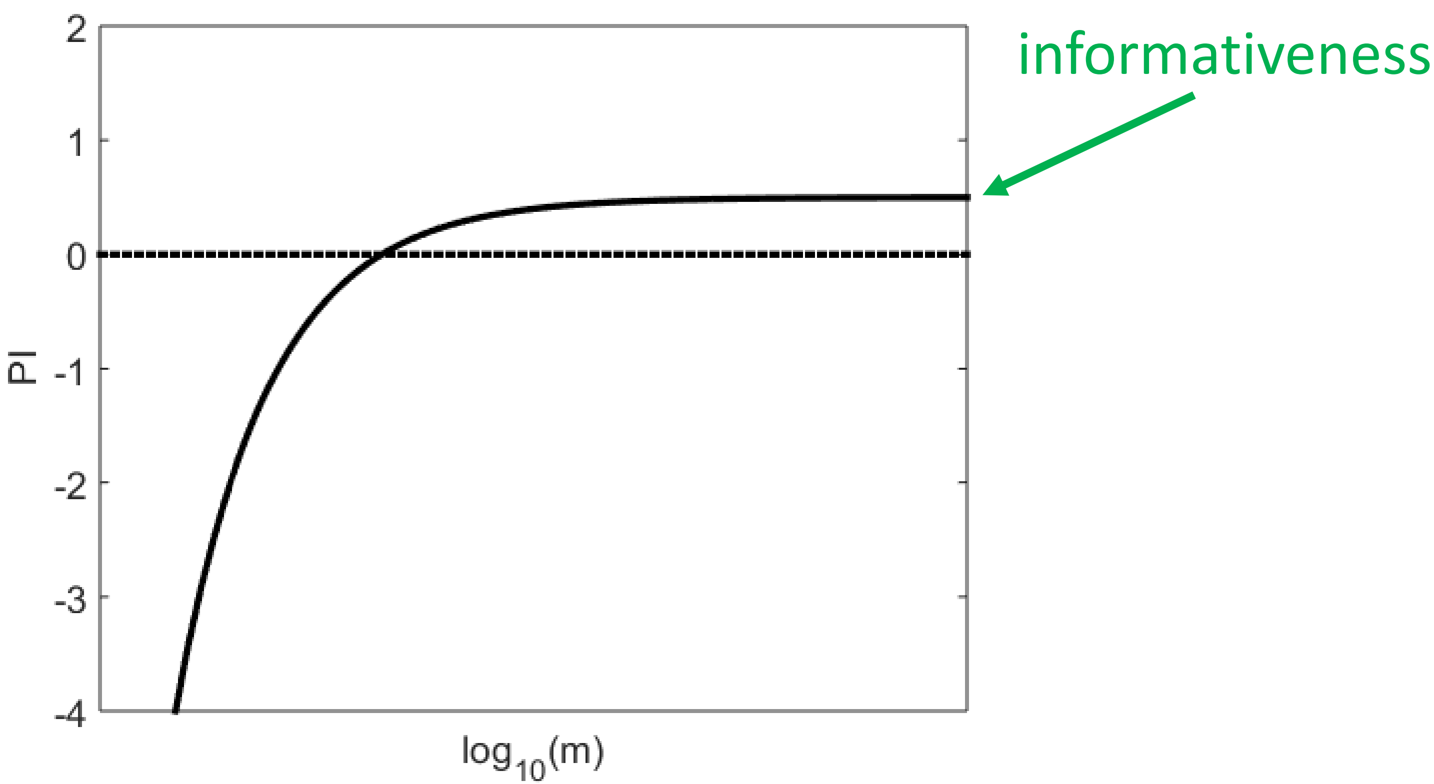


Efficiency gain (II)

- The PI curve allows "getting rid of" the *n* axis
  $\Rightarrow$ it is faster to estimate than the SR surface

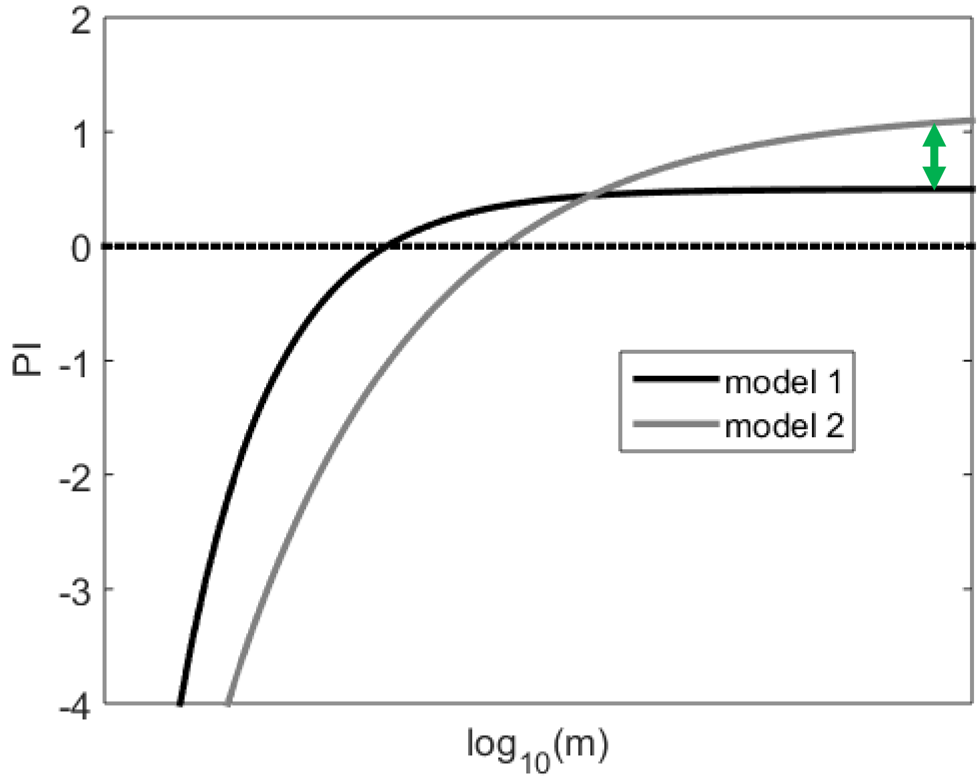


- The PI curve allows "getting rid of" the *n* axis
  ⇒ it is faster to estimate than the SR surface



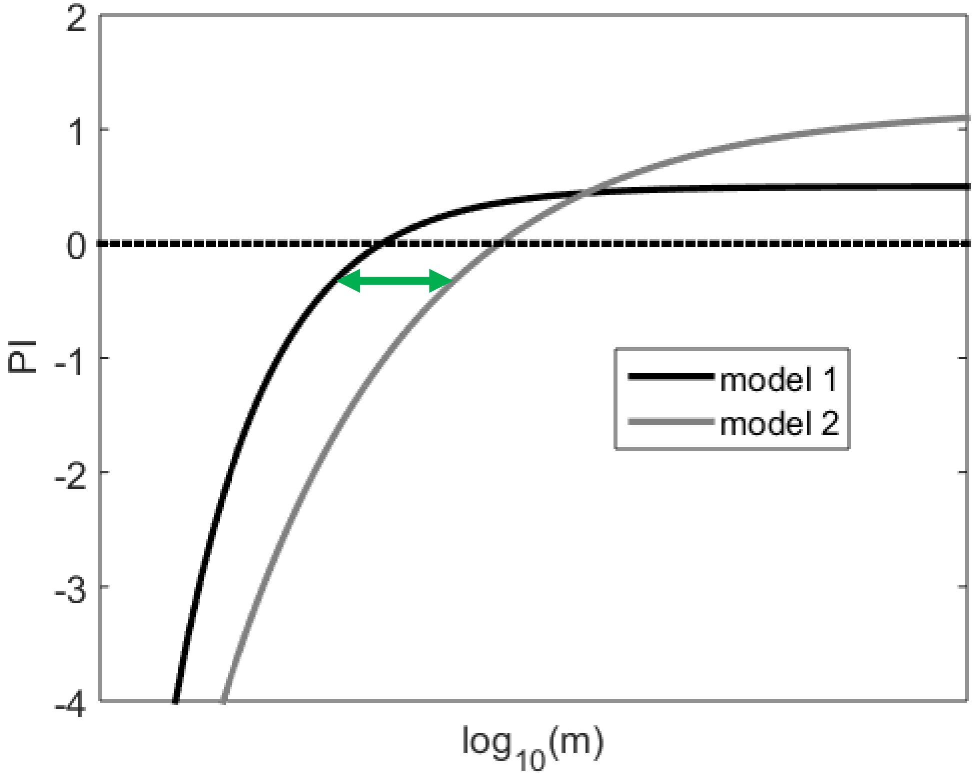
Interpretation (I)

- Allows sound & efficient comparison of models
  - e.g., model 2 more informative than model 1



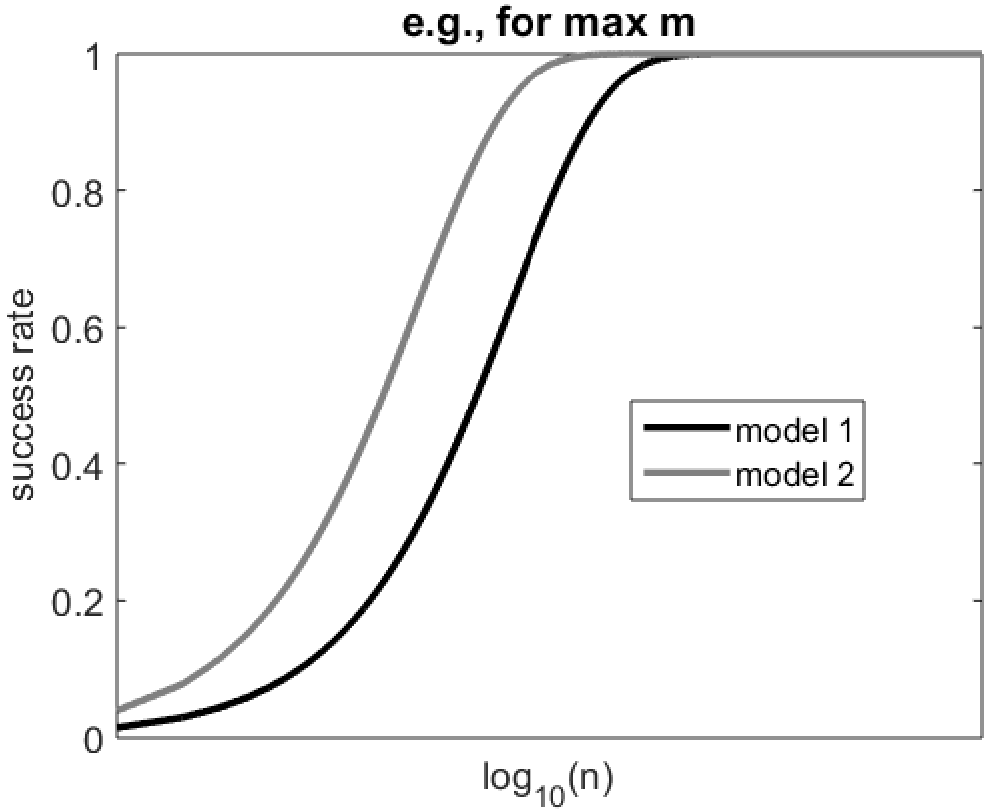
Interpretation (I)

- Allows sound & efficient comparison of models
  - and model 1 converges faster than model 2



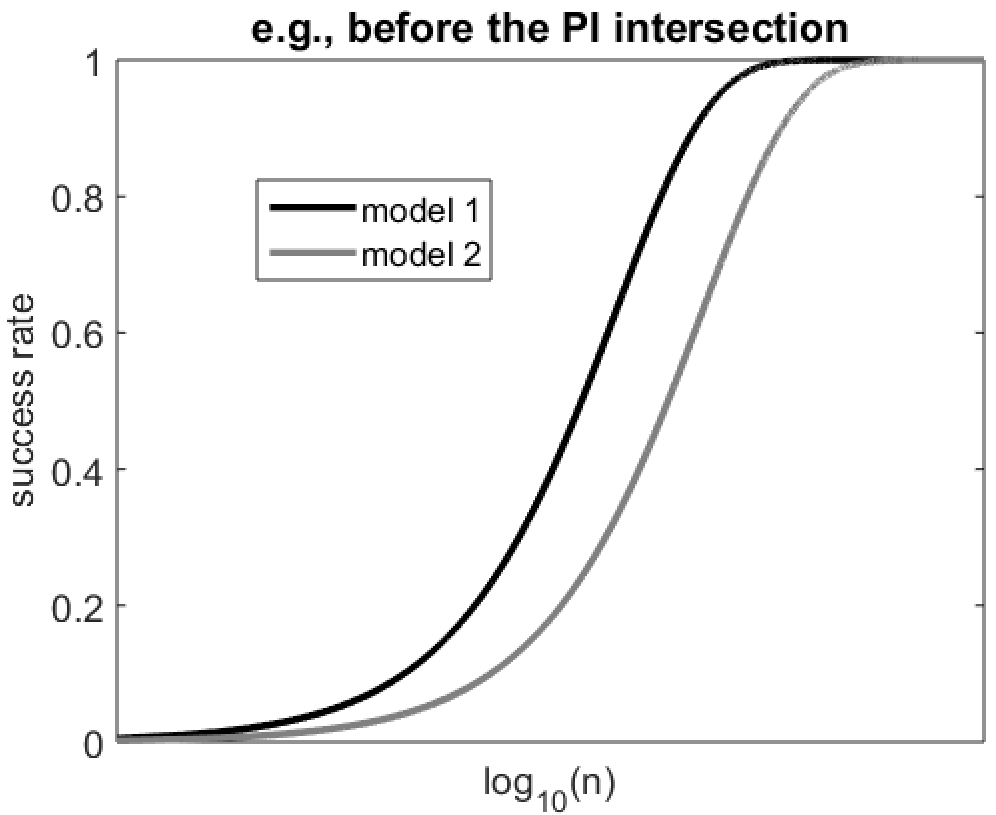
Interpretation (II)

- For a given *m*, one can always compute the success rate curves to gain concrete intuition



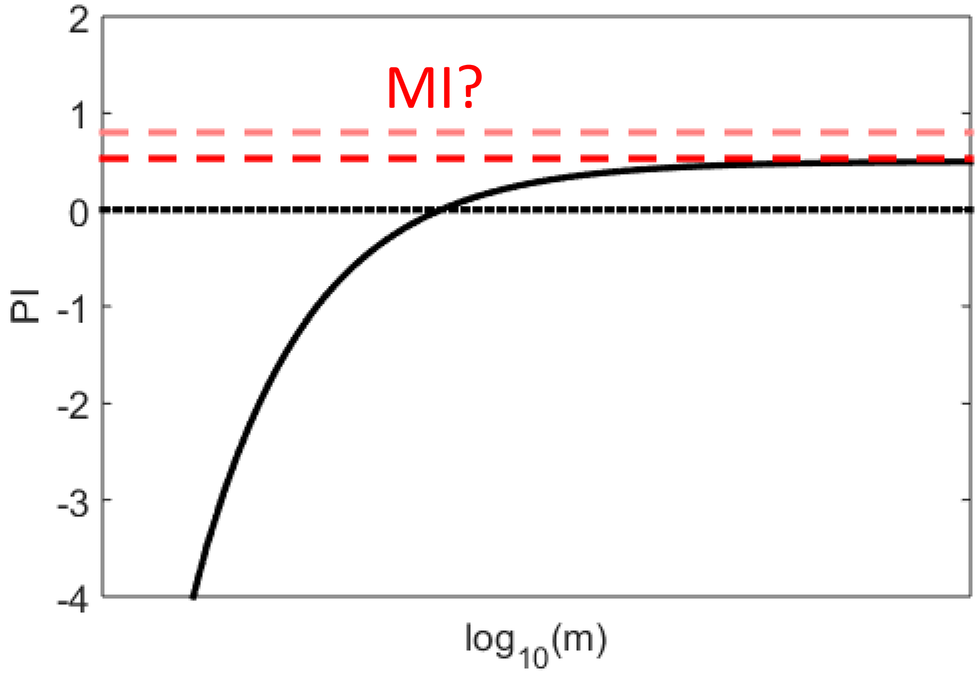
Interpretation (II)

- For a given *m*, one can always compute the success rate curves to gain concrete intuition



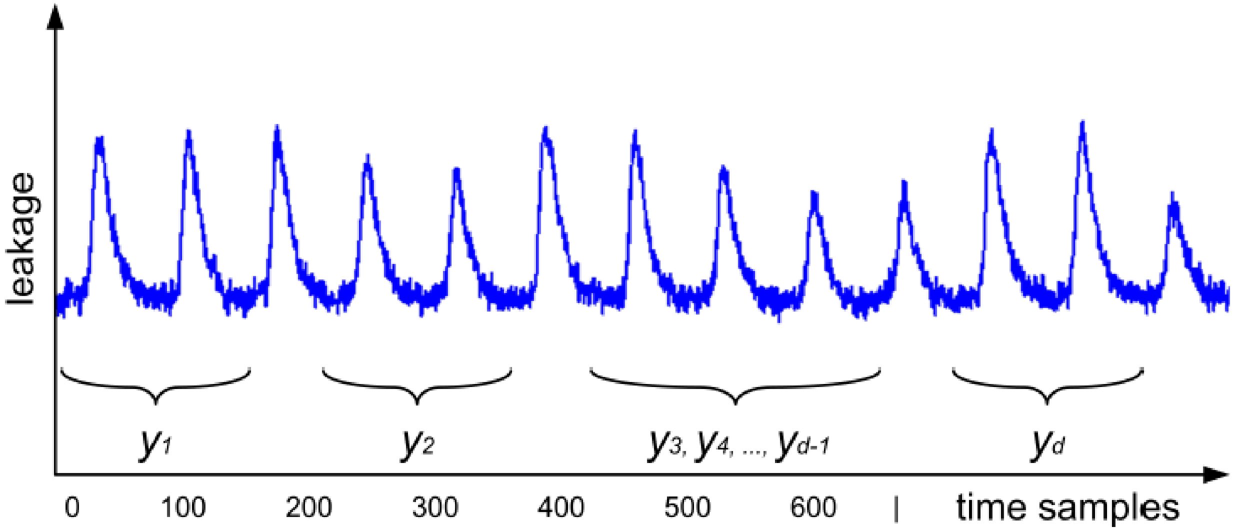
Optimality guarantees?

- Remaining question: how far are we from the MI?



- *Leakage certification* allows answering this!

- Let $z = S(x \oplus k) = S(y)$ be a leaking S-box
- Let $y = y_1 \oplus y_2 \oplus \cdots \oplus y_d$ be a sharing of y



- Perform computations on “shared” variables

More generally (II)

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

More generally (II)

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

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff*: Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427

More generally (II)

- 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

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff*: Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427

More generally (II)

- 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

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

partial products refreshing

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff: *Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427

More generally (II)

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

- Multiplications: $c = a \times b$ 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

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff: *Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427

More generally (II)

- 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

$\Longrightarrow$ Quadratic overheads & randomness

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff: *Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427

More generally (II)

- 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

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

$\Rightarrow$ Quadratic overheads & randomness

$\Rightarrow$ Composable (from gadgets to circuits)

Yuval Ishai, Amit Sahai, David Wagner: *Private Circuits: Securing Hardware against Probing Attacks*. CRYPTO 2003: 463-481. Matthieu Rivain, Emmanuel Prouff*: Provably Secure Higher-Order Masking of AES*. CHES 2010: 413-427. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, Rébecca Zucchini: *Strong Non-Interference and Type-Directed Higher-Order Masking*. ACM Conference on Computer and Communications Security 2016: 116-129

Main theorem (informal)

- Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  - $\mathrm{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (why d? – or d^2 in proofs)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements

- Then: $\mathrm{SR} \leq 1 - \left(1 - \mathrm{MI}(Z_i; L_{Z_i})^d\right)^m$

Alexandre Duc, Sebastian Faust, François-Xavier Standaert: Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. EUROCRYPT (1) 2015: 401-429

Main theorem (informal)

- Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  - $\mathrm{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (multiplications)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements

- Then: $\mathrm{SR} \leq 1 - \left(1 - \mathrm{MI}(Z_i; L_{Z_i})^d\right)^m$

Alexandre Duc, Sebastian Faust, François-Xavier Standaert: *Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device*. EUROCRYPT (1) 2015: 401-429

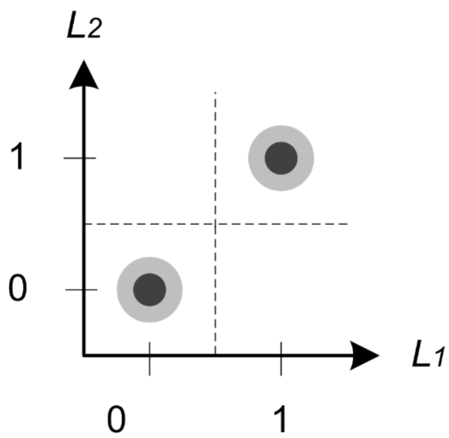
Main theorem (informal)

- Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  - $\mathrm{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (multiplications)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements
- Then: $\mathrm{SR} \leq 1 - \left(1 - \mathrm{MI}(Z_i; L_{Z_i})^d\right)^m$
- For $m = 1$, $\mathrm{SR} \leq \mathrm{MI}(Z_i; L_{Z_i})^d \propto (\sigma_N^2)^d$
- (Intuitively ≈ "noisy" piling up lemma)

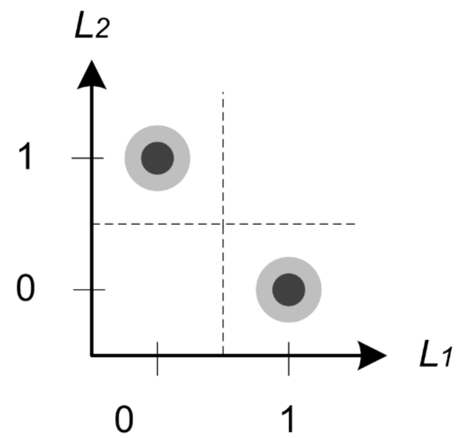
Alexandre Duc, Sebastian Faust, François-Xavier Standaert: Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. EUROCRYPT (1) 2015: 401-429

Statistical intuition

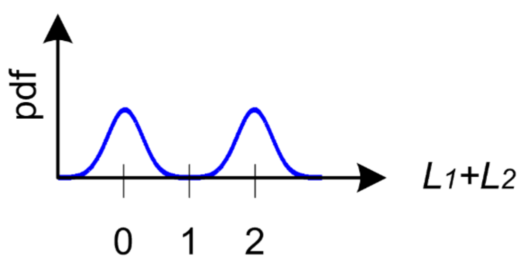
- 1-bit, 2-shares example



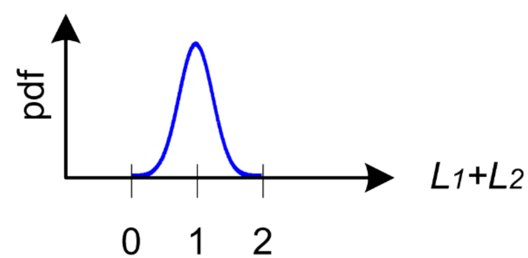
(a) $Z = 0$, serial.



(b) $Z = 1$, serial.



(c) $Z = 0$, parallel.

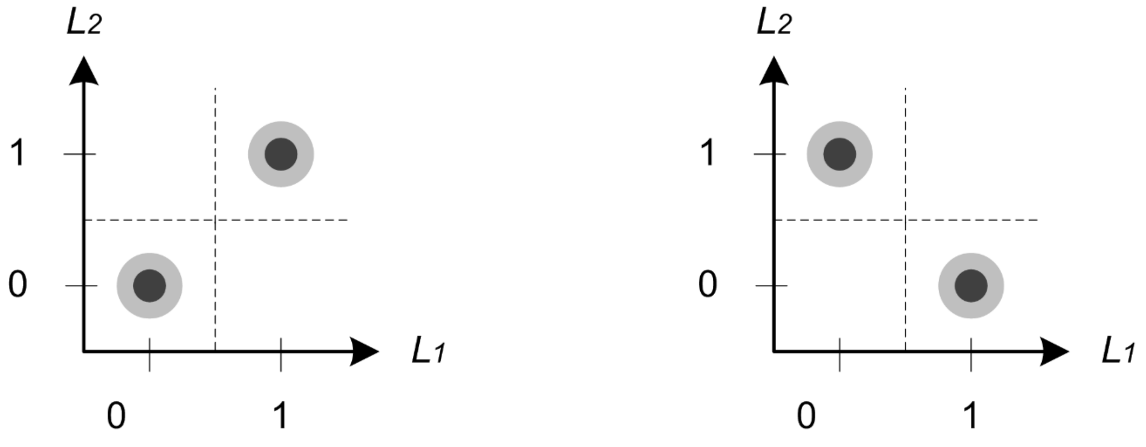


(d) $Z = 1$, parallel.

Statistical intuition

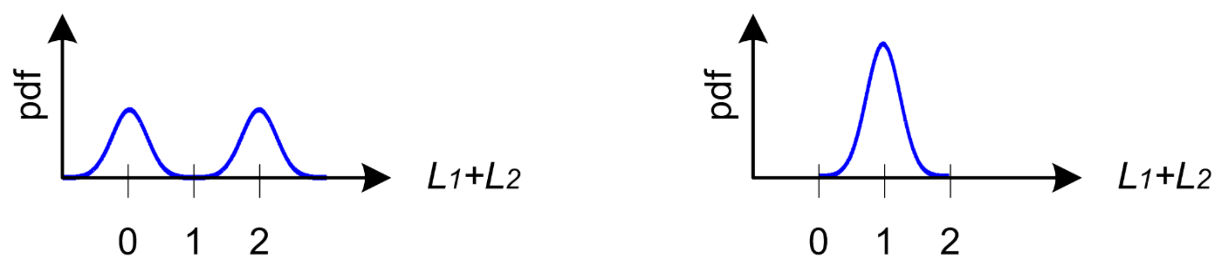
- 1-bit, 2-shares example

key-independent means



(a) Z = 0, serial.

(b) Z = 1, serial.

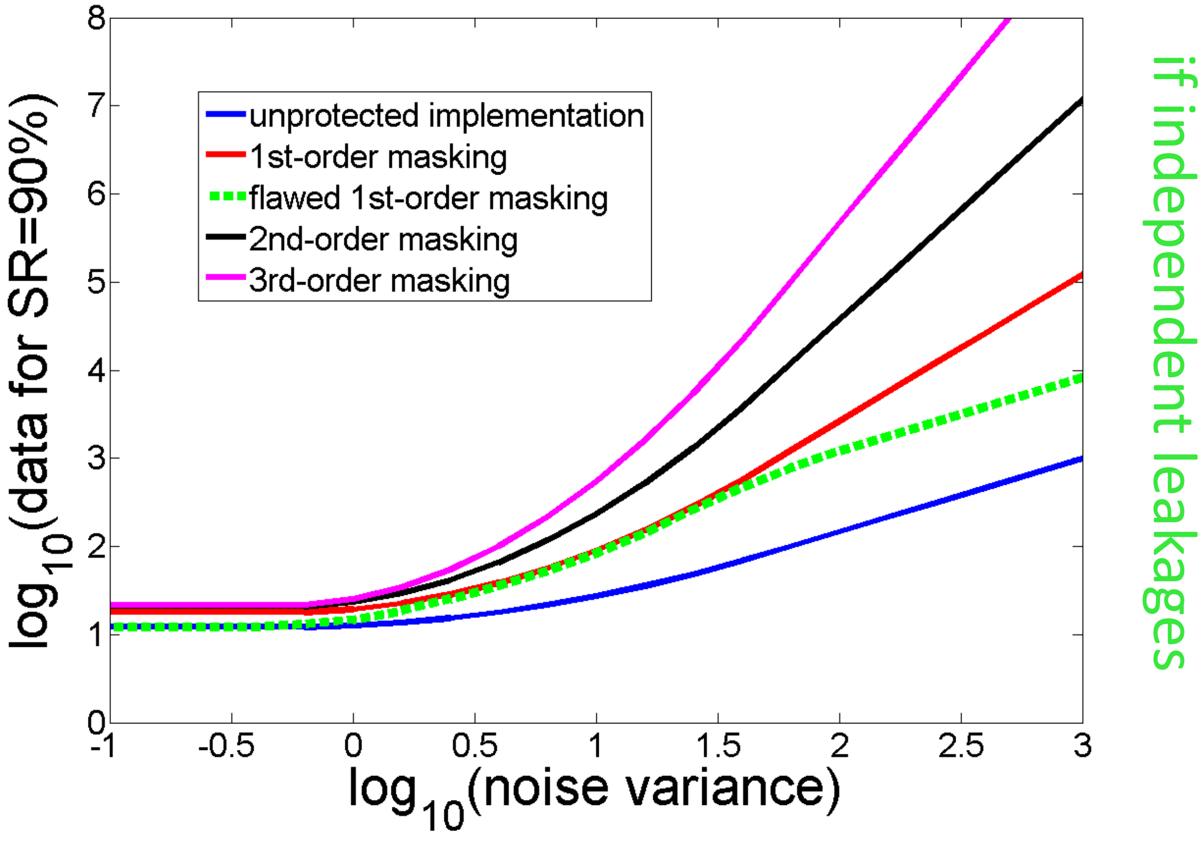


(c) Z = 0, parallel.

(d) Z = 1, parallel.

Information theoretic intuition

- Slope of the IT curves = d (i.e., security order)
  - e.g., for information leakage of an encoding



- As masking order increases, the # of *d*-tuples of informative samples increases (say by d)

⇒ the gap between "simple" attacks targeting one *d*-tuple and d ones increase by a factor d

- As masking order increases, the # of d-tuples of informative samples increases (say by d)

⇒ the gap between “simple” attacks targeting one d-tuple and d ones increase by a factor d

- If shares are re-used (allowing averaging before combination) this factor becomes d^d

Beware of the *d(imensionality)*

- As masking order increases, the # of d-tuples of informative samples increases (say by d)

⇒ the gap between "simple" attacks targeting one d-tuple and d ones increase by a factor d

- If shares are re-used (allowing averaging before combination) this factor becomes d^d

⇒ It means security depends on efficiency (in cycles), e.g., parallelism reduces # of leaking tuples

- And that t-tests become irrelevant with large #dim.

Outline

- Introduction
  - Side-channel analysis (attack steps)
  - Heuristic vs. optimal separation
- Measurement & preprocessing
  - Filtering, leakage/POI detection, dimension. reduction
- Predictions & modeling
  - Profiled vs. non-profiled separation, leakage certification
- Exploitation
  - Soft Analytical Side-Channel Attacks
- Post-processing
  - Key enumeration, rank estimation
- Future trends
  - Security without obscurity
  - Exploiting (tight) proofs

Wrapping up!

Conclusions

- For some parts, verifiably fair (i.e., close to worst-case) security evaluations are possible
  - But measurements & preprocessing remain essentially based on engineering knowledge
  - & there remain challenges for highly multivariate and (very) high-order side-channel attacks

Conclusions

- For some parts, verifiably fair (i.e., close to worst-case) security evaluations are possible
  - But measurements & preprocessing remain essentially based on engineering knowledge
  - & there remain challenges for highly multivariate and (very) high-order side-channel attacks
- Transparency is needed for high security
  - e.g., HW with 2^{80} security should be open source

Conclusions

- For some parts, verifiably fair (i.e., close to worst-case) security evaluations are possible
  - But measurements & preprocessing remain essentially based on engineering knowledge
  - & there remain challenges for highly multivariate and (very) high-order side-channel attacks
- Transparency is needed for high security
  - e.g., HW with 2^{80} security should be open source
- First focus should be on understanding (adv.'s practicality comes only afterwards)
  - e.g., thing about linear cryptanalysis

Open problems

- Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification

$\Rightarrow$ Empirically verifiable (falsifiable) assumptions
$\Rightarrow$ Systematic ways to deal with hardware defaults (or constructions that are less demanding)
$\Rightarrow$ Tight proofs in (reasonably) realistic models

Open problems

- Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification

$\Rightarrow$ Empirically verifiable (falsifiable) assumptions
$\Rightarrow$ Systematic ways to deal with hardware defaults (or constructions that are less demanding)
$\Rightarrow$ Tight proofs in (reasonably) realistic models

- Tools, formal methods, design automation

Open problems

- Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification

⇒ Empirically verifiable (falsifiable) assumptions
⇒ Systematic ways to deal with hardware defaults (or constructions that are less demanding)
⇒ Tight proofs in (reasonably) realistic models

- Tools, formal methods, design automation

- We need both theoretical works to lay out foundations & experimental case studies

THANKS

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

http://perso.uclouvain.be/fstandae/PUBLIS/183.pdf