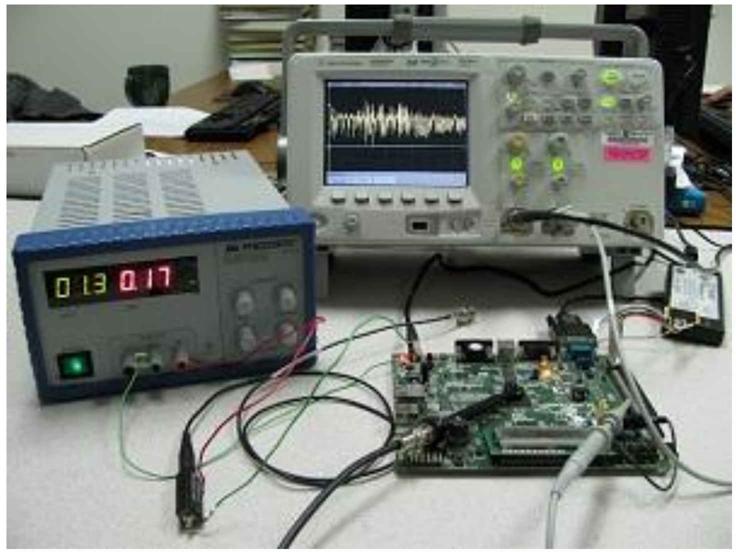
How to Certify the Leakage of a Chip?



François-Xavier Standaert

UCL Crypto Group, Belgium

CryptArchi, Fréjus, France, July 2013

Outline

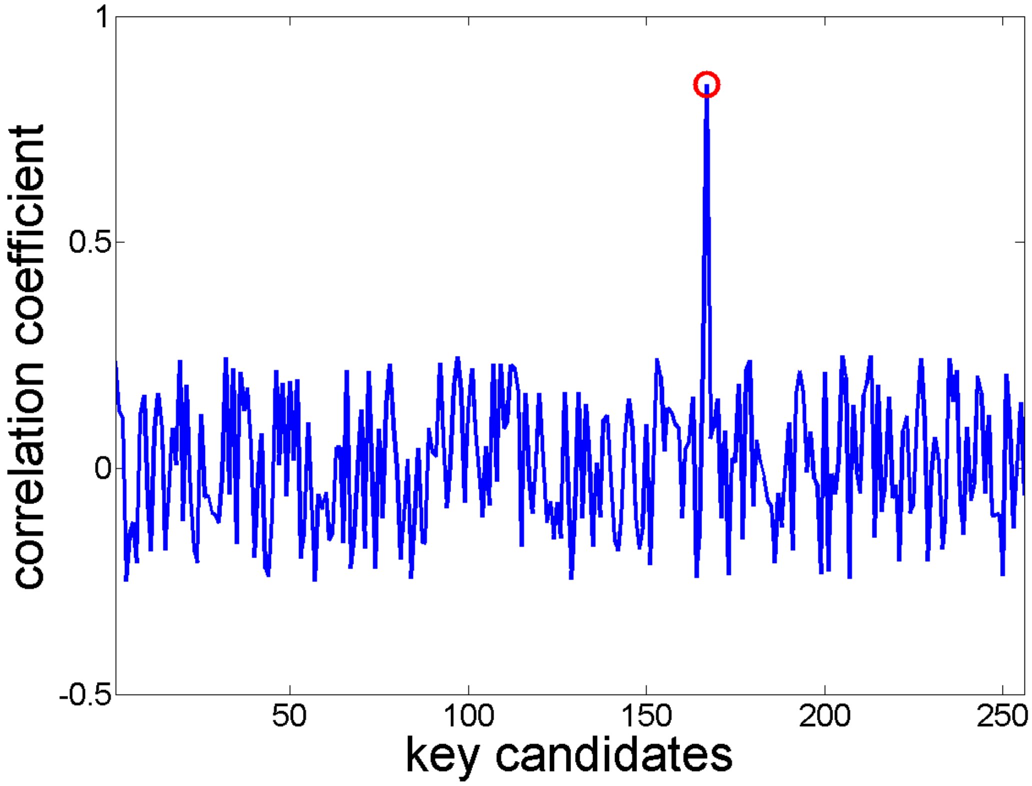
- The Eurocrypt 2009 framework revisited
- New results towards information leakage bounds
- Security analyzes and time complexity

Outline

- The Eurocrypt 2009 framework revisited
- New results towards information leakage bounds
- Security analyzes and time complexity

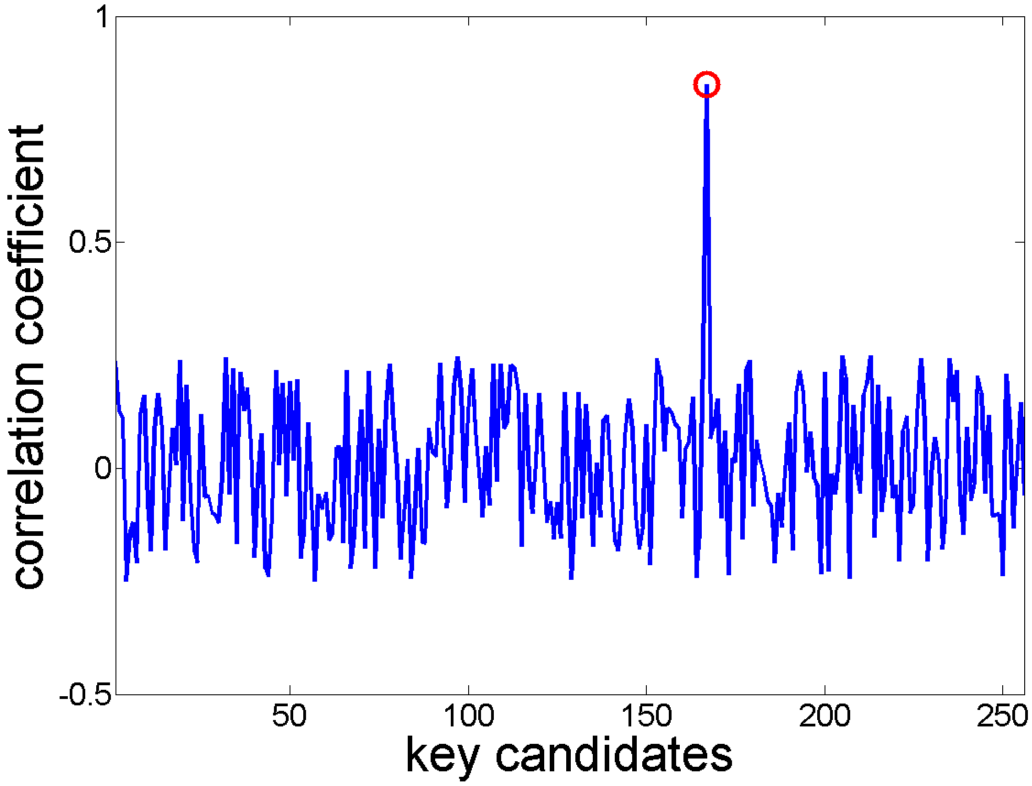
How not to evaluate

- Launch a single attack with an arbitrary distinguisher



How not to evaluate

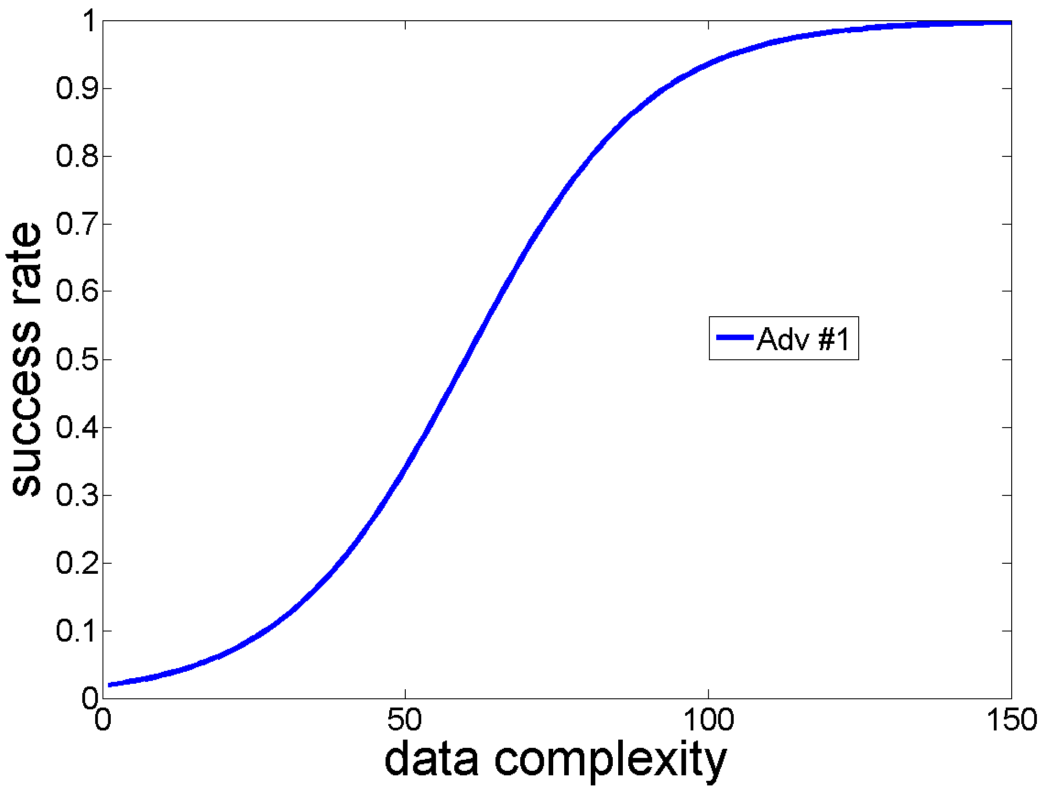
- Launch a single attack with an arbitrary distinguisher



- First issue: no statistical confidence in evaluation

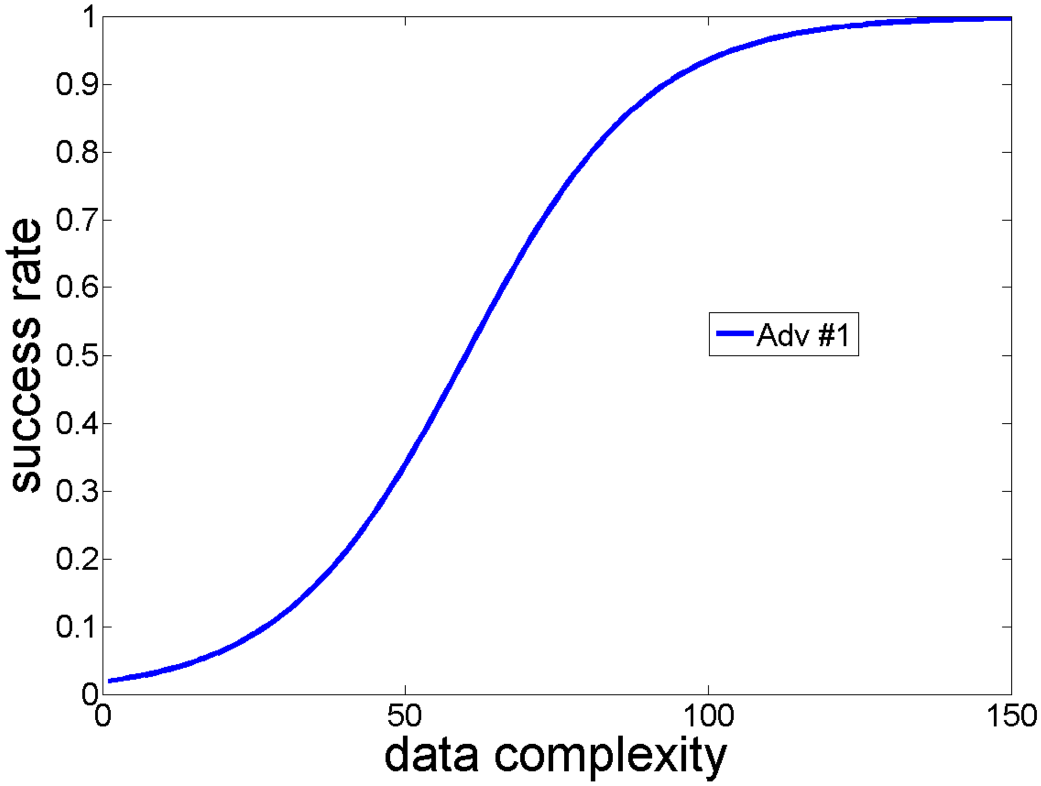
A first improvement

- Repeat the attack and estimate (e.g.) a success rate



A first improvement

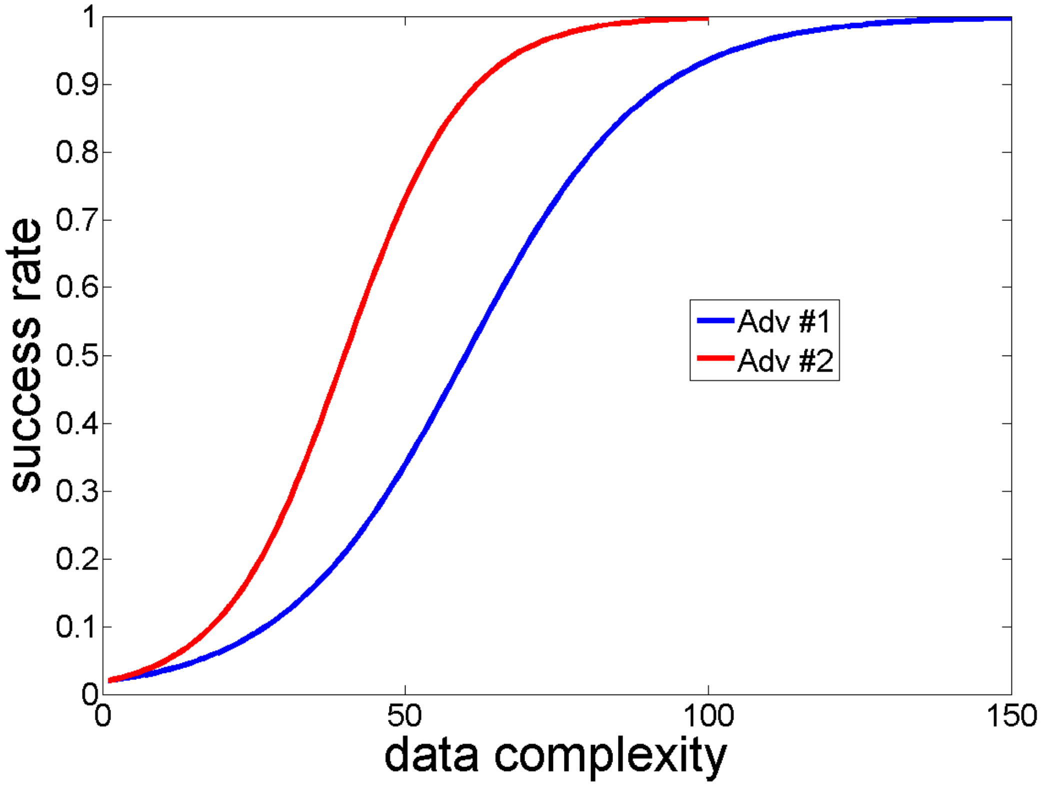
- Repeat the attack and estimate (e.g.) a success rate



- Second issue: arbitrary adversary (maybe suboptimal)

A first improvement

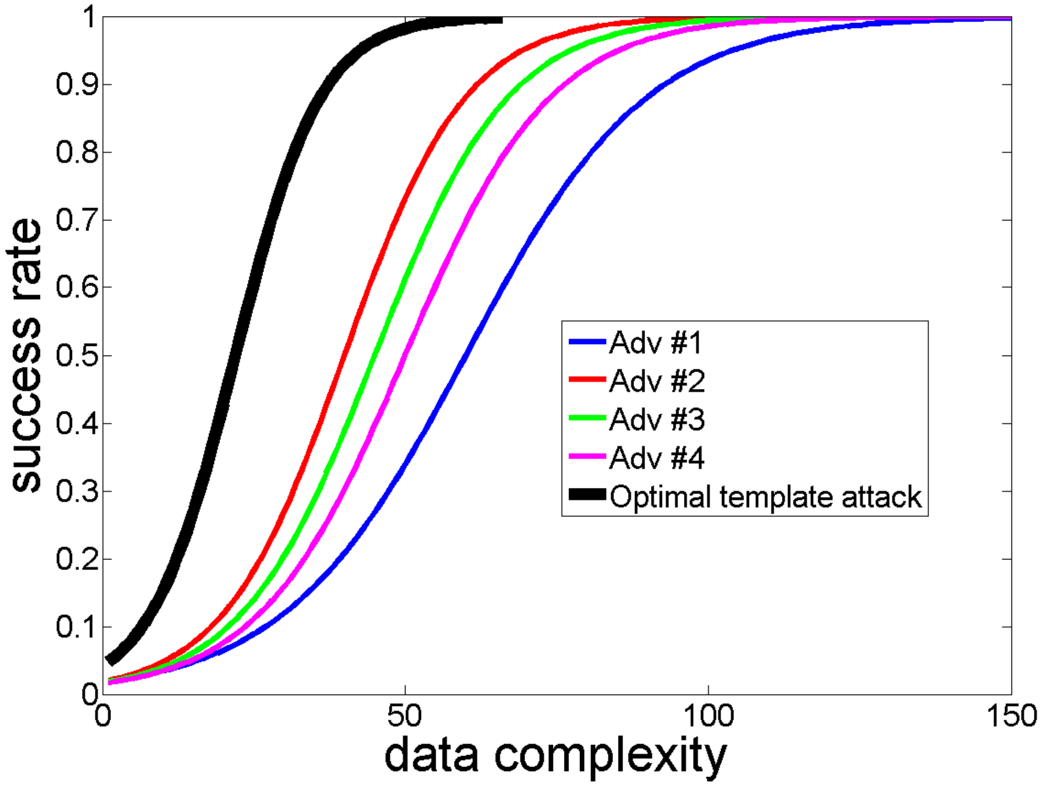
- Repeat the attack and estimate (e.g.) a success rate



- A stronger adversary may invalidate the evaluation

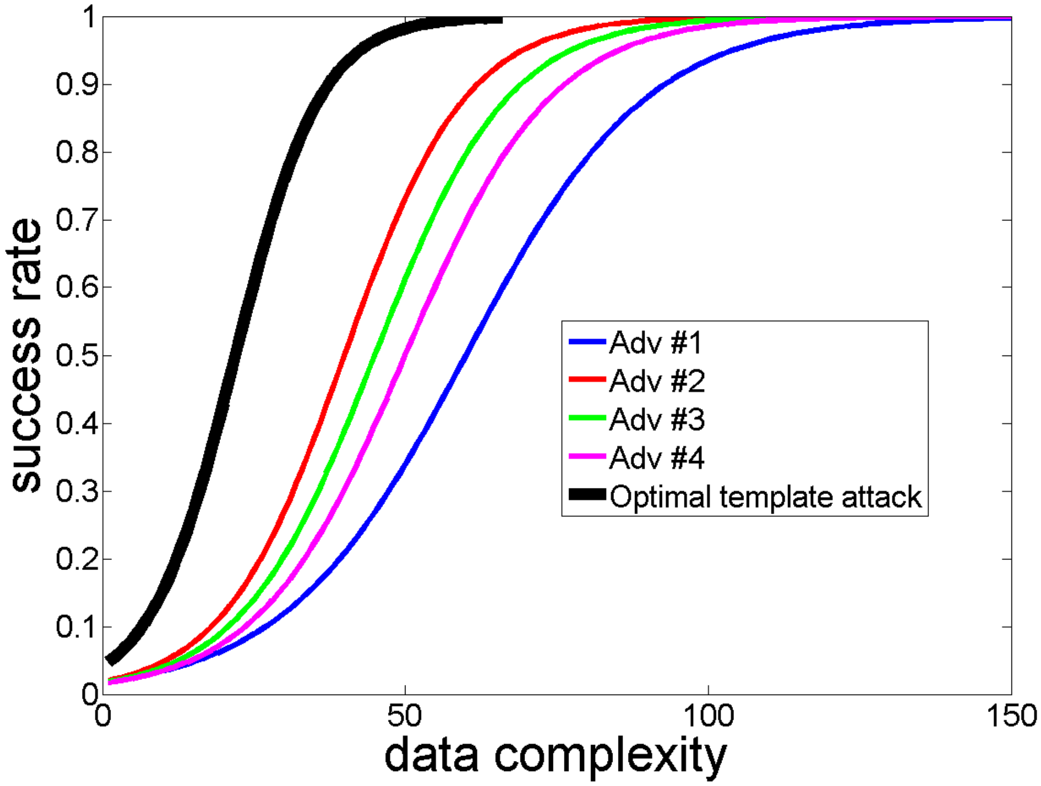
A second improvement

- Apply an “optimal” template attack



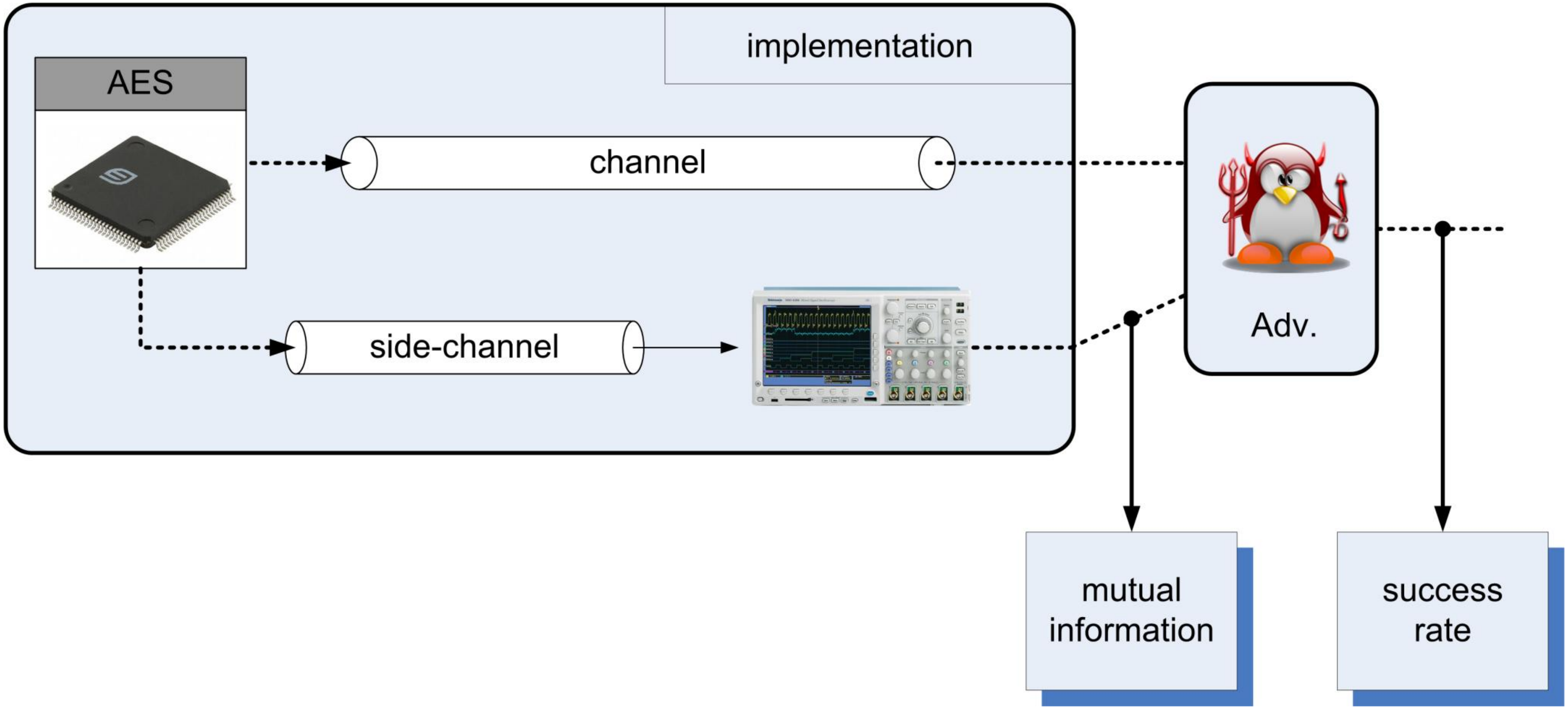
A second improvement

- Apply an "optimal" template attack

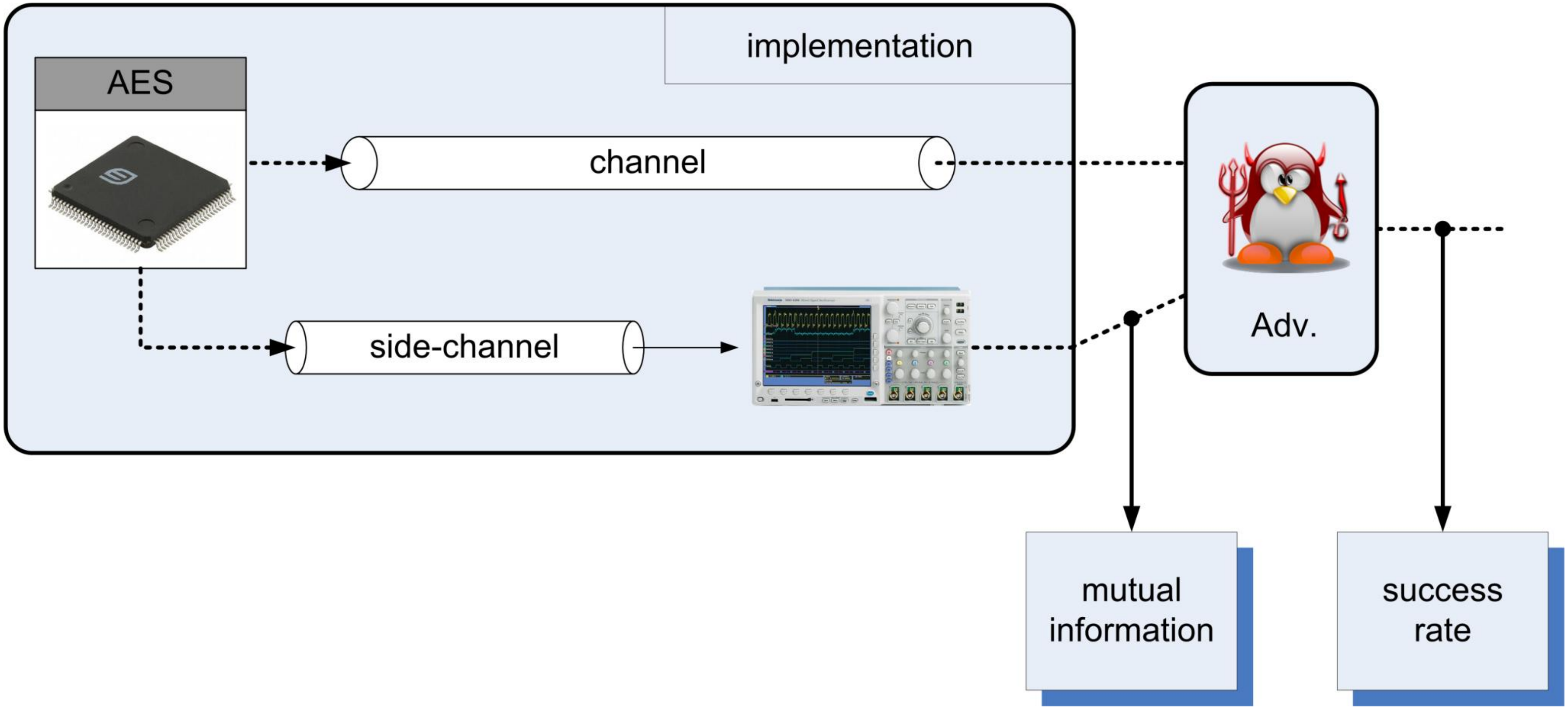


- Of course nobody know what is generally "optimal"!

Background: EC09 Framework [1]



Background: EC09 Framework [1]



- More generally: evaluate implementations with IT metrics, evaluate adversaries with security metrics

Fair evaluation of side-channel leakage

- Leakage certification is first concerned with IT metrics (i.e. aims at estimating the information leakage independent of the adversary)

Fair evaluation of side-channel leakage

- Leakage certification is first concerned with IT metrics (i.e. aims at estimating the information leakage independent of the adversary)

- *But estimating the mutual information between arbitrary distributions is notoriously hard!*

Fair evaluation of side-channel leakage

- Leakage certification is first concerned with IT metrics (i.e. aims at estimating the information leakage independent of the adversary)

- *But estimating the mutual information between arbitrary distributions is notoriously hard!*

- Good news: side-channel attacks need a model
  - i.e. an estimation of the leakage distribution

Fair evaluation of side-channel leakage

- Leakage certification is first concerned with IT metrics (i.e. aims at estimating the information leakage independent of the adversary)
- *But estimating the mutual information between arbitrary distributions is notoriously hard!*
- Good news: side-channel attacks need a model
  - i.e. an estimation of the leakage distribution
- Main idea: estimate the mutual information from the “best available” profiled model (i.e. worst case)

- Information leakage on the secret key

$$\mathrm{H}[K] - \sum_{k} \mathrm{Pr}[k] \sum_{l} \mathrm{Pr}_{chip}\,[l|k]\ .\log_2 \widehat{\mathrm{Pr}}_{model}\,[k|l]$$

- where $\widehat{\mathrm{Pr}}_{model}\,[k|l]$ is obtained by profiling
- and $\mathrm{Pr}_{chip}\,[l|k]$ is obtained by sampling

In practice: two-step process

- *Step 1*: estimate the leakage model $\widehat{\Pr}_{model}\,[k|l]$
  - e.g. with Gaussian templates, linear regression, Gaussian mixtures, Kernel density estimation, …

In practice: two-step process

- *Step 1*: estimate the leakage model $\widehat{\Pr}_{model}[k|l]$
  - e.g. with Gaussian templates, linear regression, Gaussian mixtures, Kernel density estimation, ….

- *Step 2*: estimate the information leakage by sampling $\Pr_{chip}[l|k]$ (i.e. perform measurements)

In practice: two-step process

- *Step 1*: estimate the leakage model $\widehat{\Pr}_{model}[k|l]$
  - e.g. with Gaussian templates, linear regression, Gaussian mixtures, Kernel density estimation, ….
- *Step 2*: estimate the information leakage by sampling $\Pr_{chip}[l|k]$ (i.e. perform measurements)
- Note: measurements to estimate the leakage model and the IT metric must be independent!

Example

- 4 key candidates with correct key k=1

Example

- 4 key candidates with correct key k=1
- $\sum_l \mathrm{Pr}_{chip}\,[l|k=1]\ .\log_2 \widehat{\mathrm{Pr}}_{model}\,[k=1|l]$

Example

- 4 key candidates with correct key k=1
- $\sum_l \Pr_{chip}[l|k=1] \,.\log_2 \widehat{\Pr}_{model}[k=1|l]$

	$k=0$	$k=1$	$k=2$	$k=3$
l_1	p_{10}	p_{11}	p_{12}	p_{13}

- 4 key candidates with correct key k=1
- $\sum_l \Pr_{chip}[l|k=1] \,.\log_2 \widehat{\Pr}_{model}[k=1|l]$

	k=0	*k=1*	*k=2*	*k=3*
l_1	p_{10}	p_{11}	p_{12}	p_{13}
l_2	p_{20}	p_{21}	p_{22}	p_{23}

Example

- 4 key candidates with correct key k=1
- $\sum_l \Pr_{chip}[l|k=1]\ .\log_2 \widehat{\Pr}_{model}[k=1|l]$

	k=0	*k=1*	*k=2*	*k=3*
l_1	p_{10}	p_{11}	p_{12}	p_{13}
l_2	p_{20}	p_{21}	p_{22}	p_{23}
l_3	p_{30}	p_{31}	p_{32}	p_{3}

Example

- 4 key candidates with correct key k=1
- $\sum_l \Pr_{chip}[l|k=1] \,.\log_2 \widehat{\Pr}_{model}[k=1|l]$

	k=0	*k=1*	*k=2*	*k=3*
l_1	p_{10}	p_{11}	p_{12}	p_{13}
l_2	p_{20}	p_{21}	p_{22}	p_{23}
l_3	p_{30}	p_{31}	p_{32}	p_{3}
…	…	…	…	…
l_N	p_{N0}	p_{N1}	p_{N2}	p_{N3}

Example

- 4 key candidates with correct key k=1
- $\sum_l \Pr_{chip}[l|k=1] \,.\log_2 \widehat{\Pr}_{model}[k=1|l]$

	$k=0$	$k=1$	$k=2$	$k=3$
l_1	p_{10}	p_{11}	p_{12}	p_{13}
l_2	p_{20}	p_{21}	p_{22}	p_{23}
l_3	p_{30}	p_{31}	p_{32}	p_{3}
…	…	…	…	…
l_N	p_{N0}	p_{N1}	p_{N2}	p_{N3}

$$\Rightarrow \frac{1}{N}\sum_{i=1}^{N} \log_2 pi_1$$

Two cases can happen [2]

- Case #1 (ideal): perfect profiling phase
- i.e. $\widehat{\Pr}_{model}[k|l] = \Pr_{chip}[l|k]$

$$\widehat{\mathrm{MI}}(K;L) = \mathrm{H}[K] - \sum_{k} \Pr[k] \sum_{l} \Pr_{chip}[l|k] \,.\log_2 \Pr_{chip}[l|k]$$

Two cases can happen [2]

- Case #1 (ideal): perfect profiling phase
- i.e. $\widehat{\Pr}_{model}[k|l] = \Pr_{chip}[l|k]$

$$\widehat{\mathrm{MI}}(K;L) = \mathrm{H}[K] - \sum_k \Pr[k] \sum_l \Pr_{chip}[l|k] \,.\log_2 \Pr_{chip}[l|k]$$

- Case #2 (actual): bounded profiling phase
- i.e. $\widehat{\Pr}_{model}[k|l] \neq \Pr_{chip}[l|k]$

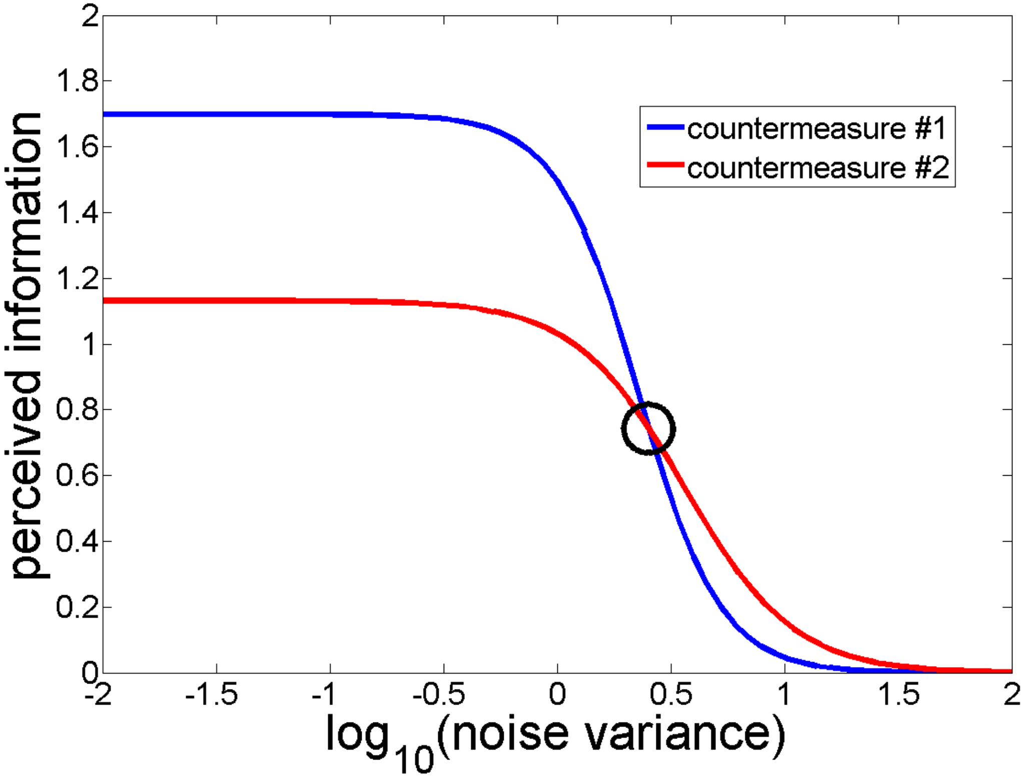
$$\widehat{\mathrm{PI}}(K;L) = \mathrm{H}[K] - \sum_k \Pr[k] \sum_l \Pr_{chip}[l|k] \,.\log_2 \widehat{\Pr}_{model}[k|l]$$

Main theorem (informal)

- PI(K;L) is directly proportional to the success rate of an adversary using $\widehat{\Pr}_{model}[k|l]$ as template

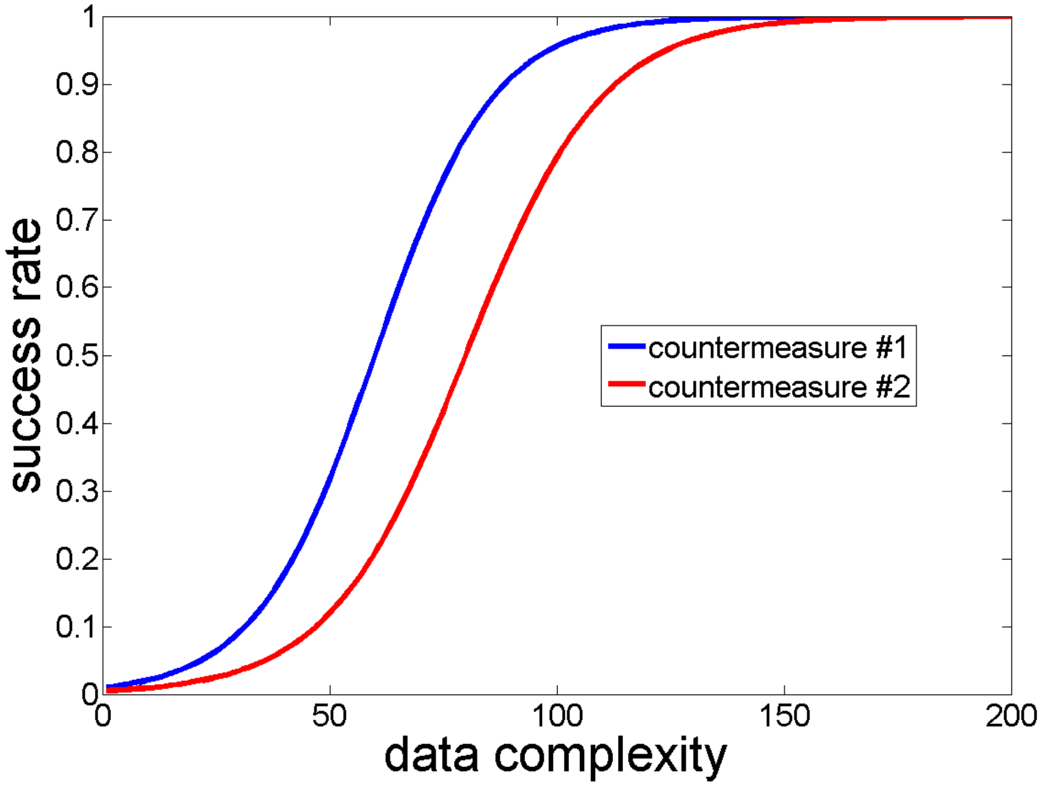
Main theorem (informal)

- PI(K;L) is directly proportional to the success rate of an adversary using $\widehat{\Pr}_{model}[k|l]$ as template
- e.g. PI(K;L) in function of the noise variance



As a result

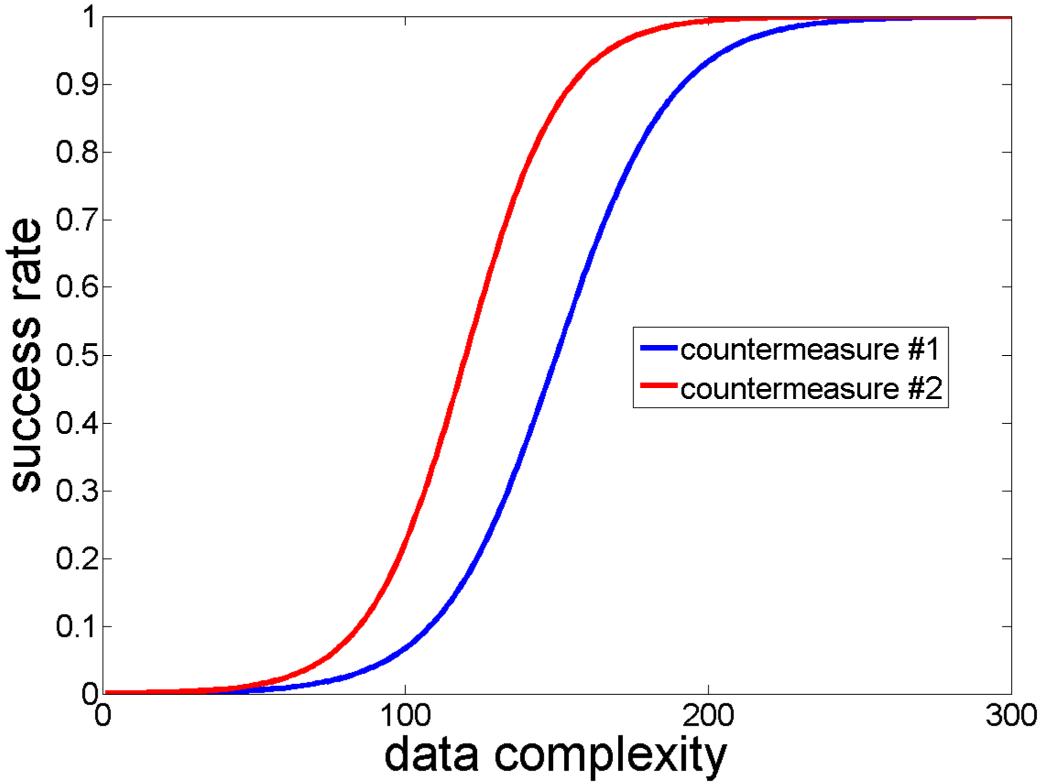
- Left of the intersection



- Countermeasure #2 more secure than first one

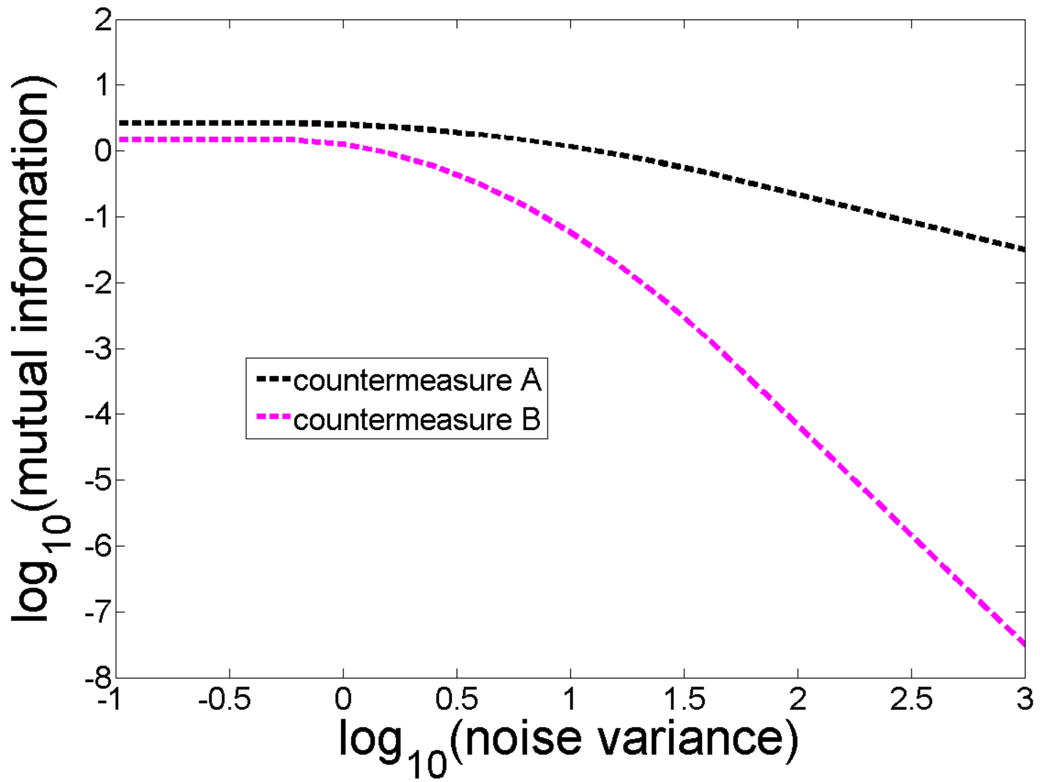
As a result

- Right of the intersection



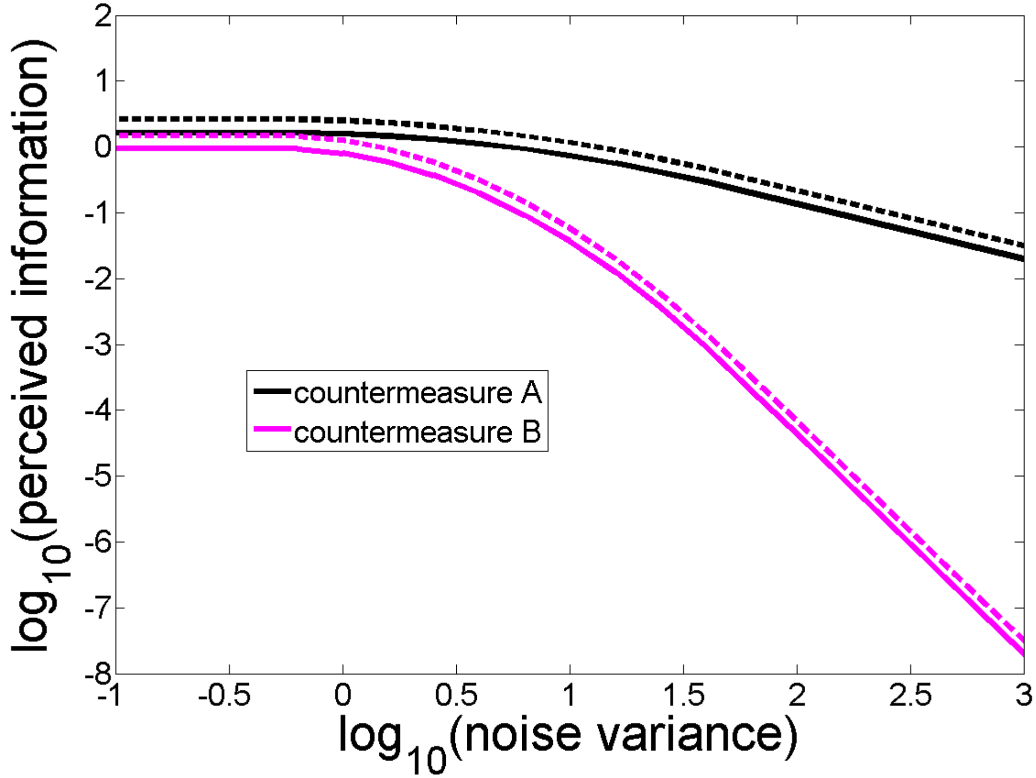
- Countermeasure #1 more secure than second one

- MI(K;L) measures the worst case data complexity

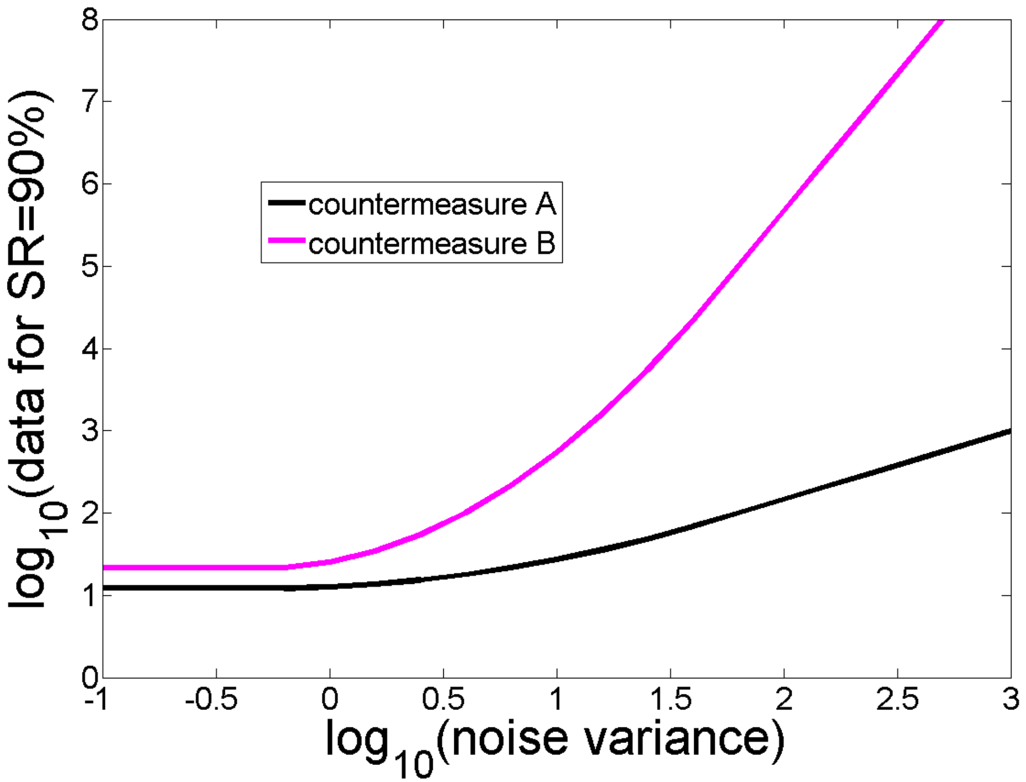


In other words

- PI(K;L) is the evaluator's best estimate



Relation with data complexity



- Theorem only proven in very specific cases
- But holds surprisingly well in real-world settings

Example: masking [3]

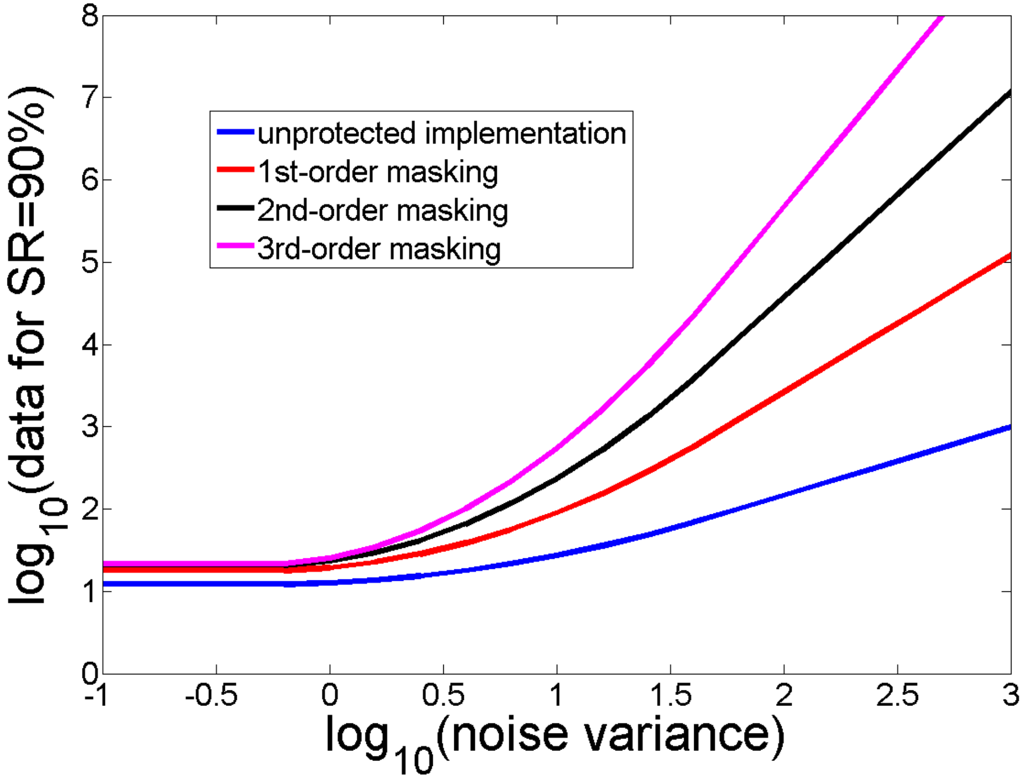
- Main idea: split the sensitive data in *r* shares

Example: masking [3]

- Main idea: split the sensitive data in *r* shares

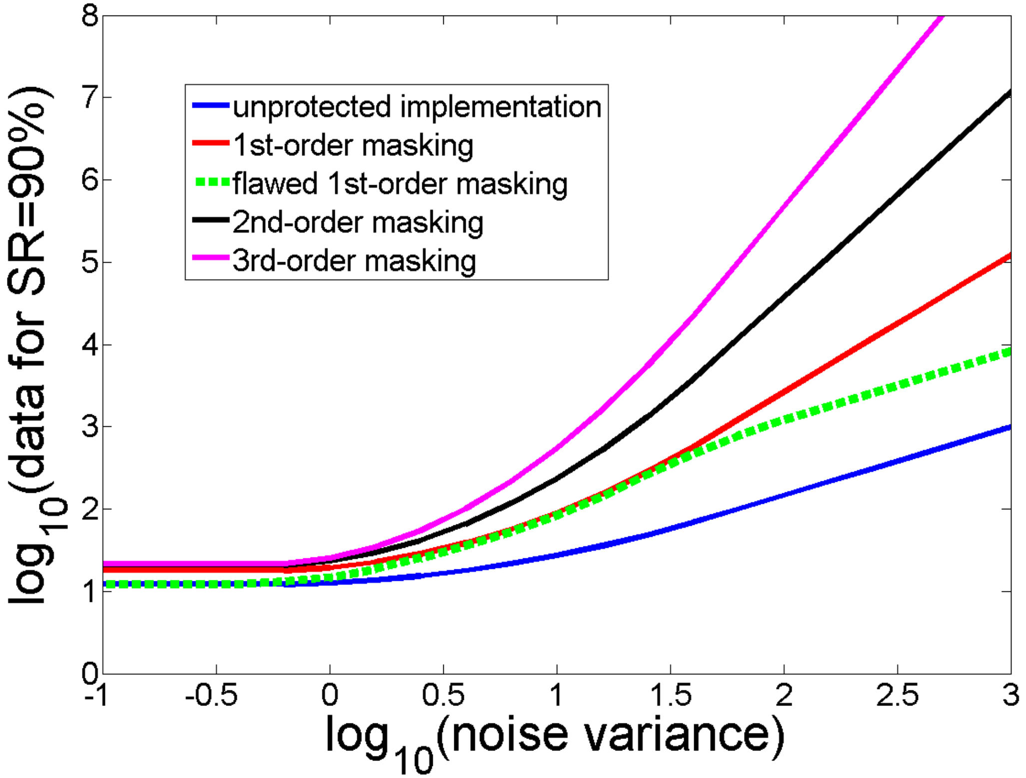
- If “perfect” implementation, the data complexity to break masking is proportional to $(\sigma_n^2)^r$
  - Perfect ≈ if the smallest-order key-dependent moment in the leakage distribution is *r*
  - Essentially depends on the hardware (e.g. glitches may make the implementation imperfect)

Information theoretic intuition



- Smallest-order key-dept. moment = curve slope

Information theoretic intuition



- Flaws due to physical defaults can be detected

How to evaluate the metrics?

- Implies to select good statistical tools
  - Critical point: PDF estimation problem

How to evaluate the metrics?

- Implies to select good statistical tools
  - Critical point: PDF estimation problem

- Tools are highly dependent on the contexts
  - So is the distance between MI and PI (and hence, the relevance of security evaluations)

How to evaluate the metrics?

- Implies to select good statistical tools
  - Critical point: PDF estimation problem
- Tools are highly dependent on the contexts
  - So is the distance between MI and PI (and hence, the relevance of security evaluations)
- A few examples next…

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Different implementations and countermeasures
- Which cases are “easy to evaluate”?

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Most distinguishers are asymtotically equivalent [4]
- … if provided with the same leakage model

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- PCA, LDA, … useful in the profiled case [5]
- Dimension reduction uneasy in non-profiled case

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Same tools as for unprotected devices work well
- Non-linear leakage functions require profiling [6]

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Uneasy to evaluate for both type of attacks
- Signal proc. can cancel countermeasures [7,8]

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Becomes measurement intensive as r increases
- No solution is always optimal in non-profiled case

Examples

	profiled attacks	non-profiled attacks
unprotected device, univariate leakage		
unprotected device, multivariate leakage		
dual-rail pre-charged implementation		
time randomizations		
masking		
combination of countermeasures		

- Specially hard if the design is unknown
- Large distance btw. profiled & non-profiled cases

Summarizing

- PI(K;L) provide a unifying view of countermeasures
- IT curves capture most intuition regarding the data complexity of worst case side-channel attacks

Summarizing

- PI(K;L) provide a unifying view of countermeasures
- IT curves capture most intuition regarding the data complexity of worst case side-channel attacks

- Evaluator's goal: avoid "false sense of security"
  - PI(K;L) ≠ MI(K;L)
  - Significant differences may arise due to signal processing, bad assumptions on the leakage, …
  - Measurement setup also matters!

Outline

- The Eurocrypt 2009 framework revisited
- New results towards information leakage bounds
- Security analyzes and time complexity

The fundamental evaluation problem

- What is the distance between the MI and the PI?
- (i.e. how good is my leakage model?)

The fundamental evaluation problem

- What is the distance between the MI and the PI?
- (i.e. how good is my leakage model?)

- Difficult since the leakage function is unknown

=> Impossible to compute this distance directly!

The fundamental evaluation problem

- What is the distance between the MI and the PI?
- (i.e. how good is my leakage model?)

- Difficult since the leakage function is unknown

=> Impossible to compute this distance directly!

- Next: we show that indirect approaches allow answering the question quite rigorously

The fundamental evaluation problem

- What is the distance between the MI and the PI?
- (i.e. how good is my leakage model?)

- Difficult since the leakage function is unknown

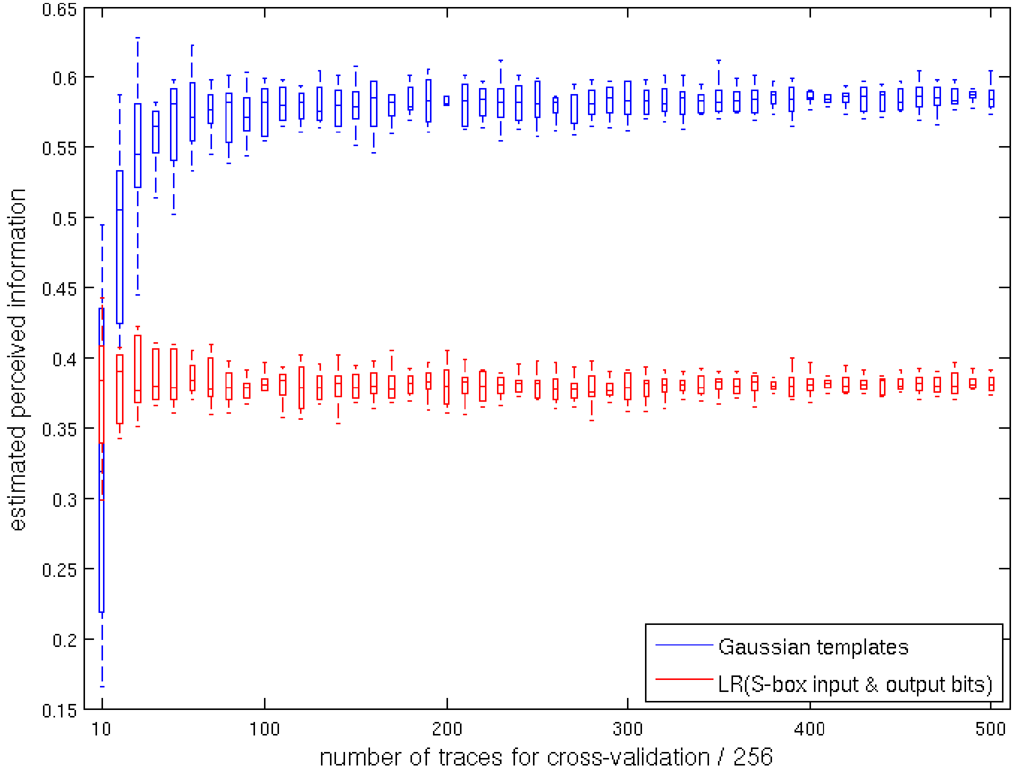
=> Impossible to compute this distance directly!

- Next: we show that indirect approaches allow answering the question quite rigorously

- Main idea: quantify *estimation* & *assumption* errors

1. Estimation errors => cross-validation

- Split traces in 10 (non-overlapping) sets, use 9/10th for profiling, 1/10th for estimating the PI
- Repeat 10 times to get average & spread



2. Assumption errors => distance sampling

- Fact: two multidimensional distributions $\mathcal{F}$ and $\mathcal{G}$ are equal if the variables X~$\mathcal{F}$ and Y~$\mathcal{G}$ generate identical distributions for the distance D(X,Y)

2. Assumption errors => distance sampling

- Fact: two multidimensional distributions $\mathcal{F}$ and $\mathcal{G}$ are equal if the variables X~$\mathcal{F}$ and Y~$\mathcal{G}$ generate identical distributions for the distance D(X,Y)

- We can compute the simulated distance

$$f_{sim}(d) = \Pr[L_1 - L_2 \leq d \mid L_1, L_2 \sim \widehat{\Pr}_{model}]$$

- Fact: two multidimensional distributions $\mathcal{F}$ and $\mathcal{G}$ are equal if the variables X~$\mathcal{F}$ and Y~$\mathcal{G}$ generate identical distributions for the distance D(X,Y)

- We can compute the simulated distance

$$f_{sim}(d) = \Pr[L_1 - L_2 \leq d \mid L_1, L_2 \sim \widehat{\Pr}_{model}]$$

- And the sampled distance

$$\hat{g}_N(d) = \Pr[l_1 - l_2 \leq d \mid l_1 \overset{N}{\Leftarrow} \widehat{\Pr}_{model}, l_2 \overset{N}{\Leftarrow} \Pr_{chip}]$$

2. Assumption errors => distance sampling

- Fact: two multidimensional distributions $\mathcal{F}$ and $\mathcal{G}$ are equal if the variables X~$\mathcal{F}$ and Y~$\mathcal{G}$ generate identical distributions for the distance D(X,Y)
- We can compute the simulated distance

$$f_{sim}(d) = \Pr[L_1 - L_2 \leq d \mid L_1, L_2 \sim \widehat{\Pr}_{model}]$$

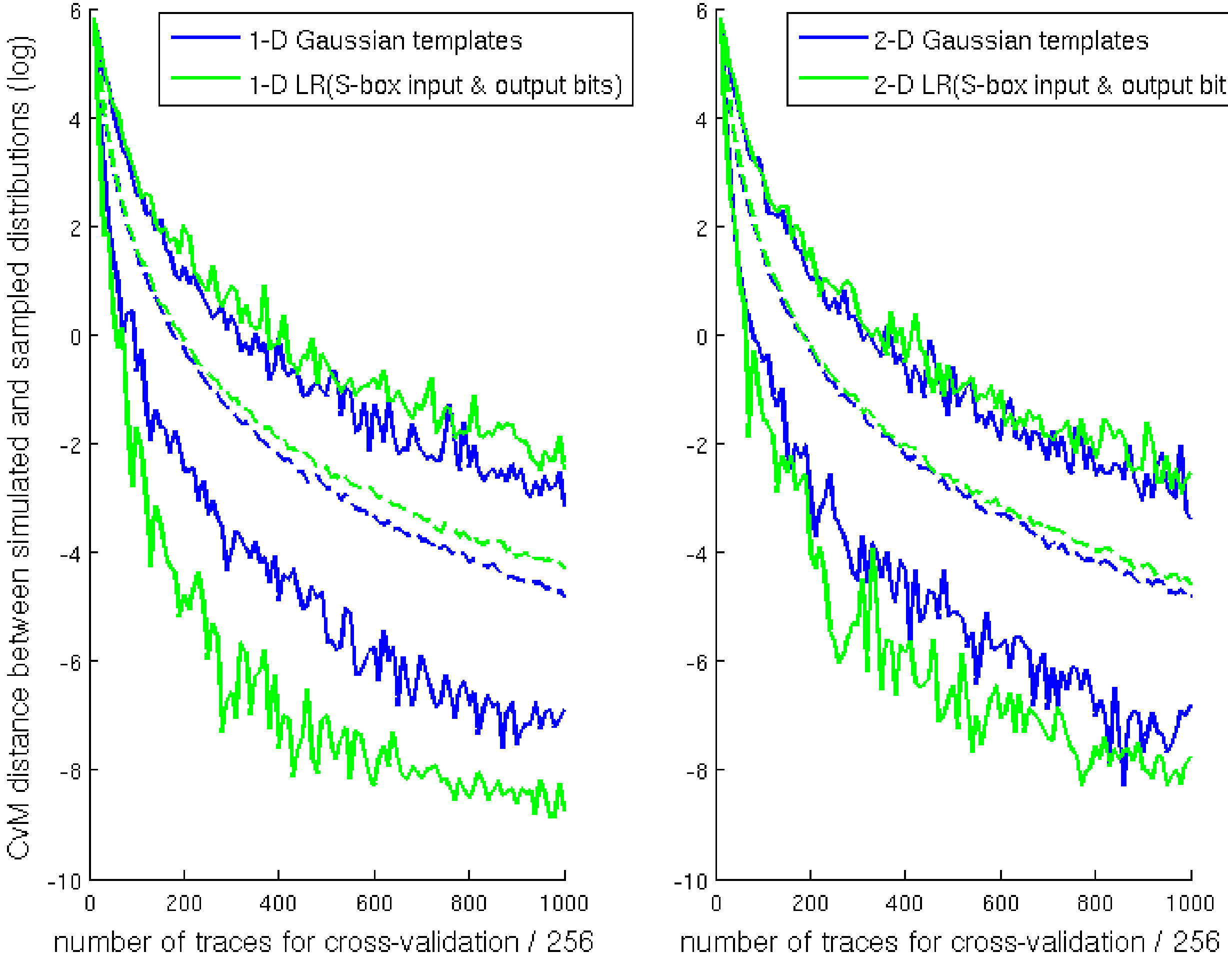
- And the sampled distance

$$\hat{g}_N(d) = \Pr[l_1 - l_2 \leq d \mid l_1 \overset{N}{\Leftarrow} \widehat{\Pr}_{model}, l_2 \overset{N}{\Leftarrow} \Pr_{chip}]$$

- And test their CvM divergence

$$\widehat{\mathrm{CvM}}\,(f_{sim}, \hat{g}_N) = \int [f_{sim}(x) - \hat{g}_N(x)]^2 dx$$

With cross-validation again, we obtain



- Any incorrect assumption => CvM saturates

3. Quantifying the information loss

- Estimation errors can be made arbitrarily small by measuring => assumption errors more damaging

3. Quantifying the information loss

- Estimation errors can be made arbitrarily small by measuring => assumption errors more damaging

- Idea: try to detect when (i.e. for which # of traces in the cross-validation set) assumption errors become significant in front of estimation ones

How to?

- Compute a sampled simulated distance

$$\hat{f}_{sim,N}(d) = \Pr[l_1 - l_2 \leq d \mid l_1, l_2 \overset{N}{\Leftarrow} \widehat{\Pr}_{model}]$$

How to?

- Compute a sampled simulated distance

$$\hat{f}_{sim,N}(d) = \Pr[l_1 - l_2 \leq d \mid l_1, l_2 \overset{N}{\Leftarrow} \widehat{\Pr}_{model}]$$

- Characterize the probability that a given divergence between f_{sim} and $\hat{f}_{sim,N}$ would be observed for a given number of traces *N*

How to?

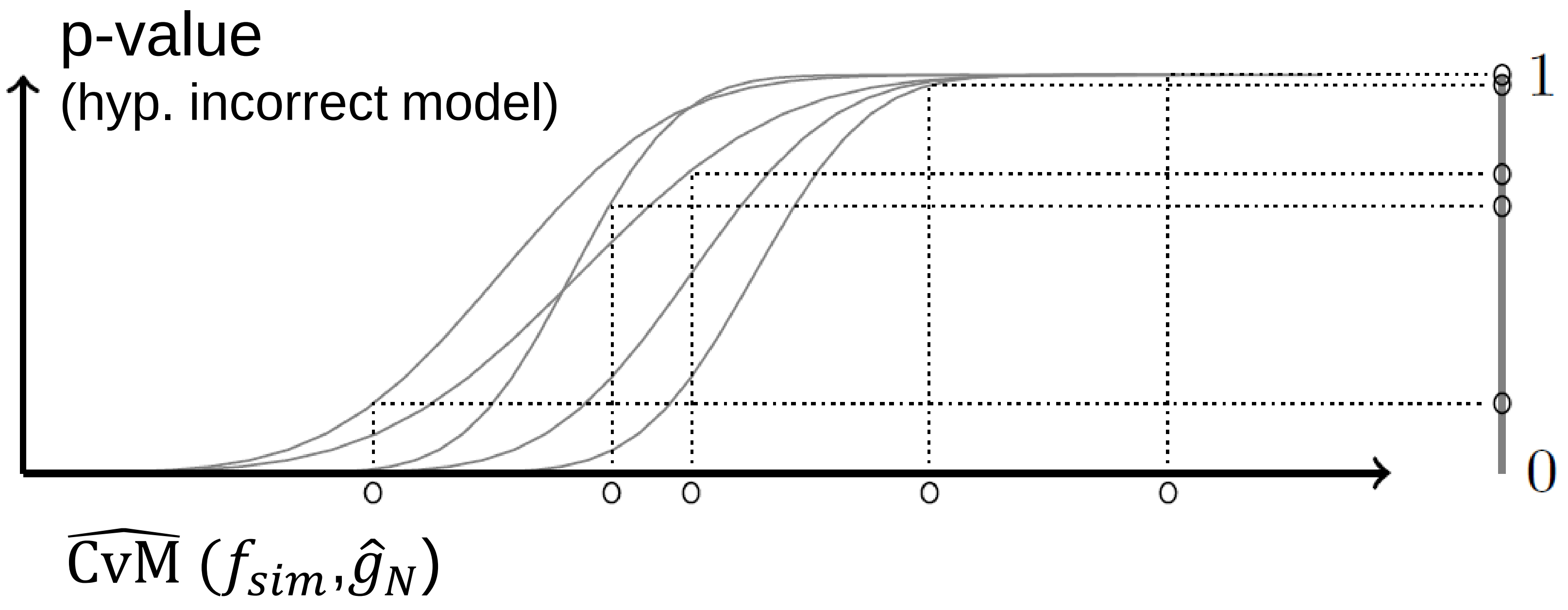
- Compute a sampled simulated distance

$$\hat{f}_{sim,N}(d) = \Pr[l_1 - l_2 \le d \mid l_1, l_2 \overset{N}{\Leftarrow} \widehat{\Pr}_{model}]$$

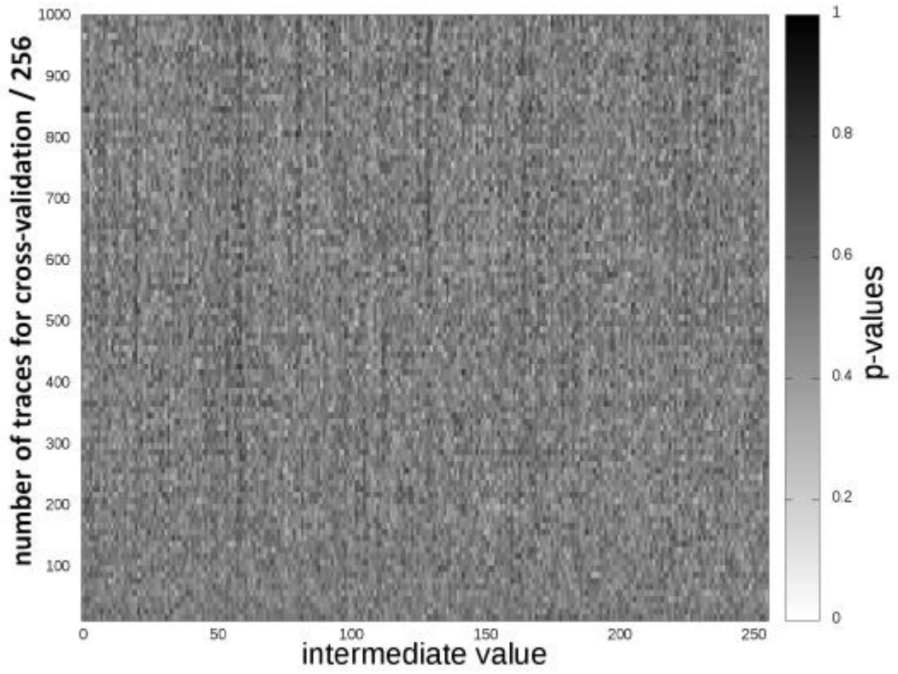
- Characterize the probability that a given divergence between f_{sim} and $\hat{f}_{sim,N}$ would be observed for a given number of traces *N*

- Look whether a given divergence between f_{sim} and $\hat{g}_N$ (the latter obtained during cross-validation again) can be due to estimation errors

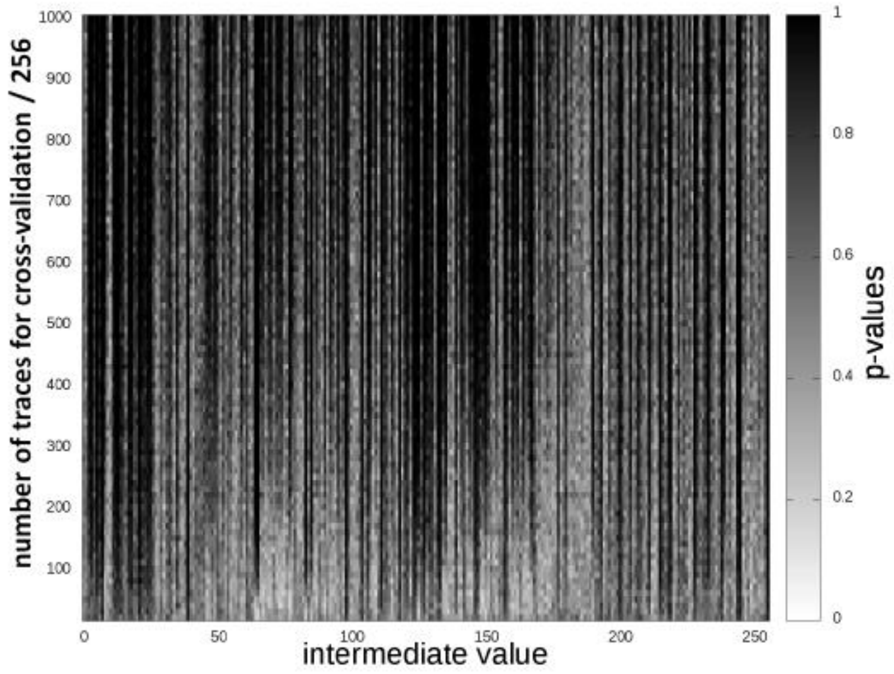
Illustration



Example



Gaussian templates



Stochastic model

Towards leakage bounds?

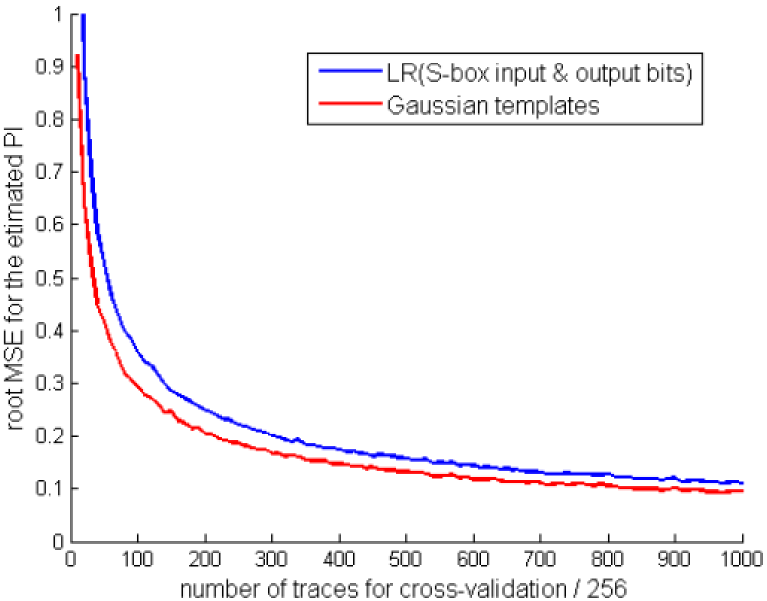
- Assume estimation errors are “small enough”
  - Which is easily obtained with enough meas.

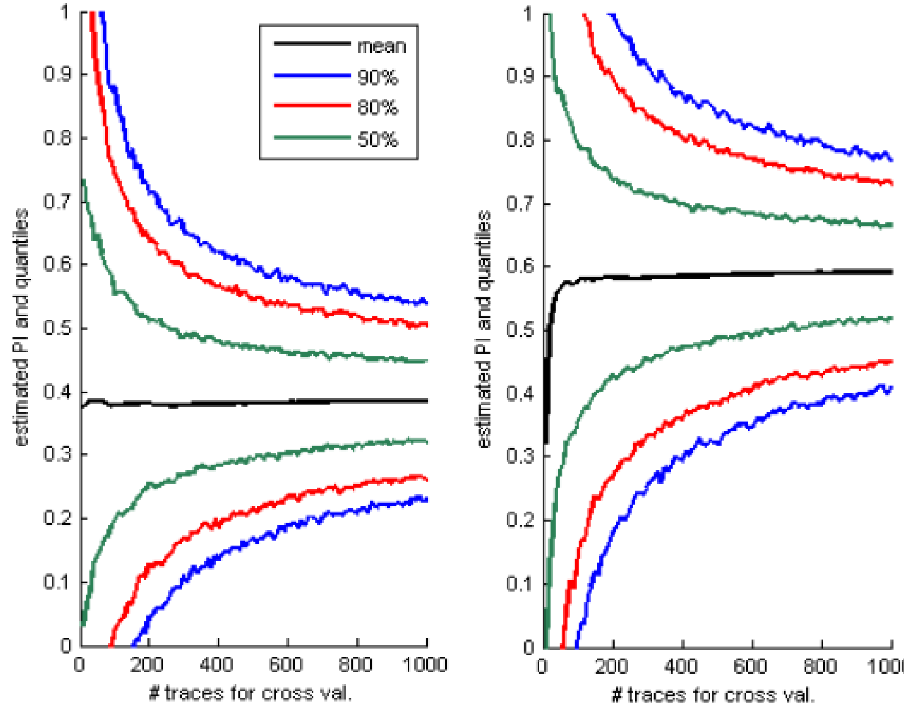
Towards leakage bounds?

- Assume estimation errors are “small enough”
  - Which is easily obtained with enough meas.

- Conjecture: For *N* such that the assumption errors are “not significant” in front of estimation errors, we can “bound” the information loss by quantifying the estimation error
  - (i.e. assumption errors that are detected for smaller *N*’s are inevitably larger)

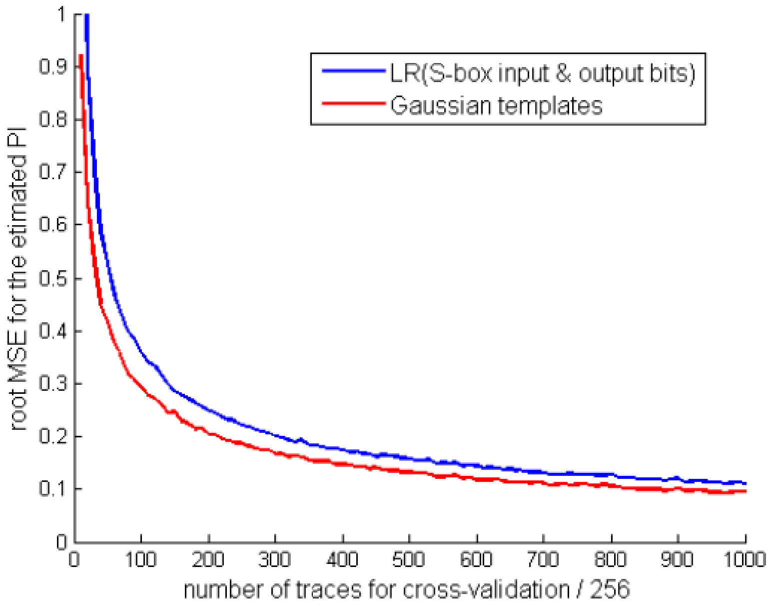
Example

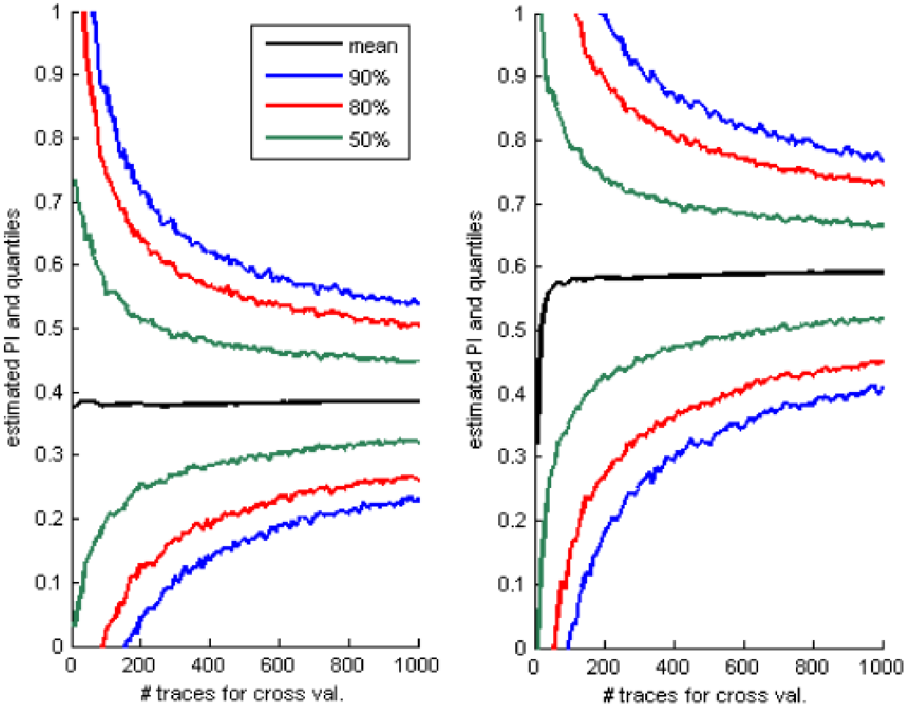




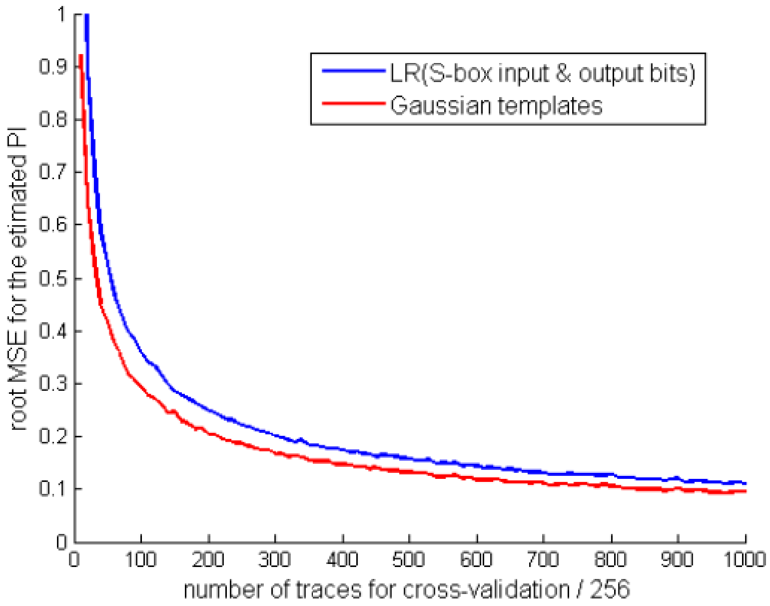
- Identified template attack with PI = 0.58

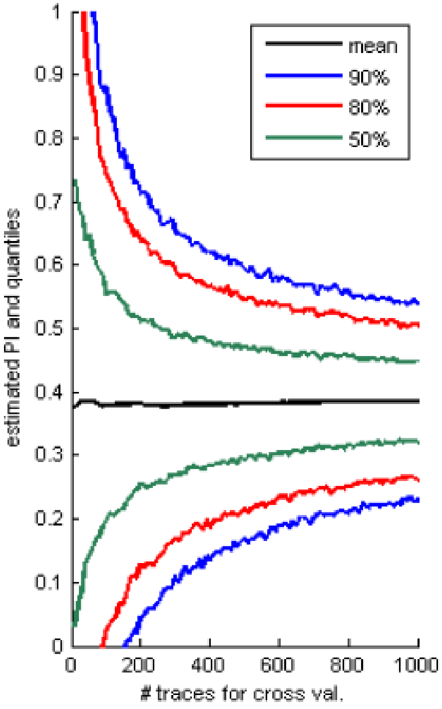
Example

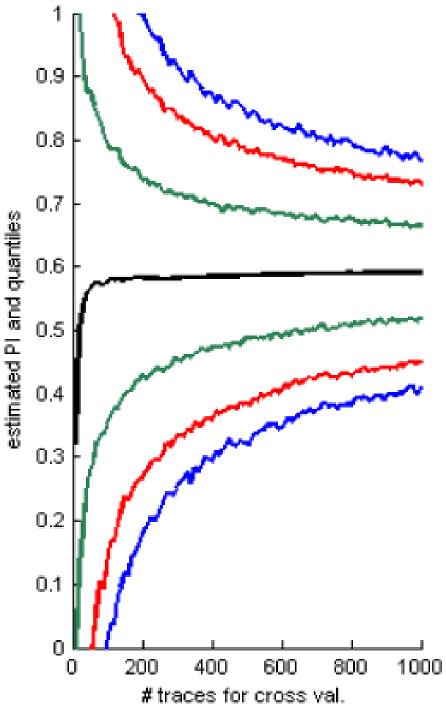




- Identified template attack with PI = 0.58
- No assumption errors for *N*=1000

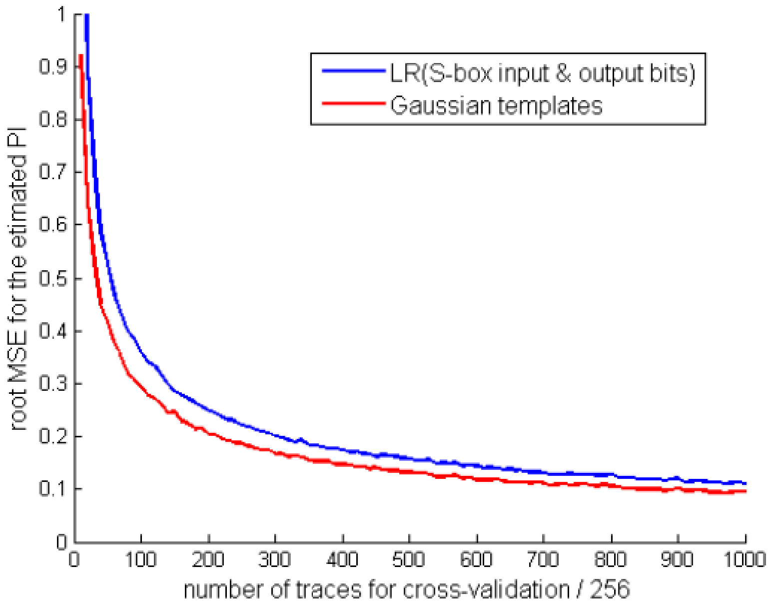


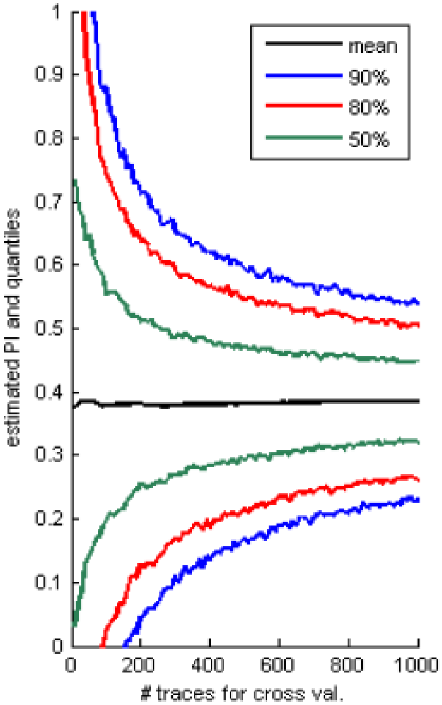


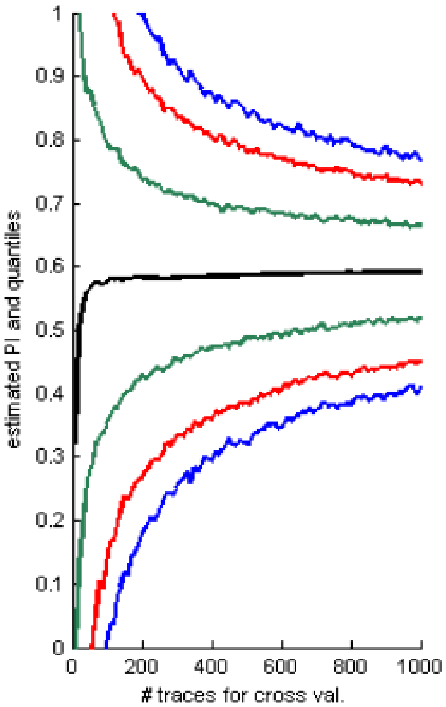


- Identified template attack with PI = 0.58
- No assumption errors for *N*=1000
- Estimation error ~ 0.11 at this point

Example





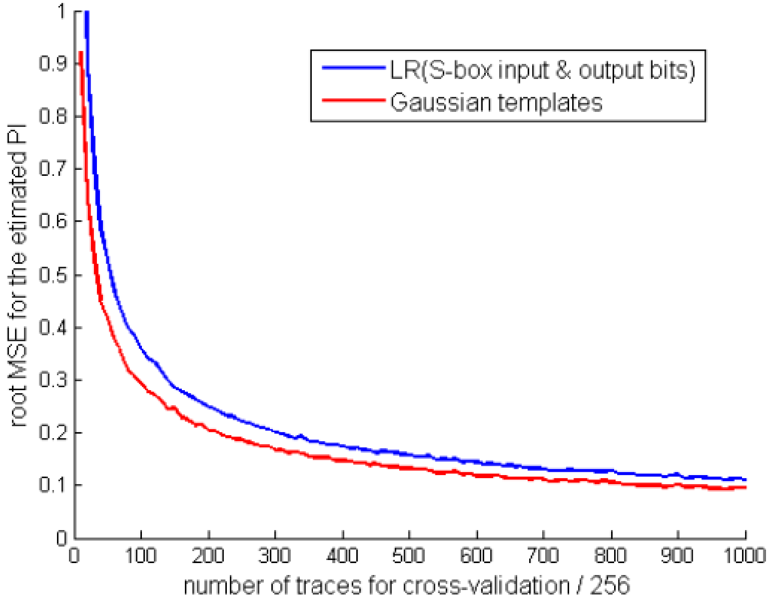


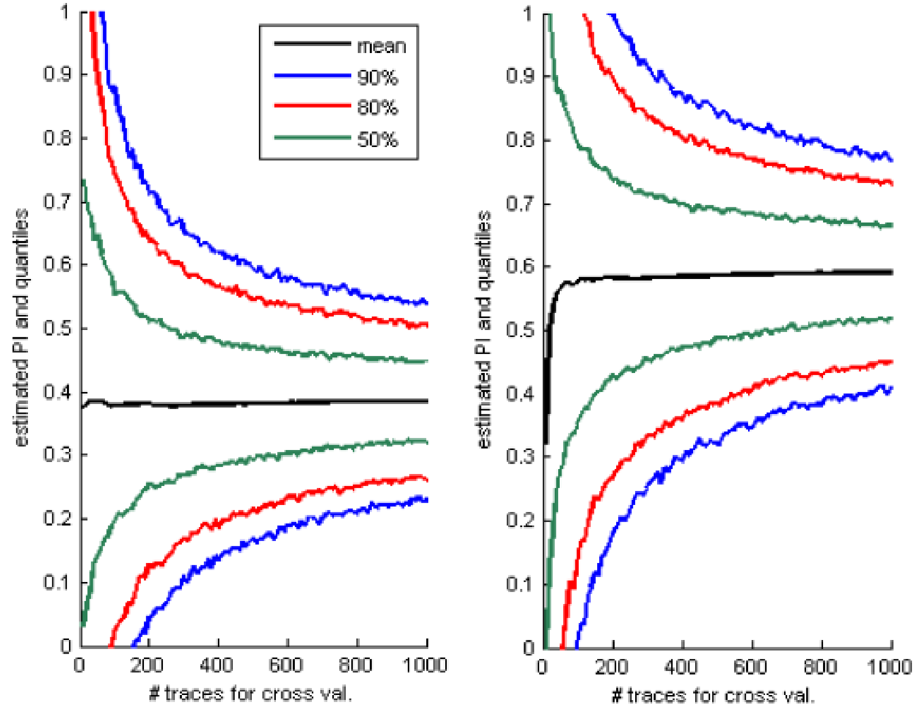
- Identified template attack with PI = 0.58
- No assumption errors for *N*=1000
- Estimation error ~ 0.11 at this point

=> With "low" confidence, no attack exist with PI>0.69

=> With "high" confidence, no attack exist with PI>0.80

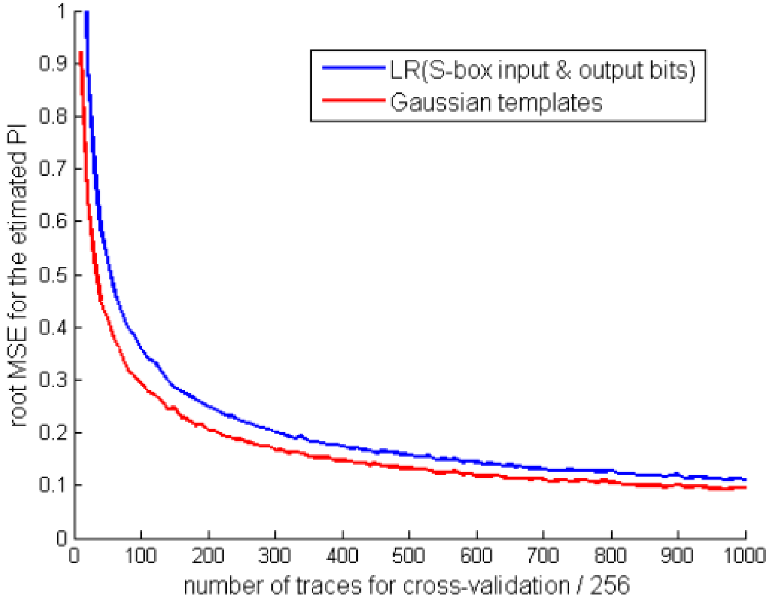
Example

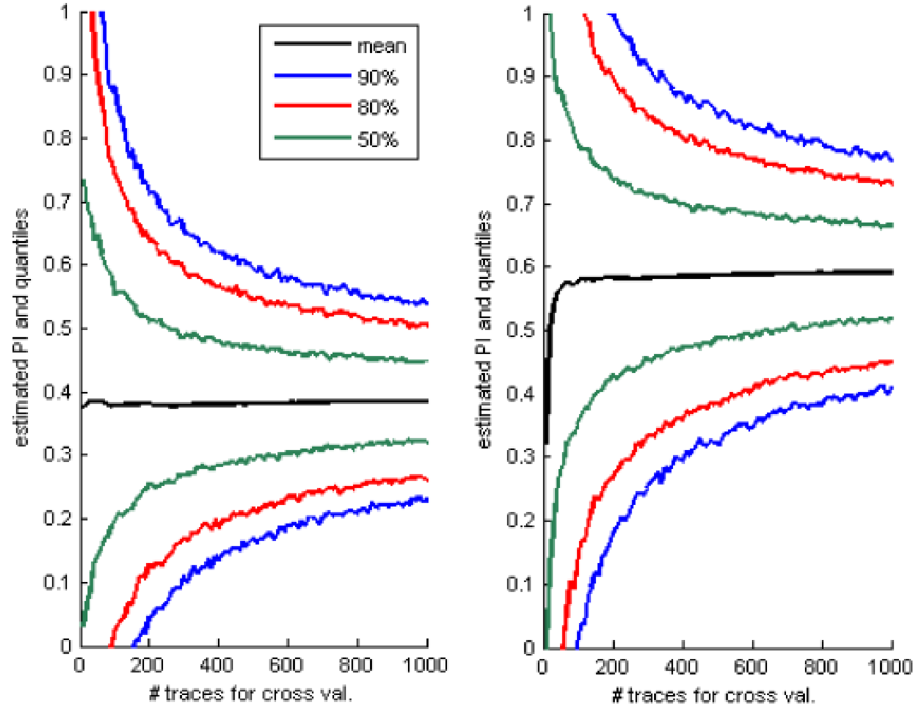




- Identified stochastic attack with PI = 0.38

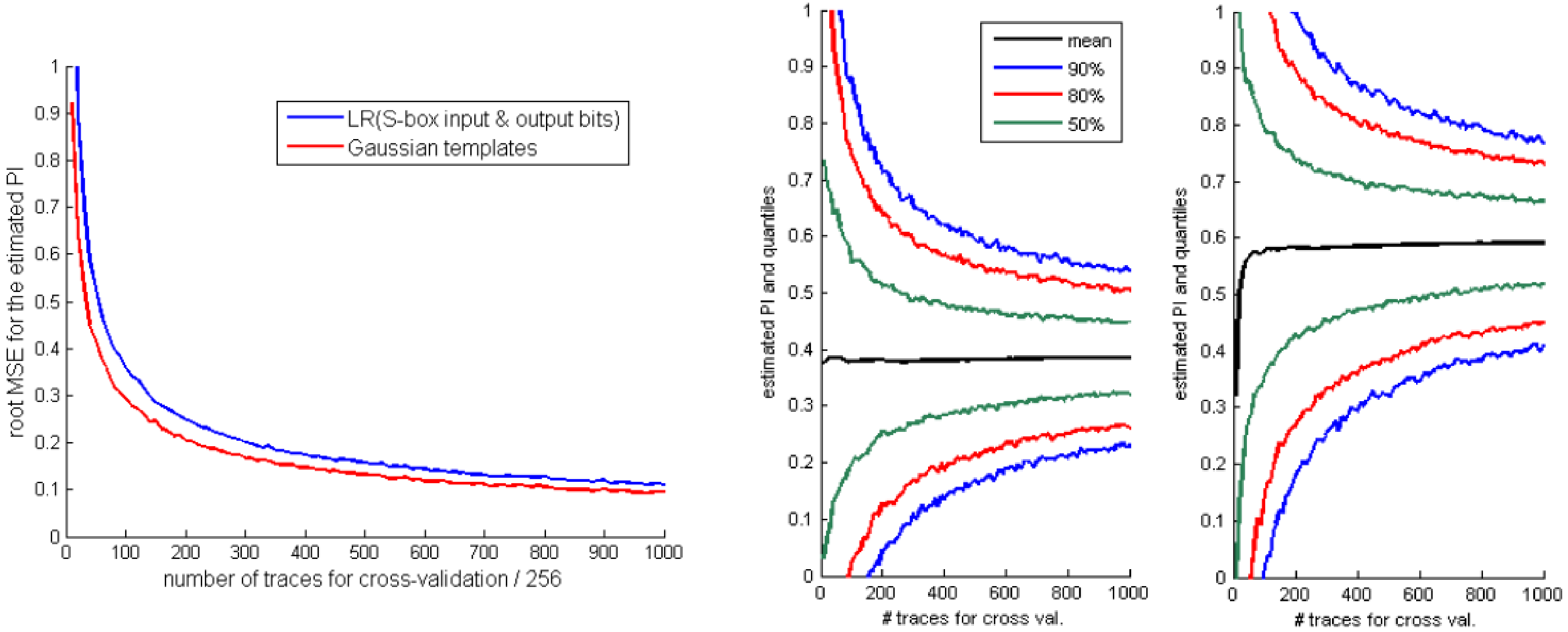
Example





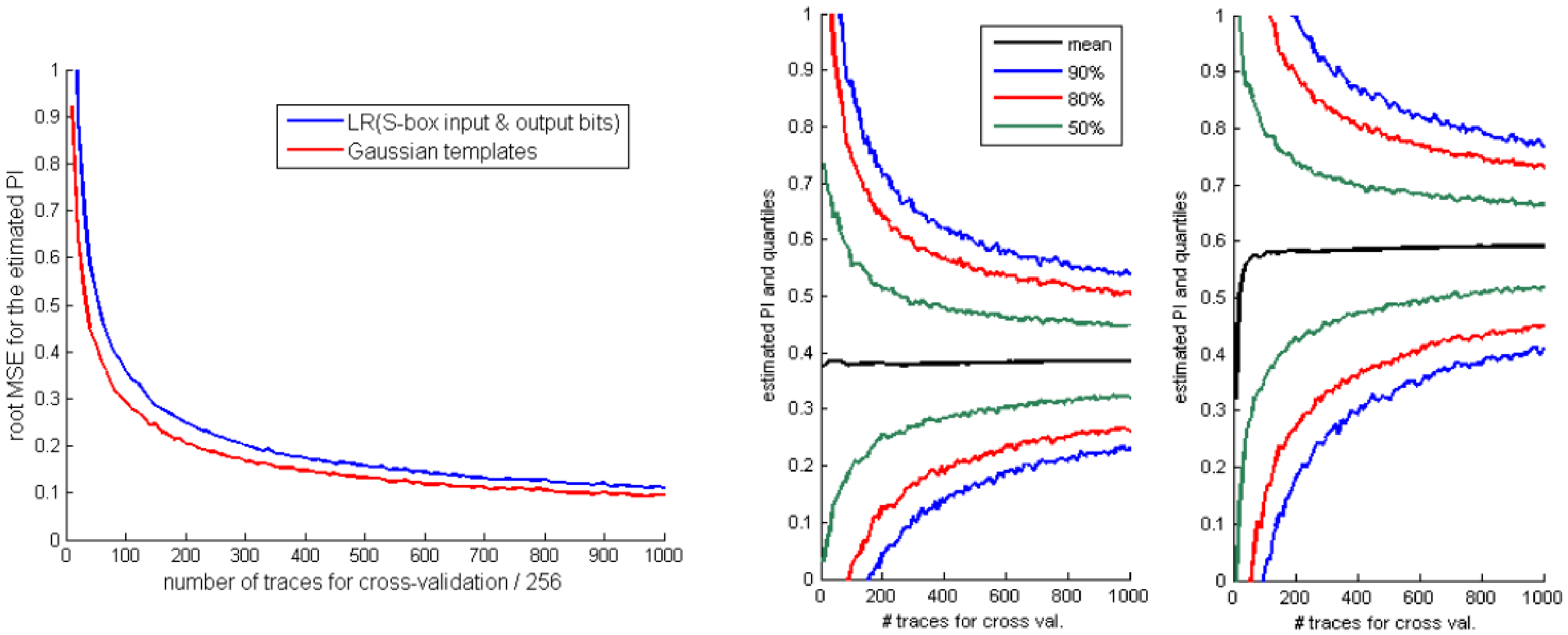
- Identified stochastic attack with PI = 0.38
- Assumption errors for N=100

Example



- Identified stochastic attack with PI = 0.38
- Assumption errors for N=100
- Estimation error ~ 0.29 at this point

Example

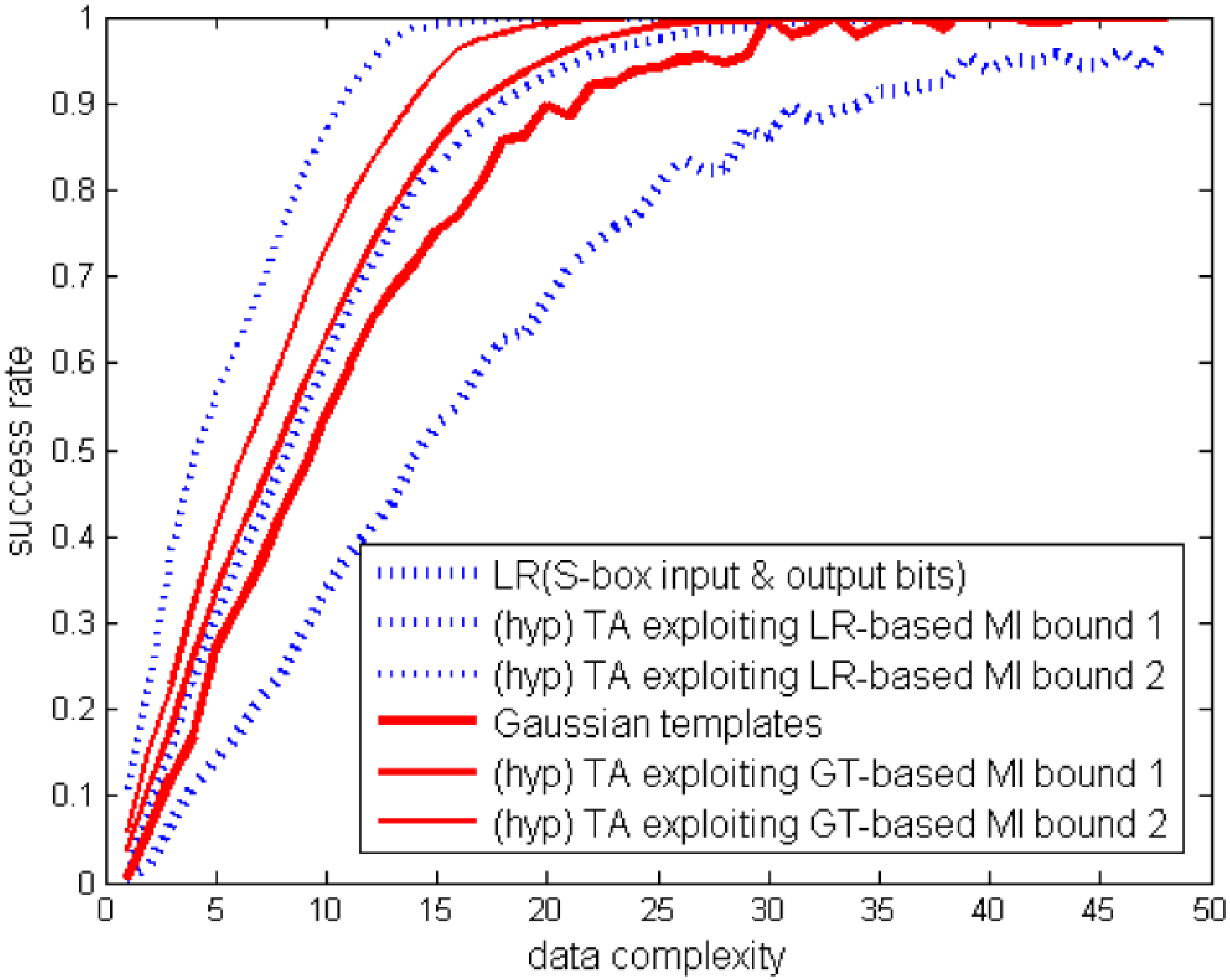


- Identified stochastic attack with PI = 0.38
- Assumption errors for *N*=100
- Estimation error ~ 0.29 at this point

=> With “low” confidence, no attack exist with PI>0.67

=> With “high” confidence, no attack exist with PI>0.96

Interpretation => success rates



Is that formally proven?

- No! (in fact there exist counterexamples)
- … but just as the PI <=> success rate connection

Is that formally proven?

- No! (in fact there exist counterexamples)
- … but just as the PI <=> success rate connection

- What can go wrong?

Is that formally proven?

- No! (in fact there exist counterexamples)
- … but just as the PI <=> success rate connection

- What can go wrong?
  - Heuristic optimization-based PDF estimation
    - (but seems OK with Gaussian templates and regression-based stochastic models)

Is that formally proven?

- No! (in fact there exist counterexamples)
- … but just as the PI <=> success rate connection

- What can go wrong?
  - Heuristic optimization-based PDF estimation
    - (but seems OK with Gaussian templates and regression-based stochastic models)
  - Very low noise levels (non-Gaussian PI estimates)
    - (but corresponds to less relevant scenarios)

Is that formally proven?

- No! (in fact there exist counterexamples)
- … but just as the PI <=> success rate connection

- What can go wrong?
  - Heuristic optimization-based PDF estimation
    - (but seems OK with Gaussian templates and regression-based stochastic models)
  - Very low noise levels (non-Gaussian PI estimates)
    - (but corresponds to less relevant scenarios)

- Good news: can be tested in simulations (since we know the true MI values in these cases!)

Outline

- The Eurocrypt 2009 framework revisited
- New results towards information leakage bounds
- Security analyzes and time complexity

Security analyzes

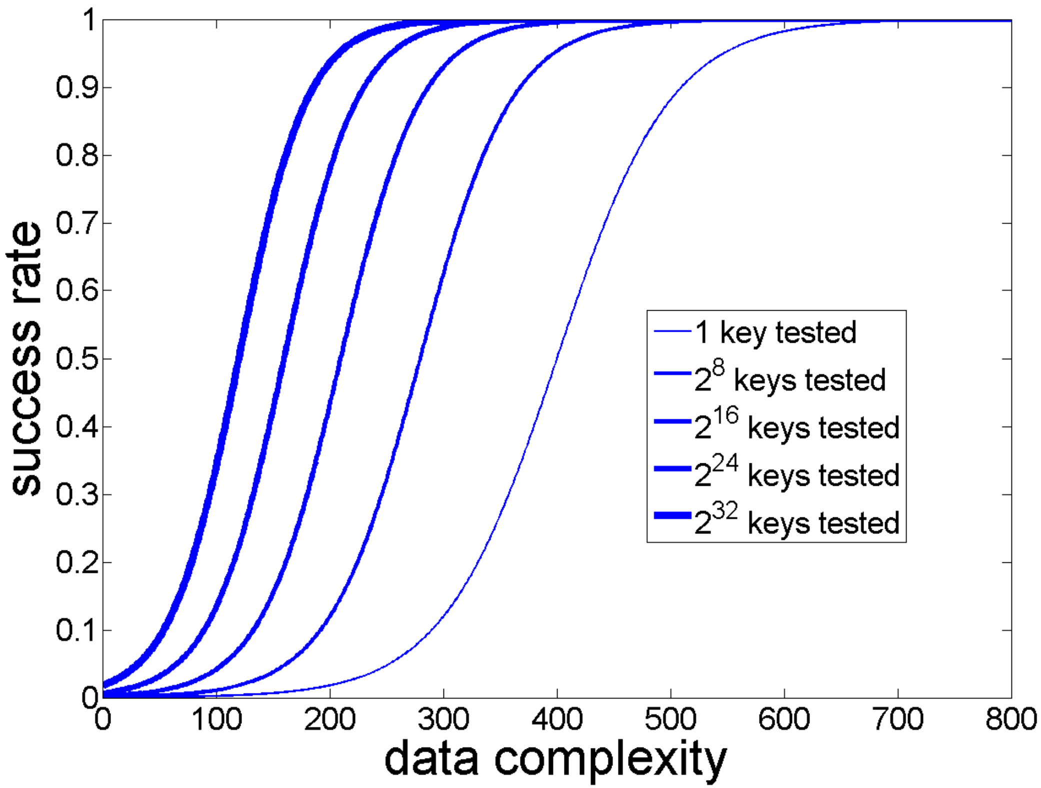
- Note: previous discussion mainly relates to the data complexity of side-channel attacks
- Time/memory complexity also matters

Security analyzes

- Note: previous discussion mainly relates to the data complexity of side-channel attacks
- Time/memory complexity also matters

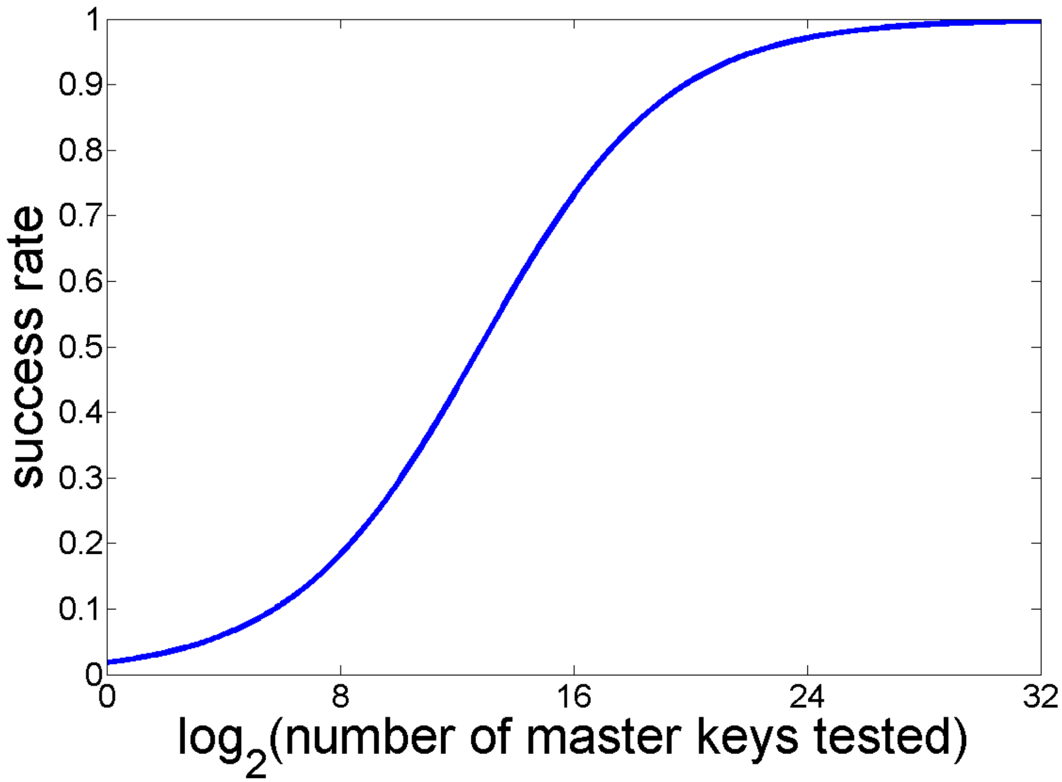
- In the context of “standard DPA”, the exploitation of computation is typically reflected by:
  - Key enumeration
  - Rank estimation

Key enumeration [9]



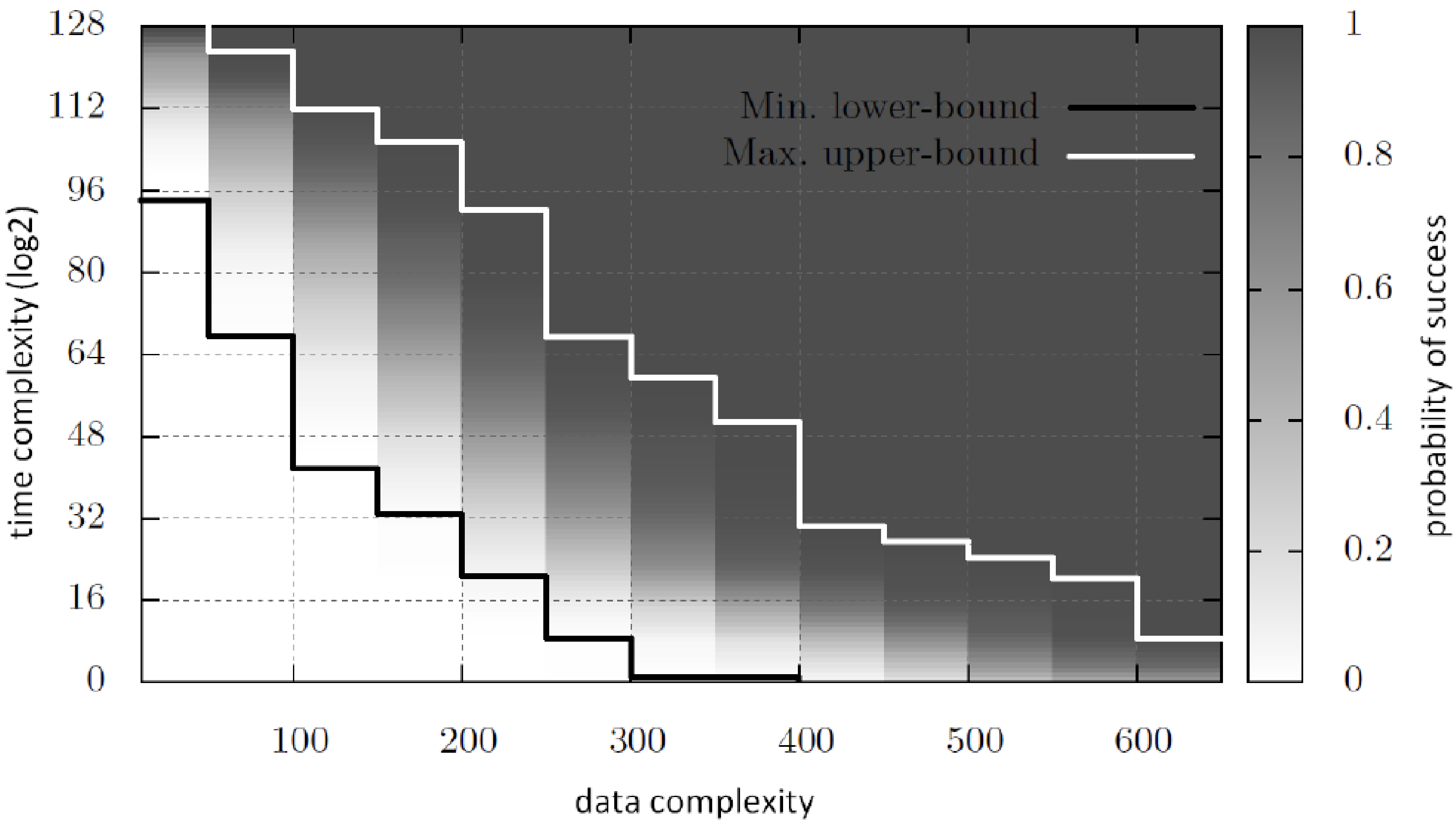
- Significant impact on the success rates!
- Very efficient attack tool (e.g. DPA contest)

Key enumeration [9]



- Missing data can always be traded for computations

Rank estimation [10]



- Evaluator's counterpart to key enumeration (the key must be known!) leading to complete security graphs

Conclusions

Main message:

- Possibility to “bound” the information leakage
- i.e. to know how far actual security evaluations computing the PI are from the true (unknown) MI
- Next: find meaningful examples/counterexamples

Conclusions

Main message:
- Possibility to “bound” the information leakage
- i.e. to know how far actual security evaluations computing the PI are from the true (unknown) MI
- Next: find meaningful examples/counterexamples

Cautionary note:
- Fair evaluations must consider both data and time
  - i.e. enumeration and rank estimation for DPA
  - But also algebraic side-channel attacks [11]

Bibliography

1. F.-X. Standaert, T.G. Malkin, M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, in the proceedings of Eurocrypt 2009, Lecture Notes in Computer Science, vol 5479, pp 443-461, Cologne, Germany, April 2009, Springer.

2. M. Renauld, F.-X. Standaert, N. Veyrat-Charvillon, D. Kamel, D. Flandre, *A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices*, in the proceedings of Eurocrypt 2011, Lecture Notes in Computer Science, vol 6632, pp 109-128, Tallinn, Estonia, May 2011, Springer.

3. F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, S. Mangard, *The World is Not Enough: Another Look on Second-Order DPA*, in the proceedings of Asiacrypt 2010, Lecture Notes in Computer Science, vol 6477, pp 112-129, Singapore, December 2010, Springer.

4. S. Mangard, E. Oswald, F.-X. Standaert, *One for All - All for One: Unifying Standard DPA Attacks*, in IET Information Security, vol 5, issue 2, pp 100-110, June 2011.

5. F.-X. Standaert, C. Archambeau, *Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages*, in the proceedings of CHES 2008, Lecture Notes in Computer Science, vol 5154, pp 411-425, Washington DC, USA, August 2008, Springer.

6. C. Whitnall, E. Oswald, F.-X. Standaert, *The Myth of Generic DPA... and the Magic of Learning*, Cryptology ePrint Archive, report 2012/038.

7. N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, F.-X. Standaert, *Shuffling Against Side-Channel Attacks: a Comprehensive Study with Cautionary Note*, in the proceedings of Asiacrypt 2012, Lecture Notes in Computer Science, vol 7658, pp 740-757, Bejing, China, December 2012, Springer.

8. F. Durvaux, M. Renauld, F.-X. Standaert, L. van Oldeneel tot Oldenzeel, N. Veyrat-Charvillon, *Efficient Removal of Random Delays from Embedded Software Implementations using Hidden Markov Models*, in the proceedings of CARDIS 2012, Lecture Notes in Computer Science, vol 7771, pp 123-140, Graz, Austria, November 2012, Springer.

9. N. Veyrat-Charvillon, B. Gerard, M. Renauld, F.-X. Standaert, *An optimal Key Enumeration Algorithm and its Application to Side-Channel Attacks*, in the proceedings of SAC 2012, Lecture Notes in Computer Science, vol 7707, pp 391-407, Windsor, Ontario, Canada, August 2012, Springer.

10. N. Veyrat-Charvillon, B. Gerard, F.-X. Standaert, *Security Evaluations Beyond Computing Power: How to Analyze Side-Channel Attacks you Cannot Mount?*, to appear in the proceedings of Eurocrypt 2013, Lecture Notes in Computer Science, vol 7881, pp 126-141, Athens, Greece, May 2013, Springer.

11. M. Renauld, F.-X. Standaert, *Algebraic Side-Channel Attacks*, in the proceedings of Inscrypt 2009, Lecture Notes in Computer Science, vol 6151, pp 393-410, Bejing, China, December 2009, Springer

THANKS

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