Introduction to Masking Protection for Symmetric Encryption

Fabrice Lozachmeur

Introduction to Masking Protection for Symmetric Encryption

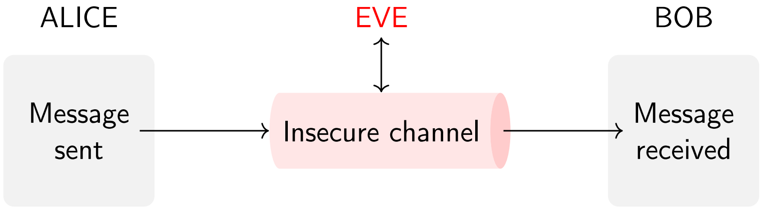
Fabrice LOZACHMEUR

Thalès – Lab-STICC

PhD thesis supervised by A. TISSERAND

May 23, 2022

The Role of Cryptography in Information Security



Context:

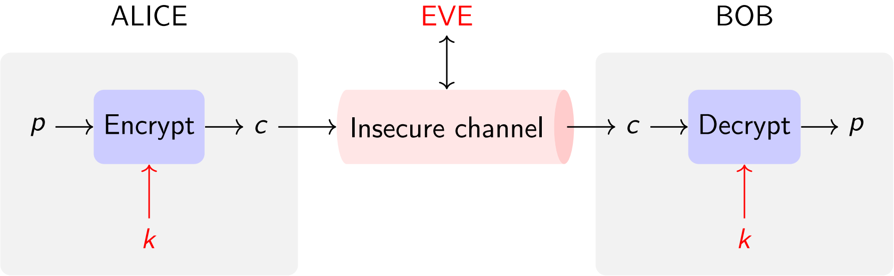
Alice communicates with Bob through an insecure channel.

Cryptography:

Provides different features in information security:

- Confidentiality
- Integrity
- Authenticity
- Non-repudiation

How to Achieve Data Confidentiality

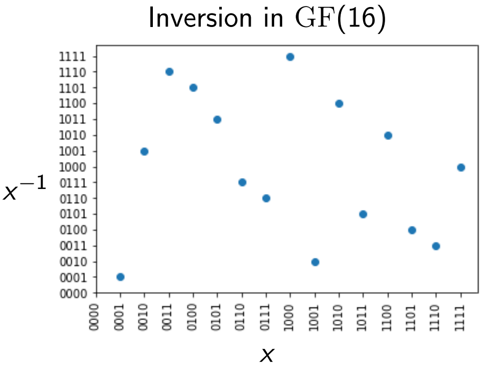


Symmetric cryptosystem:

- An encryption function encrypts a plaintext p into a ciphertext c using a secret key k
- A decryption function decrypts the ciphertext c into the plaintext p using the same secret key k

The only secret parameter is the secret key k.

Properties for a Strong Cryptosystem



Strong cryptosystem:

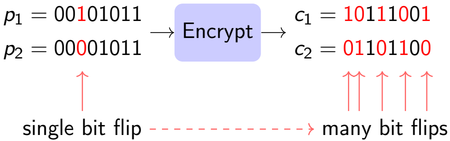
- Good confusion
- Good diffusion

Good confusion:

- The relationship between p and c is complex
- Non-linear functions

Good diffusion:

- Small modification(s) on p must impact many bits on c
- Linear functions



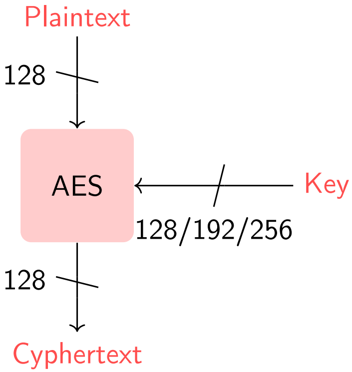
Advanced Encryption Standard (AES)

History[1]:

- Selection started in 1997 by NIST[2]
- AES algorithm selected in 2000

Requirements for AES candidates:

- Block cipher with 128-bit block size
- Supported key lengths:
  128, 192 and 256 bits
- Efficient in software and hardware
- Secure against known attacks



[1] https://competitions.cr.yp.to/aes.html
[2] National Institute of Standards and Technology

The Finite Field GF(2)

Definition:

$\mathrm{GF}(2)$ is the set $\{0, 1\}$ with operations modulo 2.

Operands		Integer operations			GF(2) operations			Logic gates	
a	b	$a + b$	$a - b$	$a \cdot b$	$a + b \bmod 2$	$a - b \bmod 2$	$a \cdot b \bmod 2$	$a \oplus b$	$a \wedge b$
0	0	0	0	0	0	0	0	0	0
0	1	1	-1	0	1	1	0	1	0
1	0	1	1	0	1	1	0	1	0
1	1	2	0	1	0	0	1	0	1

Remarks:

- Addition and subtraction modulo 2 are the same operation
- Addition modulo 2 is equivalent to a XOR gate $\oplus$
- Multiplication modulo 2 is equivalent to an AND gate $\wedge$

The Finite Field $\mathrm{GF}(2^8)$

Representation of elements:

Polynomials of degree 7 with coefficients in $\mathrm{GF}(2)$:

$$a_0 + a_1 X + a_2 X^2 + \ldots + a_7 X^7, \quad a_i \in \mathrm{GF}(2)$$

is represented by the byte $a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7$

Addition and subtraction in $\mathrm{GF}(2^8)$:

- Polynomial addition
- Bitwise XOR of bytes

Multiplication in $\mathrm{GF}(2^8)$:

Polynomial multiplication modulo an irreducible polynomial of degree 8.

Inverse in $\mathrm{GF}(2^8)$:

Every non-zero element a has an unique inverse a^{-1}.

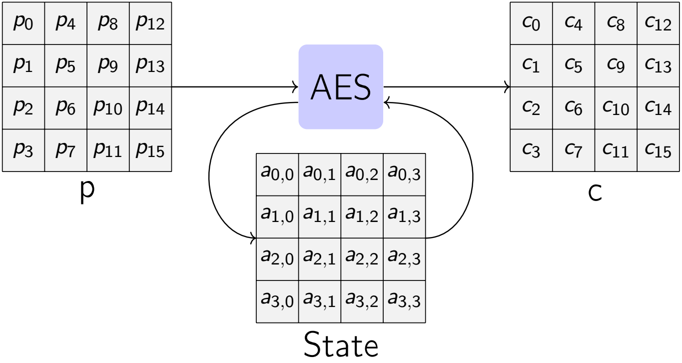
AES State [DR02]

State:

A 4×4 matrix of elements of $\mathrm{GF}(2^8)$.

Initial and final states:

- At the beginning, the state is the plaintext
- At the end, the state is the ciphertext



AES Encryption

AES principle:

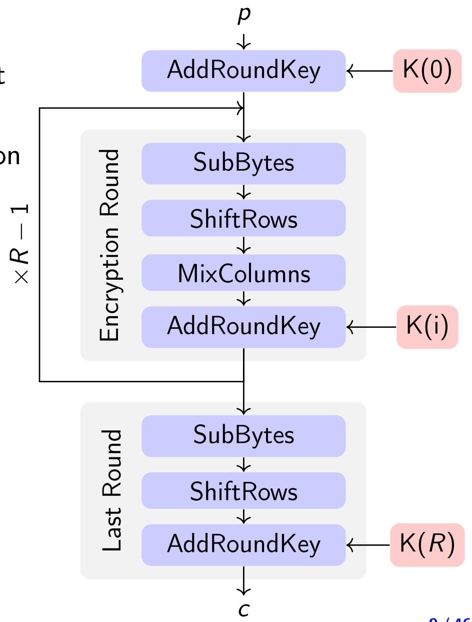
- Repeat on the state a round that brings confusion and diffusion
- The number of rounds depends on the key size

Round keys:

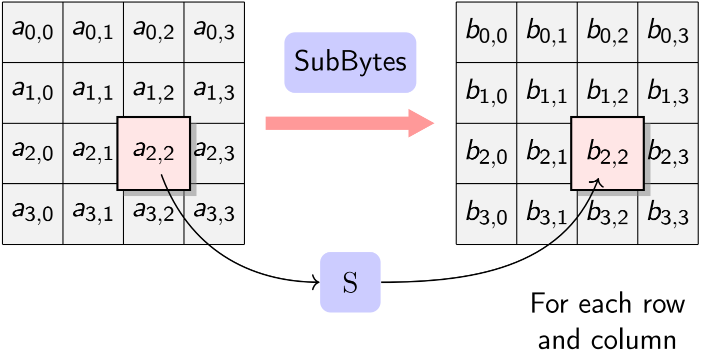
$K(i)$ are 128 bits round keys derived from the secret key.

Sub-functions:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

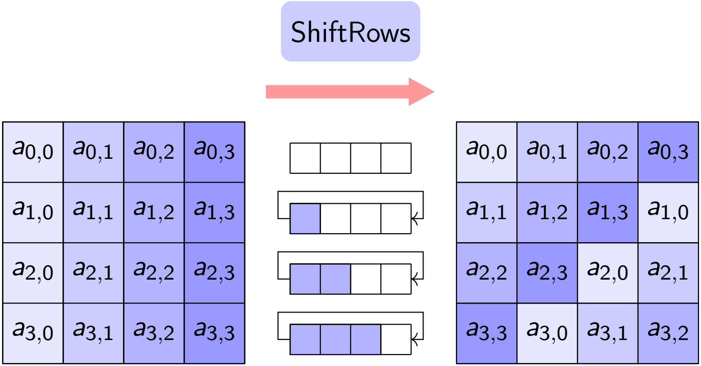


SubBytes



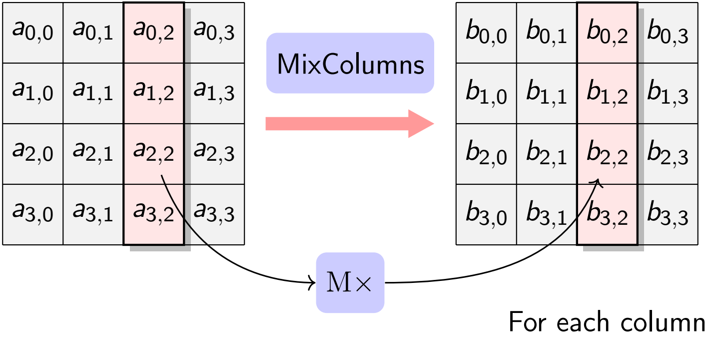
- Apply the S function to each state byte
- Provides confusion thanks to a carefully chosen nonlinear function
- S is called SBOX when it is implemented as a table.

ShiftRows



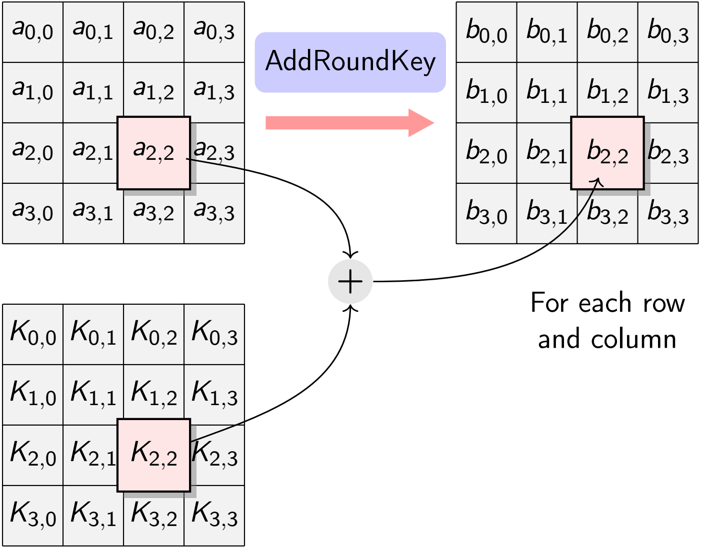
- Rows of the state matrix are shifted cyclically
- Provides inter column diffusion

MixColumns



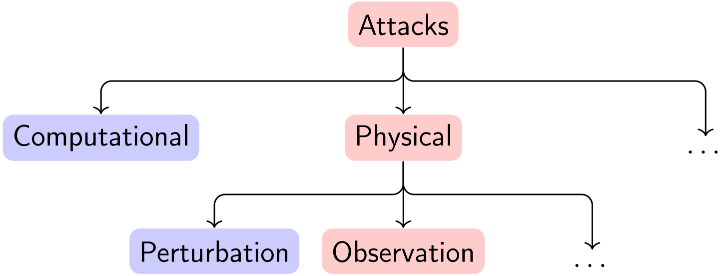
- Multiply each column by the matrix M
- Mixes each column of the state matrix

AddRoundKey



K: round key

Types of Attacks



Goal:

Guess the secret key or information on p.

Different types of attacks:

- Computational
- Physical
- ...

Available data:

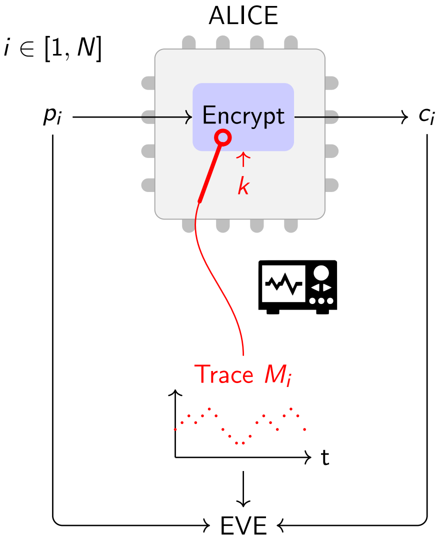
- Plaintexts and/or ciphertexts
- Physical measurements during each encryption

Measured physical quantities:

- Power consumption [KJJ99]
- Electromagnetic radiation [QS01]
- ...

Good book:

Power Analysis Attacks: Revealing the Secrets of Smart Cards [MOP07].



Power Consumption in Digital Circuits

Power Consumption:

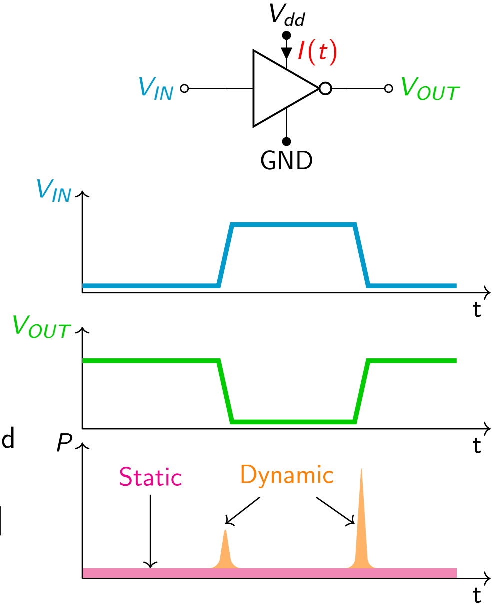
$P(t) = V_{dd} \times I(t)$

Two components:

- Static consumption
- Dynamic consumption

Remark:

- Power depends on operands and their transitions
- Can reveal information [KJJ99]



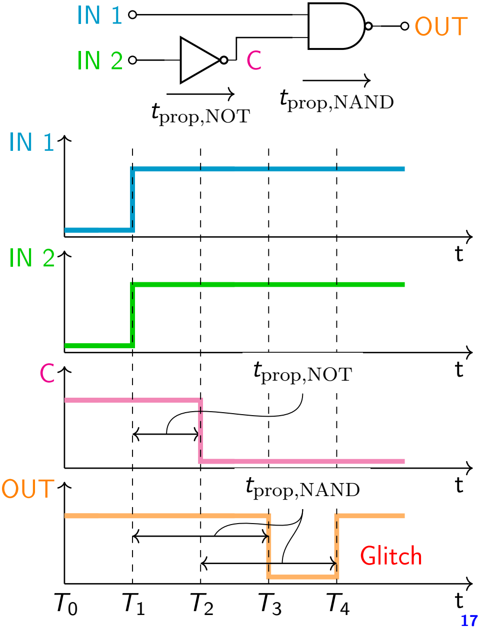
Glitch

Glitch:

Propagation delays can cause temporary modifications of output signals.

Remark:

- Glitches consume power
- Power depends on operands and their transitions
- Can reveal information [MPG05]



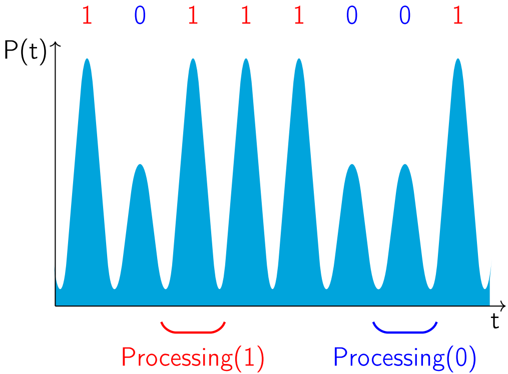
Simple Power Analysis (SPA) [KJJ99]

Instructions executed depend on the value of the key.
Exploits Correlation(s) between:

- Amplitude(s) of the measured physical signal
- Value(s) of the secret key bits

A single trace makes it possible to recover the secret key.

for $i = 1$ **to** n **do**
    Processing($K[i]$)
end for



SPA Limits

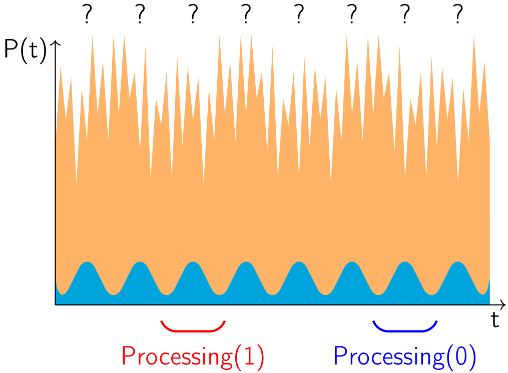
Limitations:

Small differences and too much noise $\Rightarrow$ one cannot distinguish key bits.

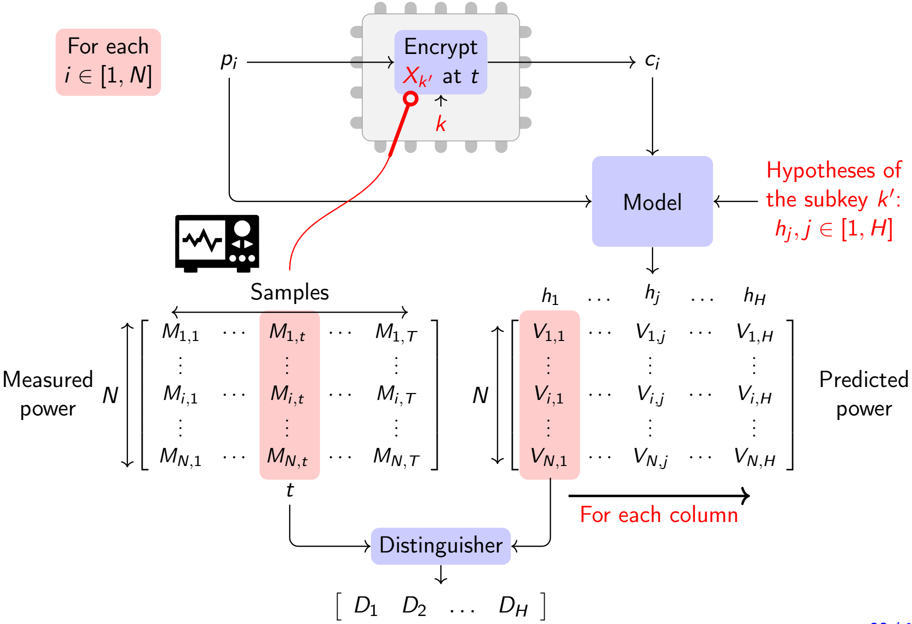
Solution:

Use statistics over several traces.

for $i = 1$ **to** n **do**
    Processing($K[i]$)
end for



Differential Power Analysis (DPA) [KJJ99]



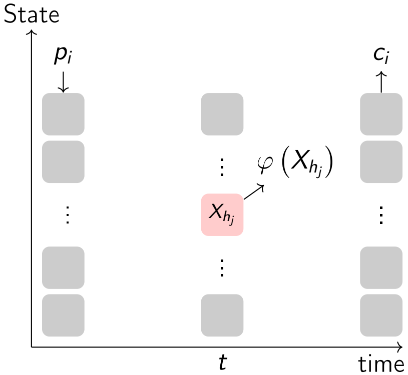
Model for Predicted Power

Predict the value of X:

Calculate $X(h_j)$ when p_i is encrypted.

Predict power consumption at t:

Use a power model φ to model the consumption from the value of X.



Power Model Examples

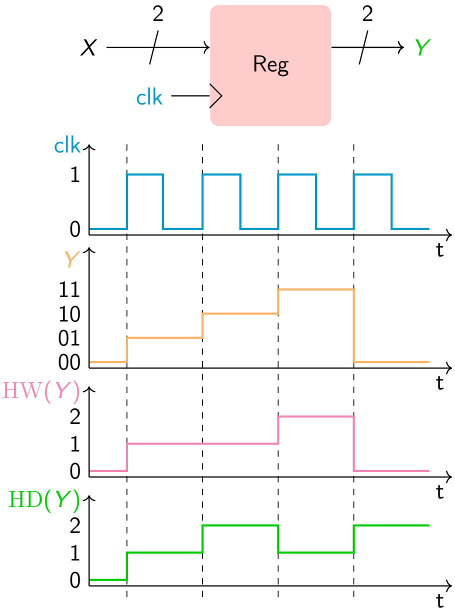
Hamming weight model [KJJ99]:

$\mathrm{HW}(Y) =$ number of 1s

Hamming distance model [BCO04]:

$\mathrm{HD}(X, Y) =$ number of transitions

$= \mathrm{HW}(X \oplus Y)$



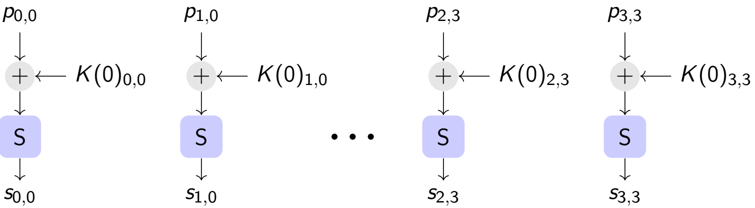
Divide and Conquer

Problem:

2^{128} key hypotheses for a 128-bit key.

Divide and conquer:

Attacking the key k piece by piece.

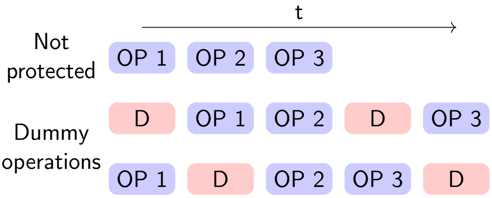


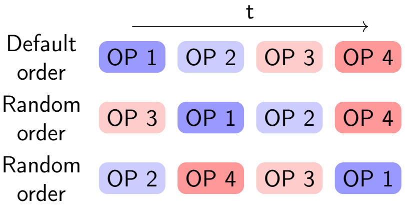
Challenge:

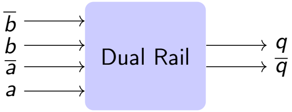
How to protect implementations against DPA attacks.

SCA Countermeasures:

- Random Insertion of Dummy Operations [TB07; CK09]
- Random shuffling [RPD09; Vey+12]
- Balanced consumption [Sok+04; PM05; Buc+06]
- Masking [Cha+99; GP99; RP10]
- ...







Masking

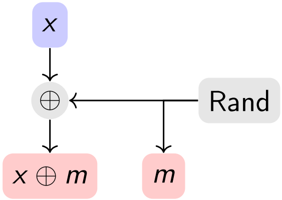
Let x to be protected.

Boolean masking:

- Combine x with an uniform random bit m:

$$x \oplus m$$

- One get two shares $x \oplus m$ and m
- Each share is independent of x
- Manipulate shares, but never x



Unmasking

Unmasking:

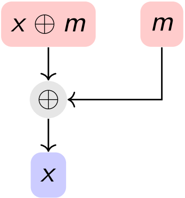
XOR of the two shares:

$$(x \oplus m) \oplus m = x$$

because $m \oplus m = 0$

Warning:

By combining the two parts, we can unmask and reveal x.

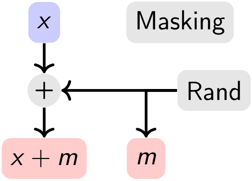


Boolean Masking in $\mathrm{GF}(2^8)$

Let $x \in \mathrm{GF}(2^8)$ a variable to be protected.

Problem:

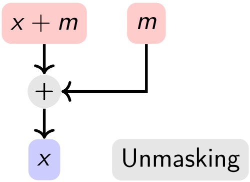
How to mask a variable in $\mathrm{GF}(2^8)$?



Boolean masking in $\mathrm{GF}(2^8)$:

Mask each bit individually:

- Generate a uniform random mask $m \in \mathrm{GF}(2^8)$
- Calculate $x + m$ in $\mathrm{GF}(2^8)$
- The shares are $(x + m, m)$



Masking a Function

Let F a function.

Method:

- Mask x into $(x + m, m)$
- Apply a masked function F'
- Unmask to get y

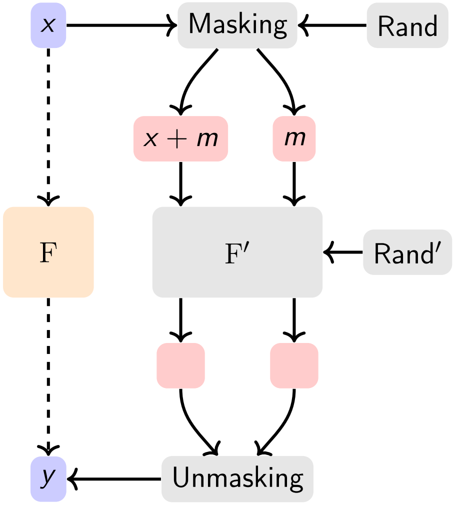
Masked function:

A function F' such that for all x:

$$\mathrm{unmask}(\mathrm{F}'(\mathrm{mask}(x))) = \mathrm{F}(x)$$

Remark:

Masked function can use other random values.



Boolean Masking for Linear Functions is Simple

Let L a linear function in $\mathrm{GF}(2^8)$.

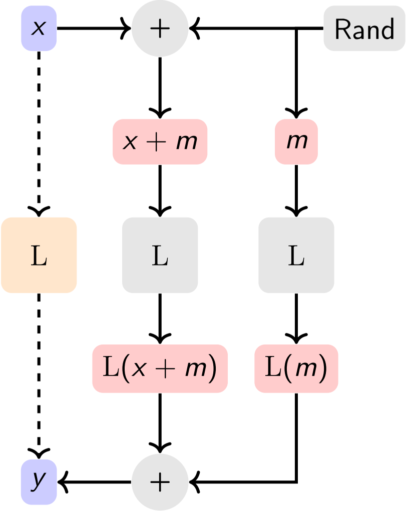
Method:

- Mask x into $(x + m, m)$
- Apply L independently to each share.
- Unmasks $(\mathrm{L}(x + m), \mathrm{L}(m))$:

$$\begin{aligned}\mathrm{L}(x + m) + \mathrm{L}(m) &= \mathrm{L}(x + m + m)\\ &= \mathrm{L}(x)\end{aligned}$$

Masked function of L:

Function that applies L independently to each share.

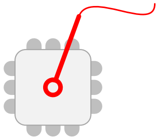


Evaluating the Security of Masked Function

Using real attacks:

Integrate the protection on a device and try to attack it:

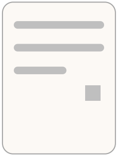
- Analyze the real leaks
- Not always possible
- No success is not a proof of security



Proof in a given theoretical model:

Model the leakage and prove the implementation security:

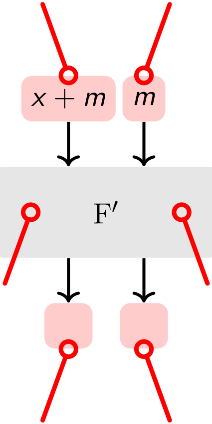
- Allows theoretical security proofs
- Actual security depends on the model quality



The Noisy Leakage Model [PR13]

Adversary model:

- Probe each intermediate variable
- The probe of a variable Z gives $Z + B$ with B an independant noise



Noisy leakage security:

F' is (σ, δ) noisy leakage secure if

- Probe noise have variance σ
- Need at least δ traces to attack

Remark:

- Realistic models
- Difficult security proof

Probing Model [ISW03]

Adversary model:

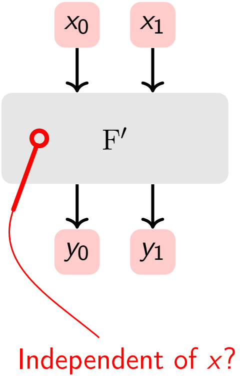
- Probe one intermediate variable
- Probes are exact value of the variable

Probing security:

F' is probing secure if any potential probe is independent from the unmasked inputs.

Security reductions [DDF14]:

Probing secure implies noisy leakage secure.

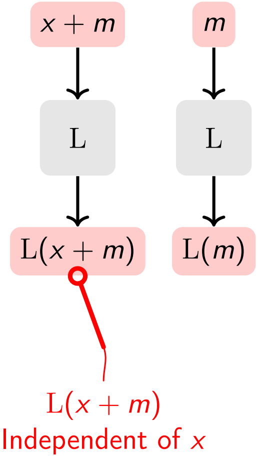


Linear Functions are Secure in Probing Model

Probing security analysis:

- $x + m$ and m are independent of x
- Other intermediate variables depend either on x+m or m
- So they are independent of x

Conclusion:
A masked linear function is probing secure.



Boolean Masking for Multiplication is More Complex

Let $x, y \in \mathrm{GF}(2^8)$, masked respectively in (x_0, x_1) and (y_0, y_1).

Method:

- Compute separately

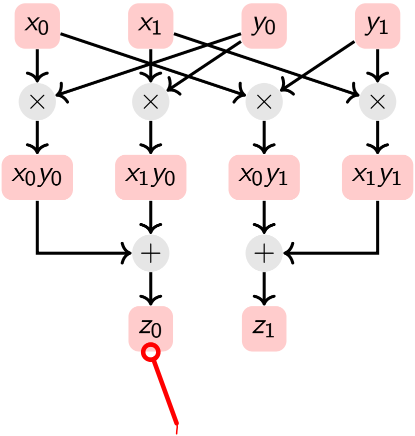
$$\begin{cases} z_0 = x_0 y_0 + x_1 y_0 \\ z_1 = x_0 y_1 + x_1 y_1 \end{cases}$$

- Get xy by unmasking (z_0, z_1):

$$\begin{aligned} z_0 + z_1 =& x_0 y_0 + x_1 y_0 + x_0 y_1 + x_1 y_1 \\ =& (x_0 + x_1)(y_0 + y_1) \\ =& xy \end{aligned}$$

Problem:

Not probing secure because z_0 depends on x.



$$\begin{aligned} x_0 y_0 + x_1 y_0 =& (x_0 + x_1) y_0 \\ =& x y_0 \end{aligned}$$

Depends on x

Boolean Masking for Multiplication: Secure Solution

Solution:

Add some randomness.

Method:

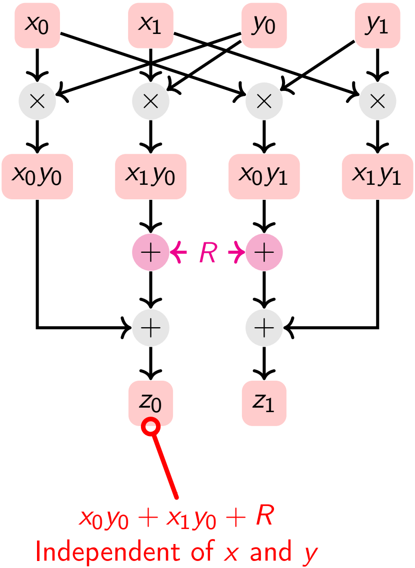
- Compute separately

$$\begin{cases} z_0 = x_0 y_0 + (x_1 y_0 + R) \\ z_1 = x_1 y_1 + (x_0 y_1 + R) \end{cases}$$

  with R a random value.
- Get xy by unmasking (z_0, z_1)

Probing security:

Masked multiplication is probing secure.



Glitch-Extended Probing Model [Fau+18]

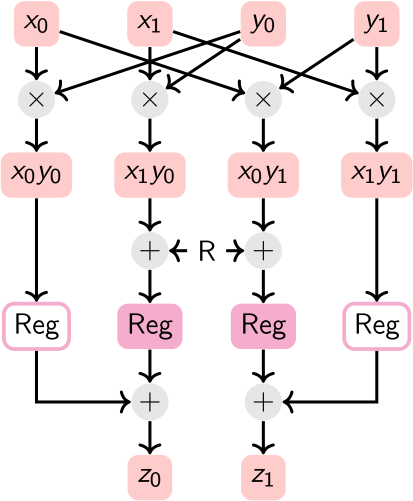
Problem:

Glitches can reveal information.

Solution [GMK16]:

Add registers to stop glitches:

- Where to insert the registers?
- How many should be inserted?



How to Mask AES? [RP10]

The approach:

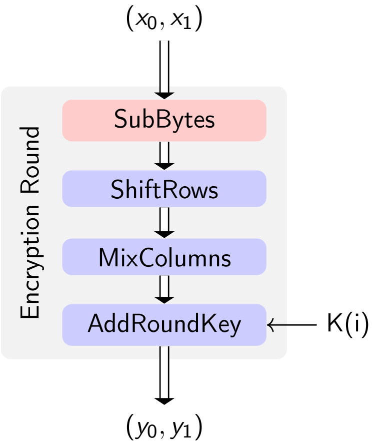
- Mask each sub-function
- Compose masked sub-functions

Mask linear sub-functions:

Apply the sub-function independently to each share.

Mask SubBytes:

How to mask SubBytes?



SubBytes Based on Inversion

S function:

Composition of the inversion in $\mathrm{GF}(2^8)$ and an affine function.

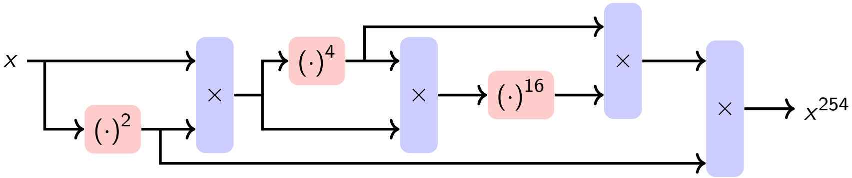
Fermat's little theorem in $\mathrm{GF}(2^8)$:

$x^{-1} = x^{254}$.

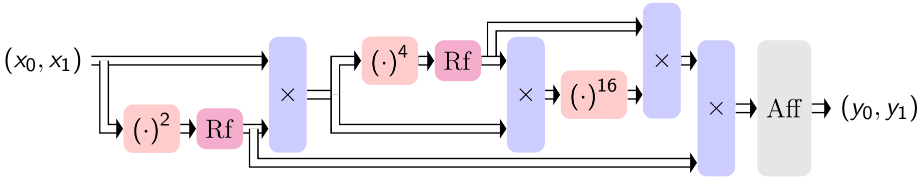
Inverse computation:

Split x^{254} into a sequence of multiplications and squares:

$$x^{-1} = x^{254} = \left[\left(x^2x\right)^4\left(x^2x\right)\right]^{16}\left(x^2x\right)^4x^2$$



Masking SubBytes Based on Masked Multiplication



The approach:

- Mask each atomic blocks
- Securely compose all blocks

The problem:

Composition of mask blocks is not always secure.

Solution [RP10]:

Add refresh blocks and add registers to avoid glitches.

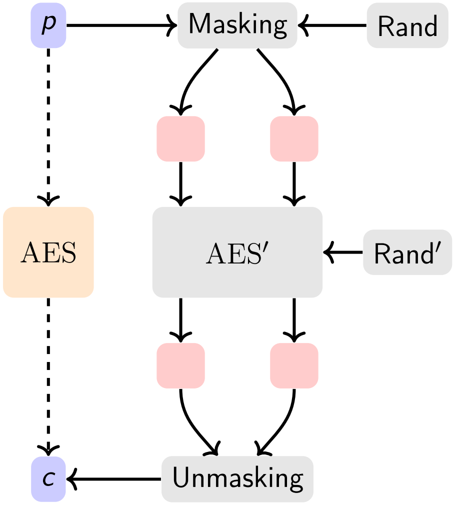
Masking of AES

Method:

- Mask p
- Apply masked AES
- Unmask to get c

Cost of masking:

- Area is at least doubled
- Registers increase latency
- Need a lot of randomness



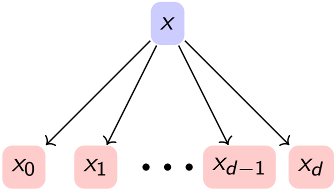
More Advanced Topics

Higher order attacks:

Attack *d* variables in the same time (not only one).

Higher order masking:

Combine a variable x with d random masks.



Cost/Performance for One Block Encryption

Hardware performance (from [MRB18]):

Source	Masked	Area [GE]	Randomness [bits/S-box]	Cycle count
[Mor+11]	No	2421	-	226
[Cnu+16]	Yes	7682	54	276
[GMK16]	Yes	7337	18	246
[MRB18]	Yes	6557	19	256

GE = gate equivalent

Software performance (from [Gao+21]):

Source	Masked	ISE	Instruction count	Cycle count	Instruction footprint	Data footprint
[Ber+02]	No	No	1932	2427	2148	524
[RP10]	Yes	No	59823	64200	14416	1356

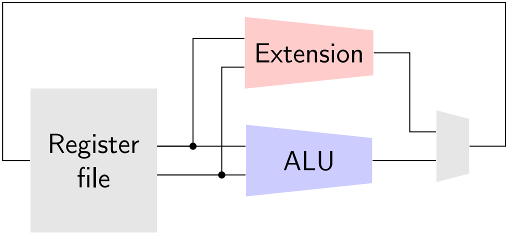
Instruction Set Extensions (ISE)

What is it?

Set of new assembly instructions.

Why extensions?

- Compromise between full software and hardware implementations
- Trade-off between execution time, circuit area, programming cost and security



Cost/Performance using AES ISE

Extract from table 5 of [Gao+21]:

Source	Masked	ISE	Instruction count	Cycle count	Instruction footprint	Data footprint
[Ber+02]	No	No	1932	2427	2148	524
[Mar+20]	No	Yes	238	291	730	10
[RP10]	Yes	No	59823	64200	14416	1356
[Gao+21]	Yes	Yes	1012	1113	968	84

Objectives of my PhD Thesis

Thesis title:

Cryptographic extensions for embedded processors.

Objectives:

Design, prototype and evaluate cryptographic extensions for:

- Higher-order masking schemes
- Post quantum cryptosystems

End of Presentation

Thank you for your attention

Do you have any questions?

Bibliography I

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. "Correlation Power Analysis with a Leakage Model". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2004, pp. 16–29. DOI: 10.1007/978-3-540-28632-5_2.

[Ber+02] Guido Bertoni et al. "Efficient Software Implementation of AES on 32-Bit Platforms". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2002, pp. 159–171. DOI: 10.1007/3-540-36400-5_13.

[Buc+06] Marco Bucci et al. "Three-Phase Dual-Rail Pre-charge Logic". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2006, pp. 232–241. DOI: 10.1007/11894063_19.

[Cha+99] Suresh Chari et al. "Towards Sound Approaches to Counteract Power-Analysis Attacks". In: Proc. Annual Cryptology Conference (CRYPTO). Springer, Aug. 1999, pp. 398–412. DOI: 10.1007/3-540-48405-1_26.

[CK09] Jean-Sébastien Coron and Ilya Kizhvatov. "An Efficient Method for Random Delay Generation in Embedded Software". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Sept. 2009, pp. 156–170. DOI: 10.1007/978-3-642-04138-9_12.

[Cnu+16] Thomas de Cnudde et al. "Masking AES with $d + 1$ Shares in Hardware". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2016, pp. 194–212. DOI: 10.1007/978-3-662-53140-2_10.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. "Unifying Leakage Models: From Probing Attacks to Noisy Leakage". In: Proc. Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT). Springer, May 2014, pp. 423–440. DOI: 10.1007/978-3-642-55220-5_24.

[DR02] Joan Daemen and Vincent Rijmen. The Design of Rijndael. 1st ed. Springer, 2002. ISBN: 978-3-540-42580-9. DOI: 10.1007/978-3-662-04722-4.

Bibliography II

[Fau+18] Sebastian Faust et al. "Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model". In: Transactions on CHES (Aug. 2018), pp. 89–120. DOI: 10.13154/tches.v2018.i3.89-120.

[Gao+21] Si Gao et al. "An Instruction Set Extension to Support Software-Based Masking". In: Transactions on CHES (Aug. 2021), pp. 283–325. DOI: 10.46586/tches.v2021.i4.283-325.

[GMK16] Hannes Gross, Stefan Mangard, and Thomas Korak. Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. IACR Cryptology ePrint Archive. Nov. 2016. URL: https://eprint.iacr.org/2016/486.

[GP99] Louis Goubin and Jacques Patarin. "DES and Differential Power Analysis The "Duplication" Method". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 1999, pp. 158–172. DOI: 10.1007/3-540-48059-5_15.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. "Private Circuits: Securing Hardware against Probing Attacks". In: Proc. Annual Cryptology Conference (CRYPTO). Springer, Aug. 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential Power Analysis". In: Proc. Annual Cryptology Conference (CRYPTO). Springer, Aug. 1999, pp. 388–397. DOI: 10.1007/3-540-48405-1_25.

[Mar+20] Ben Marshall et al. "The design of scalar AES Instruction Set Extensions for RISC-V". In: Transactions on CHES (Dec. 2020), pp. 109–136. DOI: 10.46586/tches.v2021.i1.109-136.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. Power Analysis Attacks: Revealing the Secrets of Smart Cards. 1st ed. Springer, 2007. ISBN: 978-0-387-38162-6. DOI: 10.1007/978-0-387-38162-6.

[Mor+11] Amir Moradi et al. "Pushing the Limits: A Very Compact and a Threshold Implementation of AES". In: Proc. Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT). Springer, May 2011, pp. 69–88. DOI: 10.1007/978-3-642-20465-4_6.

Bibliography III

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. "Side-Channel Leakage of Masked CMOS Gates". In: Proc. Cryptographers' Track RSA Conference (CT-RSA). Springer, Feb. 2005, pp. 351–365. DOI: 10.1007/978-3-540-30574-3_24.

[MRB18] Lauren de Meyer, Oscar Reparaz, and Begül Bilgin. "Multiplicative Masking for AES in Hardware". In: Transactions on CHES (Aug. 2018), pp. 431–468. DOI: 10.13154/tches.v2018.i3.431-468.

[PM05] Thomas Popp and Stefan Mangard. "Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2005, pp. 172–186. DOI: 10.1007/11545262_13.

[PR13] Emmanuel Prouff and Matthieu Rivain. "Masking against Side-Channel Attacks: A Formal Security Proof". In: Proc. Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT). Springer, May 2013, pp. 142–159. DOI: 10.1007/978-3-642-38348-9_9.

[QS01] Jean-Jacques Quisquater and David Samyde. "ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards". In: Proc. International Conference on Research in Smart Cards (E-smart). Springer, Sept. 2001, pp. 200–210. DOI: 10.1007/3-540-45418-7_17.

[RP10] Matthieu Rivain and Emmanuel Prouff. "Provably Secure Higher-Order Masking of AES". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2010, pp. 413–427. DOI: 10.1007/978-3-642-15031-9_28.

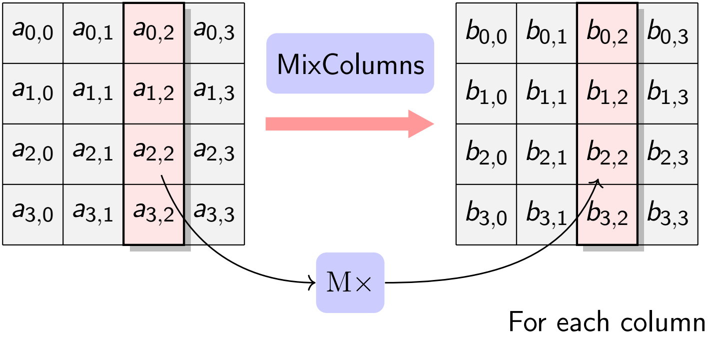
[RPD09] Matthieu Rivain, Emmanuel Prouff, and Julien Doget. "Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Sept. 2009, pp. 171–188. DOI: 10.1007/978-3-642-04138-9_13.

[Sok+04] Danil Sokolov et al. "Improving the Security of Dual-Rail Circuits". In: Proc. International Workshop on Cryptographic Hardware and Embedded Systems (CHES). Springer, Aug. 2004, pp. 282–297. DOI: 10.1007/978-3-540-28632-5_21.

[TB07] Michael Tunstall and Olivier Benoit. "Efficient Use of Random Delays in Embedded Software". In: Proc. International Workshop on Information Security Theory and Practices (WISTP). Springer, May 2007, pp. 27–38. DOI: 10.1007/978-3-540-72354-7_3.

[Vey+12] Nicolas Veyrat-Charvillon et al. "Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note". In: Proc. International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT). Springer, Dec. 2012, pp. 740–757. DOI: 10.1007/978-3-642-34961-4_44.

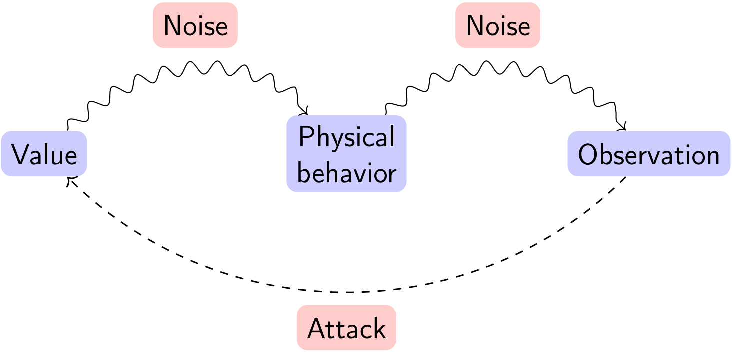
MixColumns



Multiply each column of the state by the matrix M:

$$\mathrm{M} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

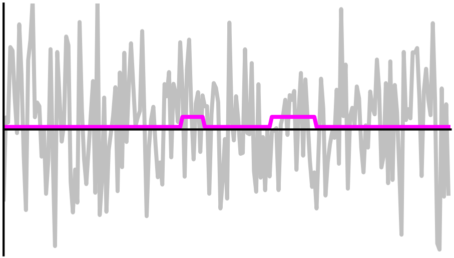
Principles of Observation Attacks



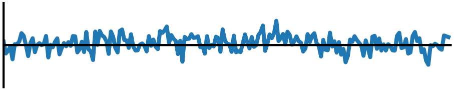
Due to noise, several observations may be necessary (stats, ML, ...).

Basic Statistics to the Rescue

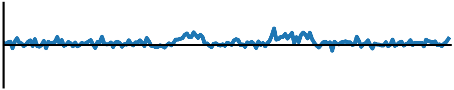
Secret signal and
noisy trace



Mean of
10 traces



Mean of
100 traces



Mean of
1000 traces

Independence of Shares

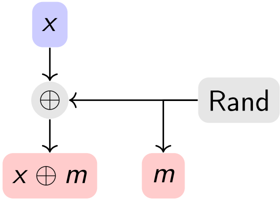
Let x to be protected.

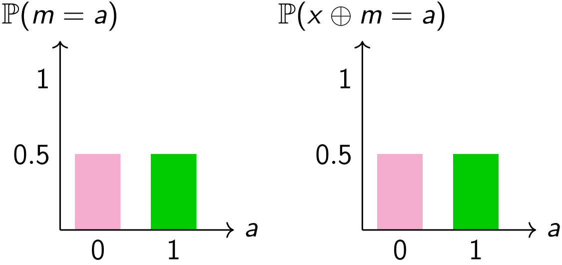
Boolean masking:

Add to x a bit m uniform and independent of x.

Result:

$x \oplus m$ is uniform and independent of x.





Other Masking Schemes

Let $x \in \mathrm{GF}(2^8)$ a variable to be protected and $m \in \mathrm{GF}(2^8)$ a random.

Boolean masking:

- A boolean masking of x is $(x + m, m)$
- Linear functions are easy to mask

Multiplicative masking:

- A multiplicative masking of x is $(x \times m, m)$
- Multiplications are easy to mask

Shamir masking:

- Let $P(X) = x + mX$
- Evaluate P in two non-zero values a_1 and a_2
- A Shamir masking of x is $(x + m \times a_1, x + m \times a_2)$
- Generalization of boolean masking

Precomputed Look-up Table

The challenge:

Mask a S-Box $S : \mathrm{GF}(2^8) \to \mathrm{GF}(2^8)$.

Solution:

Precomputed look-up table T:

$$T(x_0, x_1) = S(x_0 + x_1) + x_1$$

So $S(x) + m = T[x + m, m]$

Problem:

- T have 2^{16} entries
- The table is very large