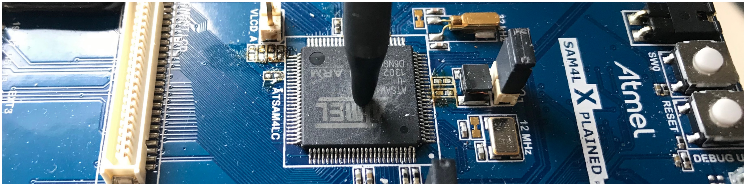


Side-channel Analysis of Cryptographic Implementations

Evaluation & Counter-Measures

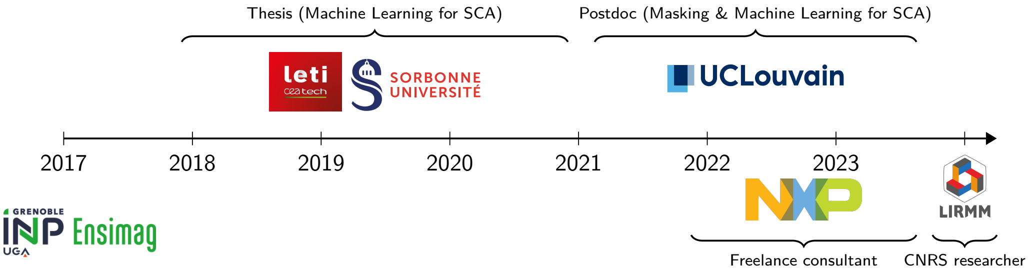
Loïc Masure (loic.masure@lirmm.fr)

PQ-TLS Summer School, Anglet, 20 – 21 June 2024

Acknowledgements

This work received funding from the France 2030 program, managed by the French National Research Agency under grant agreement No. ANR-22-PETQ-0008 PQ-TLS

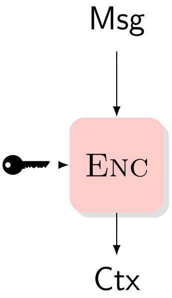
Who am I?


Thesis (Machine Learning for SCA)
Postdoc (Masking & Machine Learning for SCA)
2017
2018
2019
2020
2021
2022
2023
Freelance consultant
CNRS researcher

Agenda

Content

Context : Side-Channel Analysis (SCA)

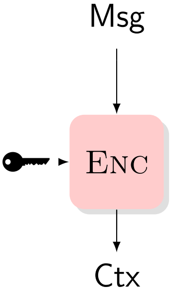


: N bits

Black-box cryptanalysis: 2^N

Context : Side-Channel Analysis (SCA)

"Cryptographic algorithms don't run on paper,

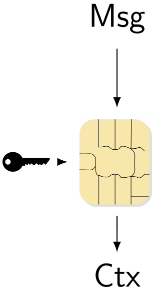


: N bits

Black-box cryptanalysis: 2^N

Context : Side-Channel Analysis (SCA)

"Cryptographic algorithms don't run on paper, they run on physical devices"

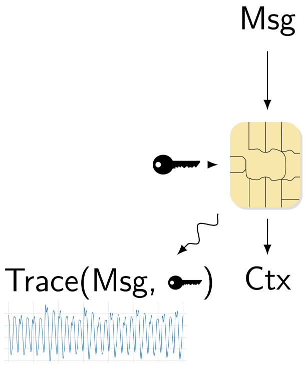


: N bits

Black-box cryptanalysis: 2^N

Context : Side-Channel Analysis (SCA)

"Cryptographic algorithms don't run on paper, they run on physical devices"



: N bits

Black-box cryptanalysis: 2^N

Side Channel = Unintended Communication Channel

Example: the Washington Pizza Index[1]

NEWS

CRUSTY D.C. VETERAN SAYS WAR IS NEAR

By Cox News Service
Chicago Tribune • Published: Jan 16, 1991 at 12:00 am

WASHINGTON — The pizza index indicates military action is imminent in the Persian Gulf, a Domino`s delivery official said Tuesday.

Record numbers of late-night pizzas have been delivered this week to the White House, Pentagon and State Department, said Frank Meeks, owner of several Washington-area Domino`s outlets.

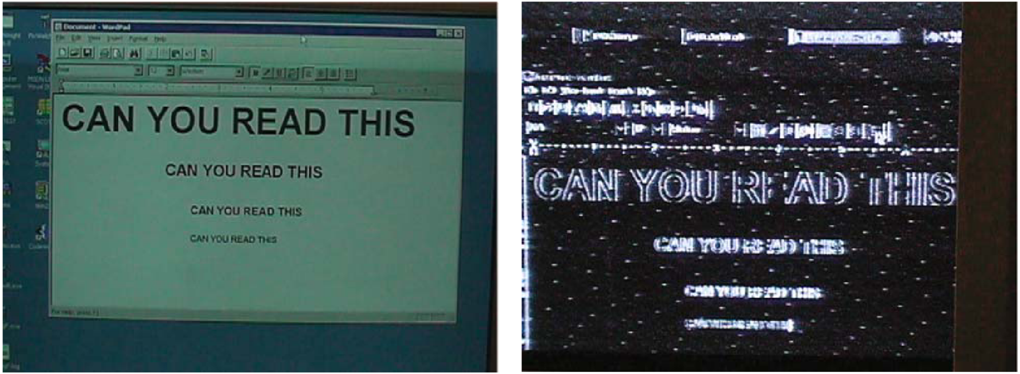
Similar order patterns came immediately before the invasions of Panama and Grenada, Meeks said.

The increase in pizza orders at key government buildings after 10 p.m. is ''very unusual,'' Meeks said. ''I don`t think they`re sitting around watching Redskins reruns.''

Figure: Chicago Tribune, **Jan. 16 1991**, the day before *Desert Storm* operation began.

[1] Reality questioned: `http://home.xnet.com/~warinner/pizzacites.html`

What is a Side Channel? A First Example



(a) A good old monitor (b) Reconstruction from EM field

Figure: An example from Koç, *Cryptographic Engineering*.

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square
Step 1: $\times M^{k_{N-1}}$ then square
...
Step i: $\times M^{k_{N-i}}$ then square

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square

Step 1: if $k_{N-1} = 1, \times M$ then square

...

Step i: if $k_{N-i} = 1, \times M$ then square

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square

Step 1: if $k_{N-1} = 1$, $\times M$ then square

...

Step i: if $k_{N-i} = 1$, $\times M$ then square

Op.	Guess
$\times M$	
square	
square	
square	
$\times M$	
square	
$\times M$	
square	

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square

Step 1: if $k_{N-1} = 1$, $\times M$ then square

...

Step i: if $k_{N-i} = 1$, $\times M$ then square

Op.	Guess
$\times M$	$k_N = 1$
square	
square	
square	
$\times M$	
square	
$\times M$	
square	

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square
Step 1: if $k_{N-1} = 1$, $\times M$ then square
...
Step i: if $k_{N-i} = 1$, $\times M$ then square

Op.	Guess
$\times M$	$k_N = 1$
square	
square	$k_{N-1} = 0$
square	
$\times M$	
square	
$\times M$	
square	

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square

Step 1: if $k_{N-1} = 1$, $\times M$ then square

...

Step i: if $k_{N-i} = 1$, $\times M$ then square

Op.	Guess
$\times M$	$k_N = 1$
square	
square	$k_{N-1} = 0$
square	$k_{N-2} = 0$
$\times M$	
square	
$\times M$	
square	

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square
Step 1: if $k_{N-1} = 1, \times M$ then square
...
Step i: if $k_{N-i} = 1, \times M$ then square

Op.	Guess
$\times M$	$k_N = 1$
square	
square	$k_{N-1} = 0$
square	$k_{N-2} = 0$
$\times M$	$k_{N-3} = 1$
square	
$\times M$	
square	

An Exemplary SCA on Crypto

RSA: Modular exponentiation over large ($\approx$ 2000-bit wide) integers

Square-and-Multiply:

$$M^k = M^{\sum_i k_i \cdot 2^i} = \prod_i (M^{k_i})^{2^i}$$

Step 0: M^{k_N} then square

Step 1: if $k_{N-1} = 1, \times M$ then square

...

Step i: if $k_{N-i} = 1, \times M$ then square

Op.	Guess
$\times M$ square	$k_N = 1$
square	$k_{N-1} = 0$
square	$k_{N-2} = 0$
$\times M$ square	$k_{N-3} = 1$
$\times M$ square	$k_{N-4} = 1$

Can you guess the key from the Oscilloscope?

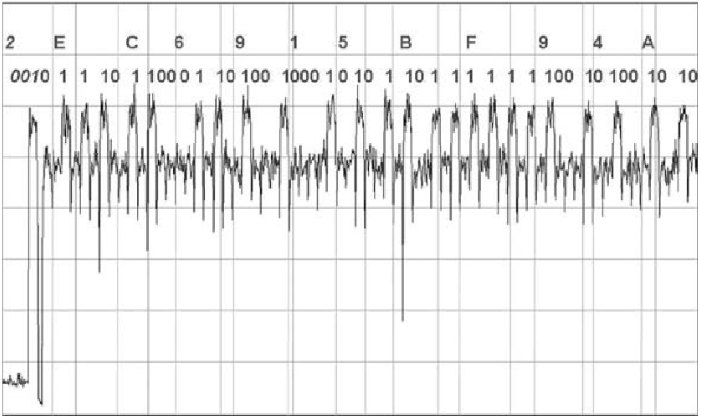
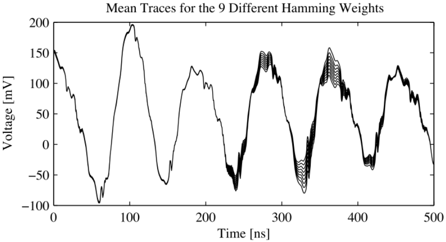


Figure: Power consumption. Illustration from Koç, *Cryptographic Engineering*

Power Analysis on Symmetric Key

Power consumption: each bit x_i of a data chunk X stored in a register[2]

$x_i = 0 \implies$ register voltage $= 0$
$x_i = 1 \implies$ register voltage $\neq 0$
Overall consumption of X is proportional to $\mathrm{hw}(X) = \sum_i x_i$
hw = Hamming Weight



[2] Mangard, Oswald, and Popp, *Power analysis attacks - revealing the secrets of smart cards*.

Practical Attack on AES, with Correlation

In practice: $\mathrm{L}_{\mathrm{P}} \propto \mathsf{hw}(k \oplus \mathrm{P}) + \mathcal{N}(0, \sigma^2)$

Practical Attack on AES, with Correlation

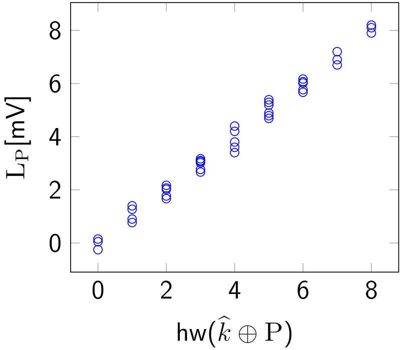
In practice: $\mathrm{L}_{\mathrm{P}} \propto \mathsf{hw}(k \oplus \mathrm{P}) + \mathcal{N}(0, \sigma^2)$
Distinguisher with a statistical test: for all key hypothesis $\widehat{k} = 0, 1, 2, \ldots$

Practical Attack on AES, with Correlation

In practice: $\mathrm{L_P} \propto \mathsf{hw}(k \oplus \mathrm{P}) + \mathcal{N}(0, \sigma^2)$
Distinguisher with a statistical test: for all key hypothesis $\widehat{k} = 0, 1, 2, \ldots$
If $\widehat{k} = k^\star$, then $\mathrm{L_P}$ should be highly correlated with $\mathsf{hw}(\widehat{k} \oplus \mathrm{P})$

$$\rho_{\widehat{k}} = \frac{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Cov}}\left(\mathrm{L_P}, \mathsf{hw}(\widehat{k} \oplus \mathrm{P})\right)}{\sqrt{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Var}}\left(\mathrm{L_P}\right)} \cdot \sqrt{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Var}}\left(\mathsf{hw}(\widehat{k} \oplus \mathrm{P})\right)}} \approx \pm 1$$

$\widehat{k} = k^\star$

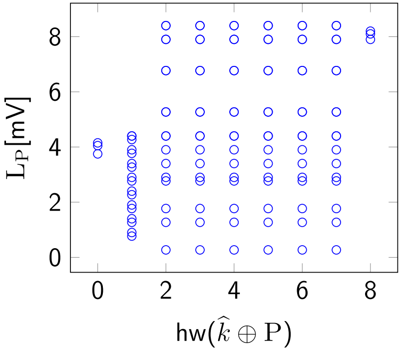


Practical Attack on AES, with Correlation

In practice: $\mathrm{L_P} \propto \mathsf{hw}(k \oplus \mathrm{P}) + \mathcal{N}(0, \sigma^2)$
Distinguisher with a statistical test: for all key hypothesis $\widehat{k} = 0, 1, 2, \ldots$
If $\widehat{k} \neq k^\star$, then $\mathrm{L_P}$ should be poorly correlated with $\mathsf{hw}(\widehat{k} \oplus \mathrm{P})$

$\widehat{k} \neq k^\star$

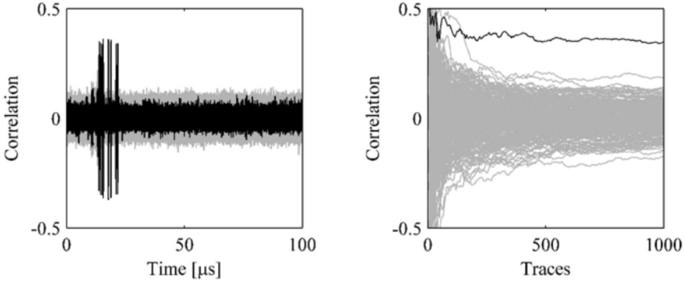
$$\rho_{\widehat{k}} = \frac{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Cov}}\left(\mathrm{L_P}, \mathsf{hw}(\widehat{k} \oplus \mathrm{P})\right)}{\sqrt{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Var}}\left(\mathrm{L_P}\right)} \cdot \sqrt{\underset{\mathrm{P},\mathcal{N}}{\mathsf{Var}}\left(\mathsf{hw}(\widehat{k} \oplus \mathrm{P})\right)}} \ll \pm 1$$



Power Analysis on Symmetric Key

Power consumption: each bit x_i of a data chunk X stored in a register[3]

Key guessed by a statistical test leveraging the correlation between the Hamming weight of data and the power consumption



[3] Mangard, Oswald, and Popp, *Power analysis attacks - revealing the secrets of smart cards*.

It's Demo Time

Application of the Correlation Attack on a `ChipWhisperer`

`CW` = Target device (8-bit MCU) + Oscilloscope

It's Demo Time

Application of the Correlation Attack on a `ChipWhisperer`
`CW` = Target device (8-bit MCU) + Oscilloscope

It's Demo Time

Application of the Correlation Attack on a `ChipWhisperer`
`CW` = Target device (8-bit MCU) + Oscilloscope

- **Q1**: What is the attack complexity ?

It's Demo Time

Application of the Correlation Attack on a `ChipWhisperer`
`CW` = Target device (8-bit MCU) + Oscilloscope

- **Q1**: What is the attack complexity ?

  One n-bit key chunk: $\mathcal{O}(2^n)$

It's Demo Time

Application of the Correlation Attack on a `ChipWhisperer`

`CW` = Target device (8-bit MCU) + Oscilloscope

- **Q1**: What is the attack complexity ?

  One n-bit key chunk: $\mathcal{O}\left(2^n\right)$

  N-bit full key: divide-and-conquer $\implies \mathcal{O}\left(\frac{N}{n} \cdot 2^n\right) \approx$ "quantum" break

Counter-Measure

Q2: Can you find a simple counter-measure for this attack ?

[4] Rivain, Prouff, and Doget, "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers".

[5] Coron and Kizhvatov, "An Efficient Method for Random Delay Generation in Embedded Software".

Counter-Measure

Q2: Can you find a simple counter-measure for this attack ?

→ Shuffling t independent operations (*e.g.*, ARK or SubBytes)[4] or inserting t dummy operations[5]

[4] Rivain, Prouff, and Doget, "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers".

[5] Coron and Kizhvatov, "An Efficient Method for Random Delay Generation in Embedded Software".

Counter-Measure

Q2: Can you find a simple counter-measure for this attack ?

→ Shuffling t independent operations (*e.g.*, ARK or SubBytes)[4] or inserting t dummy operations[5]

→ Multiplies data complexity by t^2: can you prove why ?

[4] Rivain, Prouff, and Doget, "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers".

[5] Coron and Kizhvatov, "An Efficient Method for Random Delay Generation in Embedded Software".

Counter-Measure

Q2: Can you find a simple counter-measure for this attack ?

→ Shuffling t independent operations (*e.g.*, ARK or SubBytes)[4] or inserting t dummy operations[5]

→ Multiplies data complexity by t^2: can you prove why ?

$$\rho_{\text{shuffling}} = \frac{\rho_0}{t}$$

Data complexity: $N_a \propto \frac{1}{\rho^2}$

[4] Rivain, Prouff, and Doget, "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers".

[5] Coron and Kizhvatov, "An Efficient Method for Random Delay Generation in Embedded Software".

Counter-Measure

Q2: Can you find a simple counter-measure for this attack ?

→ Shuffling t independent operations (*e.g.*, ARK or SubBytes)[4] or inserting t dummy operations[5]

→ Multiplies data complexity by t^2: can you prove why ?

$$\rho_{\text{shuffling}} = \frac{\rho_0}{t}$$

Data complexity: $N_a \propto \frac{1}{\rho^2}$

→ What can the adversary do? *Integrated* attack: $\rho_{\text{shuffling,integrated}} = \frac{\rho}{\sqrt{t}}$

[4] Rivain, Prouff, and Doget, "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers".

[5] Coron and Kizhvatov, "An Efficient Method for Random Delay Generation in Embedded Software".

Contradicting Goals

For correlation attacks, we usually target AddRoundKey or the SubBytes (bijective between each other)

- **Q3**: What's the "best" target ?

→ When targeting hw($k \oplus \mathrm{P}$), $k^\star$ and $\overline{k^\star}$ undistinguishable: *ghost* peaks

→ Problem solved when targeting hw(Sbox[$k \oplus \mathrm{P}$])

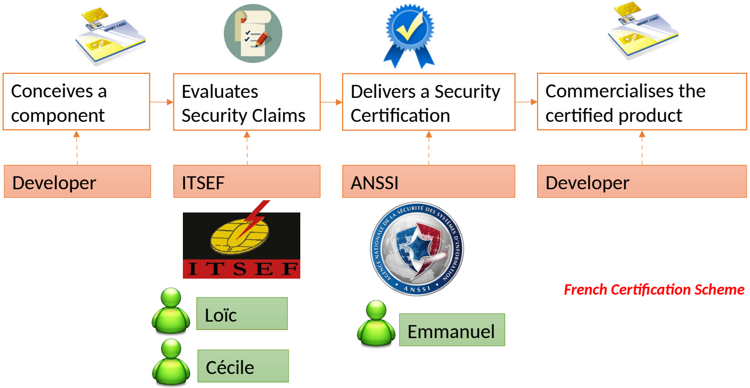
Contradicting goal: Sbox brings confusion to thwart cryptanalysis, but helps side-channel analysis[6]

[6]Prouff, "DPA Attacks and S-Boxes".

Content

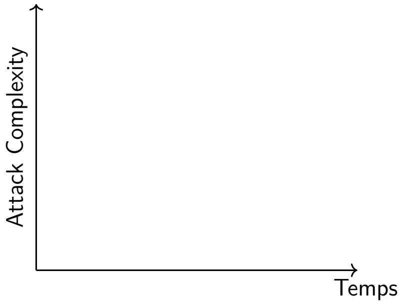
Content

Certification against SCA



Security graded w.r.t. attack complexity in terms of human, material, and financial means

Evaluate Security against Side-Channel Attacks

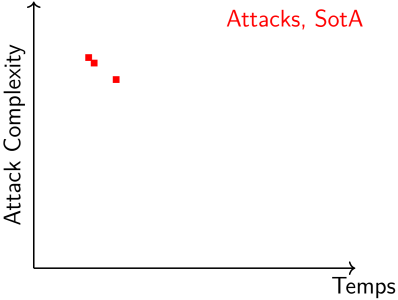


[a]

Attack approach (industry):

[a] Shamelessly stolen to O. Bronchain

Evaluate Security against Side-Channel Attacks

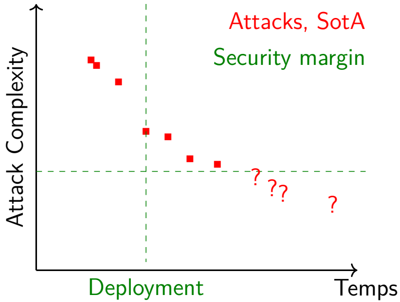


[a]

Attack approach (industry):
Current security level ✓

[a]Shamelessly stolen to O. Bronchain

Evaluate Security against Side-Channel Attacks



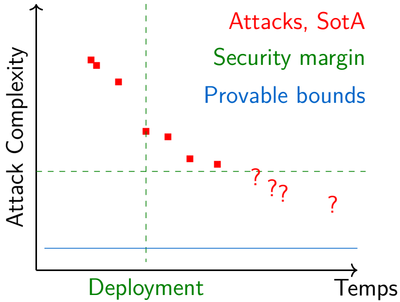
[a]

Attack approach (industry):

- Current security level ✓
- Future improvement → reevaluation ✗

[a]Shamelessly stolen to O. Bronchain

Evaluate Security against Side-Channel Attacks



[a]

Attack approach (industry):

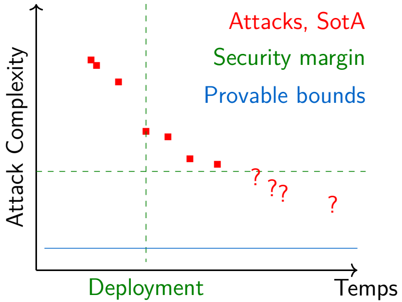
- Current security level ✓
- Future improvement → reevaluation ✗

Approach by *proofs* (academia):

- Rigorous approach ✓
- Potentially conservative ✗

[a] Shamelessly stolen to O. Bronchain

Evaluate Security against Side-Channel Attacks



[a]

Attack approach (industry):

- Current security level ✓
- Future improvement → reevaluation ✗

Approach by *proofs* (academia):

- Rigorous approach ✓
- Potentially conservative ✗

[a] Shamelessly stolen to O. Bronchain

Agenda: evaluation by attack (today), evaluation by proofs (tomorrow)

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

$\rightarrow$ Characterize analytically the *best* $\mathcal{A}$

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

$\rightarrow$ Characterize analytically the *best* $\mathcal{A}$

$\rightarrow$ Decompose each attack step

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

→ Characterize analytically the *best* $\mathcal{A}$

→ Decompose each attack step

→ Quantify the complexity of each step

[7] Analogy with real estate

How to Evaluate Efficiently? Interlude

A good *evaluator* $\mathcal{E} \neq$ A good *adversary* $\mathcal{A}$

Evaluating = guessing how much the best $\mathcal{A}$ must pay to succeed[7]

- Naive way: instantiate all possible $\mathcal{A}$ from the literature (CPA, DoM, stochastic attacks, template attacks, ...) ✗
- Smarter way: yet another D&C approach ✓

$\rightarrow$ Characterize analytically the *best* $\mathcal{A}$

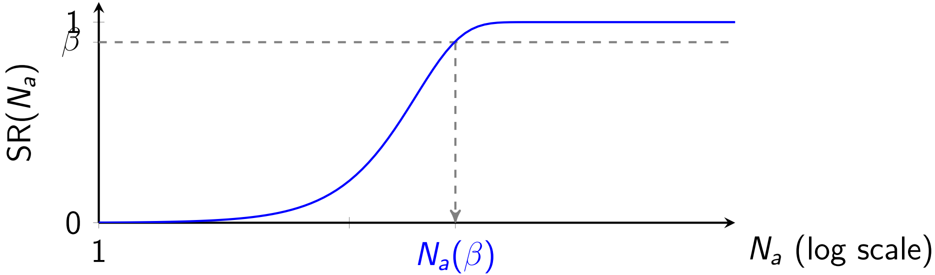
$\rightarrow$ Decompose each attack step

$\rightarrow$ Quantify the complexity of each step

$\Longrightarrow$ Finding evaluation shortcuts

[7] Analogy with real estate

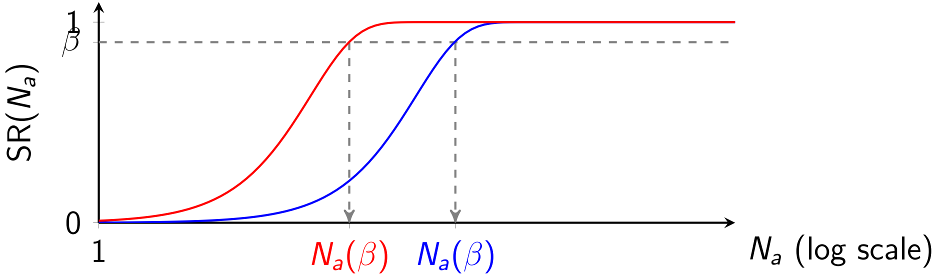
Assessing an Attack: the Success Rate[8]



- SR: probability to succeed the attack within N_a queries to the target

[8] In the following, we focus on *data* complexity only. All known attacks are computationally efficient.

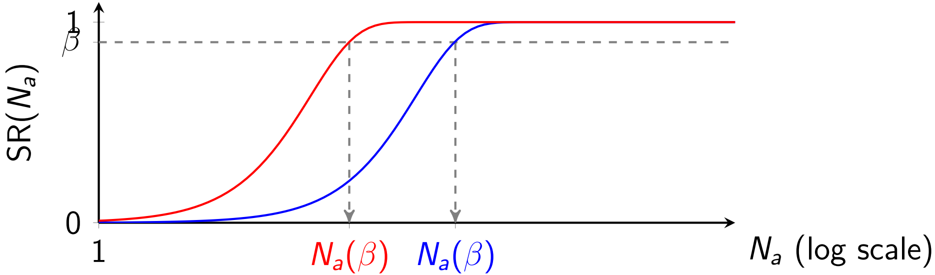
Assessing an Attack: the Success Rate[8]



- SR: probability to succeed the attack within N_a queries to the target
- Allows to compare attacks: $\mathcal{A}_1 \underset{\beta}{<} \mathcal{A}_2$ iff for a fixed $N_a(\beta) > N_a(\beta)$

[8] In the following, we focus on *data* complexity only. All known attacks are computationally efficient.

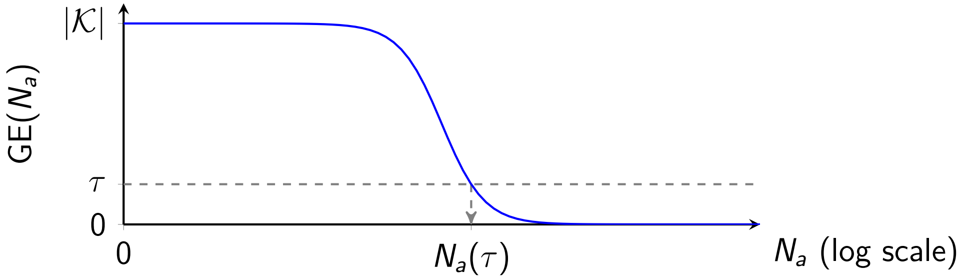
Assessing an Attack: the Success Rate[8]



- SR: probability to succeed the attack within N_a queries to the target
- Allows to compare attacks: $\mathcal{A}_1 \underset{\beta}{<} \mathcal{A}_2$ iff for a fixed $N_a(\beta) > N_a(\beta)$
- **Q4 [full key to D&C]**: Prove that $N_a(\beta, \mathrm{full}) = N_a(\beta^{\frac{n}{|\mathcal{K}|}}, \mathrm{word})$

[8] In the following, we focus on *data* complexity only. All known attacks are computationally efficient.

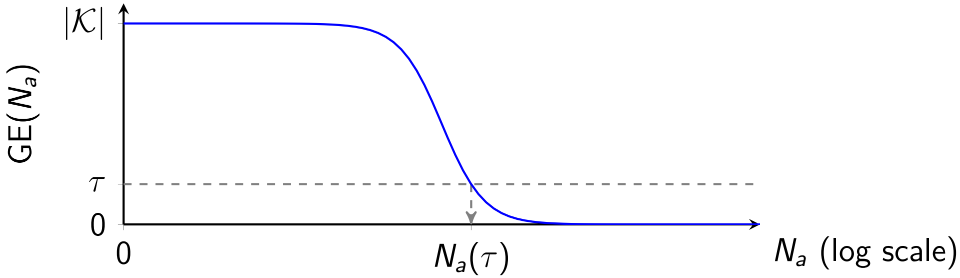
Assessing an Attack: the Guessing Entropy



- GE: average rank of the right key, among all key hypotheses

[9] David and Wool, *A Bounded-Space Near-Optimal Key Enumeration Algorithm for Multi-Dimensional Side-Channel Attacks*.

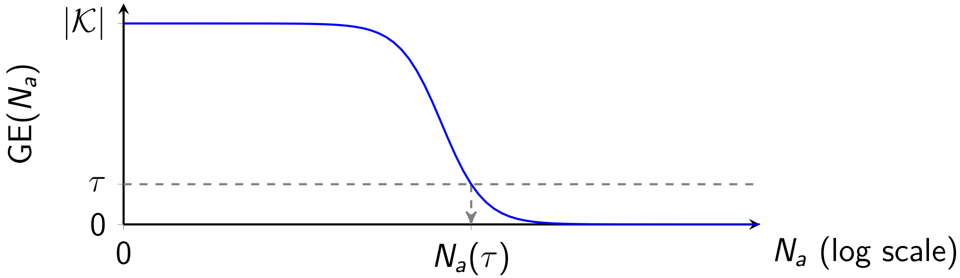
Assessing an Attack: the Guessing Entropy



- GE: average rank of the right key, among all key hypotheses
- Allows to quantify the key remaining *enumeration work*

[9] David and Wool, *A Bounded-Space Near-Optimal Key Enumeration Algorithm for Multi-Dimensional Side-Channel Attacks*.

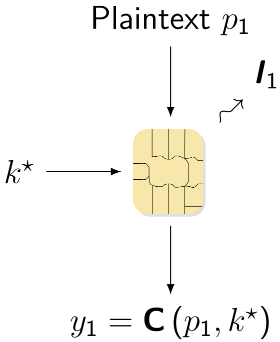
Assessing an Attack: the Guessing Entropy



- GE: average rank of the right key, among all key hypotheses
- Allows to quantify the key remaining *enumeration work*
- **Q5 [full key to D&C]**:[9] Prove that $N_a(\tau, \mathrm{full}) \geq N_a(\tau^{\frac{n}{|\mathcal{K}|}}, \mathrm{word})$

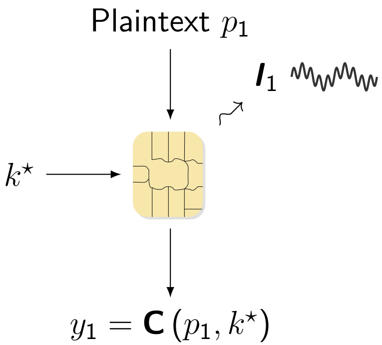
[9] David and Wool, *A Bounded-Space Near-Optimal Key Enumeration Algorithm for Multi-Dimensional Side-Channel Attacks*.

Optimal Attack: Maximum Likelihood[10]



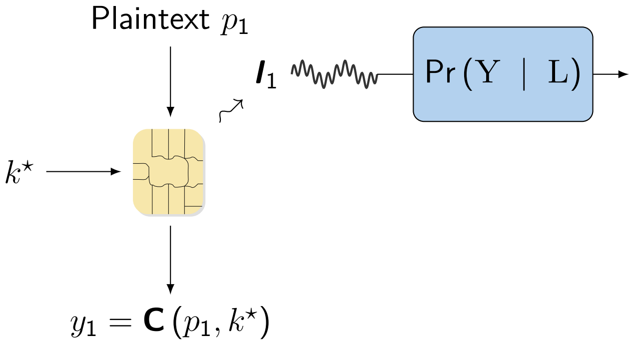
[10] Heuser, Rioul, and Guilley, “Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory”.

Optimal Attack: Maximum Likelihood[10]



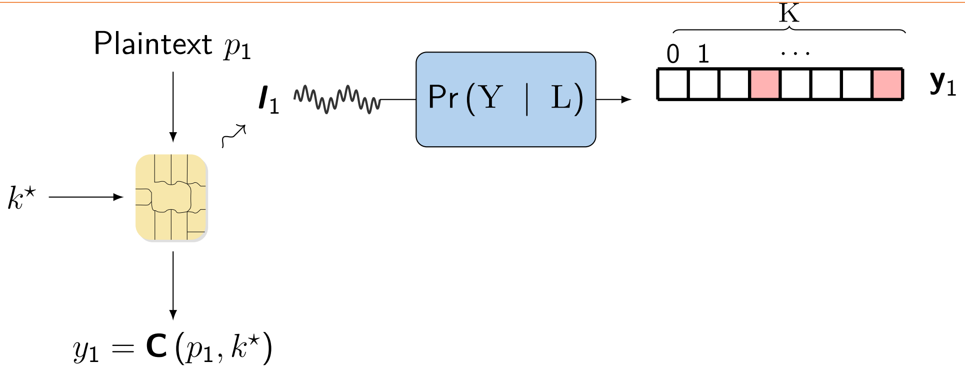
[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



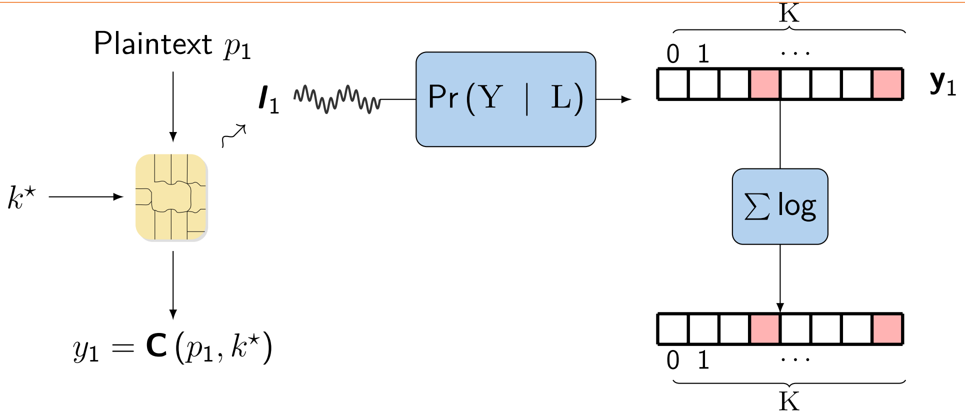
[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



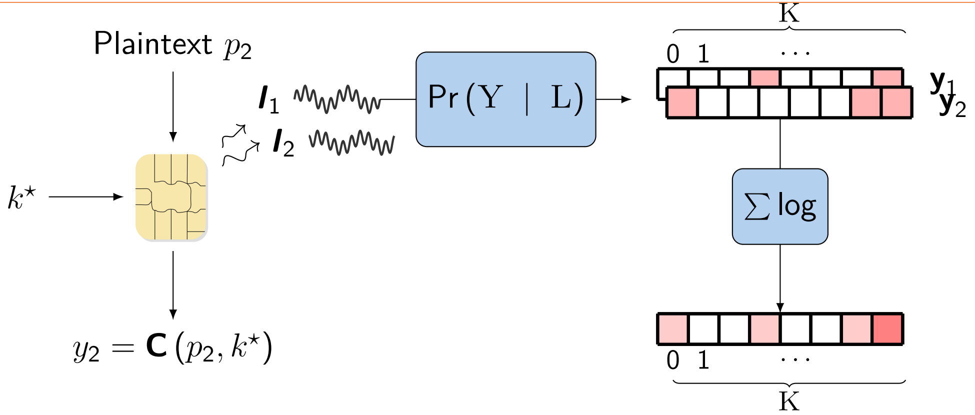
[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



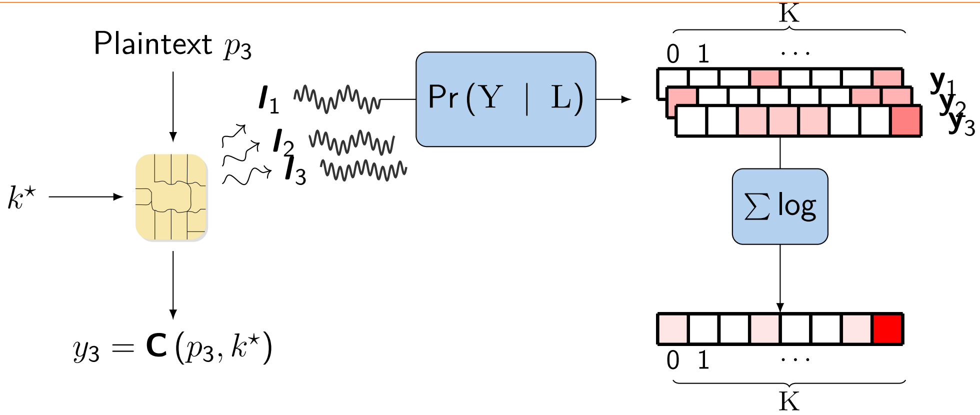
[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



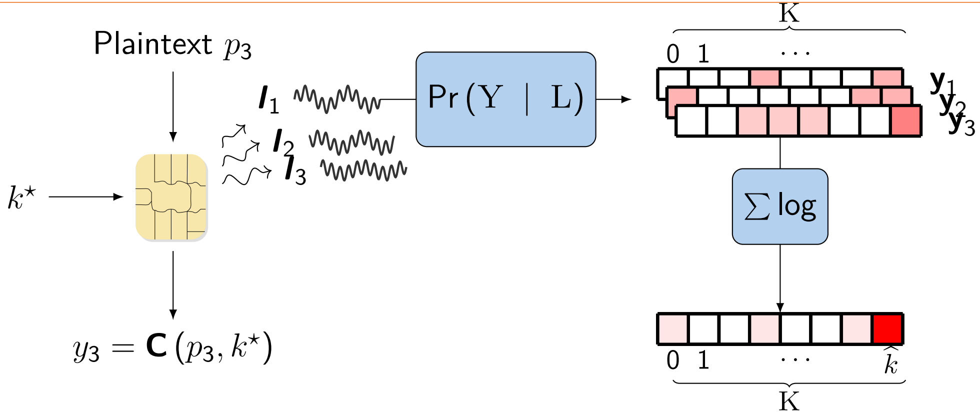
[10]Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



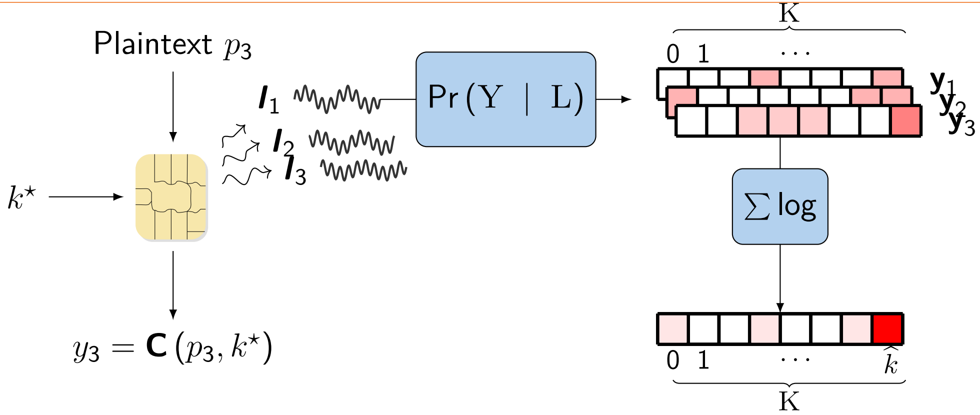
[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]



[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Optimal Attack: Maximum Likelihood[10]

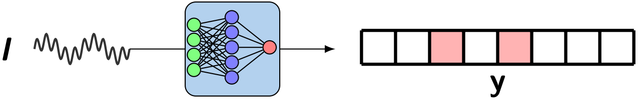


Problem: $\Pr(Y \mid \mathbf{L})$ unknown (device-dependent)

[10] Heuser, Rioul, and Guilley, "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory".

Content

Deep Learning (DL) for SCA



Deep Neural Network (DNN)

General way to modelize, *i.e.*, to convert leakage into probabilities

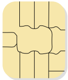
$$F : \left| \begin{array}{ccc} \mathcal{L} & \longrightarrow & \mathcal{P}(\mathcal{Y}) \\ \boldsymbol{l} & \longmapsto & \mathbf{y} = F(\boldsymbol{l}) \approx \Pr\left(\mathrm{Y} \mid \mathbf{L} = \boldsymbol{l}\right) \end{array} \right. \tag{1}$$

$F(\boldsymbol{l})$: output of a Directed Acyclic Graph (DAG) of computation:

- Each node: elementary function $f_i(\cdot, \theta_i)$
- θ_i: *parameters* fully describing f_i

Shape of the DAG, nature of the classes of functions: *architecture* of the DNN.

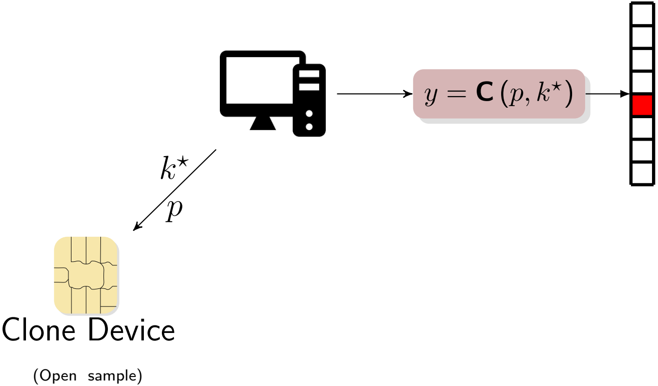
Profiled SCA = Supervised Learning Problem



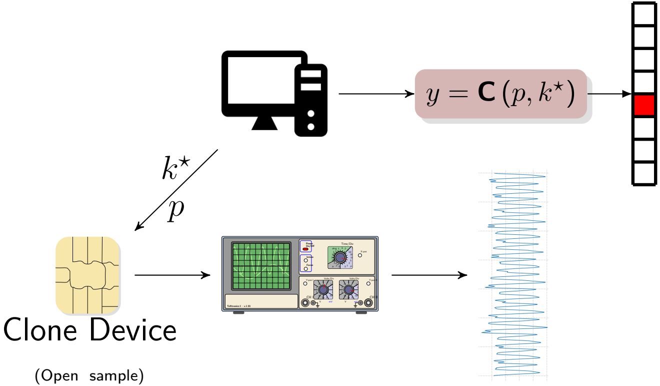
Clone Device

(Open sample)

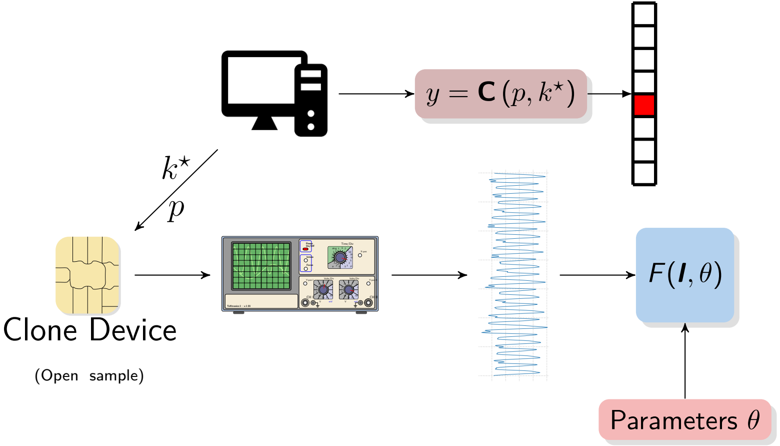
Profiled SCA = Supervised Learning Problem



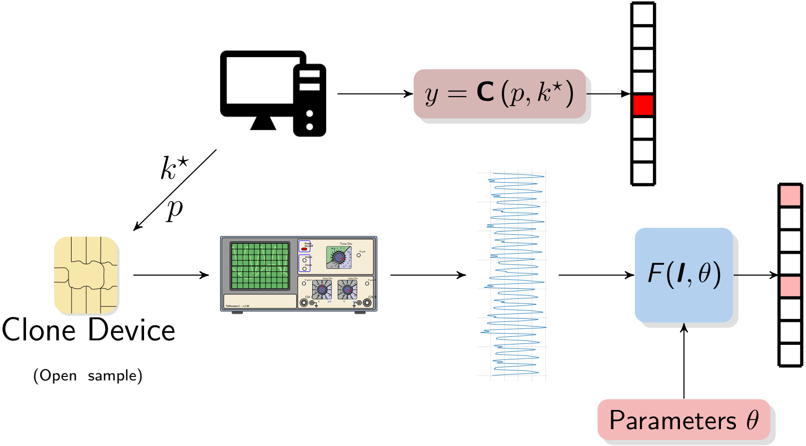
Profiled SCA = Supervised Learning Problem



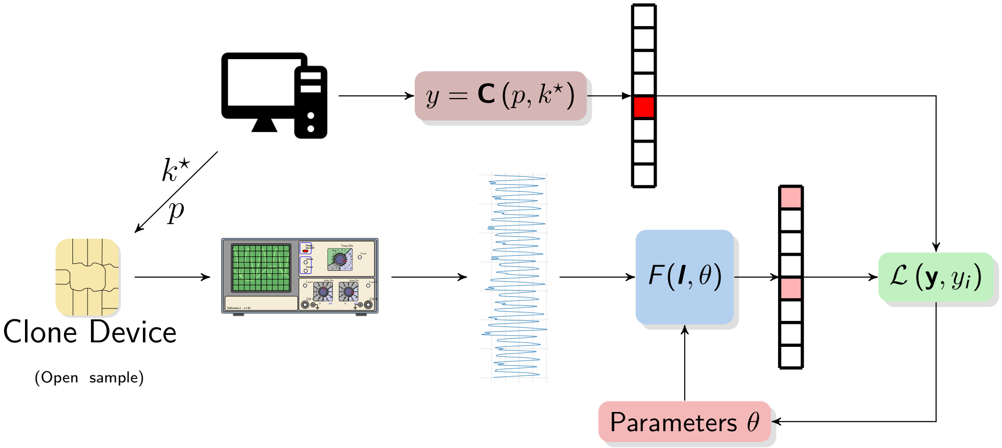
Profiled SCA = Supervised Learning Problem



Profiled SCA = Supervised Learning Problem



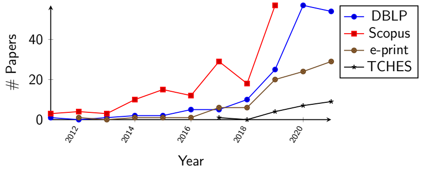
Profiled SCA = Supervised Learning Problem



$\mathcal{L}()$: loss function to minimize, with *gradient descent*

The Deep Learning (DL) hype in SCA

- ➔ **Space 2016:** DL breaks *masking*[11]
- ➔ **Ches 2017:** CNNs efficiently tackles *misalignment*[12]
- ➔ **Ches 2019:** non-profiled attacks[13]
- ➔ *De facto* standard for evaluations
- ➔ Dedicated sessions in conferences



[11] Maghrebi, Portigliatti, and Prouff, "Breaking Cryptographic Implementations Using Deep Learning Techniques"

[12] Cagli, Dumas, and Prouff, "Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures - Profiling Attacks Without Pre-processing"

[13] Timon, "Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis"

Content

Use Case: SCA against code polymorphism I

Implementation of AES from mbedTLS on ARM-Cortex M4 architecture

- T-table implementation with 32 bit variables

$100,000$ traces acquired for each target ($\leq$ a day)

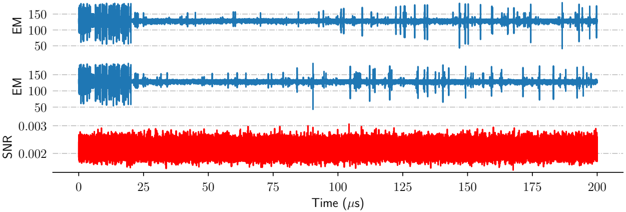


Figure: Two examples of traces (blue) and the Signal-to-Noise Ratio (SNR) (red)

No SNR peak $\implies$ a layman attacker fails, even with $N_a = 10^5$ traces

Use Case:SCA against code polymorphism II[14]

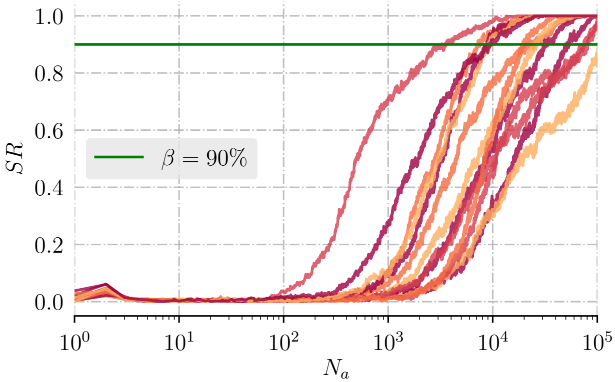


Figure: Layman Attacker with Re-alignment

[14] Masure et al., "Deep Learning Side-Channel Analysis on Large-Scale Traces - A Case Study on a Polymorphic AES".

Use Case:SCA against code polymorphism II[14]

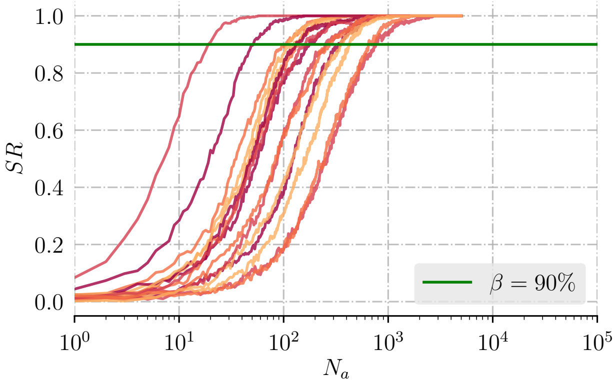


Figure: Attacker with Re-alignment and a Clone device

[14] Masure et al., "Deep Learning Side-Channel Analysis on Large-Scale Traces - A Case Study on a Polymorphic AES".

Use Case:SCA against code polymorphism II[14]

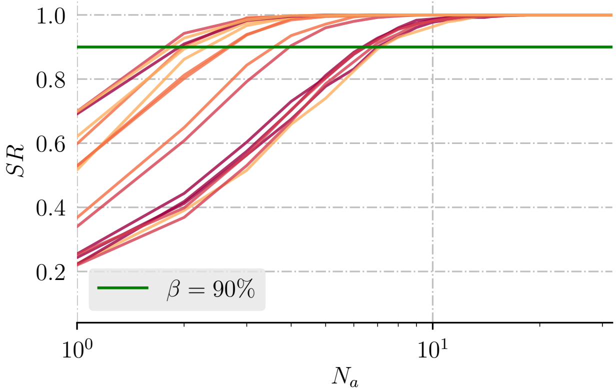


Figure: Attacker without Re-alignment but with a Clone device and deep learning

[14] Masure et al., "Deep Learning Side-Channel Analysis on Large-Scale Traces - A Case Study on a Polymorphic AES".

Post-Mortem Sensitivity Analysis

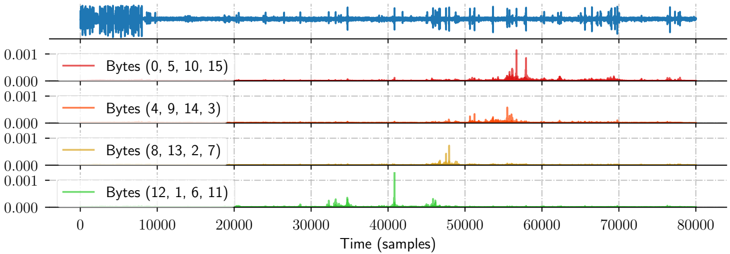


Figure: Gradient Visualization against code polymorphism

Forensics: *"Where does my leakage come from"*?

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

X0X1X2X3

(a) AES state after the first AddRoundKey

0	4	8	12
5	9	13	1
10	14	2	6
15	3	7	11

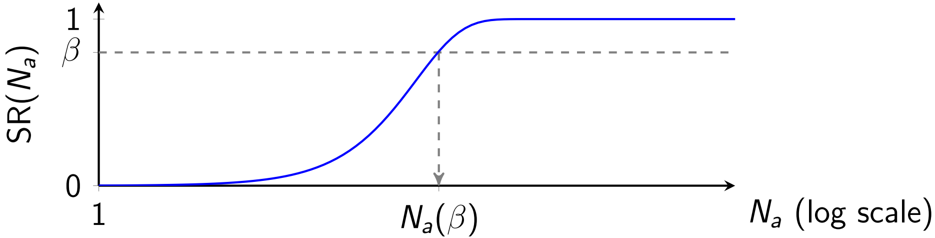
X0X1X2X3

(b) AES state at the end of the ShiftRows

Content

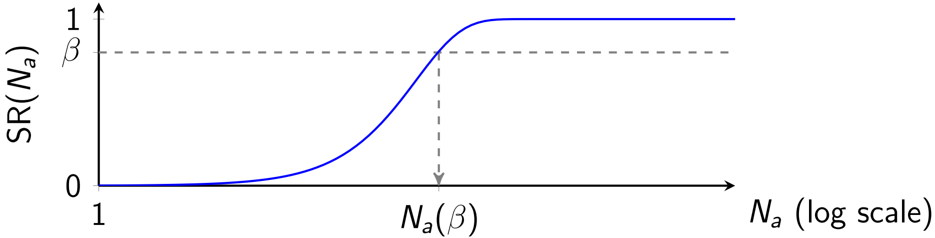
Expensive Metrics to Estimate

Needs to estimate the whole Success Rate (SR) curve to derive $N_a(\beta)$



Expensive Metrics to Estimate

Needs to estimate the whole SR curve to derive $N_a(\beta)$

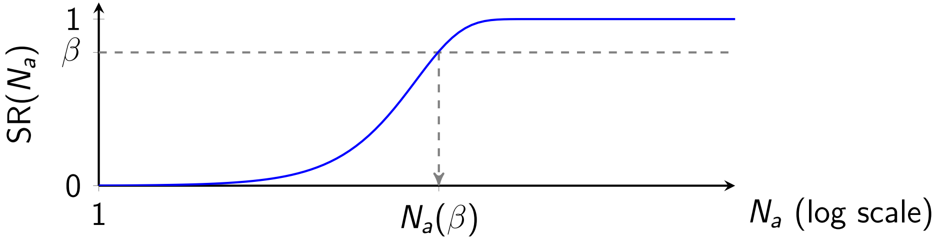


Naive est. of $N_a(\beta)$ expensive:

- Requires $N_v \geq N_a(\beta)$ ✓
- Complexity $\mathcal{O}\left(N_{est} \cdot N_v\right)$ ✗

Expensive Metrics to Estimate

Needs to estimate the whole SR curve to derive $N_a(\beta)$



Naive est. of $N_a(\beta)$ expensive:

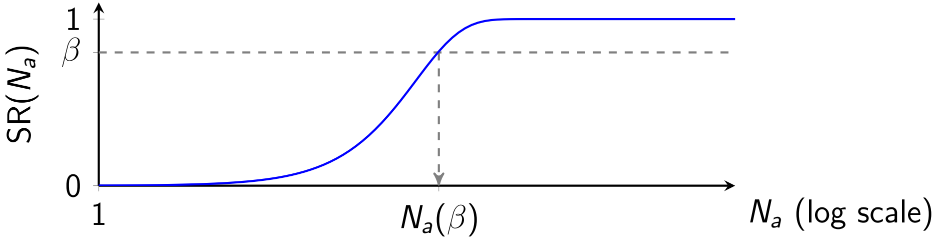
- Requires $N_v \geq N_a(\beta)$ ✓
- Complexity $\mathcal{O}(N_{est} \cdot N_v)$ ✗

Cheaper alternative: re-sampling

- Requires $N_v \gg N_a(\beta)$ ✓
- Biased method ✓

Expensive Metrics to Estimate

Needs to estimate the whole SR curve to derive $N_a(\beta)$



Naive est. of $N_a(\beta)$ expensive:

- Requires $N_v \geq N_a(\beta)$ ✓
- Complexity $\mathcal{O}(N_{est} \cdot N_v)$ ✗

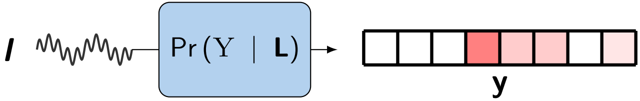
Cheaper alternative: re-sampling

- Requires $N_v \gg N_a(\beta)$ ✓
- Biased method ✓

Wish: find a shortcut to estimate $N_a(\beta)$

Shortcut

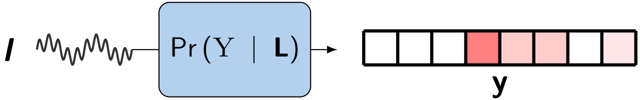
Solution: characterize the predictions of the adversary's model



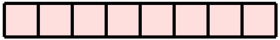
[15] D: Kullback - Leibler (KL) divergence, total variation, Euclidean norm, ...

Shortcut

Solution: characterize the predictions of the adversary's model



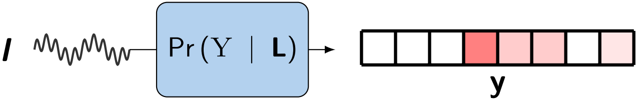
If, the adversary gets:



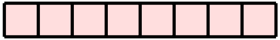
[15] D: KL divergence, total variation, Euclidean norm, ...

Shortcut

Solution: characterize the predictions of the adversary's model



If, the adversary gets:



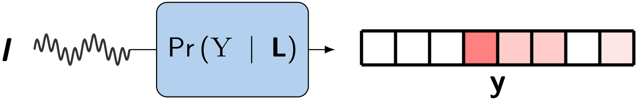
Very noisy leakage

k unpredictable: $N_a(\beta) \to \infty$, if $\beta > \frac{1}{|\mathcal{K}|}$

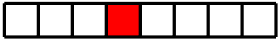
[15] D: KL divergence, total variation, Euclidean norm, ...

Shortcut

Solution: characterize the predictions of the adversary's model



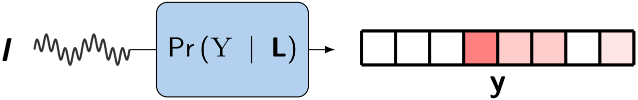
If, the adversary gets:



[15] D: KL divergence, total variation, Euclidean norm, ...

Shortcut

Solution: characterize the predictions of the adversary's model



If, the adversary gets:

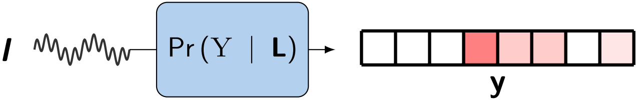
Low-noise leakage

Exact prediction: $N_a(\beta) = 1$

[15] D: KL divergence, total variation, Euclidean norm, ...

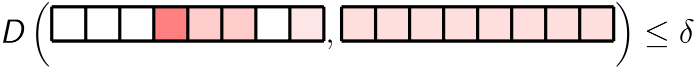
Shortcut

Solution: characterize the predictions of the adversary's model



δ-noisy adversary

All the p.m.f.s accessed by the adversary are δ-close[15] to the uniform:



[15] D: KL divergence, total variation, Euclidean norm, ...

A First Attempt: the Statistical Distance (SD)

Definition (Statistical Distance (SD))

Statistical Distance (SD) upper bounds the probability to distinguish two leakage distributions given two different keys (useful for cryptographers):

$$\mathsf{SD}\left(\mathrm{Y};\mathbf{L}\right) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{TV}\left(\mathsf{p}_{\mathrm{Y}\mid\mathbf{L}};\mathsf{p}_{\mathrm{Y}}\right)\right], \text{ where } \mathsf{TV}\left(\mathsf{p};\mathsf{m}\right) = \frac{1}{2}\sum_{y\in\mathcal{Y}}\left|\mathsf{p}\left(y\right)-\mathsf{m}\left(y\right)\right|$$

[16] Lower bound: tensorization of SD. Upper bound: Chernoff inequality (+ Slud's for tightness).

A First Attempt: the Statistical Distance (SD)

Definition (Statistical Distance (SD))

SD upper bounds the probability to distinguish two leakage distributions given two different keys (useful for cryptographers):

$$\mathsf{SD}(\mathrm{Y};\mathbf{L}) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{TV}\left(\mathsf{p}_{\mathrm{Y}\mid\mathbf{L}};\mathsf{p}_{\mathrm{Y}}\right)\right], \text{ where } \mathsf{TV}(\mathsf{p};\mathsf{m}) = \frac{1}{2}\sum_{y\in\mathcal{Y}}|\mathsf{p}(y)-\mathsf{m}(y)|$$

Lemma (Tensorization)

Denote $\mathsf{SD}(\mathrm{Y};\mathbf{L})$ *by* δ, *then the following bounds are tight (**Q6**: prove it):*[16]

[16] Lower bound: tensorization of SD. Upper bound: Chernoff inequality (+ Slud's for tightness).

A First Attempt: the Statistical Distance (SD)

Definition (Statistical Distance (SD))

SD upper bounds the probability to distinguish two leakage distributions given two different keys (useful for cryptographers):

$$\mathsf{SD}\left(\mathrm{Y}; \mathbf{L}\right) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{TV}\left(\mathsf{p}_{\mathrm{Y} \mid \mathbf{L}}; \mathsf{p}_{\mathrm{Y}}\right)\right], \text{ where } \mathsf{TV}\left(\mathsf{p}; \mathsf{m}\right) = \frac{1}{2}\sum_{y \in \mathcal{Y}} \left|\mathsf{p}\left(y\right) - \mathsf{m}\left(y\right)\right|$$

Lemma (Tensorization)

Denote $\mathsf{SD}\left(\mathrm{Y}; \mathbf{L}\right)$ *by* δ*, then the following bounds are tight (****Q6****: prove it):*[16]

$$\Omega\left(\frac{\beta}{\delta}\right) \leq N_a(\beta)$$

[16] Lower bound: tensorization of SD. Upper bound: Chernoff inequality (+ Slud's for tightness).

A First Attempt: the Statistical Distance (SD)

Definition (Statistical Distance (SD))

SD upper bounds the probability to distinguish two leakage distributions given two different keys (useful for cryptographers):

$$\mathsf{SD}\left(\mathrm{Y};\mathbf{L}\right)=\mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{TV}\left(\mathsf{p}_{\mathrm{Y}\mid\mathbf{L}};\mathsf{p}_{\mathrm{Y}}\right)\right],\ \text{where } \mathsf{TV}\left(\mathsf{p};\mathsf{m}\right)=\frac{1}{2}\sum_{y\in\mathcal{Y}}\left|\mathsf{p}\left(y\right)-\mathsf{m}\left(y\right)\right|$$

Lemma (Tensorization)

Denote $\mathsf{SD}\left(\mathrm{Y};\mathbf{L}\right)$ *by* δ*, then the following bounds are tight (**Q6**: prove it):*[16]

$$\Omega\left(\frac{\beta}{\delta}\right)\leq N_a(\beta)\leq\mathcal{O}\left(\frac{\beta}{\delta^{2}}\right)$$

[16] Lower bound: tensorization of SD. Upper bound: Chernoff inequality (+ Slud's for tightness).

A Second Attempt: the Mutual Information (MI)

Definition (Mutual Information (MI))

$$\mathsf{MI}(\mathrm{Y}; \mathbf{L}) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{D}(\mathsf{p}_{\mathrm{Y} \mid \mathbf{L}} \parallel \mathsf{p}_{\mathrm{Y}})\right], \text{ where } \mathsf{D}(\mathsf{p} \parallel \mathsf{m}) = \sum_{y \in \mathcal{Y}} \mathsf{p}(y) \log\left(\frac{\mathsf{p}(y)}{\mathsf{m}(y)}\right)$$

Lemma (Fano Inequality)

Denote $\mathsf{MI}(\mathrm{Y}; \mathbf{L})$ *by* δ*, then the following inequality is tight (Shannon's coding theorem):*[17]

$$\frac{f(\beta)}{\delta} \leq N_a(\beta)$$

[17] Cover and Thomas, *Elements of information theory (2. ed.)*

Use Cases with Univariate Leakage, Gaussian Noise

Leakage model of shape $\mathrm{L} = \delta(\mathcal{Y}) + N$

$\rightarrow$ Upper bound of MI from Signal-to-Noise Ratio (SNR)

$$N_a(\beta) \geq \frac{f(\beta)}{\mathsf{MI}(\mathrm{Y};\mathrm{L})}$$

Use Cases with Univariate Leakage, Gaussian Noise

Leakage model of shape $\mathrm{L} = \delta(\mathcal{Y}) + N$

$\rightarrow$ Upper bound of MI from Signal-to-Noise Ratio (SNR)

$$N_a(\beta) \geq \frac{f(\beta)}{\frac{1}{2}\log\left(1 + \frac{\underset{\mathrm{Y}}{\mathsf{Var}}\left(\mathbb{E}[\mathrm{L} \mid \mathrm{Y}]\right)}{\underset{\mathrm{Y}}{\mathbb{E}}\left[\mathsf{Var}(\mathrm{L} \mid \mathrm{Y})\right]}\right)}$$

Use Cases with Univariate Leakage, Gaussian Noise

Leakage model of shape $\mathrm{L} = \delta(\mathcal{Y}) + N$

$\rightarrow$ Upper bound of MI from Signal-to-Noise Ratio (SNR)

$$N_a(\beta) \geq \frac{f(\beta)}{\frac{1}{2}\log\left(1 + \frac{\mathsf{Var}_{\mathrm{Y}}\left(\mathbb{E}[\mathrm{L} \mid \mathrm{Y}]\right)}{\mathbb{E}_{\mathrm{Y}}\left[\mathsf{Var}(\mathrm{L} \mid \mathrm{Y})\right]}\right)}$$

$\rightarrow$ Generalizes the data complexity for correlation attack:

$$N_a^{corr} \approx \frac{28}{\rho^2}$$

Reduction to MI estimation

Estimating a Mutual Information is generally hard:

[18] Paninski, "Estimation of Entropy and Mutual Information".
[19] Masure et al., "Information Bounds and Convergence Rates for Side-Channel Security Evaluators".

Reduction to MI estimation

Estimating a Mutual Information is generally hard:

✗ No unbiased estimator[18]

[18] Paninski, "Estimation of Entropy and Mutual Information".

[19] Masure et al., "Information Bounds and Convergence Rates for Side-Channel Security Evaluators".

Reduction to MI estimation

Estimating a Mutual Information is generally hard:

✗ No unbiased estimator[18]

✗ *Empirical* estimator:

→ Positively biased in average ✓

→ Suffers from *curse of dimensionality*[19] ✗

[18] Paninski, "Estimation of Entropy and Mutual Information".

[19] Masure et al., "Information Bounds and Convergence Rates for Side-Channel Security Evaluators".

Reduction to MI estimation

Estimating a Mutual Information is generally hard:

- ✗ No unbiased estimator[18]
- ✗ *Empirical* estimator:
  - → Positively biased in average ✓
  - → Suffers from *curse of dimensionality*[19] ✗

Definition (PI)

$$\mathsf{MI}\left(\mathrm{Y};\mathbf{L}\right) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{D}(\mathsf{p}_{\mathrm{Y}\,|\,\mathbf{L}} \parallel \mathsf{p}_{\mathrm{Y}})\right]$$

[18] Paninski, "Estimation of Entropy and Mutual Information".

[19] Masure et al., "Information Bounds and Convergence Rates for Side-Channel Security Evaluators".

Reduction to MI estimation

Estimating a Mutual Information is generally hard:

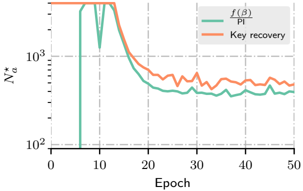
- ✗ No unbiased estimator[18]
- ✗ *Empirical* estimator:
  - → Positively biased in average ✓
  - → Suffers from *curse of dimensionality*[19] ✗

Definition (PI)

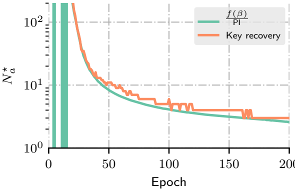
$$\mathsf{PI}\left(\mathrm{Y}; \mathbf{L}; \theta\right) = \mathop{\mathbb{E}}_{\mathbf{L}}\left[\mathsf{D}(F(\mathbf{L}, \theta) \parallel \mathsf{p}_{\mathrm{Y}})\right] \leq \mathsf{MI}\left(\mathrm{Y}; \mathbf{L}\right)$$

[18] Paninski, "Estimation of Entropy and Mutual Information".
[19] Masure et al., "Information Bounds and Convergence Rates for Side-Channel Security Evaluators".

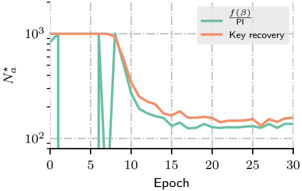
Examples



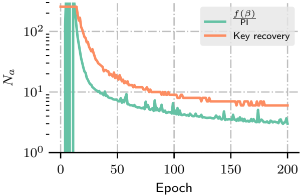
(a) AES FPGA



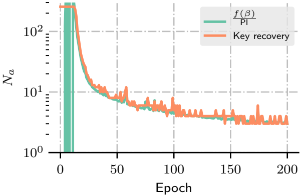
(b) AES with misalignment



(c) AES soft with masking



(d) Polymorphic AES



(e) Polymorphic AES

PI and other Surrogates

PI can be well[20] estimated/optimized over the open sample:

- Estimation error $\epsilon = \mathcal{O}\left(\frac{poly(\mathcal{H})}{N_p}\right)$, where

$\rightarrow$ $N_p = \#$profiling traces

$\rightarrow$ $\mathcal{H}$: class of models (neural network, #parameters, ...)

[20] Ito, Ueno, and Homma, "Perceived Information Revisited: New Metrics to Evaluate Success Rate of Side-Channel Attacks", Might suffer from inconsistencies in rare cases.

PI and other Surrogates

PI can be well[20] estimated/optimized over the open sample:

- Estimation error $\epsilon = \mathcal{O}\left(\frac{poly(\mathcal{H})}{N_p}\right)$, where

 $\rightarrow$ $N_p = \#$profiling traces

 $\rightarrow$ $\mathcal{H}$: class of models (neural network, #parameters, ...)

- We want the estimation error $\epsilon \lessapprox \mathsf{MI}\left(\mathrm{Y}; \mathrm{L}\right) \implies$

$$N_p(\epsilon) \geq \Omega\left(\frac{poly(\mathcal{H})}{\mathsf{MI}\left(\mathrm{Y}; \mathrm{L}\right)}\right) \gtrapprox N_a(\beta)$$

 $\rightarrow$ "Profiling is a costly as attacking"

[20] Ito, Ueno, and Homma, "Perceived Information Revisited: New Metrics to Evaluate Success Rate of Side-Channel Attacks", Might suffer from inconsistencies in rare cases.

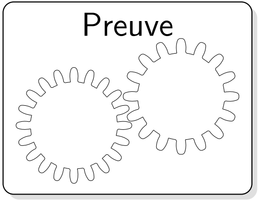
Wrap-Up

Side-Channel Analysis is a threat as powerful (but cheaper) as quantum computers

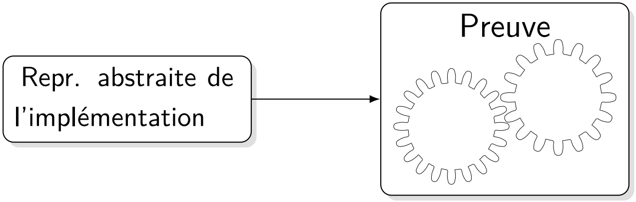
Need to assess the security level against SCA in an affordable manner
$\Longrightarrow$ evaluation shortcuts

Tomorrow: presentation of masking, how to implement it, security analysis

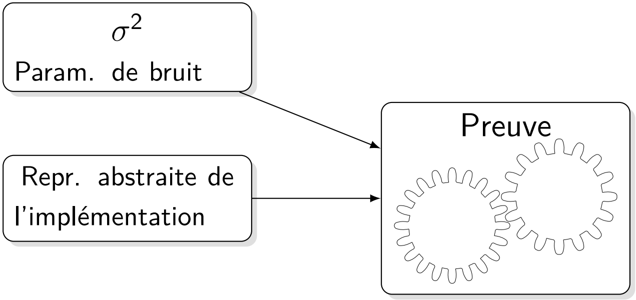
Preuves de Sécurité pour Masquage



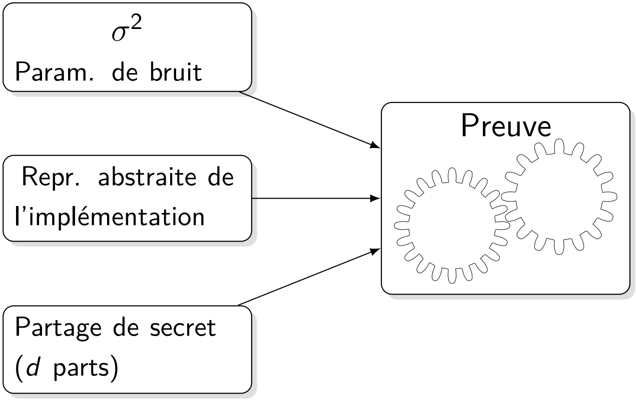
Preuves de Sécurité pour Masquage



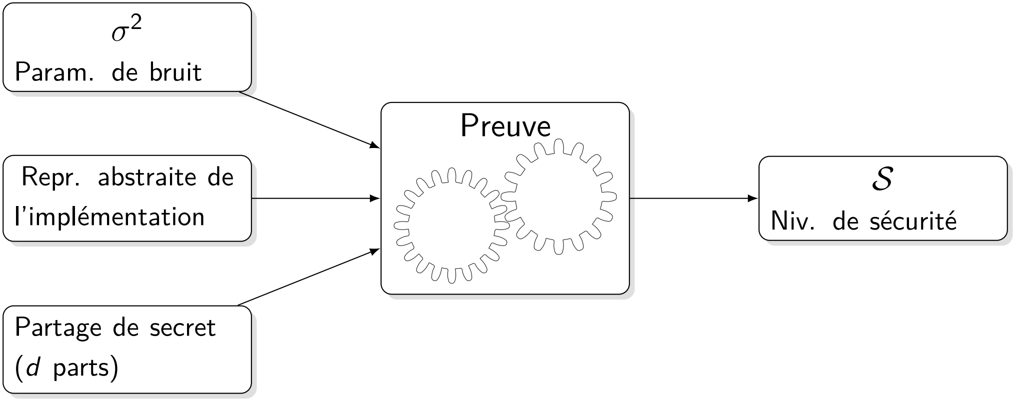
Preuves de Sécurité pour Masquage



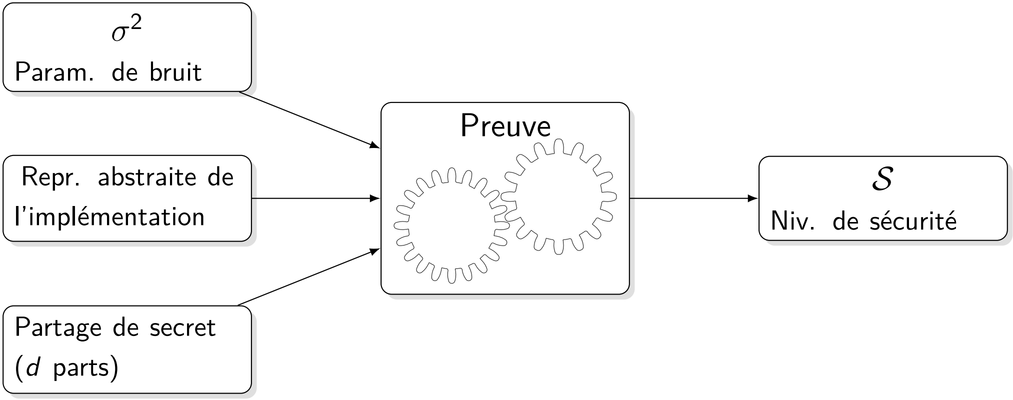
Preuves de Sécurité pour Masquage



Preuves de Sécurité pour Masquage



Preuves de Sécurité pour Masquage



"Toute attaque nécessite $\mathcal{S}$ observations"

Content

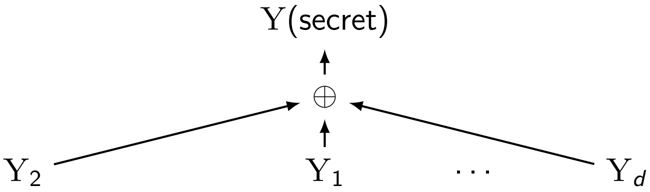
Masking: what is that ?

Masking, a.k.a. *MPC on silicon*:[21][22] secret sharing over a finite field $(\mathbb{F}, \oplus, \otimes)$

$$\Upsilon(\text{secret})$$

Masking: what is that ?

Masking, a.k.a. *MPC on silicon*:[21,22] secret sharing over a finite field $(\mathbb{F}, \oplus, \otimes)$



Masking: what is that ?

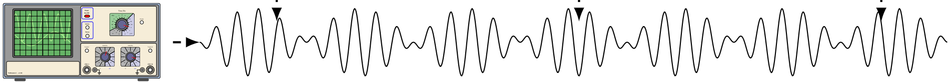
Masking, a.k.a. *MPC on silicon*:[21][22] secret sharing over a finite field $(\mathbb{F}, \oplus, \otimes)$

$$\mathrm{Y}(\text{secret})$$

$$\oplus$$

$$\mathrm{Y}_2 \qquad \mathrm{Y}_1 \qquad \cdots \qquad \mathrm{Y}_d$$

$$\mathrm{L}(\mathrm{Y}_2) = \delta(\mathrm{Y}_2) + N \qquad \mathrm{L}(\mathrm{Y}_1) = \delta(\mathrm{Y}_1) + N \qquad \mathrm{L}(\mathrm{Y}_d) = \delta(\mathrm{Y}_d) + N$$

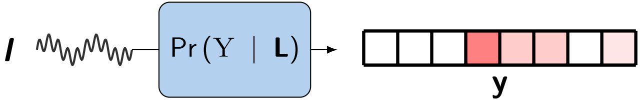


[21] Chari et al., "Towards Sound Approaches to Counteract Power-Analysis Attacks".
[22] Goubin and Patarin, "DES and Differential Power Analysis (The "Duplication" Method)".

Content

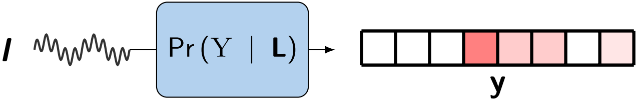
The Noisy Leakage Model

In this model, for each intermediate computation, the adversary gets a probability distribution about its operands:

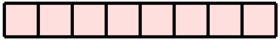


The Noisy Leakage Model

In this model, for each intermediate computation, the adversary gets a probability distribution about its operands:

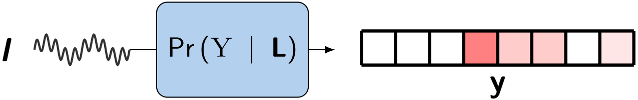


If, the adversary gets:

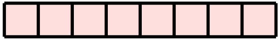


The Noisy Leakage Model

In this model, for each intermediate computation, the adversary gets a probability distribution about its operands:



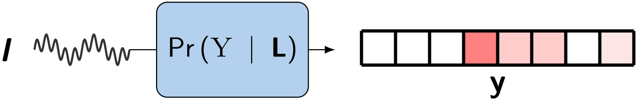
If, the adversary gets:



Very noisy
Sensitive computation unpredictable

The Noisy Leakage Model

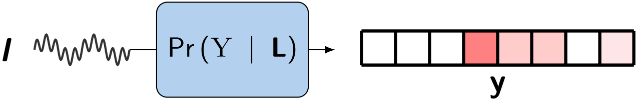
In this model, for each intermediate computation, the adversary gets a probability distribution about its operands:



If, the adversary gets:

The Noisy Leakage Model

In this model, for each intermediate computation, the adversary gets a probability distribution about its operands:



If, the adversary gets:

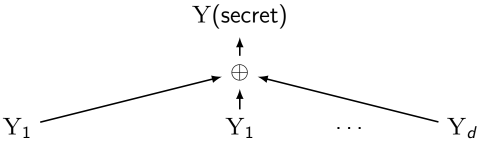
Low-noise

Exact prediction of the sensitive computation

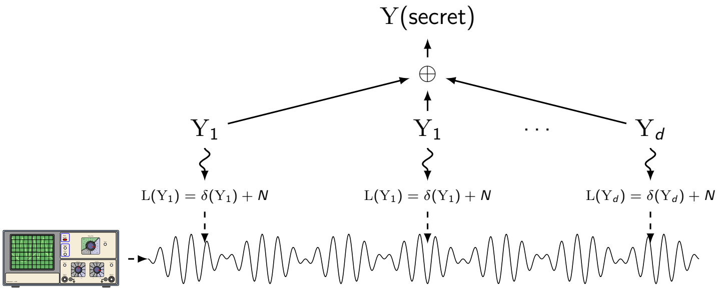
The Effect of Masking

Y(secret)

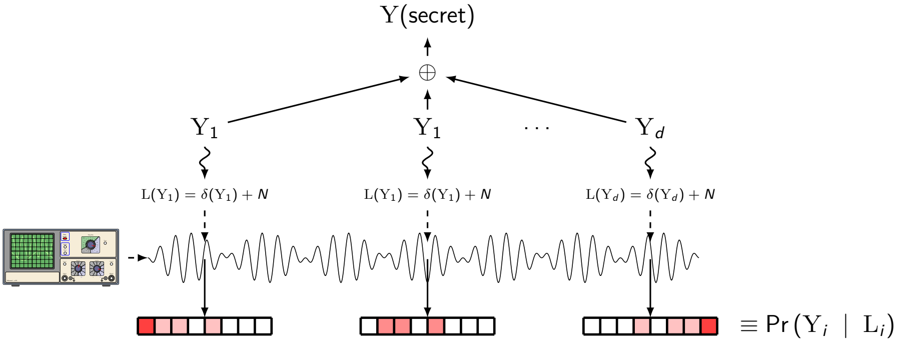
The Effect of Masking



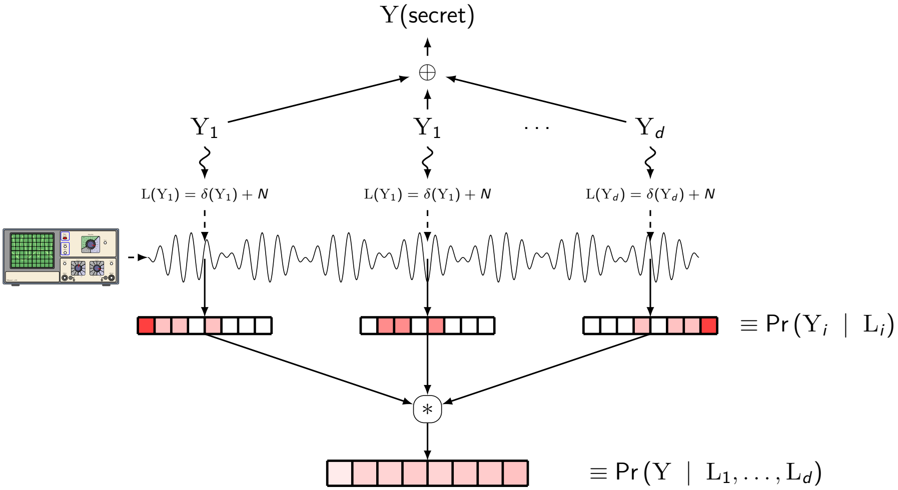
The Effect of Masking



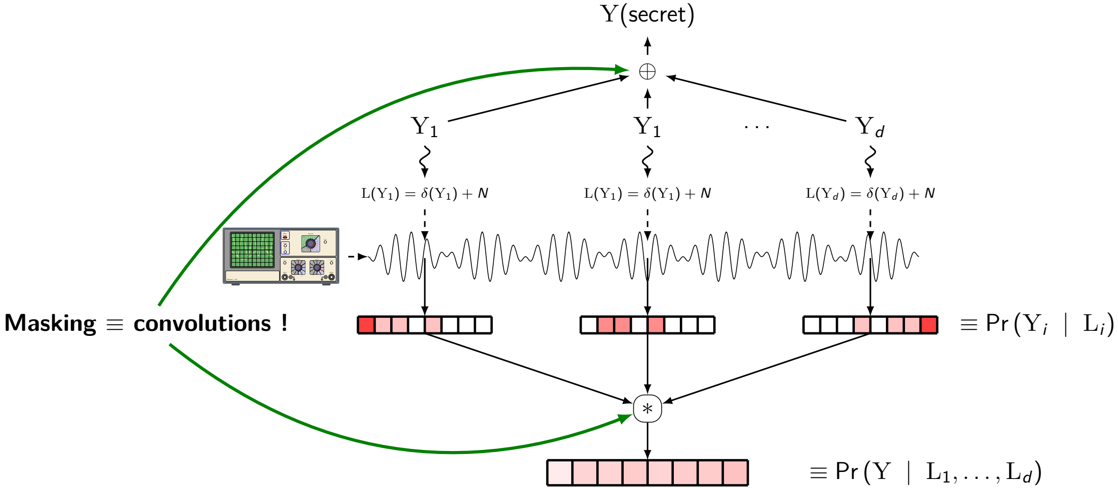
The Effect of Masking



The Effect of Masking

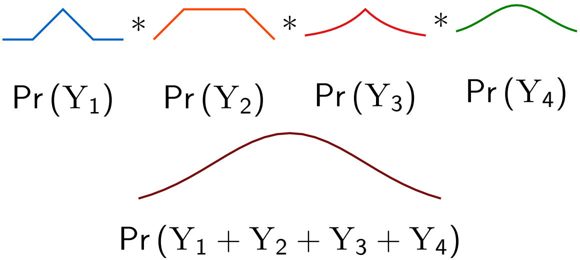


The Effect of Masking



The Secret Power of Convolutions

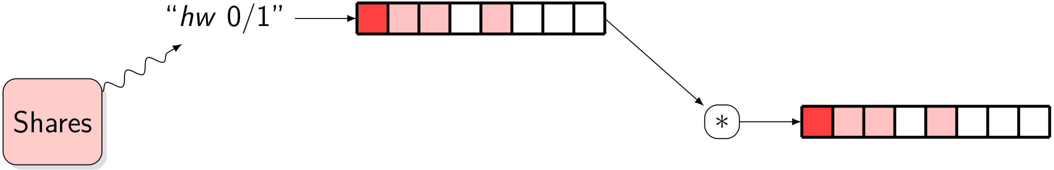
Central Limit Theorem: Assume real-valued random variables Y_i



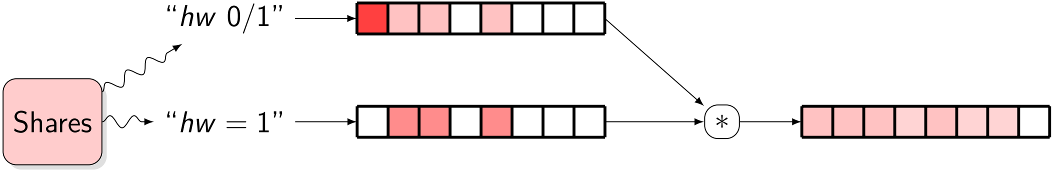
Then the sum is (approximately) distributed like a Gaussian[23]
Interesting property of Gaussian: maximizes the entropy (*i.e.*, uncertainty)[24]

[23] With mild assumptions, but we'll get back to that ...
[24] Out of all Probability Density Functions (p.d.f.s) of same mean and variance

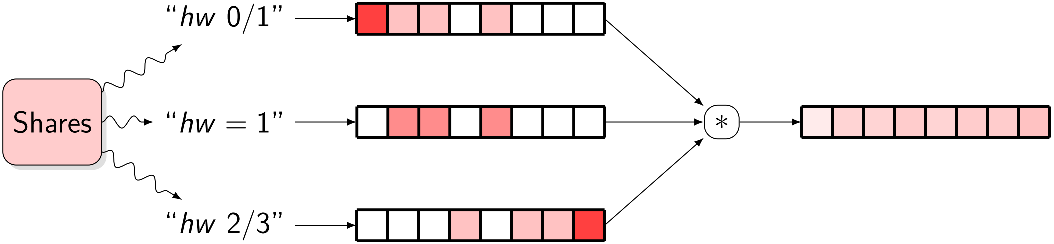
CLT also Works in Finite Groups/Fields !



CLT also Works in Finite Groups/Fields !



CLT also Works in Finite Groups/Fields !



	$Y_i \in \mathbb{R}$	$Y_i \in \mathbb{F}$
Limit Entropy	Gaussian Maximized	Uniform Maximized

Fast Fourier Transform also apply over finite fields !

Quantitative version of CLT

Theorem (Mrs. Gerber's Lemma[25]*)*

Given $\mathrm{Y} = \mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d$, *and each* Y_i *with (indep.) side information* $\mathrm{L}_1, \ldots, \mathrm{L}_d$, *then for* $\eta^{-1} = 2\log(2)$*:*

[25] Béguinot et al., "Removing the Field Size Loss from Duc et al.'s Conjectured Bound for Masked Encodings".

Quantitative version of CLT

Theorem (Mrs. Gerber's Lemma[25])

Given $\mathrm{Y} = \mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d$, *and each* Y_i *with (indep.) side information* $\mathrm{L}_1, \ldots, \mathrm{L}_d$, *then for* $\eta^{-1} = 2\log(2)$*:*

$$\mathsf{MI}\left(\mathrm{Y}; \mathbf{L}\right) \leq \prod_{i=1}^{d} \frac{\mathsf{MI}\left(\mathrm{Y}_i; \mathrm{L}_i\right)}{\eta} + \mathcal{O}\left(\prod_{i=1}^{d} \mathsf{MI}\left(\mathrm{Y}_i; \mathrm{L}_i\right)^2\right) \quad \textit{in } \mathbb{F}_{2^n}$$

[25] Béguinot et al., "Removing the Field Size Loss from Duc et al.'s Conjectured Bound for Masked Encodings".

Quantitative version of CLT

Theorem (Mrs. Gerber's Lemma[25])

Given $\mathrm{Y} = \mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d$, *and each* Y_i *with (indep.) side information* $\mathrm{L}_1, \ldots, \mathrm{L}_d$, *then for* $\eta^{-1} = 2\log(2)$*:*

$$\mathsf{MI}\left(\mathrm{Y}; \mathbf{L}\right) \leq \prod_{i=1}^{d} \frac{\mathsf{MI}\left(\mathrm{Y}_i; \mathrm{L}_i\right)}{\eta} + \mathcal{O}\left(\prod_{i=1}^{d} \mathsf{MI}\left(\mathrm{Y}_i; \mathrm{L}_i\right)^2\right) \quad \textit{in } \mathbb{F}_{2^n}$$

$\rightarrow$ Security $\propto \frac{1}{\mathsf{MI}(\mathrm{Y};\mathbf{L})} \implies$ increases **exponentially fast** with d ✓

[25] Béguinot et al., "Removing the Field Size Loss from Duc et al.'s Conjectured Bound for Masked Encodings".

Quantitative version of CLT

Theorem (Mrs. Gerber's Lemma[25])

Given $\mathrm{Y} = \mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d$, *and each* Y_i *with (indep.) side information* $\mathrm{L}_1, \ldots, \mathrm{L}_d$, *then for* $\eta^{-1} = 2\log(2)$:

$$\mathsf{MI}(\mathrm{Y}; \mathbf{L}) \leq \prod_{i=1}^{d} \frac{\mathsf{MI}(\mathrm{Y}_i; \mathrm{L}_i)}{\eta} + \mathcal{O}\left(\prod_{i=1}^{d} \mathsf{MI}(\mathrm{Y}_i; \mathrm{L}_i)^2\right) \quad \textit{in } \mathbb{F}_{2^n}$$

→ Security $\propto \frac{1}{\mathsf{MI}(\mathrm{Y};\mathbf{L})} \implies$ increases **exponentially fast** with d ✓

→ Independent of the adversary ✓

[25] Béguinot et al., "Removing the Field Size Loss from Duc et al.'s Conjectured Bound for Masked Encodings".

Convolution = Noise Amplification

Simulation, for $\mathbb{F}_{2^n}$: $L(Y_i) = hw(Y_i) + \mathcal{N}(0; \sigma^2)$, hw = Hamming weight

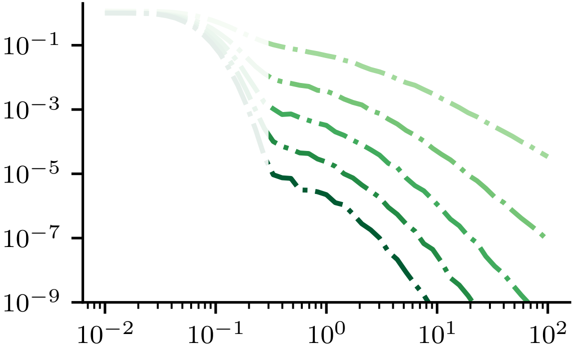


Figure: $\mathrm{MI}(\mathrm{Y}; \mathbf{L})$ vs. σ^2, $2 \leq d \leq 6$

Masking in a Low-Noise Setting

Does masking always work in a low-noise setting ?

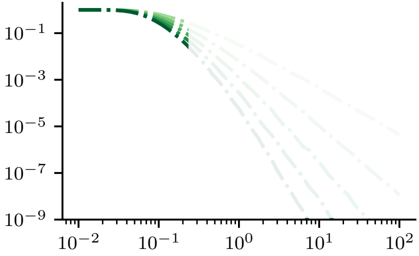


Figure: $MI(Y; Trace)$ vs. σ^2, $2 \leq d \leq 6$

Observation:
Secret always leaks > 1 bit, regardless of d
Explanation:
$\mathsf{lsb}(\mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d) = \mathsf{lsb}(\mathrm{Y}_1) \oplus \ldots \oplus \mathsf{lsb}(\mathrm{Y}_d)$

Masking in a Low-Noise Setting

Does masking always work in a low-noise setting ?

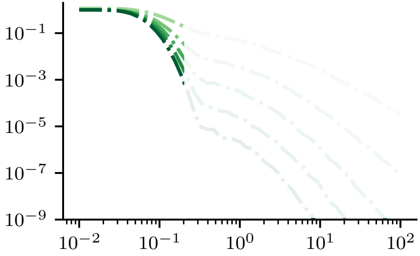


Figure: $MI(Y; Trace)$ vs. σ^2, $2 \leq d \leq 6$

Observation:
Secret always leaks > 1 bit, regardless of d
Explanation:
$\mathrm{hw}(\mathrm{Y}_1 \oplus \ldots \oplus \mathrm{Y}_d) = \sum_i \mathrm{hw}(\mathrm{Y}_i) - 2 \cdot \left(\ldots \right)$
Parity of $\mathrm{hw}(\mathrm{Y})$: **cosets of** $\mathbb{F}_{2^n}$
Corollary: parallelism is no cure either

Masking in a Low-Noise Setting

Does masking always work in a low-noise setting ?

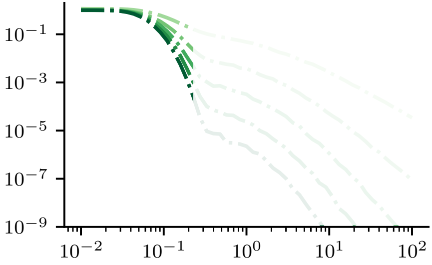
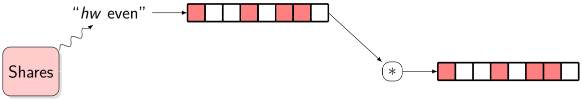


Figure: $MI(Y; Trace)$ vs. σ^2, $2 \leq d \leq 6$

Explanation:

Parity of $\mathrm{hw}(\mathrm{Y})$: **cosets of** $\mathbb{F}_{2^n}$



Masking in a Low-Noise Setting

Does masking always work in a low-noise setting ?

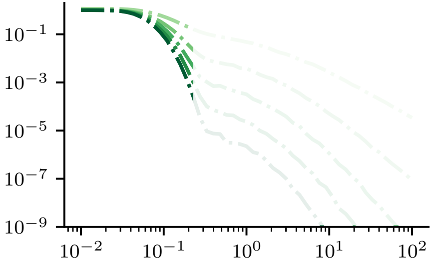
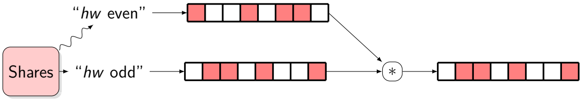


Figure: $MI(Y; Trace)$ vs. σ^2, $2 \leq d \leq 6$

Explanation:

Parity of $\mathrm{hw}(Y)$: **cosets of** $\mathbb{F}_{2^n}$



Masking in a Low-Noise Setting

Does masking always work in a low-noise setting ?

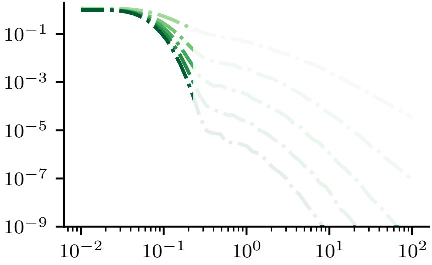
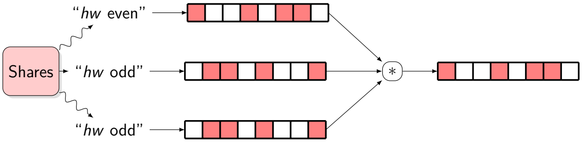


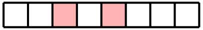
Figure: $MI(Y; Trace)$ vs. σ^2, $2 \leq d \leq 6$

Explanation:

Parity of $\mathrm{hw}(\mathrm{Y})$: **cosets of** $\mathbb{F}_{2^n}$

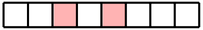


Conditions for Sound Masking

What conditions the distributions  of each share must fit?

[26] Stromberg, "Probabilities on a Compact Group".

Conditions for Sound Masking

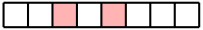
What conditions the distributions  of each share must fit?

"Central Limit Theorem" (qualitative)[26]

Conv. to uniform $\iff$ support *not* contained in any non-trivial coset of $\mathbb{F}$

[26] Stromberg, "Probabilities on a Compact Group".

Conditions for Sound Masking

What conditions the distributions  of each share must fit?

"Central Limit Theorem" (qualitative)[26]

Conv. to uniform $\iff$ support *not* contained in any non-trivial coset of $\mathbb{F}$

In $\mathbb{R}$: mild assumption

$\rightarrow$ Only $\mathbb{Z}$ and $\mathbb{Q}$ (and their respective subgroups)

$\rightarrow$ Negligible measure over $\mathbb{R}$

In finite $\mathbb{F}$: no longer mild in finite fields ...

[26] Stromberg, "Probabilities on a Compact Group".

Two Solutions

Two Solutions

Solution 1: Make sure to leak < 1 bit per share:

- Support of PMF always larger than any coset
- Work with any $\mathbb{F}$ (usually chosen to fit the cipher) ✓
- **Leakage-dependent: not always verified** ✗

Two Solutions

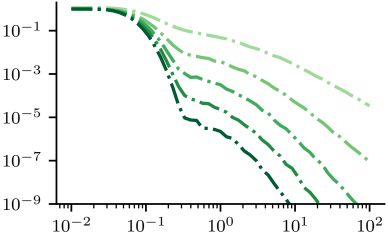
Solution 2: Choose $\mathbb{F}$ without any non-trivial subgroup, *i.e.*, $\mathbb{F}_p$, p prime:

- No assumption on the leakage ✓
- Major change of paradigm:

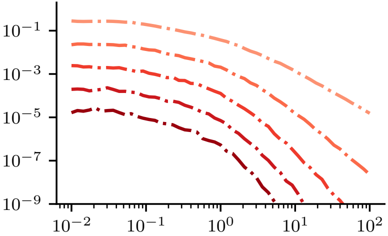
  Fix $\mathbb{F}$ masking-friendly first,

  Then build crypto upon it ✓

Comparing Binary and Prime Fields: a Simulation



(a) $\mathbb{F}_{2^7}$



(b) $\mathbb{F}_{2^7-1}$

Figure: Comparing binary and prime fields.

Content

Threshold Probing Model

Definition (t-privacy)

Any tuple of t intermediate values $\perp$ secrets

[27] Not $\iff$, see Bordes, "Security of symmetric primitives and their implementations", Example 5.5

Threshold Probing Model

Definition (t-privacy)

Any tuple of t intermediate values $\perp$ secrets

Definition (Simulatability)

A set of probes $\mathcal{P}$ in a circuit $\mathbb{C}$ can be simulated with the input shares $\mathcal{I}$ if there exists an algorithm $\mathcal{S}$ (the *simulator*) such that

$$\mathcal{P} \overset{d}{=} \mathcal{S}(\mathcal{I})$$

[27] Not $\iff$, see Bordes, "Security of symmetric primitives and their implementations", Example 5.5

Threshold Probing Model

Definition (t-privacy)

Any tuple of t intermediate values $\perp$ secrets

Definition (Simulatability)

A set of probes $\mathcal{P}$ in a circuit $\mathbb{C}$ can be simulated with the input shares $\mathcal{I}$ if there exists an algorithm $\mathcal{S}$ (the *simulator*) such that

$$\mathcal{P} \stackrel{d}{=} \mathcal{S}(\mathcal{I})$$

Definition (t-non-interference (NI))

$\mathbb{C}$ is t-NI if *any* set of t probes is simulatable by *at most* t shares of each input

[27] Not $\iff$, see Bordes, "Security of symmetric primitives and their implementations", Example 5.5

Threshold Probing Model

Definition (t-privacy)

Any tuple of t intermediate values $\perp$ secrets

Definition (Simulatability)

A set of probes $\mathcal{P}$ in a circuit $\mathbb{C}$ can be simulated with the input shares $\mathcal{I}$ if there exists an algorithm $\mathcal{S}$ (the *simulator*) such that

$$\mathcal{P} \stackrel{d}{=} \mathcal{S}(\mathcal{I})$$

Definition (t-non-interference (NI))

$\mathbb{C}$ is t-NI if *any* set of t probes is simulatable by *at most t* shares of each input

Q8: For a circuit with d shares, prove that d-NI $\implies$ d-privacy[27]

[27] Not $\iff$, see Bordes, "Security of symmetric primitives and their implementations", Example 5.5

Threshold Probing Model

Definition (t-privacy)

Any tuple of t intermediate values $\perp$ secrets

Definition (Simulatability)

A set of probes $\mathcal{P}$ in a circuit $\mathbb{C}$ can be simulated with the input shares $\mathcal{I}$ if there exists an algorithm $\mathcal{S}$ (the *simulator*) such that

$$\mathcal{P} \overset{d}{=} \mathcal{S}(\mathcal{I}) \iff \mathcal{P} \perp \text{all inputs except } \mathcal{I}$$

Definition (t-non-interference (NI))

$\mathbb{C}$ is t-NI if *any* set of t probes is simulatable by *at most t* shares of each input

Q8: For a circuit with d shares, prove that d-NI $\implies$ d-privacy[27]

[27] Not $\iff$, see Bordes, "Security of symmetric primitives and their implementations", Example 5.5

The Composition Paradigm

Idea to make a circuit NI:

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

→ Logical circuit made of `not`, `and` gates[28]

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

→ Logical circuit made of `not`, `and` gates[28]

→ Arithmetical circuit made of $\oplus, \otimes$ gates[29]

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

$\rightarrow$ Logical circuit made of `not`, `and` gates[28]

$\rightarrow$ Arithmetical circuit made of $\oplus, \otimes$ gates[29]

- Replace each gate by a masked *gadget* NI

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

→ Logical circuit made of `not`, `and` gates[28]

→ Arithmetical circuit made of $\oplus, \otimes$ gates[29]

- Replace each gate by a masked *gadget* NI
- Et voilà !

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

The Composition Paradigm

Idea to make a circuit NI:

- View your algorithm as a logical/arithmetical circuit

$\rightarrow$ Logical circuit made of `not`, `and` gates[28]

$\rightarrow$ Arithmetical circuit made of $\oplus, \otimes$ gates[29]

- Replace each gate by a masked *gadget* NI
- Et voilà !

Issue: NI is not composable[30] ✗**Q8**: example on white board

[28] Ishai, Sahai, and Wagner, "Private Circuits: Securing Hardware against Probing Attacks".
[29] Rivain and Prouff, "Provably Secure Higher-Order Masking of AES".
[30] Coron et al., "Higher-Order Side Channel Security and Mask Refreshing".

Strong Non-Interference[32]

Definition (t-Strong Non-Interference)

A gadget is t-SNI if any set of t_1 internal probes and t_2 output probes can be simulated with t_1 shares of each input sharing, and

$$t = t_1 + t_2$$

$\rightarrow$ SNI $\implies$ NI $\implies$ privacy

Other composable notions: SNIo, PINI[31], robust probing, glitch-extended, ...

[31] Cassiers and Standaert, "Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference".

[32] Barthe et al., "Strong Non-Interference and Type-Directed Higher-Order Masking".

Masked addition gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \oplus \left(\sum_i B_i\right)$$

Masked addition gadget

Inputs:

$$\begin{aligned}[\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d)\end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \oplus \left(\sum_i B_i\right)$$

SecAdd algorithm:

$$\begin{aligned}C_1 &= A_1 \oplus B_1 \\ &\vdots \\ C_d &= A_d \oplus B_d\end{aligned}$$

Masked addition gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \oplus \left(\sum_i B_i\right)$$

SecAdd algorithm:

$$\begin{aligned} C_1 &= A_1 \oplus B_1 \\ &\vdots \\ C_d &= A_d \oplus B_d \end{aligned}$$

- NI, but not SNI ✗

Masked addition gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \oplus \left(\sum_i B_i\right)$$

SecAdd algorithm:

$$\begin{aligned} C_1 &= A_1 \oplus B_1 \\ &\vdots \\ C_d &= A_d \oplus B_d \end{aligned}$$

- NI, but not SNI ✗
- t-NI + t-SNI refresh $\implies$ t-SNI ✓

Masked addition gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \oplus \left(\sum_i B_i\right)$$

SecAdd algorithm:

$$\begin{aligned} C_1 &= A_1 \oplus B_1 \\ &\vdots \\ C_d &= A_d \oplus B_d \end{aligned}$$

- NI, but not SNI ✗
- t-NI + t-SNI refresh $\implies$ t-SNI ✓
- Generalization: share-wise application of any affine map

Masked multiplication gadget

Inputs:

$$\begin{aligned}[\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d)\end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \otimes \left(\sum_i B_i\right)$$

Masked multiplication gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \otimes \left(\sum_i B_i\right)$$

BadMult algorithm:

$$\begin{aligned} C_1 &= (A_1 \otimes B_1 \quad) \oplus (A_1 \otimes B_2 \quad) \oplus (A_1 \otimes B_3 \quad) \\ C_2 &= (A_2 \otimes B_1 \quad) \oplus (A_2 \otimes B_2 \quad) \oplus (A_2 \otimes B_3 \quad) \\ C_3 &= (A_3 \otimes B_1 \quad) \oplus (A_3 \otimes B_2 \quad) \oplus (A_3 \otimes B_3 \quad) \end{aligned}$$

Correct, but not 2-NI. **Q7**: Why ?

Masked multiplication gadget

Inputs:

$$
\begin{aligned}
[\![A]\!] &= (A_1, \ldots, A_d) \\
[\![B]\!] &= (B_1, \ldots, B_d)
\end{aligned}
$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \otimes \left(\sum_i B_i\right)$$

BadMult algorithm:

$$
\begin{aligned}
C_1 &= (A_1 \otimes B_1 \quad) \oplus (A_1 \otimes B_2 \quad) \oplus (A_1 \otimes B_3 \quad) \\
C_2 &= (A_2 \otimes B_1 \quad) \oplus (A_2 \otimes B_2 \quad) \oplus (A_2 \otimes B_3 \quad) \\
C_3 &= (A_3 \otimes B_1 \quad) \oplus (A_3 \otimes B_2 \quad) \oplus (A_3 \otimes B_3 \quad)
\end{aligned}
$$

Correct, but not 2-NI. **Q7**: Why ?

Masked multiplication gadget

Inputs:

$$\begin{aligned} [\![A]\!] &= (A_1, \ldots, A_d) \\ [\![B]\!] &= (B_1, \ldots, B_d) \end{aligned}$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \otimes \left(\sum_i B_i\right)$$

SecMult algorithm:

$$\begin{aligned} C_1 &= (A_1 \otimes B_1 \quad\quad) \oplus (A_1 \otimes B_2 \oplus R_1) \oplus (A_1 \otimes B_3 \oplus R_2) \\ C_2 &= (A_2 \otimes B_1 \ominus R_1) \oplus (A_2 \otimes B_2 \quad\quad) \oplus (A_2 \otimes B_3 \oplus R_3) \\ C_3 &= (A_3 \otimes B_1 \ominus R_2) \oplus (A_3 \otimes B_2 \ominus R_3) \oplus (A_3 \otimes B_3 \quad\quad) \end{aligned}$$

- SecMult is $(d-1)$-SNI ✓

Masked multiplication gadget

Inputs:

$$
\begin{aligned}
[\![A]\!] &= (A_1, \ldots, A_d) \\
[\![B]\!] &= (B_1, \ldots, B_d)
\end{aligned}
$$

Output:

$$[\![C]\!] = (C_1, \ldots, C_d)$$

such that

$$\sum_i C_i = \left(\sum_i A_i\right) \otimes \left(\sum_i B_i\right)$$

SecMult algorithm:

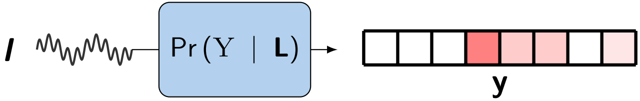
$$
\begin{aligned}
C_1 &= (A_1 \otimes B_1) \oplus (A_1 \otimes B_2 \oplus R_1) \oplus (A_1 \otimes B_3 \oplus R_2) \\
C_2 &= (A_2 \otimes B_1 \ominus R_1) \oplus (A_2 \otimes B_2) \oplus (A_2 \otimes B_3 \oplus R_3) \\
C_3 &= (A_3 \otimes B_1 \ominus R_2) \oplus (A_3 \otimes B_2 \ominus R_3) \oplus (A_3 \otimes B_3)
\end{aligned}
$$

- SecMult is $(d-1)$-SNI ✓
- If $[\![B]\!] = (1, 0, \ldots, 0)$, then SecMult$([\![A]\!], [\![B]\!])$ = Refresh$([\![A]\!])$ ✓

Content

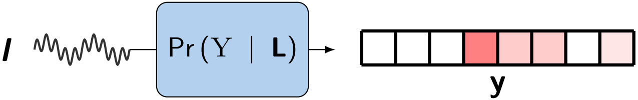
Recall on Noisy Leakage Model

Yesterday:

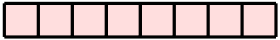


Recall on Noisy Leakage Model

Yesterday:

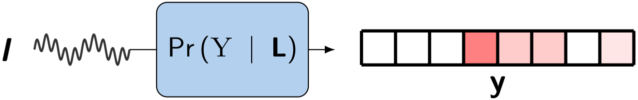


If, the adversary gets:

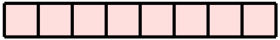


Recall on Noisy Leakage Model

Yesterday:



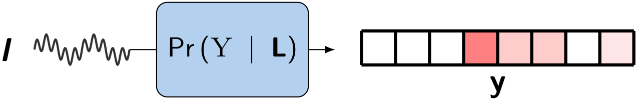
If, the adversary gets:



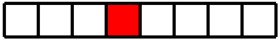
Very noisy leakage
Y indistinguishable from blind guess

Recall on Noisy Leakage Model

Yesterday:

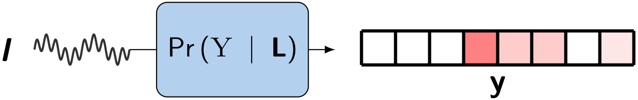


If, the adversary gets:



Recall on Noisy Leakage Model

Yesterday:



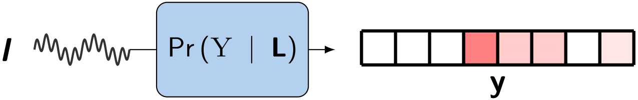
If, the adversary gets:

Low-noise leakage

Exact prediction for Y

Recall on Noisy Leakage Model

Yesterday:



δ-NOISY ADVERSARY

Any intermediate computation Y leaks $\mathrm{L}(\mathrm{Y})$ such that:

$$\mathsf{SD}\left(\mathrm{Y};\mathrm{L}\right) = \mathop{\mathbb{E}}_{\mathrm{L}}\left[TV \left(\underbrace{\qquad\qquad}_{\mathsf{Pr}(\mathrm{Y}\mid\mathrm{L})}, \underbrace{\qquad\qquad}_{\mathsf{Pr}(\mathrm{Y})} \right)\right] \leq \delta$$

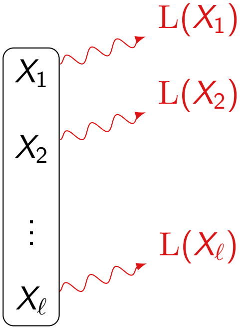
Security Proof for a Gadget

Consider a gadget with ℓ intermediate computations:

$$\boxed{\begin{matrix} X_1 \\ X_2 \\ \vdots \\ X_\ell \end{matrix}}$$

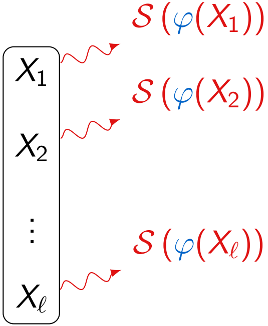
Security Proof for a Gadget

Consider a gadget with ℓ δ-noisy intermediate computations:



Security Proof for a Gadget

Consider a gadget with ℓ δ-noisy intermediate computations:



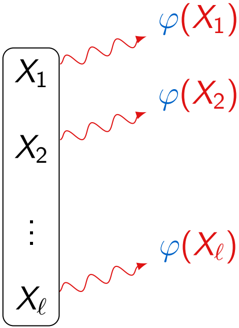
Lemma (Simulatability by RP)

The leakage function L *can be simulated from a random probing adversary:* $\varphi(x)$ *exactly reveals* x *with probability*
$\epsilon = 1 - \sum_l \min_x \Pr\left(\mathrm{L}(x) = l\right) \leq \delta \cdot |\mathbb{F}|$.[a]

[a] Duc, Dziembowski, and Faust, "Unifying Leakage Models: From Probing Attacks to Noisy Leakage".

Security Proof for a Gadget

Consider a gadget with ℓ δ-noisy intermediate computations:



We may reduce to an adversary observing $\varphi(X)$ instead of $\mathcal{S}(\varphi(X))$ (Data Processing Inequality)

Proof of the Core Lemma (I)

Assume there exists such a simulator $\mathcal{S}$,

Proof of the Core Lemma (I)

Assume there exists such a simulator $\mathcal{S}$, we need to construct it for all inputs:

$$\begin{aligned} \Pr(\mathcal{S}(x) = l) &= \ldots, \text{ for all } x \\ \Pr(\mathcal{S}(\perp) = l) &= \ldots \end{aligned}$$

Constraints:

Proof of the Core Lemma (I)

Assume there exists such a simulator $\mathcal{S}$, we need to construct it for all inputs:

$$\begin{aligned}\Pr(\mathcal{S}(x) = l) &= \ldots, \text{ for all } x \\ \Pr(\mathcal{S}(\perp) = l) &= \ldots\end{aligned}$$

Constraints:

$\rightarrow$ For all input x, $\Pr(\mathcal{S}(x))$ should be a p.m.f. $(2 \cdot |\mathbb{F}|$ (in)equations)

Proof of the Core Lemma (I)

Assume there exists such a simulator $\mathcal{S}$, we need to construct it for all inputs:

$$\begin{aligned} \Pr(\mathcal{S}(x) = l) &= \ldots, \text{ for all } x \\ \Pr(\mathcal{S}(\perp) = l) &= \ldots \end{aligned}$$

Constraints:

→ For all input x, $\Pr(\mathcal{S}(x))$ should be a p.m.f. $(2 \cdot |\mathbb{F}|$ (in)equations)

→ For the input $\perp$, $\Pr(\mathcal{S}(\perp))$ should be a p.m.f. (2 (in)equations)

Proof of the Core Lemma (I)

Assume there exists such a simulator $\mathcal{S}$, we need to construct it for all inputs:

$$\begin{aligned} \Pr\left(\mathcal{S}\left(x\right) = l\right) &= \ldots, \text{ for all } x \\ \Pr\left(\mathcal{S}\left(\perp\right) = l\right) &= \ldots \end{aligned}$$

Constraints:

→ For all input x, $\Pr\left(\mathcal{S}\left(x\right)\right)$ should be a p.m.f. ($2 \cdot |\mathbb{F}|$ (in)equations)

→ For the input $\perp$, $\Pr\left(\mathcal{S}\left(\perp\right)\right)$ should be a p.m.f. (2 (in)equations)

→ For any x, l, $\Pr\left(\mathcal{S}\left(\varphi(x)\right) = l\right) = \Pr\left(\mathrm{L}(x) = l\right)$ ($|\mathbb{F}| \times |\mathcal{L}|$ equations)

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\Pr\left(\mathrm{L}(x) = l\right) = \Pr\left(\mathcal{S}\left(\varphi(x)\right) = l\right)$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned} \Pr(\mathrm{L}(x) = l) &= \Pr(\mathcal{S}(\varphi(x)) = l) \\ &= \Pr(\varphi(x) = x) \cdot \Pr(\mathcal{S}(x) = l) + \Pr(\varphi(x) = \perp) \cdot \Pr(\mathcal{S}(\perp) = l) \end{aligned}$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned}
\Pr\left(\mathrm{L}(x) = l\right) &= \Pr\left(\mathcal{S}\left(\varphi(x)\right) = l\right) \\
&= \Pr\left(\varphi(x) = x\right) \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + \Pr\left(\varphi(x) = \perp\right) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right) \\
&= \epsilon \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + (1 - \epsilon) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right)
\end{aligned}$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned}
\Pr\left(\mathrm{L}(x) = l\right) &= \Pr\left(\mathcal{S}\left(\varphi(x)\right) = l\right) \\
&= \Pr\left(\varphi(x) = x\right) \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + \Pr\left(\varphi(x) = \perp\right) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right) \\
&= \epsilon \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + (1 - \epsilon) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right)
\end{aligned}$$

Hence,

$$0 \leq \Pr\left(\mathcal{S}\left(\perp\right) = l\right) = \frac{\overbrace{\Pr\left(\mathrm{L}(x) = l\right) - \epsilon \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right)}^{\text{Should not depend on } X}}{1 - \epsilon}$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned}
\Pr(\mathrm{L}(x) = l) &= \Pr(\mathcal{S}(\varphi(x)) = l) \\
&= \Pr(\varphi(x) = x) \cdot \Pr(\mathcal{S}(x) = l) + \Pr(\varphi(x) = \perp) \cdot \Pr(\mathcal{S}(\perp) = l) \\
&= \epsilon \cdot \Pr(\mathcal{S}(x) = l) + (1 - \epsilon) \cdot \Pr(\mathcal{S}(\perp) = l)
\end{aligned}$$

Hence,

$$0 \leq \Pr(\mathcal{S}(\perp) = l) = \frac{\overbrace{\Pr(\mathrm{L}(x) = l) - \epsilon \cdot \Pr(\mathcal{S}(x) = l)}^{\text{Should not depend on } x}}{1 - \epsilon} = \frac{\pi(l)}{1 - \epsilon} \quad (2)$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned}
\Pr\left(\mathrm{L}(x) = l\right) &= \Pr\left(\mathcal{S}\left(\varphi(x)\right) = l\right) \\
&= \Pr\left(\varphi(x) = x\right) \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + \Pr\left(\varphi(x) = \perp\right) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right) \\
&= \epsilon \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right) + (1 - \epsilon) \cdot \Pr\left(\mathcal{S}\left(\perp\right) = l\right)
\end{aligned}$$

Hence,

$$0 \leq \Pr\left(\mathcal{S}\left(\perp\right) = l\right) = \frac{\overbrace{\Pr\left(\mathrm{L}(x) = l\right) - \epsilon \cdot \Pr\left(\mathcal{S}\left(x\right) = l\right)}^{\text{Should not depend on } X}}{1 - \epsilon} = \frac{\pi(l)}{1 - \epsilon} \quad (2)$$

$$0 \leq \Pr\left(\mathcal{S}\left(x\right) = l\right) = \frac{\Pr\left(\mathrm{L}(x) = l\right) - \pi(l)}{\epsilon} \quad (3)$$

Proof of the Core Lemma (II)

Let us start from the last constraint. For any x and any l:

$$\begin{aligned}
\Pr(\mathrm{L}(x) = l) &= \Pr(\mathcal{S}(\varphi(x)) = l) \\
&= \Pr(\varphi(x) = x) \cdot \Pr(\mathcal{S}(x) = l) + \Pr(\varphi(x) = \perp) \cdot \Pr(\mathcal{S}(\perp) = l) \\
&= \epsilon \cdot \Pr(\mathcal{S}(x) = l) + (1 - \epsilon) \cdot \Pr(\mathcal{S}(\perp) = l)
\end{aligned}$$

Hence,

$$0 \leq \Pr(\mathcal{S}(\perp) = l) = \frac{\overbrace{\Pr(\mathrm{L}(x) = l) - \epsilon \cdot \Pr(\mathcal{S}(x) = l)}^{\text{Should not depend on } X}}{1 - \epsilon} = \frac{\pi(l)}{1 - \epsilon} \quad (2)$$

$$0 \leq \Pr(\mathcal{S}(x) = l) = \frac{\Pr(\mathrm{L}(x) = l) - \pi(l)}{\epsilon} \quad (3)$$

Is there any ϵ such that $\geq$ and $\geq$ are valid?

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid?

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr\left(\mathrm{L}(x) = l\right) \text{ for any } x$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr\left(\mathrm{L}(x) = l\right) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_{x} \Pr\left(\mathrm{L}(x) = l\right)$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr\left(\mathrm{L}(x) = l\right) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_x \Pr\left(\mathrm{L}(x) = l\right)$$

Furthermore, summing (2) over l, by definition of probability distributions,

$$\sum_l \pi(l) = \underbrace{\sum_l \Pr\left(\mathrm{L}(x) = l\right)}_{=1} - \epsilon \cdot \underbrace{\sum_l \Pr\left(\mathcal{S}\left(x\right) = l\right)}_{=1}$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr\left(\mathrm{L}(x) = l\right) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_x \Pr\left(\mathrm{L}(x) = l\right)$$

Furthermore, summing (2) over l, by definition of probability distributions,

$$\sum_l \pi(l) = \underbrace{\sum_l \Pr\left(\mathrm{L}(x) = l\right)}_{=1} - \epsilon \cdot \underbrace{\sum_l \Pr\left(\mathcal{S}(x) = l\right)}_{=1} = 1 - \epsilon$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr(\mathrm{L}(x) = l) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_{x} \Pr(\mathrm{L}(x) = l)$$

Furthermore, summing (2) over l, by definition of probability distributions,

$$\sum_{l} \pi(l) = \underbrace{\sum_{l} \Pr(\mathrm{L}(x) = l)}_{=1} - \epsilon \cdot \underbrace{\sum_{l} \Pr(\mathcal{S}(x) = l)}_{=1} = 1 - \epsilon$$

Hence,

$$\epsilon = 1 - \sum_{l} \pi(l) \geq 1 - \sum_{l} \min_{x} \Pr(\mathrm{L}(x) = l)$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr(\mathrm{L}(x) = l) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_x \Pr(\mathrm{L}(x) = l)$$

Furthermore, summing (2) over l, by definition of probability distributions,

$$\sum_l \pi(l) = \underbrace{\sum_l \Pr(\mathrm{L}(x) = l)}_{=1} - \epsilon \cdot \underbrace{\sum_l \Pr(\mathcal{S}(x) = l)}_{=1} = 1 - \epsilon$$

Hence, to have the smallest ϵ,

$$\epsilon = 1 - \sum_l \pi(l) = 1 - \sum_l \min_x \Pr(\mathrm{L}(x) = l)$$

Proof of the Core Lemma (III)

Is there any ϵ such that $\geq$ and $\geq$ are valid? From (2), and (3), we get

$$0 \leq \pi(l) \leq \Pr(\mathrm{L}(x) = l) \text{ for any } x$$

In other words,

$$0 \leq \pi(l) \leq \min_x \Pr(\mathrm{L}(x) = l)$$

Furthermore, summing (2) over l, by definition of probability distributions,

$$\sum_l \pi(l) = \underbrace{\sum_l \Pr(\mathrm{L}(x) = l)}_{=1} - \epsilon \cdot \underbrace{\sum_l \Pr(\mathcal{S}(x) = l)}_{=1} = 1 - \epsilon$$

Hence, to have the smallest ϵ,

$$\epsilon = 1 - \sum_l \pi(l) = 1 - \sum_l \min_x \Pr(\mathrm{L}(x) = l) \leq \delta \cdot |\mathbb{F}| \text{ } (\textbf{Q11}\text{: prove it})$$

Security against a Random Probing Adversary

To succeed, at least d out of ℓ wires must be revealed to the adversary:

$$\Pr\left(\text{Adv. wins}\right) \leq \Pr\left(\text{At least } d \text{ wires revealed}\right)$$

[33] Boucheron, Lugosi, and Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, P.24, and Ex. 2.11.

Security against a Random Probing Adversary

To succeed, at least d out of ℓ wires must be revealed to the adversary:

$$\Pr(\text{Adv. wins}) \leq \Pr(\text{At least } d \text{ wires revealed})$$

Theorem (Chernoff Concentration Inequality)

If ℓ wires, each independently revealed with proba. ϵ:

$$\Pr(\textit{At least } d \textit{ wires revealed}) \leq \left(\frac{e \cdot \ell \cdot \epsilon}{d}\right)^d$$

[33] Boucheron, Lugosi, and Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, P.24, and Ex. 2.11.

Security against a Random Probing Adversary

To succeed, at least d out of ℓ wires must be revealed to the adversary:

$$\Pr(\text{Adv. wins}) \leq \Pr(\text{At least } d \text{ wires revealed})$$

Theorem (Chernoff Concentration Inequality)

If ℓ wires, each independently revealed with proba. ϵ:

$$\Pr(\textit{At least } d \textit{ wires revealed}) \leq \left(\frac{e \cdot \ell \cdot \epsilon}{d}\right)^d$$

Q11: Prove the inequality from a particular case of Chernoff inequality[33]

[33] Boucheron, Lugosi, and Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, P.24, and Ex. 2.11.

Putting all Together

In our context, $\ell \leq \mathcal{O}\left(d^2\right)$ (for $\otimes$ gadget), and $\epsilon \leq \delta \cdot |\mathbb{F}|$:

Theorem (Security Bound)

For a single gadget with $\ell \leq \mathcal{O}\left(d^2\right)$ intermediate computations:

$$\mathsf{SD}\left(k; \mathbf{L}\right) \leq \mathcal{O}\left(\left(7e \cdot d \cdot \delta \cdot |\mathbb{F}|\right)^d\right)$$

[34] *t*-Region-probing secure: NI, with *t* probes from *each* gadget

Putting all Together

In our context, $\ell \leq \mathcal{O}\left(d^2\right)$ (for $\otimes$ gadget), and $\epsilon \leq \delta \cdot |\mathbb{F}|$:

Theorem (Security Bound)

For a single gadget with $\ell \leq \mathcal{O}\left(d^2\right)$ intermediate computations:

$$\mathsf{SD}\left(k; \mathbf{L}\right) \leq \mathcal{O}\left(\left(7e \cdot d \cdot \delta \cdot |\mathbb{F}|\right)^d\right)$$

For the whole circuit $\mathbb{C}$,

$$\mathsf{SD}\left(k; \mathbf{L}\right) \leq \mathcal{O}\left(\left(7e \cdot |\mathbb{C}| \cdot d \cdot \delta \cdot |\mathbb{F}|\right)^d\right)$$

[34] t-Region-probing secure: NI, with t probes from *each* gadget

Putting all Together

In our context, $\ell \leq \mathcal{O}\left(d^2\right)$ (for $\otimes$ gadget), and $\epsilon \leq \delta \cdot |\mathbb{F}|$:

Theorem (Security Bound)

For a single gadget with $\ell \leq \mathcal{O}\left(d^2\right)$ intermediate computations:

$$\mathsf{SD}\left(k; \mathbf{L}\right) \leq \mathcal{O}\left(\left(7e \cdot d \cdot \delta \cdot |\mathbb{F}|\right)^{d}\right)$$

For the whole circuit $\mathbb{C}$, $d/2$-region probing[34] security implies

$$\mathsf{SD}\left(k; \mathbf{L}\right) \leq \mathcal{O}\left(|\mathbb{C}| \left(7e \cdot d \cdot \delta \cdot |\mathbb{F}|\right)^{d/2}\right)$$

[34] t-Region-probing secure: NI, with t probes from *each* gadget

Remarks on the Proof

- "Exponential" security ✓

[35] Brian, Dziembowski, and Faust, "From Random Probing to Noisy Leakages Without Field-Size Dependence".

[36] Belaïd, Rivain, and Taleb, "On the Power of Expansion: More Efficient Constructions in the Random Probing Model".

Remarks on the Proof

- "Exponential" security ✓
- Bad *leakage rate* $\tau = 7e \cdot d \cdot |\mathbb{F}|$ ✗, but:

[35] Brian, Dziembowski, and Faust, "From Random Probing to Noisy Leakages Without Field-Size Dependence".

[36] Belaïd, Rivain, and Taleb, "On the Power of Expansion: More Efficient Constructions in the Random Probing Model".

Remarks on the Proof

- "Exponential" security ✓
- Bad *leakage rate* $\tau = 7e \cdot d \cdot |\mathbb{F}|$ ✗, but:

→ The $|\mathbb{F}|$ factor is a proof artifact[35] ✓

[35] Brian, Dziembowski, and Faust, "From Random Probing to Noisy Leakages Without Field-Size Dependence".

[36] Belaïd, Rivain, and Taleb, "On the Power of Expansion: More Efficient Constructions in the Random Probing Model".

Remarks on the Proof

- "Exponential" security ✓
- Bad *leakage rate* $\tau = 7e \cdot d \cdot |\mathbb{F}|$ ✗, but:

→ The $|\mathbb{F}|$ factor is a proof artifact[35] ✓

→ New constructions with better (even constant) leakage rates[36] ✓

[35] Brian, Dziembowski, and Faust, "From Random Probing to Noisy Leakages Without Field-Size Dependence".

[36] Belaïd, Rivain, and Taleb, "On the Power of Expansion: More Efficient Constructions in the Random Probing Model".

Content

Masking Post-Quantum Cryptography: Kyber

- Basic arithmetic over $\mathbb{Z}_q$, with q prime
  - ✓ Friendly with arithmetic masking
- *Fujisaki-Okamoto* transform:
  - ✗ Needs to convert masks A2B: complexity $\mathcal{O}\left(d^2 \log(d)\right)$
  - ✗ Needs to convert masks B2A: complexity $\mathcal{O}\left(d^2\right)$
  - ✗ Needs to mask hash functions: very expensive

Slow ops unmasked (NTT) become “fast” with higher-order masking ✓
Fast ops unmasked (rejection in Dilithium) become slow with higher-order masking ✗
Alternative masking-friendly signature schemes proposed (Raccoon) ✓

Content

Conclusion

Challenges for masking generally:

$\rightarrow$ Improving the reduction from noisy leakage to probing security

$\rightarrow$ Can we prove directly in the noisy model ?

$\rightarrow$ What about non-independent leakage randomness ?

Conclusion

Challenges for masking generally:

→ Improving the reduction from noisy leakage to probing security

→ Can we prove directly in the noisy model ?

→ What about non-independent leakage randomness ?

Challenges for masking in PQC:

→ Analysis in the RP/noisy model: current implementations deviate from the arithmetic circuit ✓

→ Having masking-friendly primitives

→ Make masked BIKE affordable

→ Masking MQ-like: not thoroughly explored yet ...

Pointers

Interested ?

- Coron's keynote at CARDIS 23 on masking lattice-based cryptography
- Cassiers' keynote at COSADE 23 on masking composability
- Nicolas Bordes' thesis with nice examples of probing notions.

References I

Barthe, G. et al. “Strong Non-Interference and Type-Directed Higher-Order Masking”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 116129. ISBN: 9781450341394. DOI: 10.1145/2976749.2978427. URL: https://doi.org/10.1145/2976749.2978427.

References II

Béguinot, J. et al. "Removing the Field Size Loss from Duc et al.'s Conjectured Bound for Masked Encodings". In: *Constructive Side-Channel Analysis and Secure Design - 14th International Workshop, COSADE 2023, Munich, Germany, April 3-4, 2023, Proceedings*. Ed. by E. B. Kavun and M. Pehl. Vol. 13979. Lecture Notes in Computer Science. Springer, 2023, pp. 86–104. DOI: 10.1007/978-3-031-29497-6_5. URL: https://doi.org/10.1007/978-3-031-29497-6_5.

References III

- Belaïd, S., M. Rivain, and A. R. Taleb. "On the Power of Expansion: More Efficient Constructions in the Random Probing Model". In: *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*. Ed. by A. Canteaut and F. Standaert. Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 313–343. DOI: 10.1007/978-3-030-77886-6_11. URL: https://doi.org/10.1007/978-3-030-77886-6_11.
- Bordes, N. "Security of symmetric primitives and their implementations". Theses. Université Grenoble Alpes [2020-....], Dec. 2021. URL: https://theses.hal.science/tel-03675249.

References IV

- Boucheron, S., G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN: 9780191747106. URL: https://books.google.fr/books?id=O3yoAQAACAAJ.
- Brian, G., S. Dziembowski, and S. Faust. "From Random Probing to Noisy Leakages Without Field-Size Dependence". In: *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part IV*. Ed. by M. Joye and G. Leander. Vol. 14654. Lecture Notes in Computer Science. Springer, 2024, pp. 345–374. DOI: 10.1007/978-3-031-58737-5_13. URL: https://doi.org/10.1007/978-3-031-58737-5_13.

References V

Cagli, E., C. Dumas, and E. Prouff. "Convolutional Neural Networks with Data Augmentation Against Jitter-Based Countermeasures - Profiling Attacks Without Pre-processing". In: *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*. Ed. by W. Fischer and N. Homma. Vol. 10529. Lecture Notes in Computer Science. Springer, 2017, pp. 45–68. ISBN: 978-3-319-66786-7. DOI: 10.1007/978-3-319-66787-4_3. URL: https://doi.org/10.1007/978-3-319-66787-4_3.

References VI

Cassiers, G. and F.-X. Standaert. "Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference". In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 2542–2555. DOI: 10.1109/TIFS.2020.2971153.

Chari, S. et al. "Towards Sound Approaches to Counteract Power-Analysis Attacks". In: *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Ed. by M. J. Wiener. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 398–412. ISBN: 3-540-66347-9. DOI: 10.1007/3-540-48405-1_26. URL: https://doi.org/10.1007/3-540-48405-1_26.

References VII

Clavier, C. and K. Gaj, eds. *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Vol. 5747. Lecture Notes in Computer Science. Springer, 2009. ISBN: 978-3-642-04137-2. DOI: 10.1007/978-3-642-04138-9. URL: https://doi.org/10.1007/978-3-642-04138-9.

References VIII

Coron, J. and I. Kizhvatov. "An Efficient Method for Random Delay Generation in Embedded Software". In: *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Ed. by C. Clavier and K. Gaj. Vol. 5747. Lecture Notes in Computer Science. Springer, 2009, pp. 156–170. ISBN: 978-3-642-04137-2. DOI: 10.1007/978-3-642-04138-9_12. URL: https://doi.org/10.1007/978-3-642-04138-9_12.

References IX

- Coron, J. et al. "Higher-Order Side Channel Security and Mask Refreshing". In: *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*. Ed. by S. Moriai. Vol. 8424. Lecture Notes in Computer Science. Springer, 2013, pp. 410–424. DOI: 10.1007/978-3-662-43933-3_21. URL: https://doi.org/10.1007/978-3-662-43933-3_21.
- Cover, T. M. and J. A. Thomas. *Elements of information theory (2. ed.)* Wiley, 2006. ISBN: 978-0-471-24195-9.
- David, L. and A. Wool. *A Bounded-Space Near-Optimal Key Enumeration Algorithm for Multi-Dimensional Side-Channel Attacks*. Cryptology ePrint Archive, Paper 2015/1236. https://eprint.iacr.org/2015/1236. 2015. URL: https://eprint.iacr.org/2015/1236.

References X

Duc, A., S. Dziembowski, and S. Faust. "Unifying Leakage Models: From Probing Attacks to Noisy Leakage". In: *J. Cryptology* 32.1 (2019), pp. 151–177. DOI: 10.1007/s00145-018-9284-1. URL: https://doi.org/10.1007/s00145-018-9284-1.

Goubin, L. and J. Patarin. "DES and Differential Power Analysis (The "Duplication" Method)". In: *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*. Ed. by Ç. K. Koç and C. Paar. Vol. 1717. Lecture Notes in Computer Science. Springer, 1999, pp. 158–172. DOI: 10.1007/3-540-48059-5_15. URL: https://doi.org/10.1007/3-540-48059-5_15.

References XI

Heuser, A., O. Rioul, and S. Guilley. "Good Is Not Good Enough - Deriving Optimal Distinguishers from Communication Theory". In: *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*. Ed. by L. Batina and M. Robshaw. Vol. 8731. Lecture Notes in Computer Science. Springer, 2014, pp. 55–74. ISBN: 978-3-662-44708-6. DOI: 10.1007/978-3-662-44709-3_4. URL: https://doi.org/10.1007/978-3-662-44709-3_4.

References XII

Ishai, Y., A. Sahai, and D. A. Wagner. "Private Circuits: Securing Hardware against Probing Attacks". In: *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*. Ed. by D. Boneh. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27. URL: https://doi.org/10.1007/978-3-540-45146-4_27.

Ito, A., R. Ueno, and N. Homma. "Perceived Information Revisited: New Metrics to Evaluate Success Rate of Side-Channel Attacks". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2022.4 (2022), pp. 228254. DOI: 10.46586/tches.v2022.i4.228-254. URL: https://tches.iacr.org/index.php/TCHES/article/view/9819.

References XIII

- Koç, Ç. K., ed. *Cryptographic Engineering*. Springer, 2009. ISBN: 978-0-387-71816-3. DOI: 10.1007/978-0-387-71817-0. URL: https://doi.org/10.1007/978-0-387-71817-0.
- Maghrebi, H., T. Portigliatti, and E. Prouff. "Breaking Cryptographic Implementations Using Deep Learning Techniques". In: *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*. Ed. by C. Carlet, M. A. Hasan, and V. Saraswat. Vol. 10076. Lecture Notes in Computer Science. Springer, 2016, pp. 3–26. ISBN: 978-3-319-49444-9. DOI: 10.1007/978-3-319-49445-6_1. URL: https://doi.org/10.1007/978-3-319-49445-6_1.

References XIV

- Mangard, S., E. Oswald, and T. Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007. ISBN: 978-0-387-30857-9.
- Masure, L. et al. "Deep Learning Side-Channel Analysis on Large-Scale Traces - A Case Study on a Polymorphic AES". In: *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part I*. Ed. by L. Chen et al. Vol. 12308. Lecture Notes in Computer Science. Springer, 2020, pp. 440–460. DOI: 10.1007/978-3-030-58951-6_22. URL: https://doi.org/10.1007/978-3-030-58951-6_22.

References XV

Masure, L. et al. "Information Bounds and Convergence Rates for Side-Channel Security Evaluators". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2023.3 (2023), pp. 522569. DOI: 10.46586/tches.v2023.i3.522-569. URL: https://tches.iacr.org/index.php/TCHES/article/view/10973.

Paninski, L. "Estimation of Entropy and Mutual Information". In: *Neural Comput.* 15.6 (2003), pp. 1191–1253.

References XVI

Prouff, E. "DPA Attacks and S-Boxes". In: *Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers*. Ed. by H. Gilbert and H. Handschuh. Vol. 3557. Lecture Notes in Computer Science. Springer, 2005, pp. 424–441. DOI: 10.1007/11502760_29. URL: https://doi.org/10.1007/11502760_29.

References XVII

Rivain, M. and E. Prouff. "Provably Secure Higher-Order Masking of AES". In: *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*. Ed. by S. Mangard and F. Standaert. Vol. 6225. Lecture Notes in Computer Science. Springer, 2010, pp. 413–427. DOI: 10.1007/978-3-642-15031-9_28. URL: https://doi.org/10.1007/978-3-642-15031-9_28.

References XVIII

- Rivain, M., E. Prouff, and J. Doget. "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers". In: *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Ed. by C. Clavier and K. Gaj. Vol. 5747. Lecture Notes in Computer Science. Springer, 2009, pp. 171–188. ISBN: 978-3-642-04137-2. DOI: 10.1007/978-3-642-04138-9_13. URL: https://doi.org/10.1007/978-3-642-04138-9_13.
- Stromberg, K. "Probabilities on a Compact Group". In: *Transactions of the American Mathematical Society* 94.2 (1960), pp. 295–309. ISSN: 00029947. URL: http://www.jstor.org/stable/1993313.

References XIX

Timon, B. “Non-Profiled Deep Learning-based Side-Channel attacks with Sensitivity Analysis”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.2 (2019), pp. 107–131. DOI: 10.13154/tches.v2019.i2.107-131. URL: https://tches.iacr.org/index.php/TCHES/article/view/7387.