Making Masking Security Proofs Concrete

Or How to Evaluate the Security of any Leaking Device

A. Duc, S. Faust, ***F.-X. Standaert***

EPFL, Switzerland and UCL Crypto Group, Belgium

EUROCRYPT 2015, Sofia, Bulgaria

Outline: a collection of results…

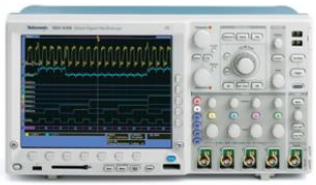
- Making masking proofs concrete
  - ≈ connecting two recent results
- Evaluating masking bounds' tightness
- Analyzing assumptions
  - Sufficiently noisy leakages
  - Independent leakages
- Quantifying computational security
- Putting things together

Outline: a collection of results…

- **Making masking proofs concrete**
  - $\approx$ **connecting two recent results**
- Evaluating masking bounds' tightness
- Analyzing assumptions
  - Sufficiently noisy leakages
  - Independent leakages
- Quantifying computational security
- Putting things together

EUROCRYPT 2014 masking proof

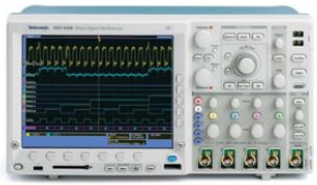
- 

+  → 

[KJJ98,…]

EUROCRYPT 2014 masking proof

- 

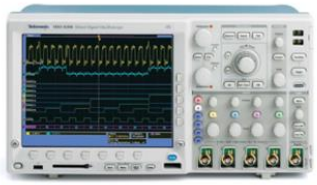
+  → 

[KJJ98,…]

circuit compiler

leakages “limited informativeness” and “independence” [ISW2004,…]

- 

+  → 

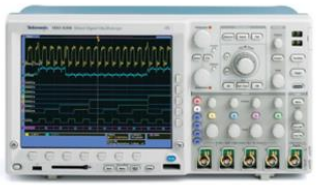
leakage-resilient

EUROCRYPT 2014 masking proof

- [DDF2014]: reduction from conceptually simple probing model to more realistic noisy leakages model [PR13]

EUROCRYPT 2009 evaluation framework

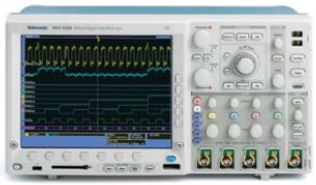
- 

+  → 

[SMY09]

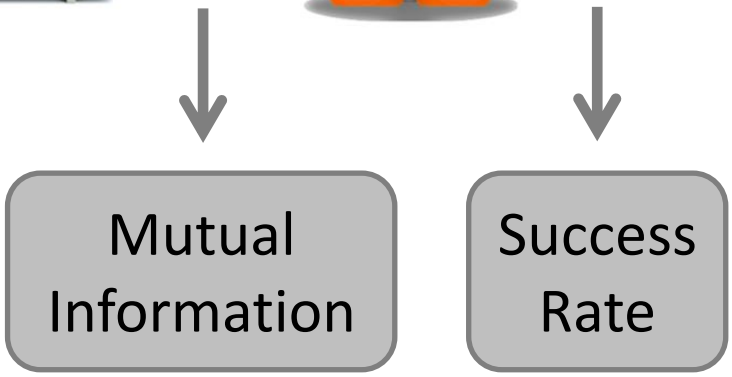
EUROCRYPT 2009 evaluation framework

- 

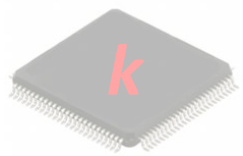
+  → 

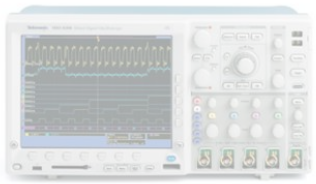
[SMY09]

- Evaluated with:



EUROCRYPT 2009 evaluation framework

- 

+  → 

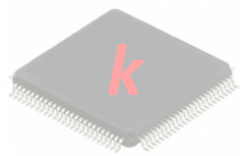
[SMY09]

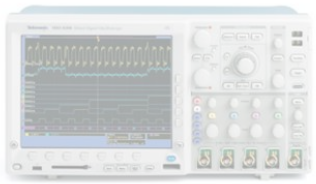
- Evaluated with:
  - Mutual Information
  - Success Rate

T1. Informative leakages => successful attacks ✓

T2. Link between mutual information and success rate ?

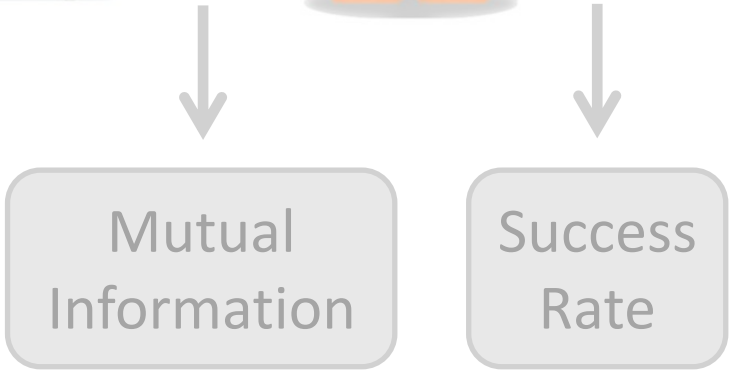
EUROCRYPT 2009 evaluation framework

- 

+  → 

[SMY09]

- Evaluated with:



T1. Informative leakages => successful attacks ✓

T2. Link between mutual information and success rate ?

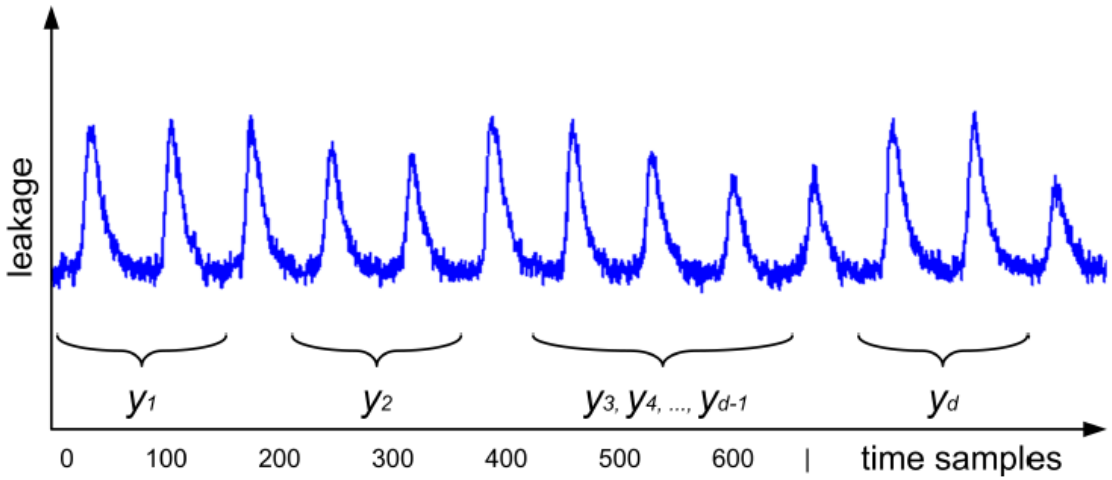
- Applied to many implementations/countermeasures

Connecting both results (example)

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

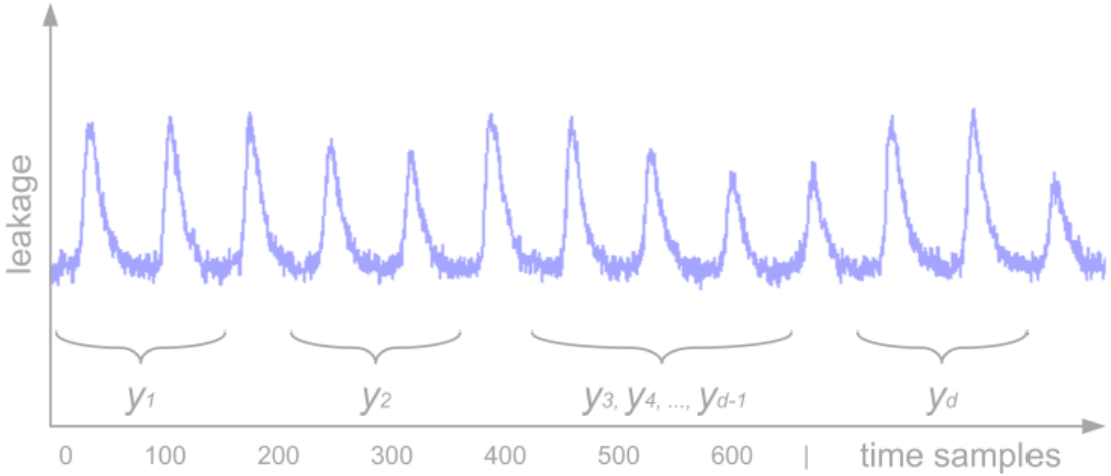
Connecting both results (example)

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



Connecting both results (example)

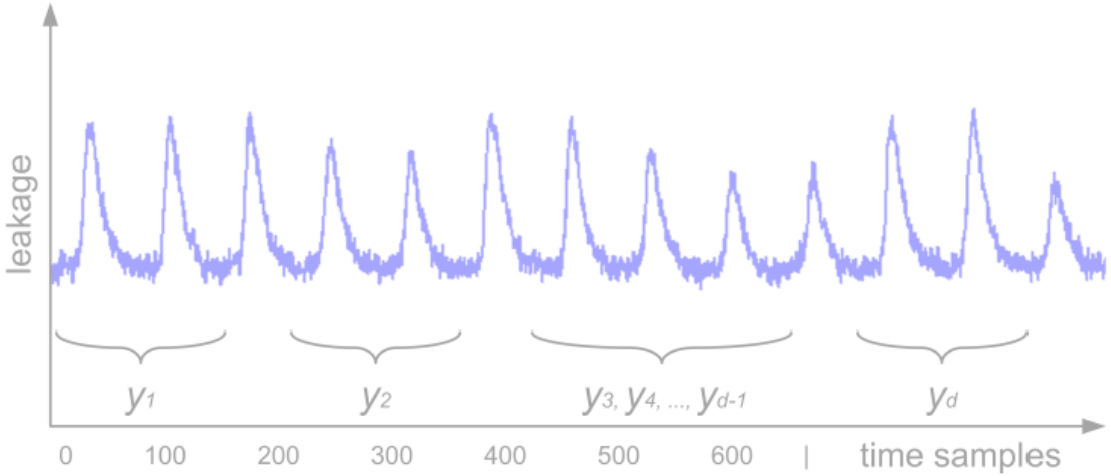
- Let $y = S(x \oplus k)$ be a leaking S-box computation
- Let $y = y_1 \oplus y_2 \oplus \cdots \oplus y_d$ be a sharing of y



- [DDF14] measures informativeness with $\mathrm{SD}(Y_i; Y_i | L_{Y_i})$
- [SMY09] measures informativeness with $\mathrm{MI}(Y_i | L_{Y_i})$

Connecting both results (example)

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



- [DDF14] measures informativeness with $\mathrm{SD}(Y_i; Y_i | L_{Y_i})$
- [SMY09] measures informativeness with $\mathrm{MI}(Y_i | L_{Y_i})$
- [D12] showed that $2 \cdot \mathrm{SD}(Y_i; Y_i | L_{Y_i})^2 \leq \mathrm{MI}(Y_i | L_{Y_i})$

Masking proof for an encoding revisited

- Assume leakage variables $L_{Y_i} = \mathrm{L}(Y_i, R_i)$ such that
  - $\mathrm{MI}(Y_i | L_{Y_i}) \leq \frac{2}{|F|^2}$ (with $|F|$ the field size)
  - The leakages of the shares are independent

Masking proof for an encoding revisited

- Assume leakage variables $L_{Y_i} = \mathrm{L}(Y_i, R_i)$ such that
  - $\mathrm{MI}(Y_i | L_{Y_i}) \leq \frac{2}{|F|^2}$ (with $|F|$ the field size)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements

Masking proof for an encoding revisited

- Assume leakage variables $L_{Y_i} = \mathrm{L}(Y_i, R_i)$ such that
  - $\mathrm{MI}(Y_i|L_{Y_i}) \leq \frac{2}{|F|^2}$ (with $|F|$ the field size)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements
- Then:

$$\mathrm{SR} \leq 1 - \left(1 - \left(|F| \sqrt{\frac{\mathrm{MI}(Y_i|L_{Y_i})}{2}}\right)^d\right)^m$$

Masking proof for an encoding revisited

- Assume leakage variables $L_{Y_i} = \mathrm{L}(Y_i, R_i)$ such that
  - $\mathrm{MI}(Y_i|L_{Y_i}) \leq \frac{2}{|F|^2}$ (with $|F|$ the field size)
  - The leakages of the shares are independent
- For a masking scheme with *d* shares
- And an adversary using *m* measurements
- Then:

$$\mathrm{SR} \leq 1 - \left(1 - \left(|F| \sqrt{\frac{\mathrm{MI}(Y_i|L_{Y_i})}{2}}\right)^d\right)^m$$

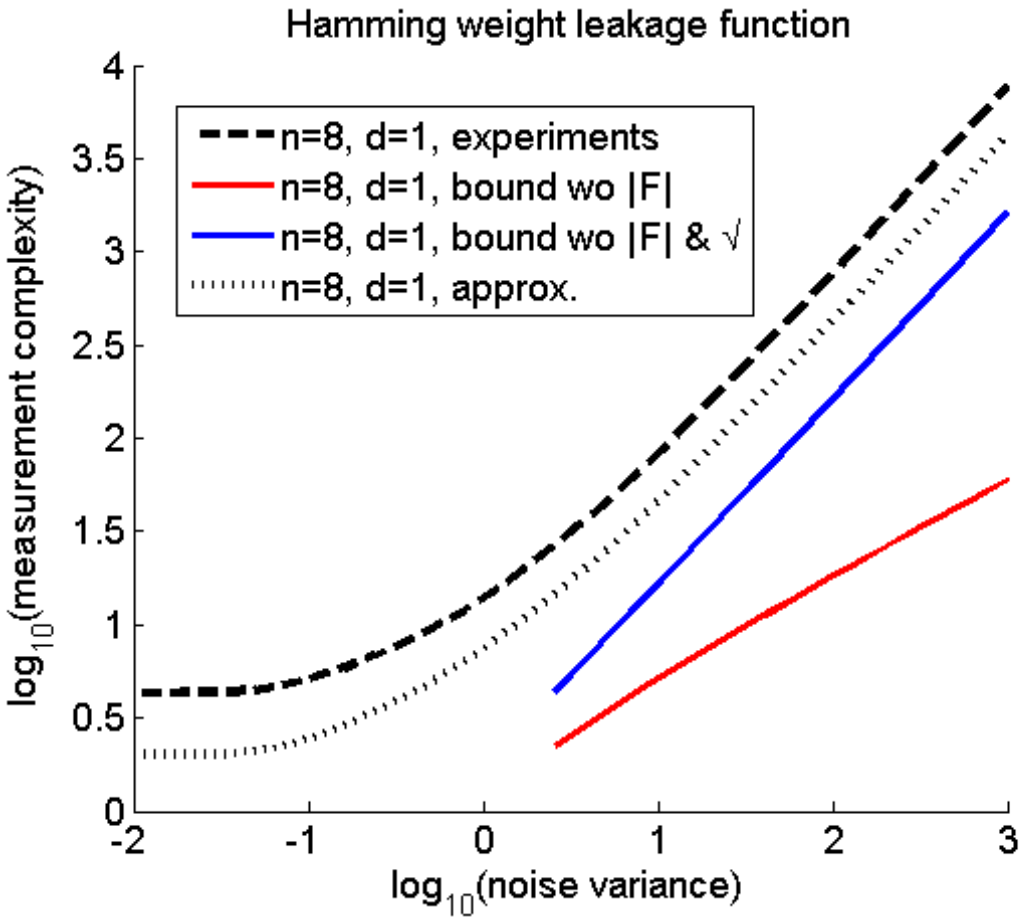
- **Which (for *d=1*) proves T2 in [SMY09]**
- We provide a bound for complete circuits in the paper

Outline: a collection of results…

- Making masking proofs concrete
  - ≈ connecting two recent results
- **Evaluating masking bounds' tightness**
- Analyzing assumptions
  - Sufficiently noisy leakages
  - Independent leakages
- Quantifying computational security
- Putting things together

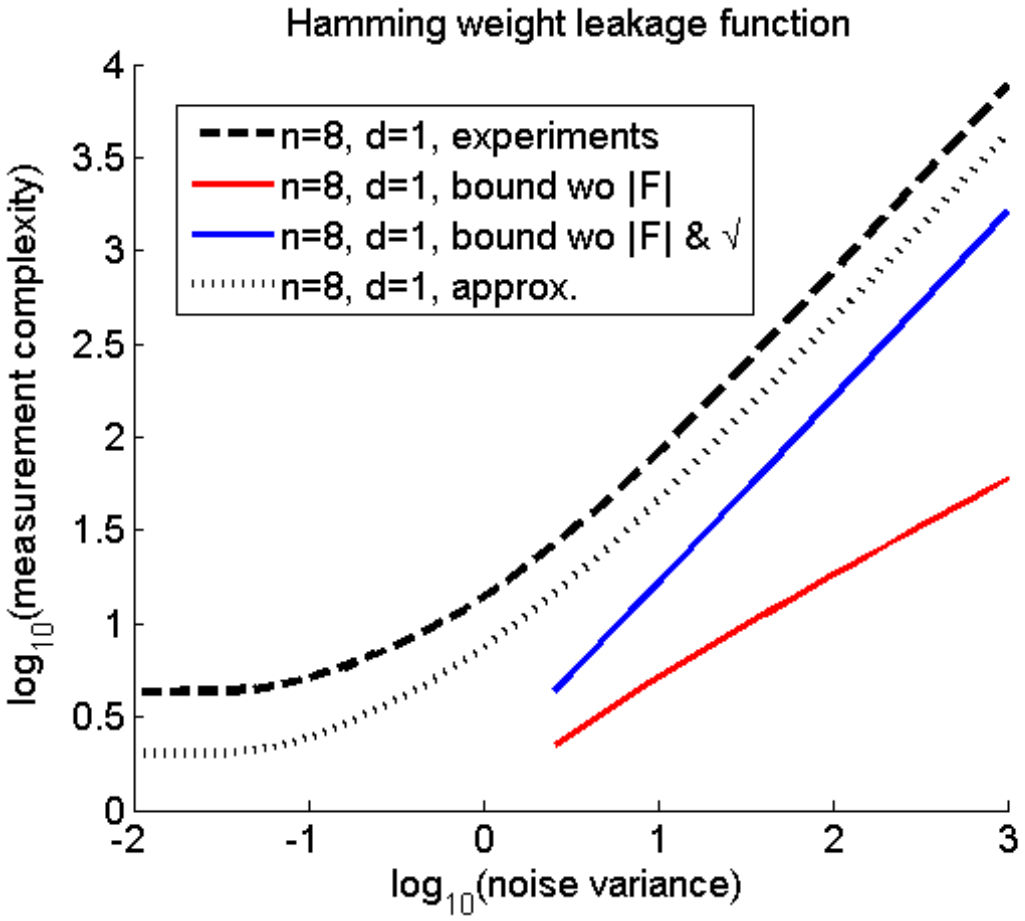
Case study: security of an encoding (I)

- $L_{Y_i} = \mathrm{HW}(Y_i) + N_i$ with N_i Gaussian-distributed



Case study: security of an encoding (I)

- $L_{Y_i} = \mathrm{HW}(Y_i) + N_i$ with N_i Gaussian-distributed



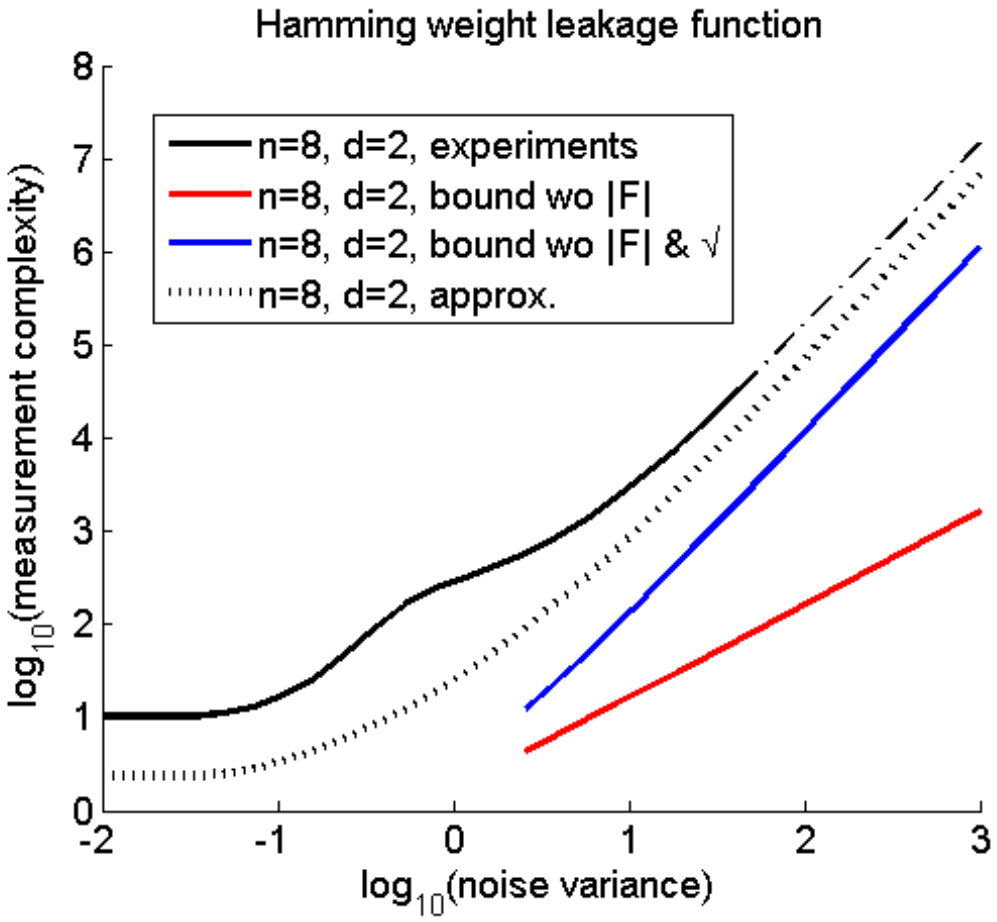
Unprotected case:

- $|F|$ factor ≈ artifact (proven in [DFS15])
- $\sqrt{\ }$ loss ≈ untight link between SD and MI
- [PR14] showed:

$$\mathrm{MI} \leq \frac{|F|}{\ln(2)} \cdot \mathrm{SD}$$

Case study: security of an encoding (II)

- $L_{Y_i} = \mathrm{HW}(Y_i) + N_i$ with N_i Gaussian-distributed



Masked case:

- Similar results

- " Slope intuition" (lowest moment of the distribution exploited)

Informal conjecture

- Under sufficiently noisy & independent leakages,

$$\mathrm{SR} \leq 1 - \left(1 - \left(|F| \sqrt{\frac{\mathrm{MI}(Y_i|L_{Y_i})}{2}}\right)^d\right)^m$$

  can be turned into: $\mathrm{SR} \leq 1 - \left(1 - \mathrm{MI}(Y_i|L_{Y_i})^d\right)^m$

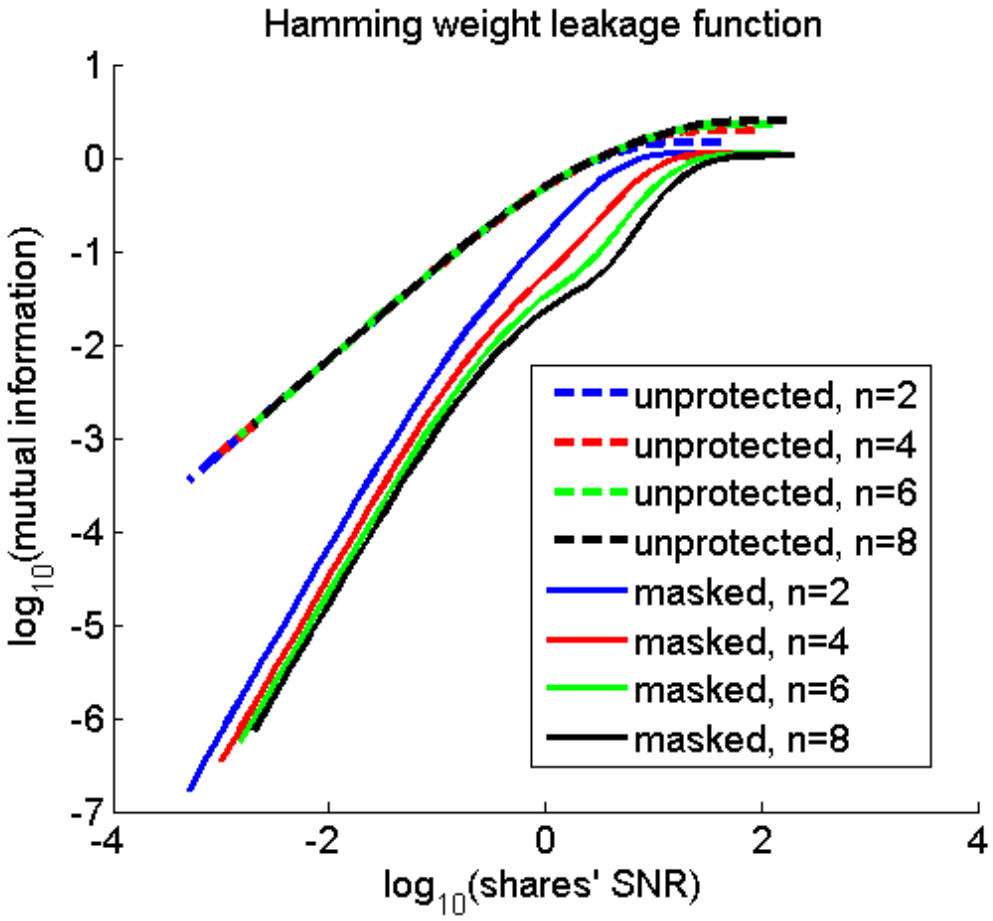
- And this is also expected to hold for complete circuits

Outline: a collection of results…

- Making masking proofs concrete
  - ≈ connecting two recent results
- Evaluating masking bounds' tightness
- **Analyzing assumptions**
  - **Sufficiently noisy leakages**
  - Independent leakages
- Quantifying computational security
- Putting things together

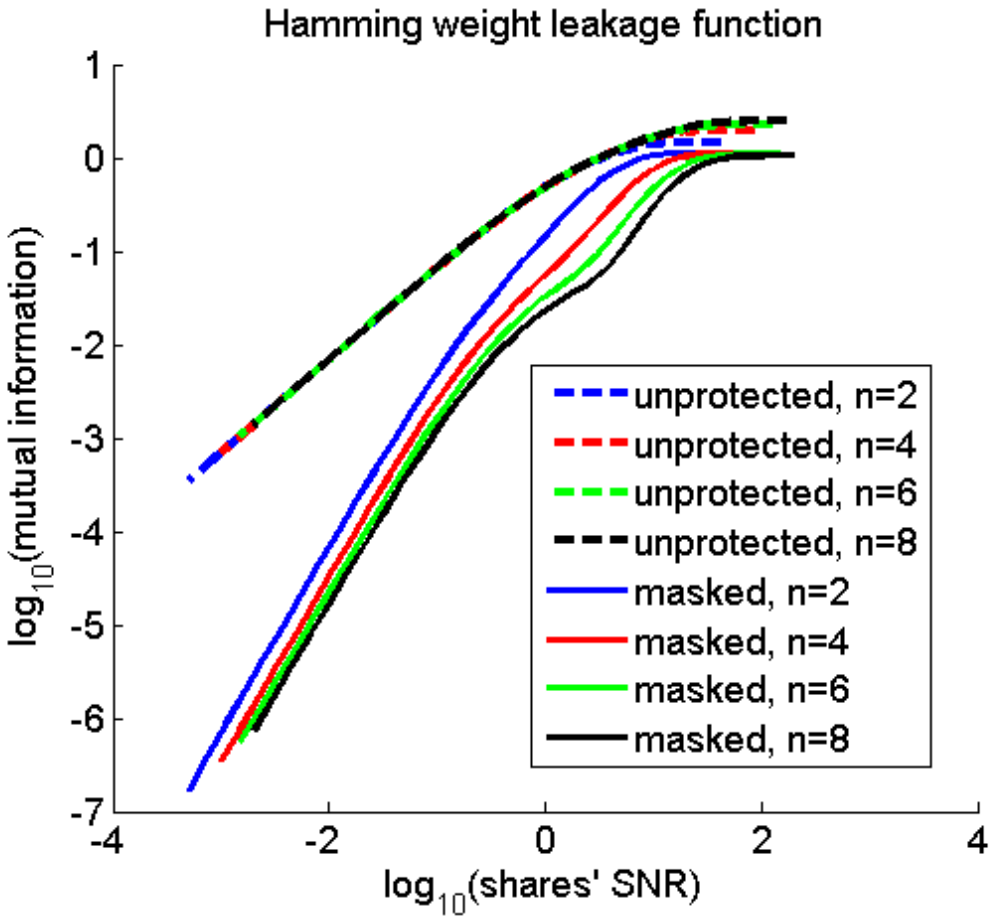
What is sufficiently noisy?

- In a simple univariate setting with Gaussian leakages, we have $\mathrm{MI}(Y_i|L_{Y_i}) \leq \frac{1}{2}\log(1+\mathrm{SNR})$ [Cover & Thomas]



What is sufficiently noisy?

- In a simple univariate setting with Gaussian leakages, we have $\mathrm{MI}(Y_i|L_{Y_i}) \leq \frac{1}{2}\log(1+\mathrm{SNR})$ [Cover & Thomas]



- " Slope intuition"

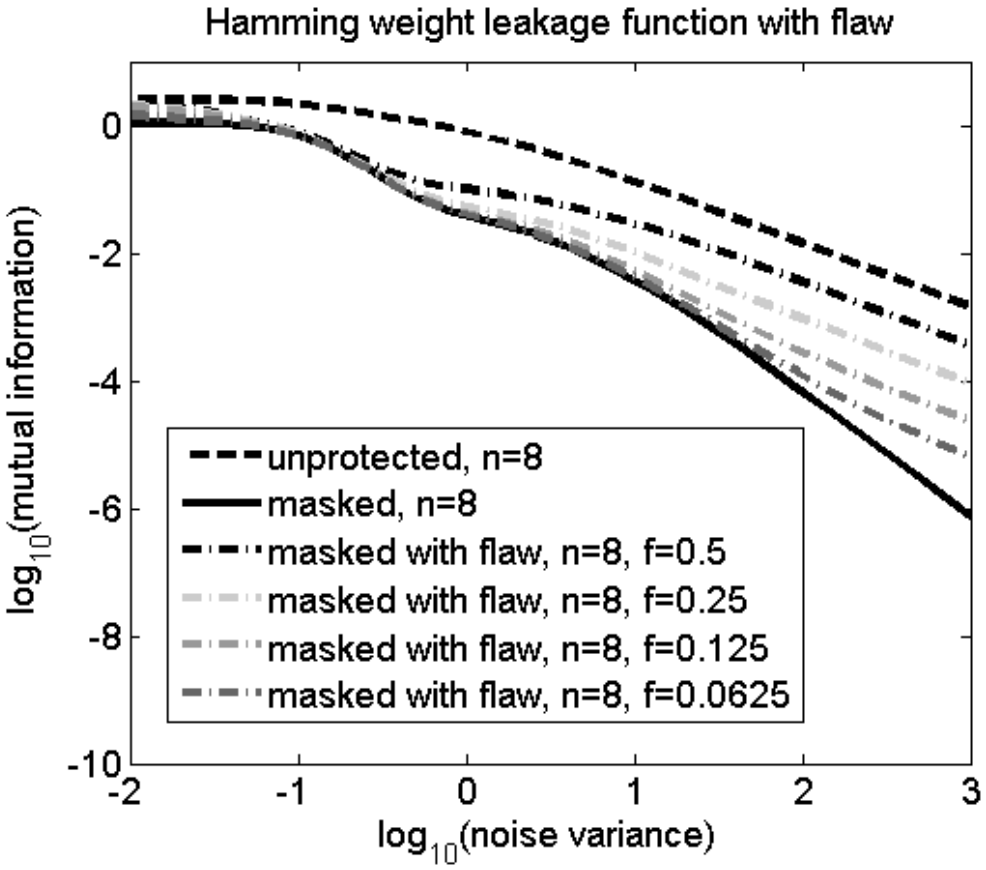
- MI is small enough when the signal becomes significantly smaller than the noise

Outline: a collection of results…

- Making masking proofs concrete
  - ≈ connecting two recent results
- Evaluating masking bounds' tightness
- **Analyzing assumptions**
  - Sufficiently noisy leakages
  - **Independent leakages**
- Quantifying computational security
- Putting things together

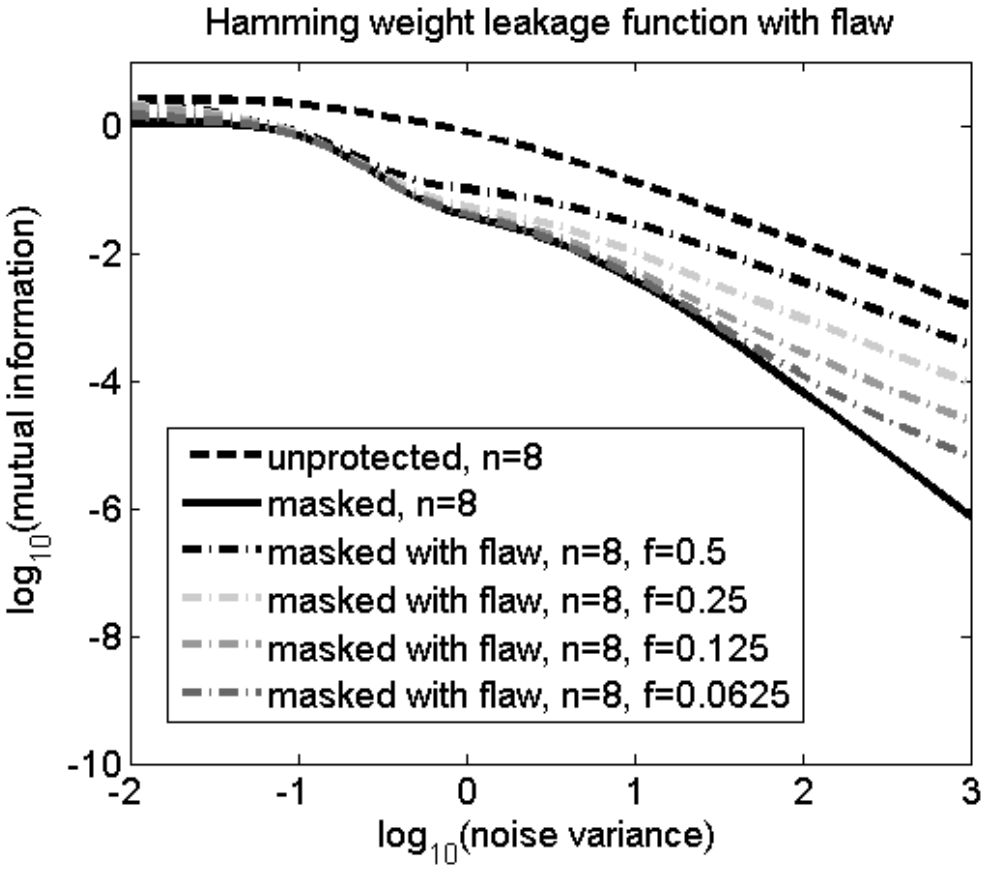
Implementation flaws?

- $L_{Y_1} = \mathrm{HW}(Y_1) + f \cdot \mathrm{HW}(Y_1 \oplus Y_2) + N_1$
- $L_{Y_2} = \mathrm{HW}(Y_2) + f \cdot \mathrm{HW}(Y_1 \oplus Y_2) + N_2$
- $f = 0$ means no flaw (independent leakages)



Implementation flaws?

- $L_{Y_1} = \mathrm{HW}(Y_1) + f \cdot \mathrm{HW}(Y_1 \oplus Y_2) + N_1$
- $L_{Y_2} = \mathrm{HW}(Y_2) + f \cdot \mathrm{HW}(Y_1 \oplus Y_2) + N_2$
- $f = 0$ means no flaw (independent leakages)



- Smaller flaws remain unnoticed until larger noises levels

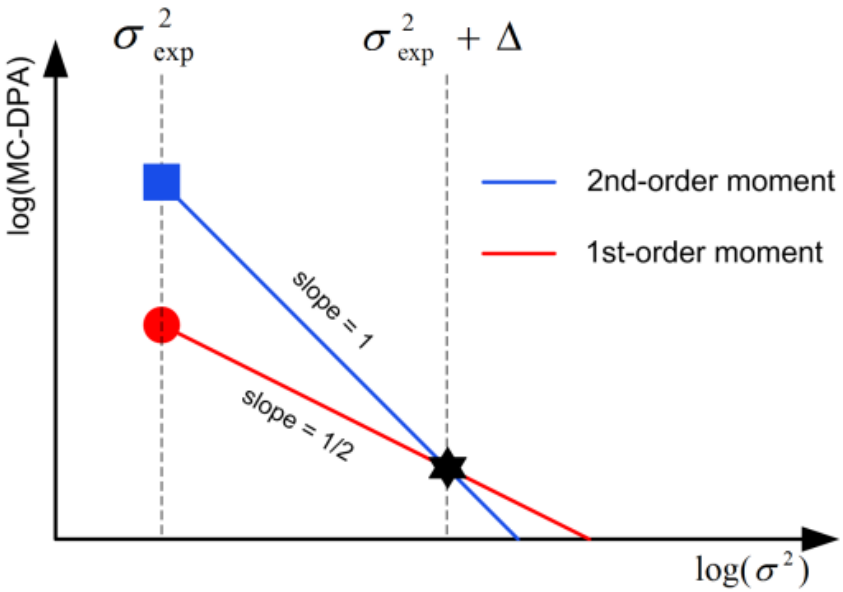
- Masking with non-independent leakages improves security until certain noise levels

- “ Slope intuition”

Evaluating flawed implementations

- For sufficiently noisy leakages (linear part of IT curves)

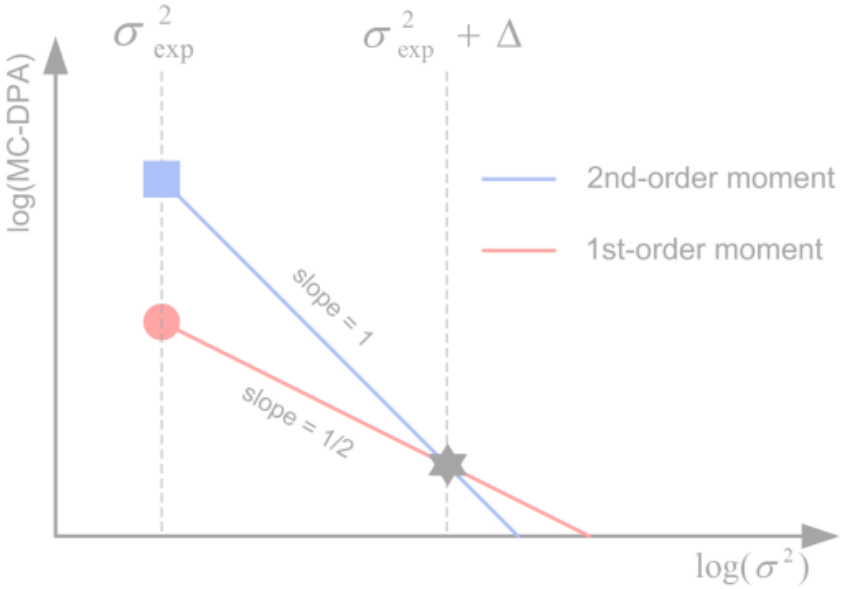
1. Evaluate the information “per moment”

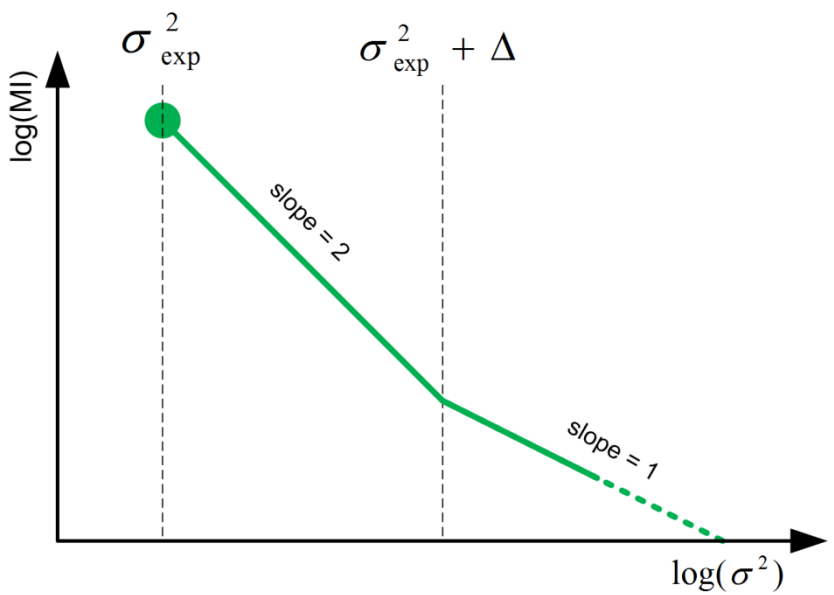


Evaluating flawed implementations

- For sufficiently noisy leakages (linear part of IT curves)

1. Evaluate the information “per moment”





2. Extrapolate the impact on the MI

Outline: a collection of results…

- Making masking proofs concrete
  - ≈ connecting two recent results
- Evaluating masking bounds' tightness
- Analyzing assumptions
  - Sufficiently noisy leakages
  - Independent leakages
- **Quantifying computational security**
- Putting things together

A motivating example

- Attacker 1 (≈ probing model)
  - Learns nothing with P=99/100
  - Learns everything with P=1/100
- Attacker 2 (≈ noisy leakage model)
  - Learns a set of 100 equally likely keys with P=1

A motivating example

- Attacker 1 (≈ probing model)
  - Learns nothing with P=99/100
  - Learns everything with P=1/100
- Attacker 2 (≈ noisy leakage model)
  - Learns a set of 100 equally likely keys with P=1
- Both attacks have the same SR
- But they highly differ w.r.t. enumeration

A motivating example

- Attacker 1 (≈ probing model)
  - Learns nothing with P=99/100
  - Learns everything with P=1/100
- Attacker 2 (≈ noisy leakage model)
  - Learns a set of 100 equally likely keys with P=1
- Both attacks have the same SR
- But they highly differ w.r.t. enumeration

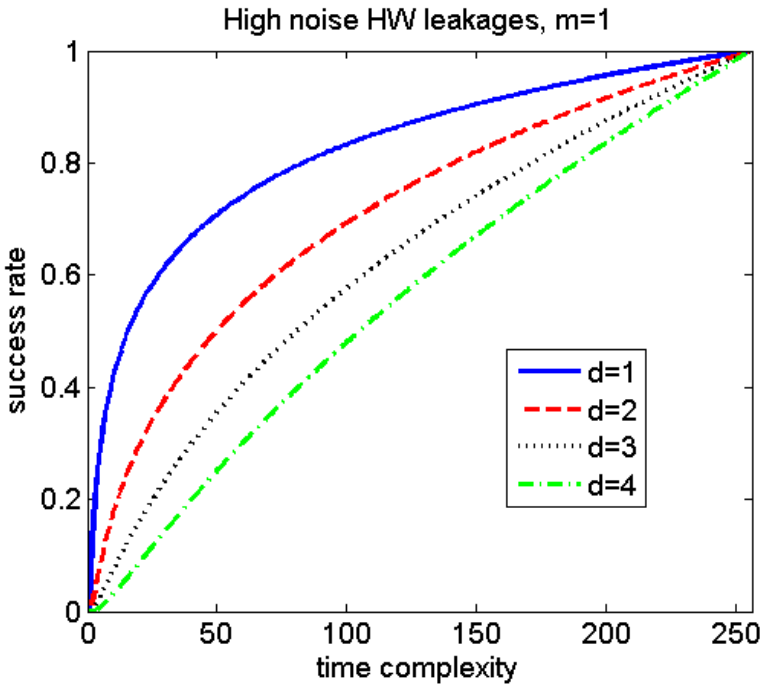
=> Despite asymptotically equivalent, the probing model is better for proofs and the noisy leakage model is better for concrete evaluations considering computing power

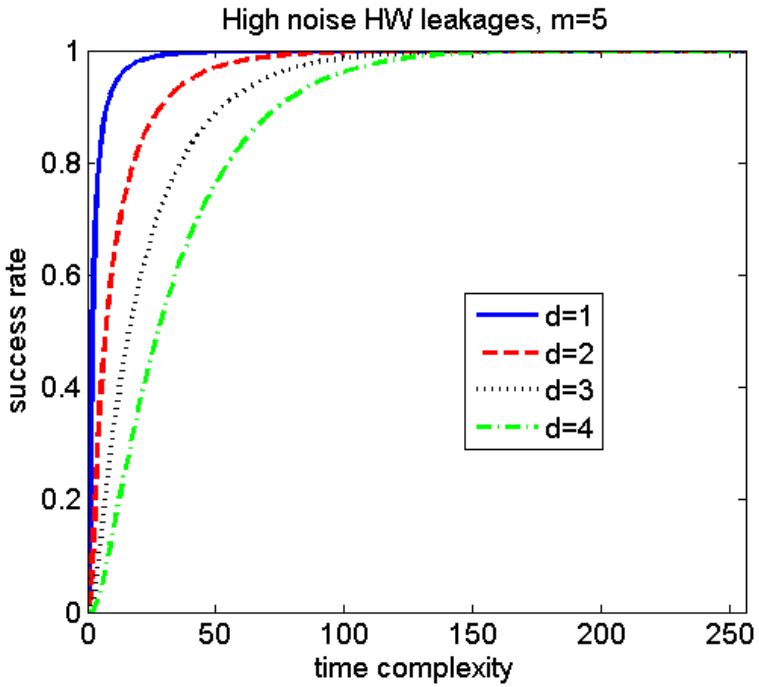
Per S-box analysis

- For each S-box, compute $\mathrm{MI}^{c}(Y_i; L_{Y_i})$,
  - i.e. the MI on the c most likely candidates

Per S-box analysis

- For each S-box, compute $\mathrm{MI}^{c}(Y_i; L_{Y_i})$,
  - i.e. the MI on the c most likely candidates
- Our theorems directly bound the c'th-order SR





Full key analysis (combining S-boxes)

- Similar to key rank estimation algorithms [VGS13]
- Problem can be written as:

$$\max_{c_1, \ldots, c_{n_s}} \quad \sum_{i=1}^{n_s} \log(\mathrm{SR}_i(m, c_i))$$

$$\text{subject to} \quad \sum_{i=1}^{n_s} \log(c_i) \leq \log(\beta)$$

- Non-linear knapsack / integer programming problem

Full key analysis (combining S-boxes)

- Similar to key rank estimation algorithms [VGS13]
- Problem can be written as:

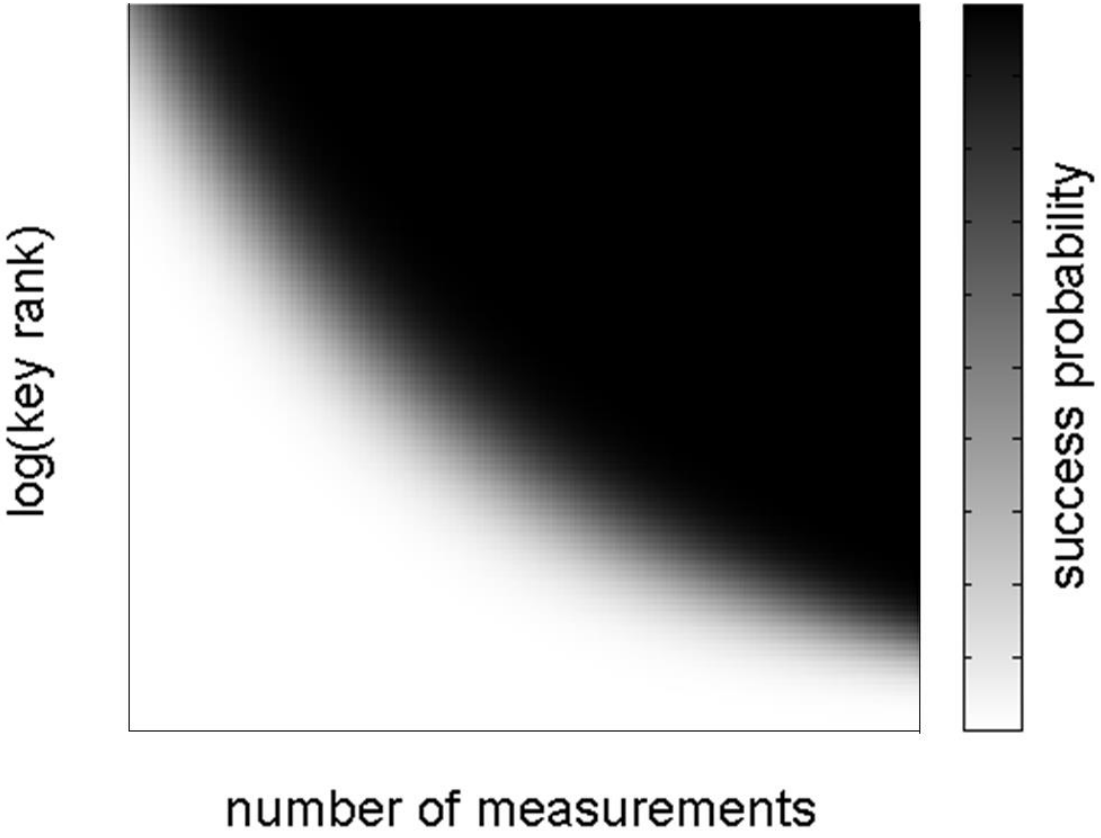
$$\max_{c_1, \ldots, c_{n_s}} \sum_{i=1}^{n_s} \log(\mathrm{SR}_i(m, c_i))$$

$$\text{subject to} \sum_{i=1}^{n_s} \log(c_i) \leq \log(\beta)$$

- Non-linear knapsack / integer programming problem
- Many solutions in the literature (to be investigated)
- Cheap heuristics work well (done in the paper)
- (Also works “online” by replacing SRs by subkey prob.)

Security graphs

- Just repeat the previous procedure for various *m*'s



Outline: a collection of results…

- Making masking proofs concrete
  - ≈ connecting two recent results
- Evaluating masking bounds' tightness
- Analyzing assumptions
  - Sufficiently noisy leakages
  - Independent leakages
- Quantifying computational security
- **Putting things together**

Why is this helpful? (Masked device evaluation)

- Existing approaches:

Why is this helpful? (Masked device evaluation)

- Existing approaches:

DPA1(N^d traces), DPA2(N^d traces), …

1. exhaustive evaluation

rank estimation #1
rank estimation #2
…
rank estimation #*Nr*

security graphs  , …

Why is this helpful? (Masked device evaluation)

- Existing approaches:

DPA1(N^d traces), DPA2(N^d traces), …

rank estimation #1
rank estimation #2
…
rank estimation #*Nr*

1. exhaustive evaluation

SNR(N traces)

S-boxes SR

2. specific shortcut [D+14,L+14]

security graphs  , …

Why is this helpful? (Masked device evaluation)

- Existing approaches:

DPA1(N^d traces), DPA2(N^d traces), …

rank estimation #1
rank estimation #2
…
rank estimation #*Nr*

1. exhaustive evaluation

security graphs  ,  , …

SNR(N traces)

S-boxes SR 

2. specific shortcut [D+14,L+14]

MI(N traces)

This paper: generic shortcut

worst-case bound for security graphs 

Conclusions & open problems

- This combination of tools allows significant reductions of the evaluation time in concrete setting
  - If you care about the worst-case security level
- Even in imperfect contexts (non-independent leakages)

Conclusions & open problems

- This combination of tools allows significant reductions of the evaluation time in concrete setting
  - If you care about the worst-case security level
- Even in imperfect contexts (non-independent leakages)

- Proofs can be useful (to estimate concrete security)

Conclusions & open problems

- This combination of tools allows significant reductions of the evaluation time in concrete setting
  - If you care about the worst-case security level
- Even in imperfect contexts (non-independent leakages)
- Proofs can be useful (to estimate concrete security)
- Open problems:
  - Investigating actual flaws (e.g. glitches)
  - Formalizing non-independent leakage
  - Maximum likelihood vs. weak maximum likelihood and nonlinear programming in rank estimation
  - Better (more secure and/or efficient) compilers

THANKS

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