Introduction to Side Channel Attacks

Florent Bruguier

Contact : florent.bruguier@lirmm.fr

March 13, 2025 – ARCHI 25

Context

Secure embedded systems



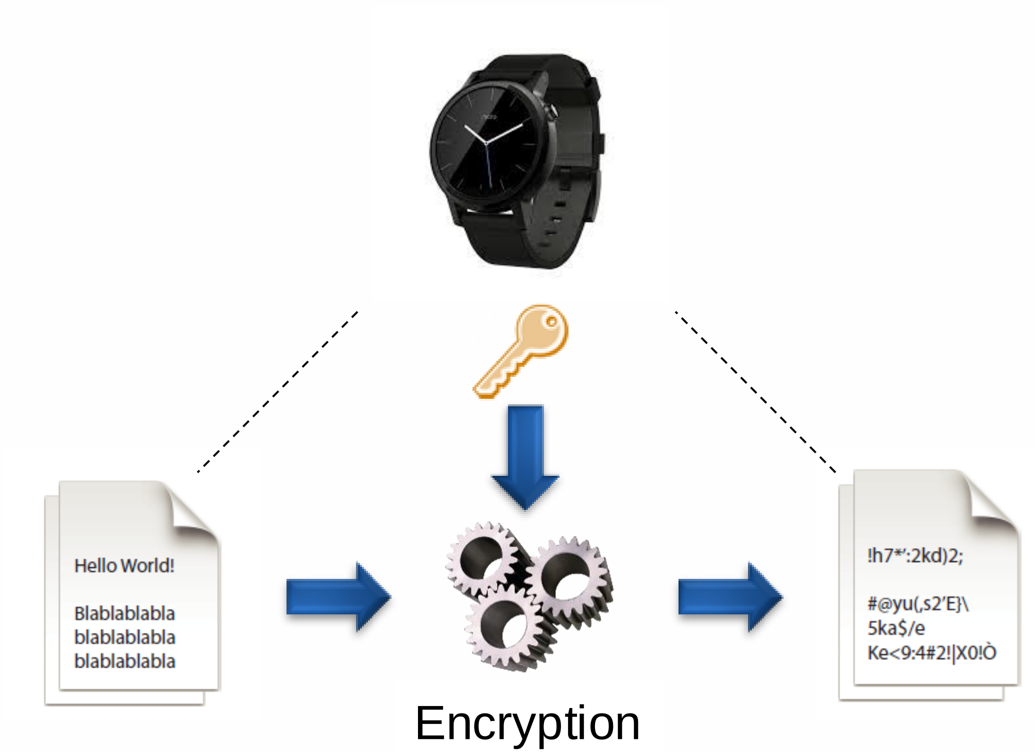
Strong cryptography from a mathematic point of view

- Used to manage sensitive data
- AES, RSA, ECC, SHA-3, GIFT-COFB, SABER…

Classical cryptography

Black box model

- Key(s) stored in the device
- Cryptographic operations computed inside the device



- The attacker has only access to pairs of plaintexts / ciphertexts

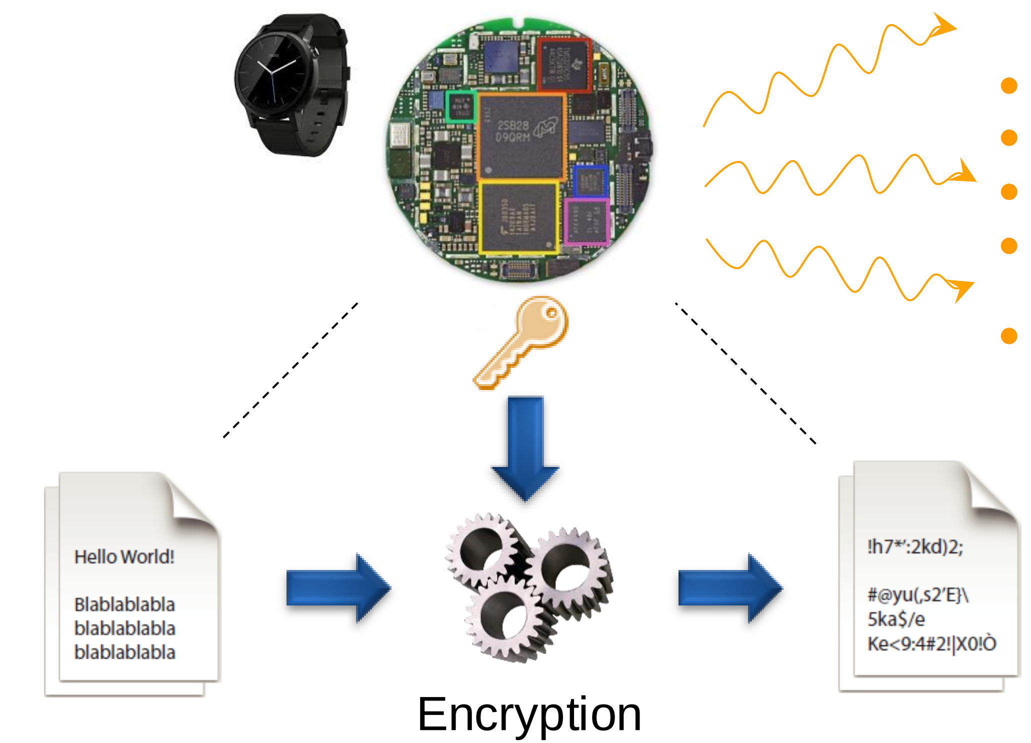
Puzzle

Which bulb is lit by which switch?

Side-Channel attacks

Grey box model

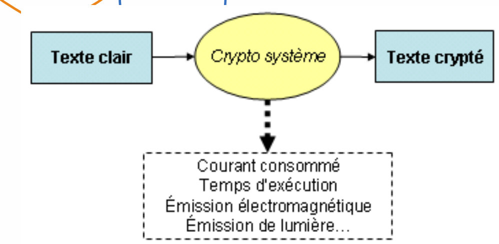
- Cryptosystems integrated in CMOS technology
- Physical leakages correlated with computed data (P. Kocher, 1996)

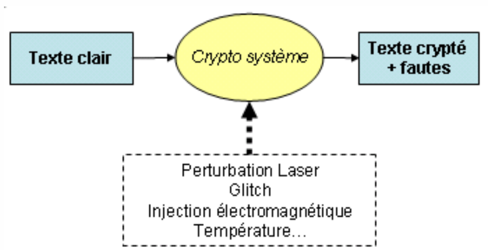


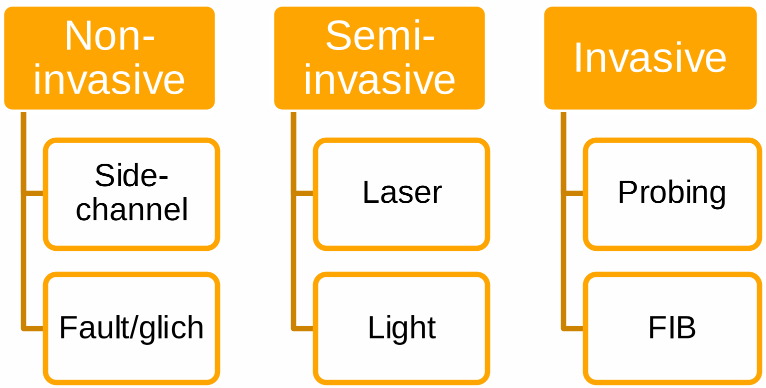
- Hit / Miss
- Execution time
- Power
- Electomagnetic emissions
- …

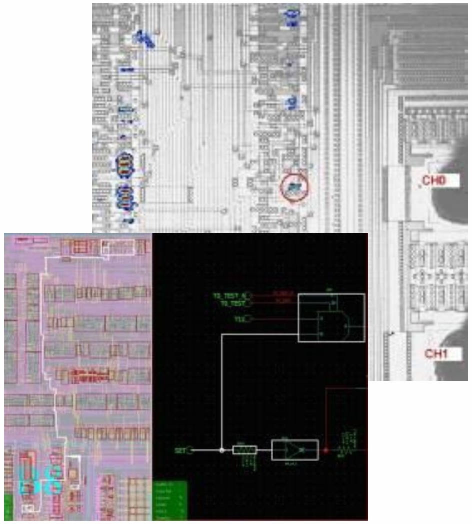
- The attacker has also access to physical leakages

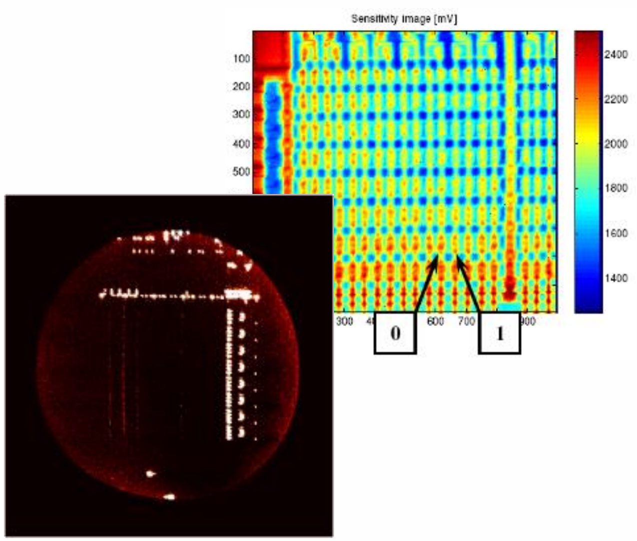
Physical side-channel

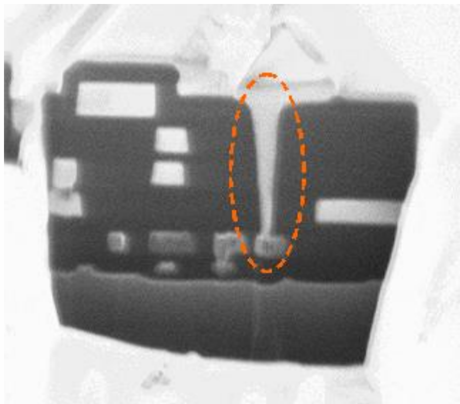








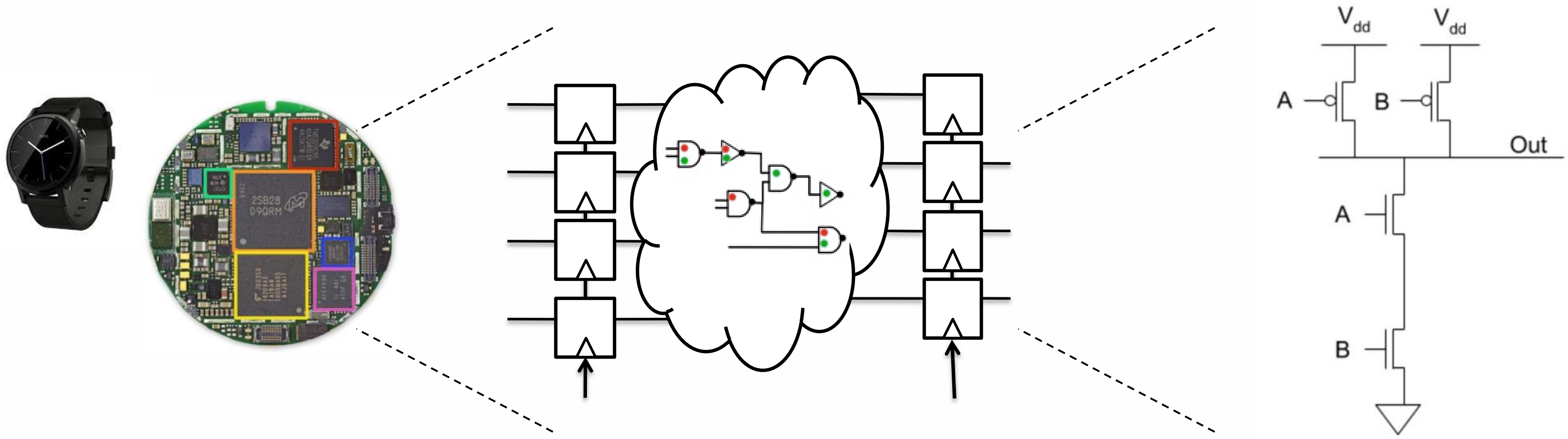




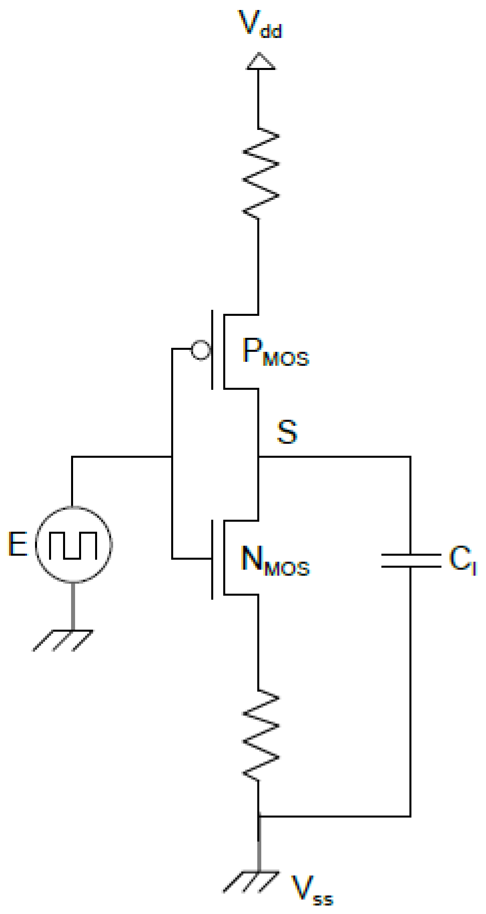
Leakages

Power SCA

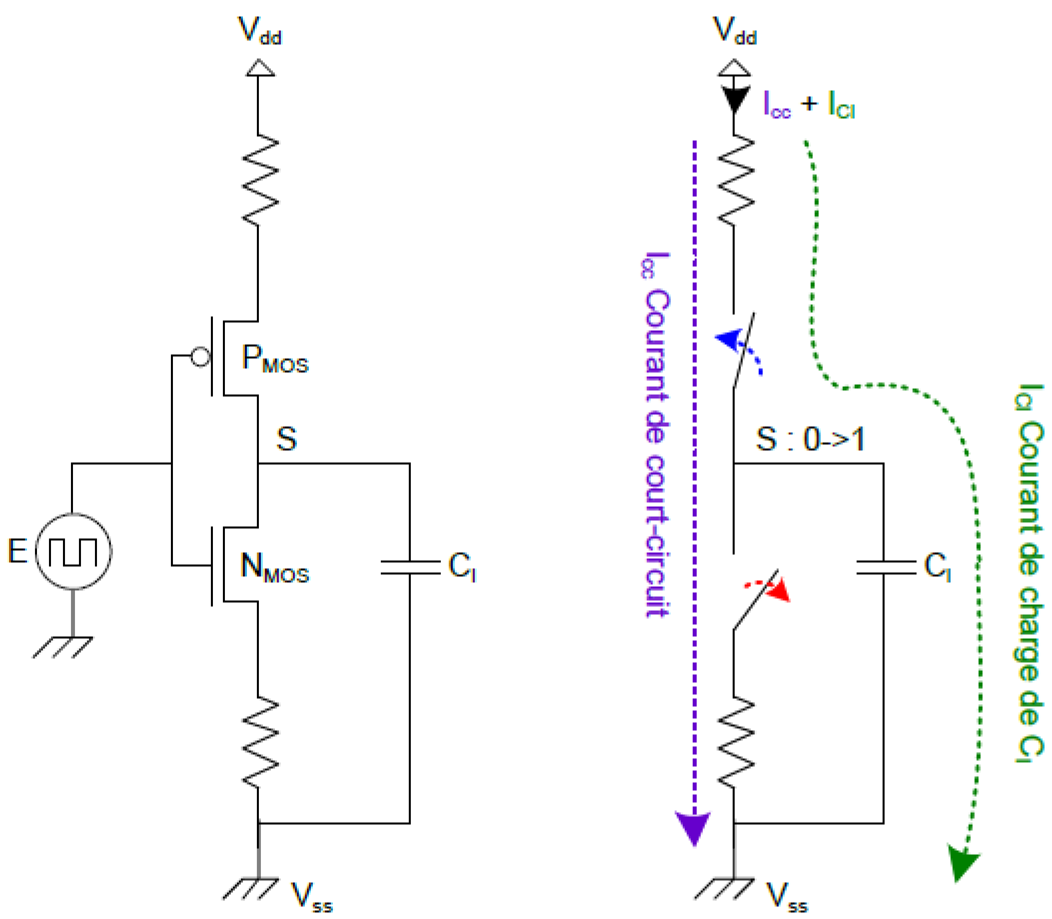
- Cryptosystems integrated in CMOS technology
- Power leakages correlated with computed data (P. Kocher, 1999)



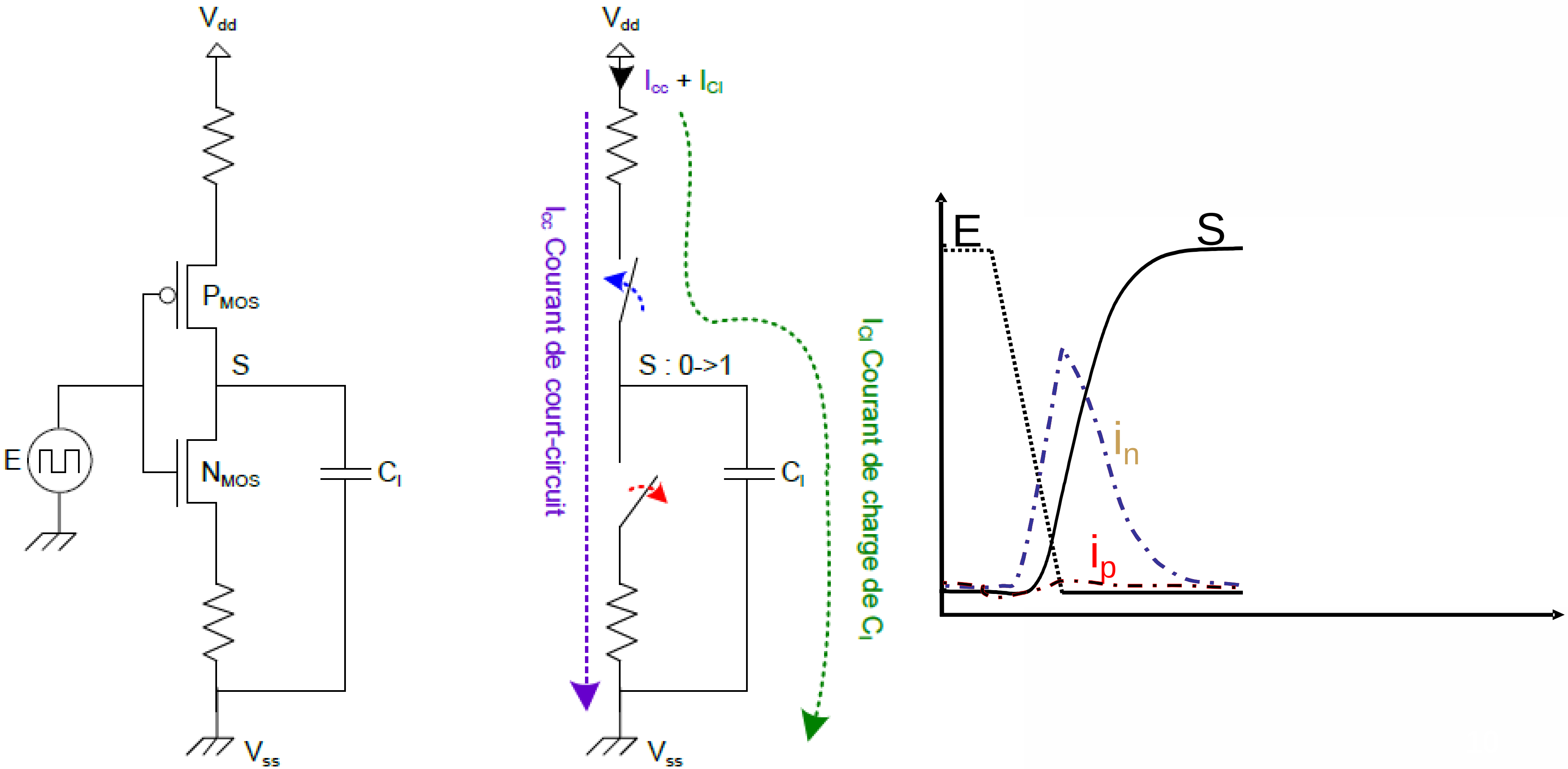
CMOS technology



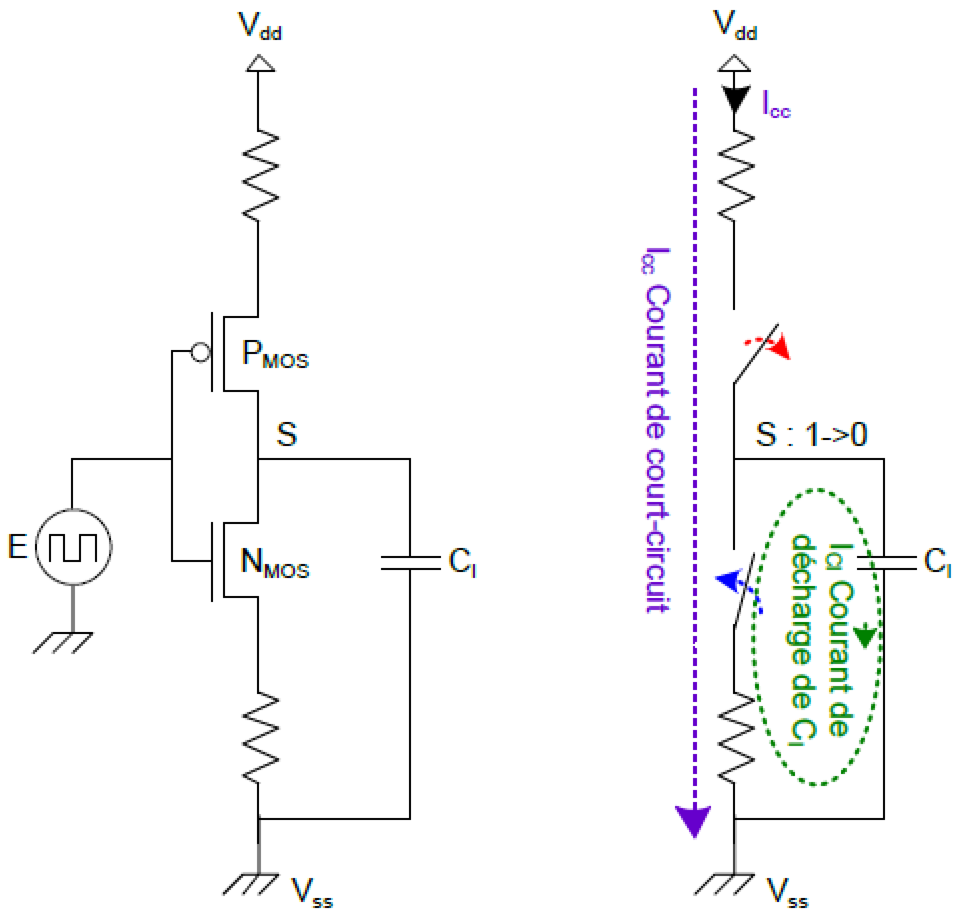
CMOS technology



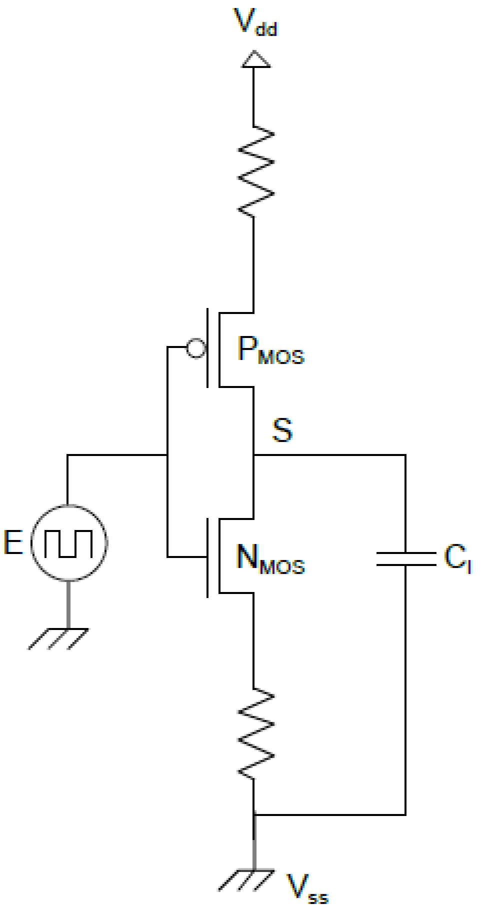
CMOS technology

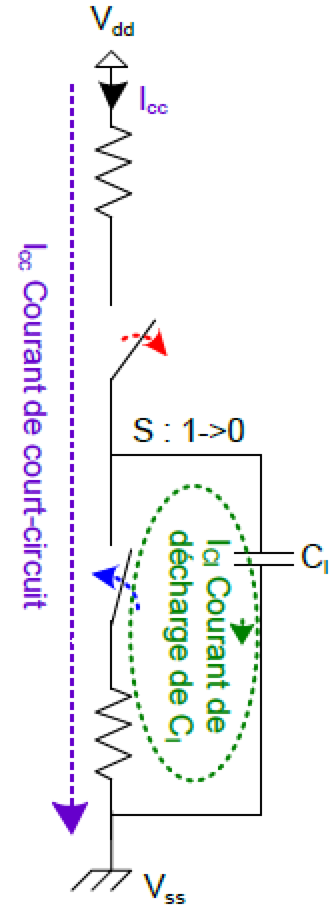


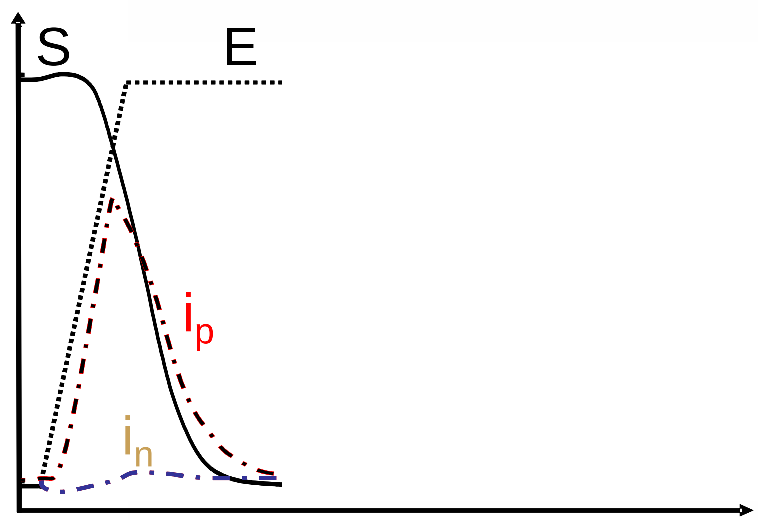
CMOS technology

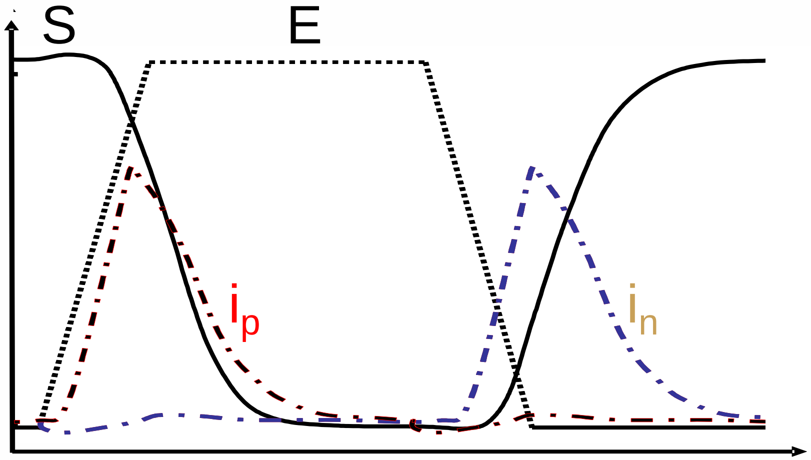


CMOS technology







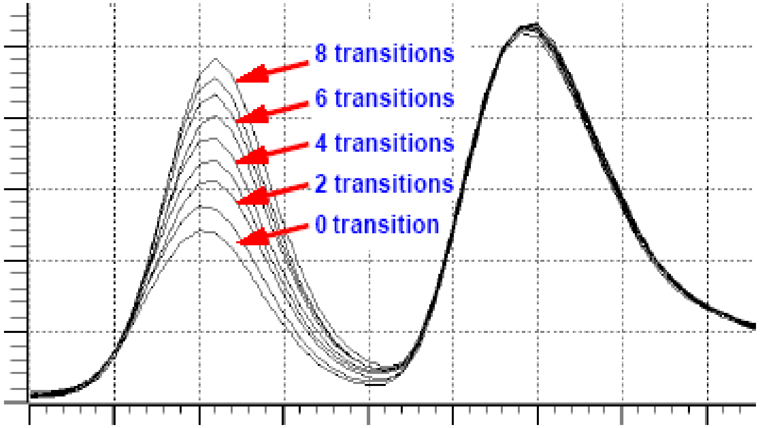


S : 0 → 1, 1 → 0

- High power consumption

S : 0 → 0, 1 → 1

- Low power consumption

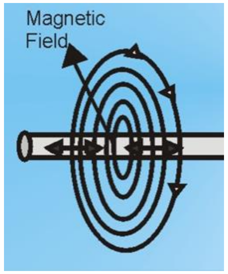


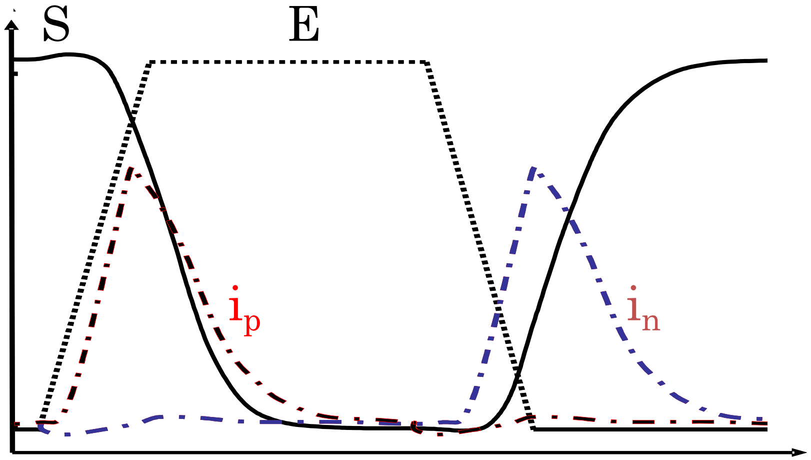
Attacks based on the power consumption

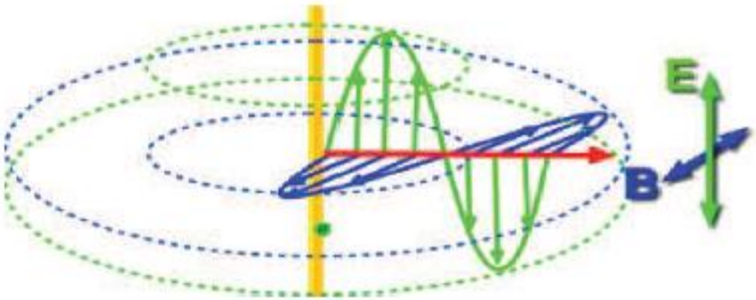
Leakages

Electromagnetic SCA

- Maxwell equations: a current flowing through a conductor induces an electromagnetic field (E. Brier 2004)



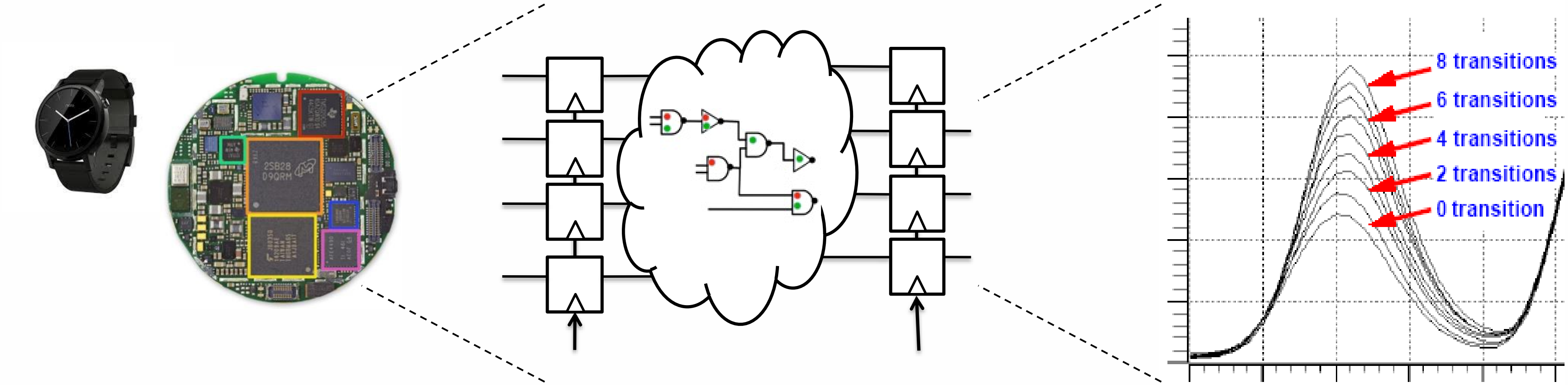




Leakages

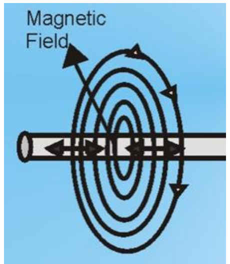
Power SCA

- Cryptosystems integrated in CMOS technology
- Power leakages correlated with computed data (P. Kocher, 1999)



Electromagnetic SCA

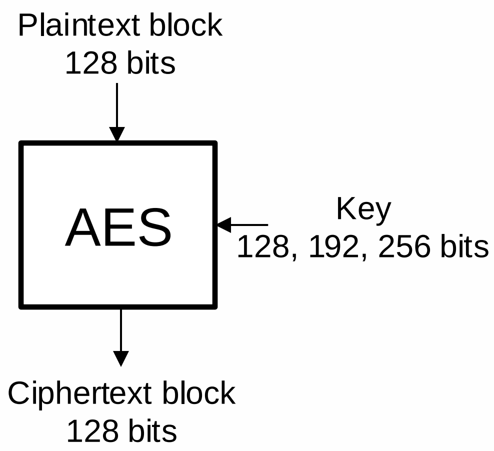
- Maxwell equations: a current flowing through a conductor induces an electromagnetic field (E. Brier 2004)



Pre-requisite

All future illustrations are based on Advanced Encryption Standard – AES

- Developed by Vincent Rijmen and Joan Daemen
- Replace the old DES
- Block cipher - 128-bit plaintexts / ciphertexts
- Three versions
  - 128-bit keys with 10 rounds
  - 192-bit keys with 12 rounds
  - 256-bit keys with 14 rounds



We consider the 128-bit keys version

How the algorithm works?

How the algorithm works?

Side-channel adversary model

In this talk, we consider the following hypotheses

- The adversary can steal the device and get full control of it
- The device has few communication interfaces
- Each communication interface exposes few commands
- There is no software vulnerability due to previous points

- Examples are done with 128-bit key AES
  - 128-bit long keys, plaintexts and ciphertexts
  - 10 rounds encryption scheme

00	11	22	33
44	55	66	77
88	99	AA	BB
CC	DD	EE	FF

Plaintext

??	??	??	??
??	??	??	??
??	??	??	??
??	??	??	??

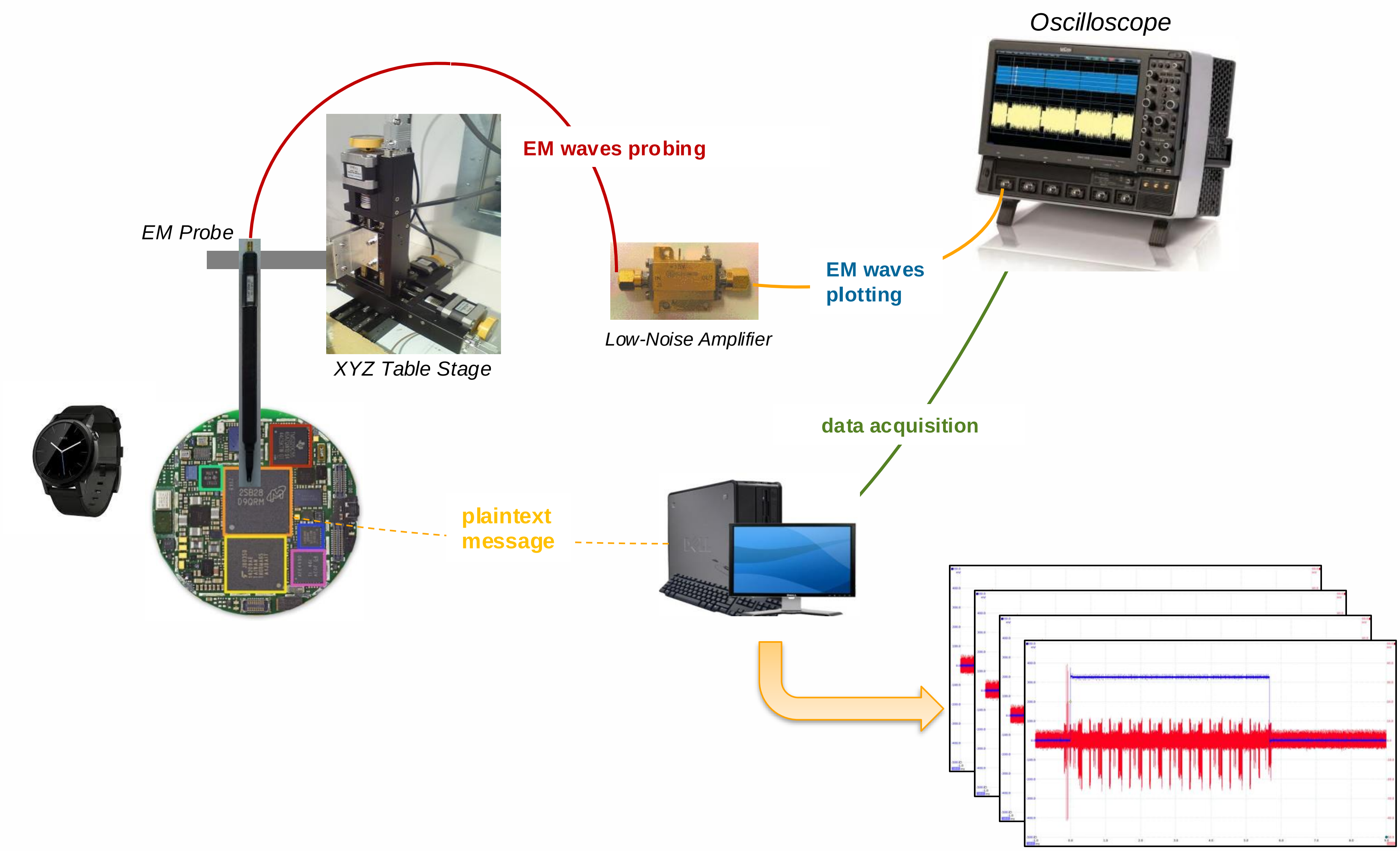
Key

AC	23	98	46
43	EF	CA	F1
32	D9	72	05
90	29	38	4F

Ciphertext

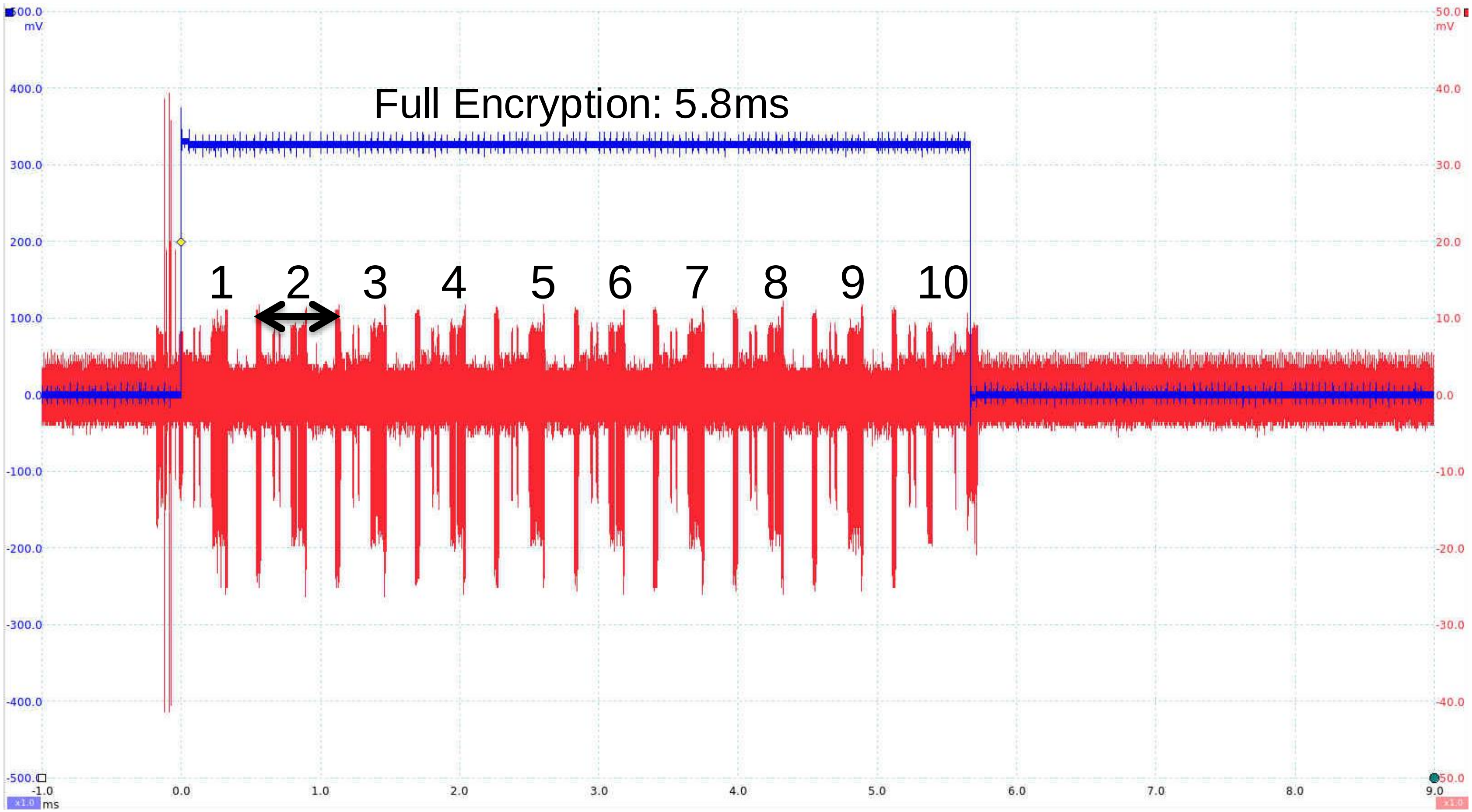
1st step: Acquisition

Electromagnetic bench example



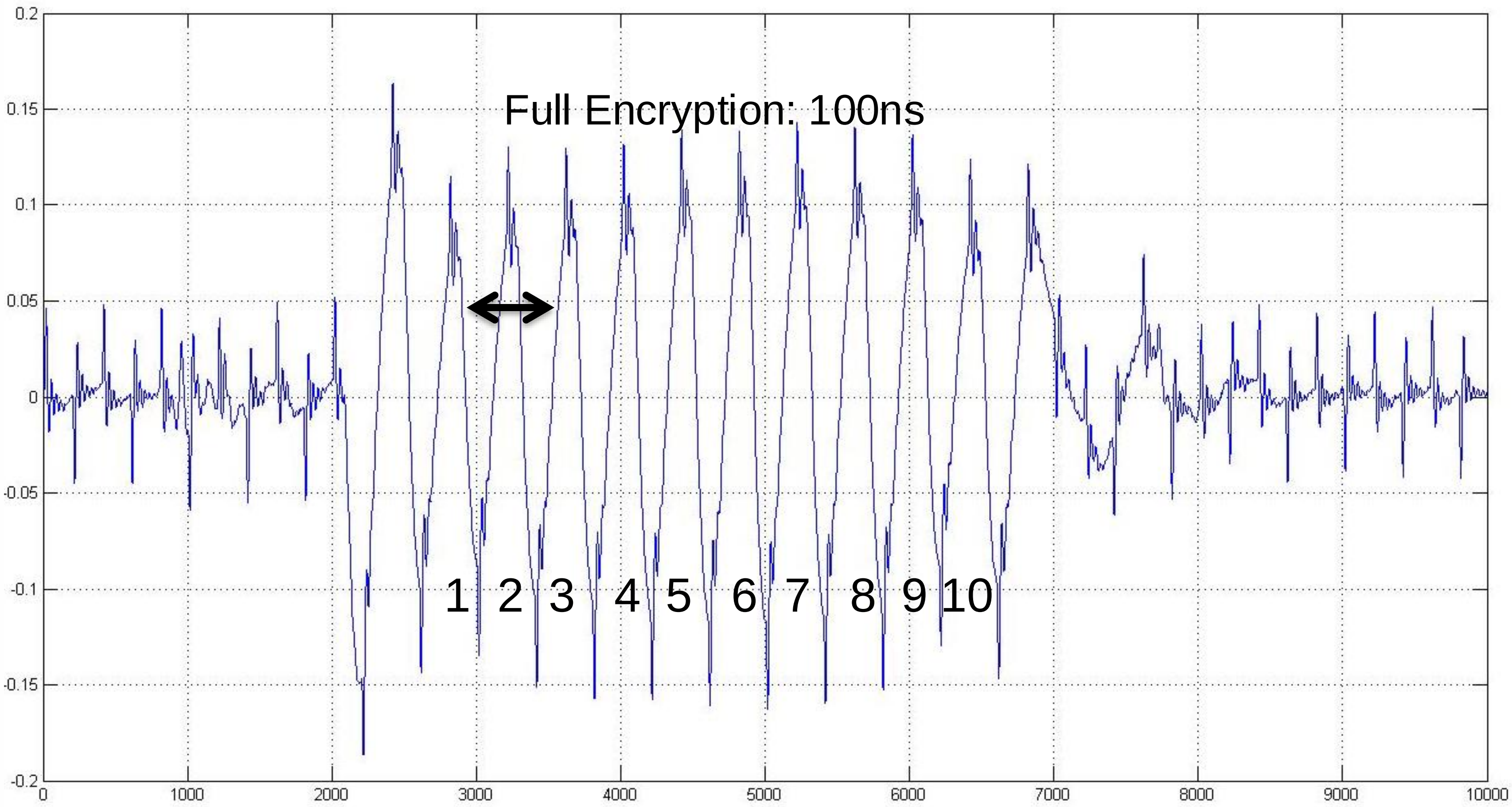
Example (1/2)

128-bit key AES executed on STM32



Example (2/2)

128-bit key AES executed on a cryptoprocessor



Disclaimer

Pre-Processing Techniques required

- Signal processing
  - Filtering
  - Resynchrnisation

- Research of Point of Interest
  - Signal-to-Noise-Ratio (SNR)
  - Variance

2nd step: selection function

Link between the leakage and the key

- The key must be mix with the plaintext/ciphertext
- Non-linearity is needed
  - Differentiate the key and the inverse of the key

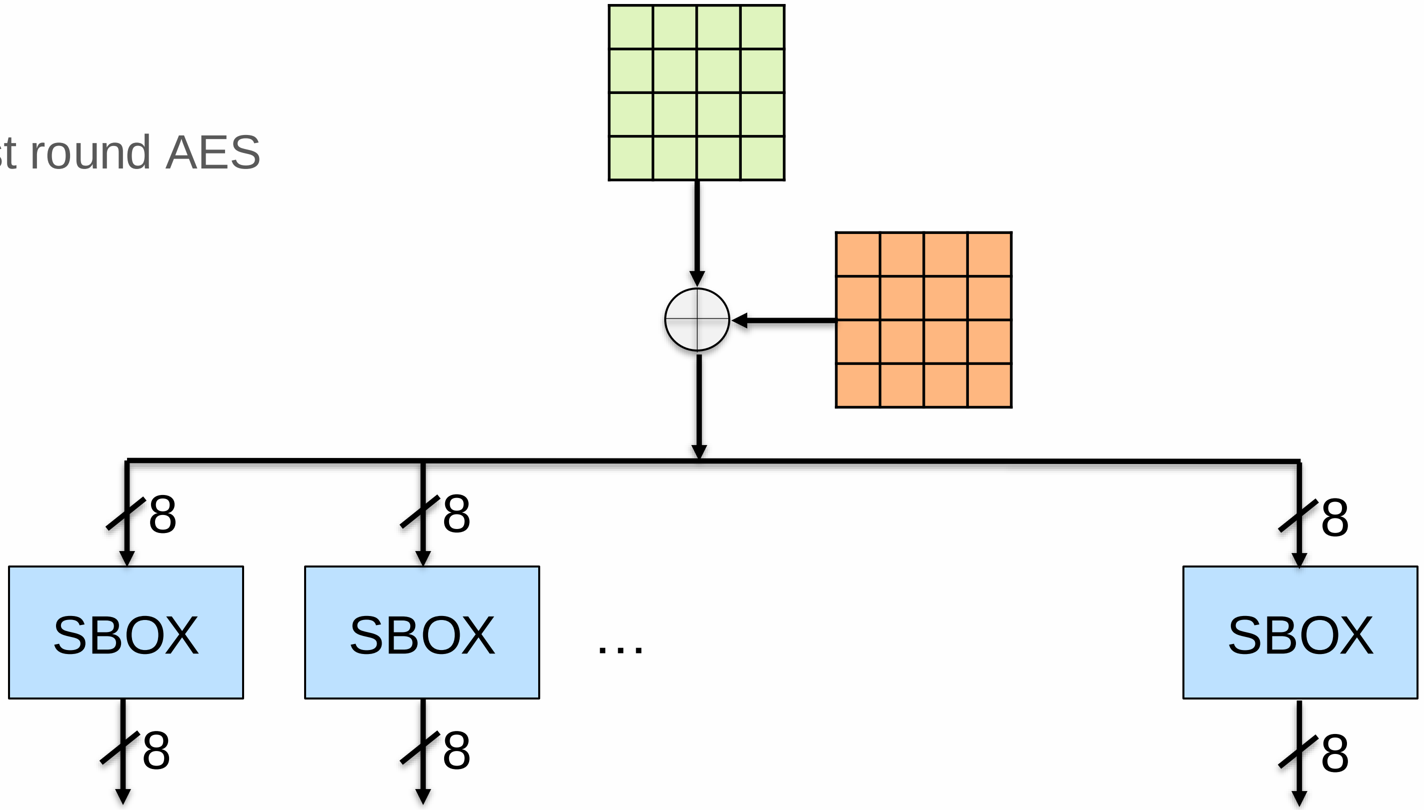
2nd step: selection function

Link between the leakage and the key

- The key must be mix with the plaintext/ciphertext
- Non-linearity is needed
  - Differentiate the key and the inverse of the key

Example

- First round AES



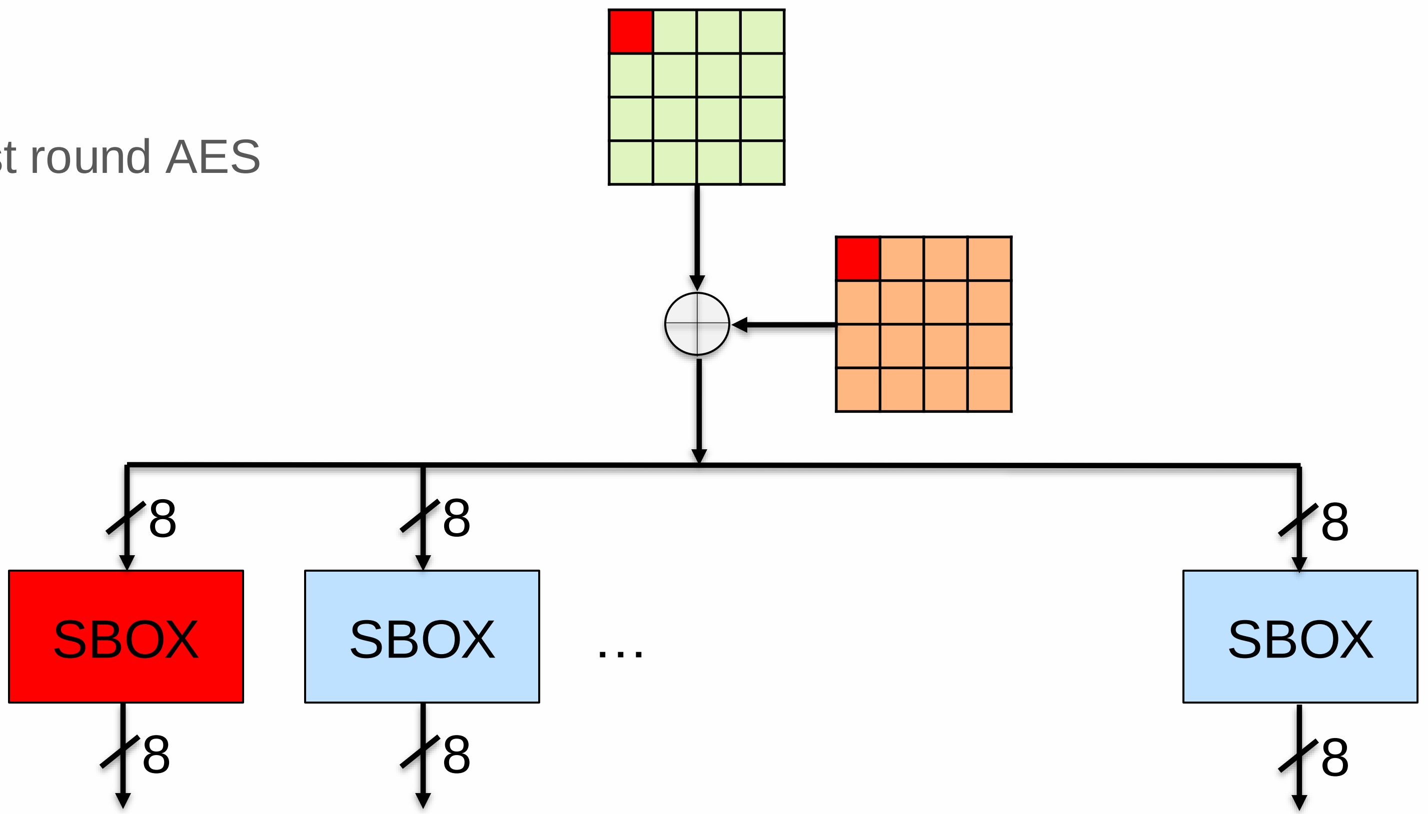
2nd step: selection function

Link between the leakage and the key

- The key must be mix with the plaintext/ciphertext
- Non-linearity is needed
  - Differentiate the key and the inverse of the key

Example

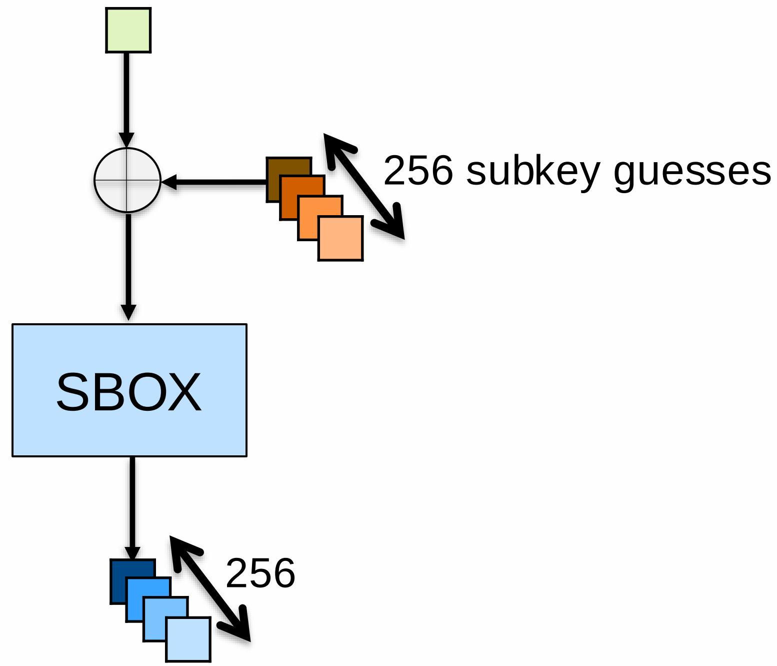
- First round AES



2nd step: selection function

Divide and conquer strategy

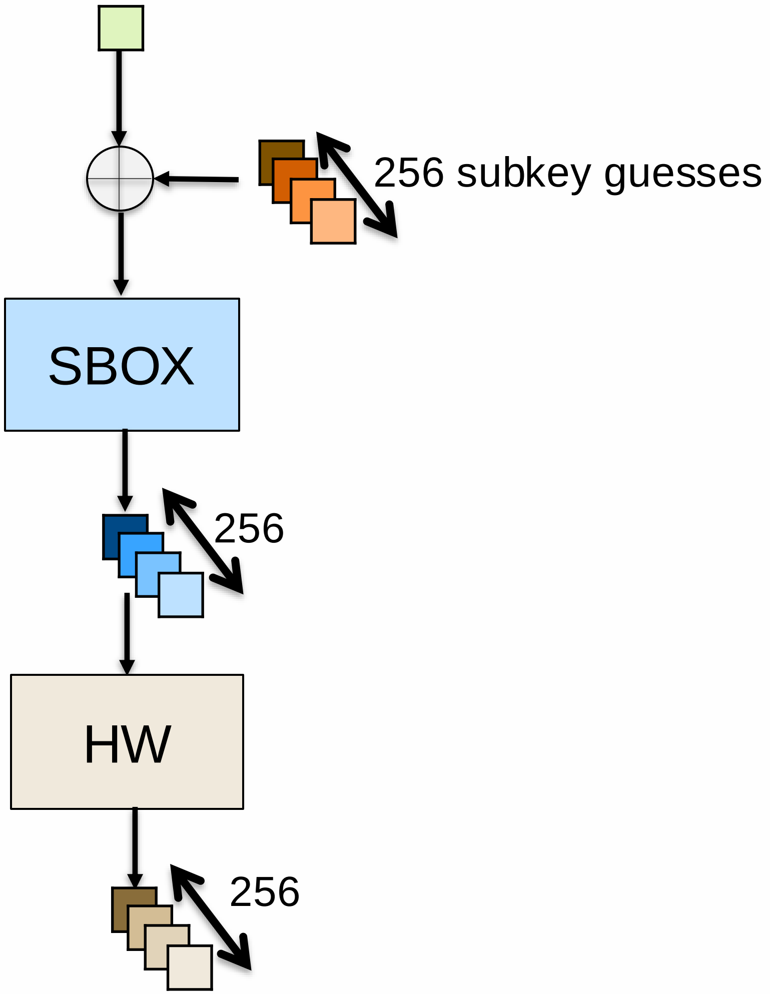
- The key could be search byte-by-byte
- 2^8 = 256 possibilities for each byte
- We consider all possibilities



2nd step: selection function

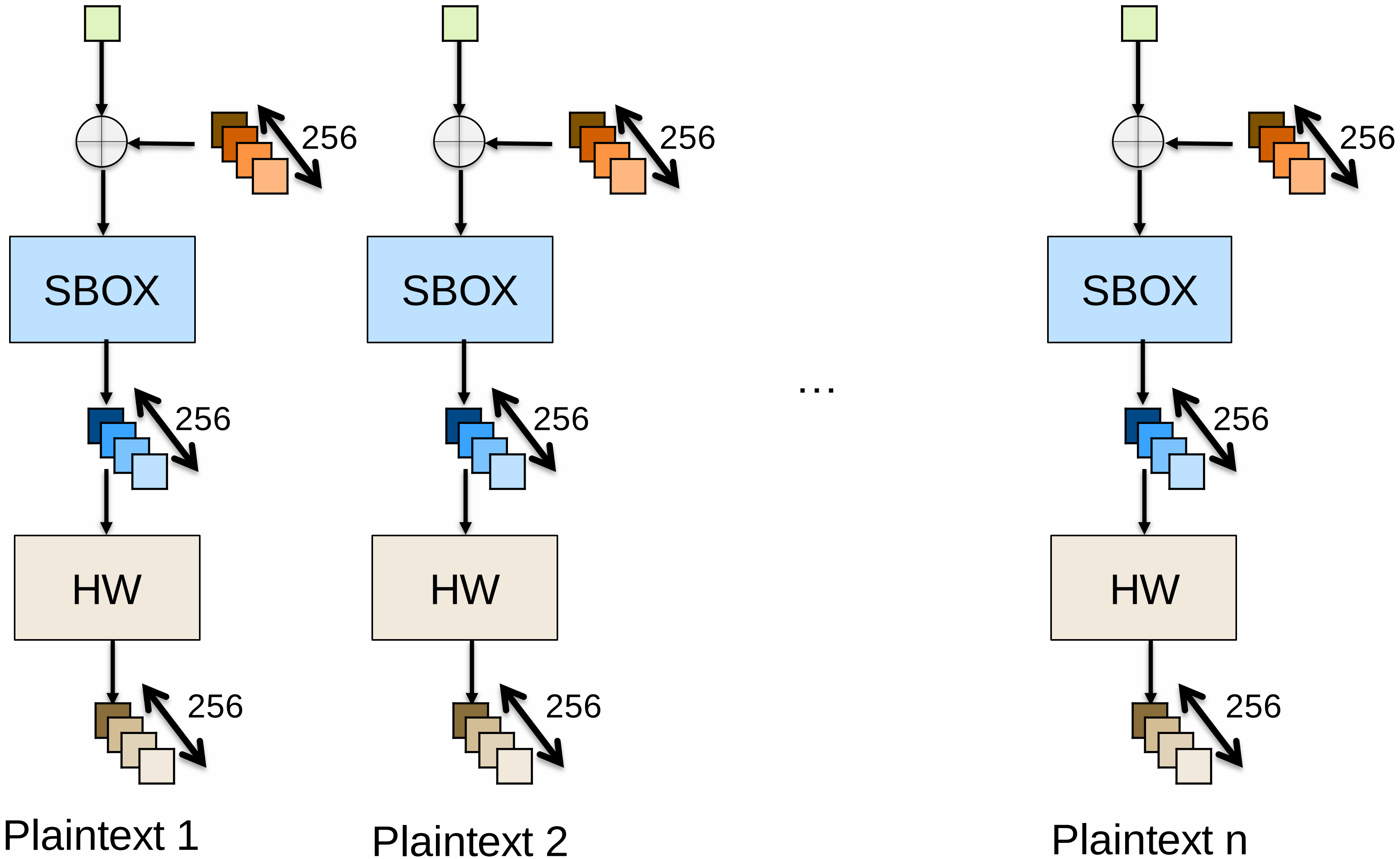
Consumption model

- e.g. circuit leaks as the Hamming Weight of the end of the SBOX



2nd step: selection function

Compute these values for each plaintext

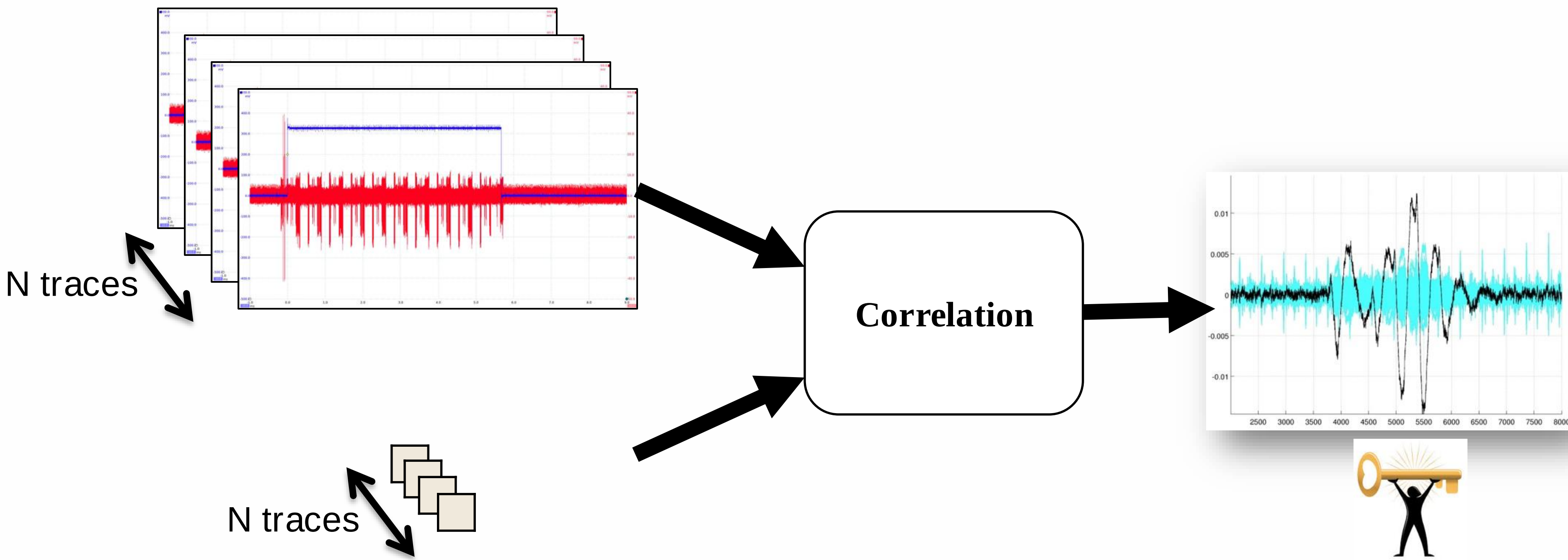


3rd step: distinguisher

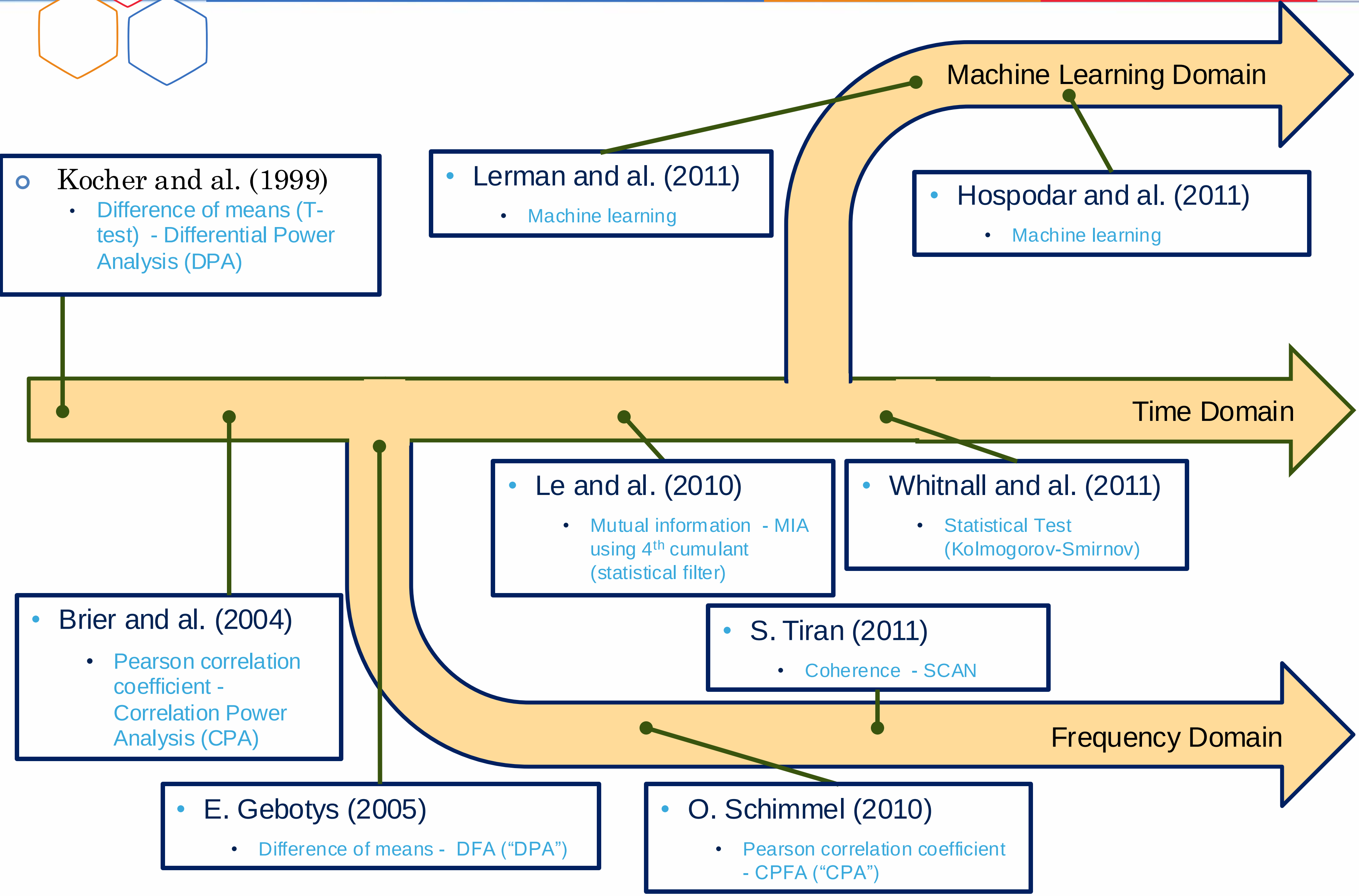
Statistical tool

- Allows to distinghish the good subkey guess from the bad ones
- e.g. Pearson Correlation

For each key guess



Most used distinguishers



Metrics

How to know if the attack works well?

- Compute the attack for a small number of traces, then add traces until the key is found

Measurement To Disclosure (MTD)

- Number of traces to find the right subkey

Measurement To Disclosure with Stability (MTDwS)

- Number of traces to find the right subkey

Percentage of Correct Guesses (PCG)

- Pourcentage de clés correctes sur la totalité des échantillons

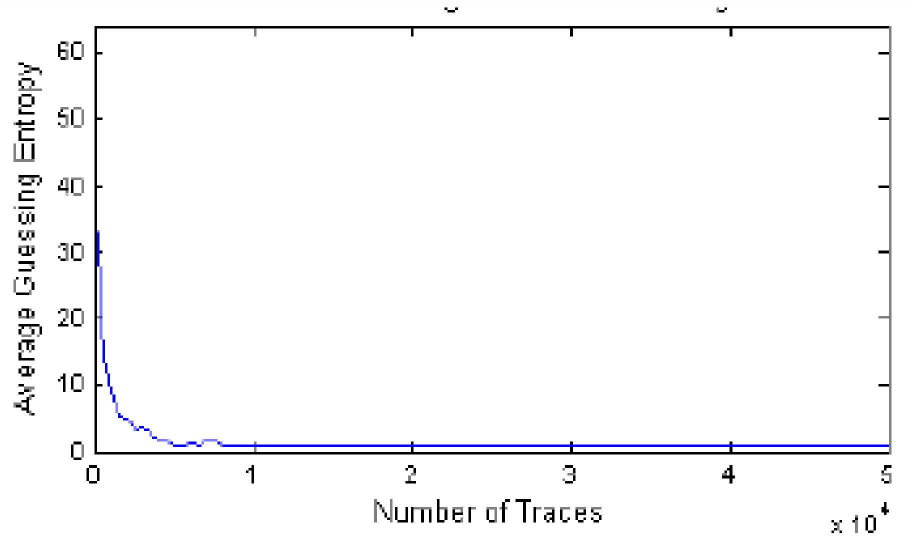
Subkey #	S1	S2	S3	S4	S5	S6	S7	S8
MTD	141	101	101	144	101	165	108	219
MTDwS	1141	1104	1168	1243	1101	1389	1164	1449
PCG	99.75%	99.80%	99.72%	99.57%	99.80%	99.39%	99.74%	99.21%
Rank	1	1	1	1	1	1	1	1
Broken	success	success	success	success	success	success	success	success

Advanced metrics

Guessing entropy

- Rank of the good subkey according to the number of traces processed
- Based on the analysis of several independent sets of traces

- Example

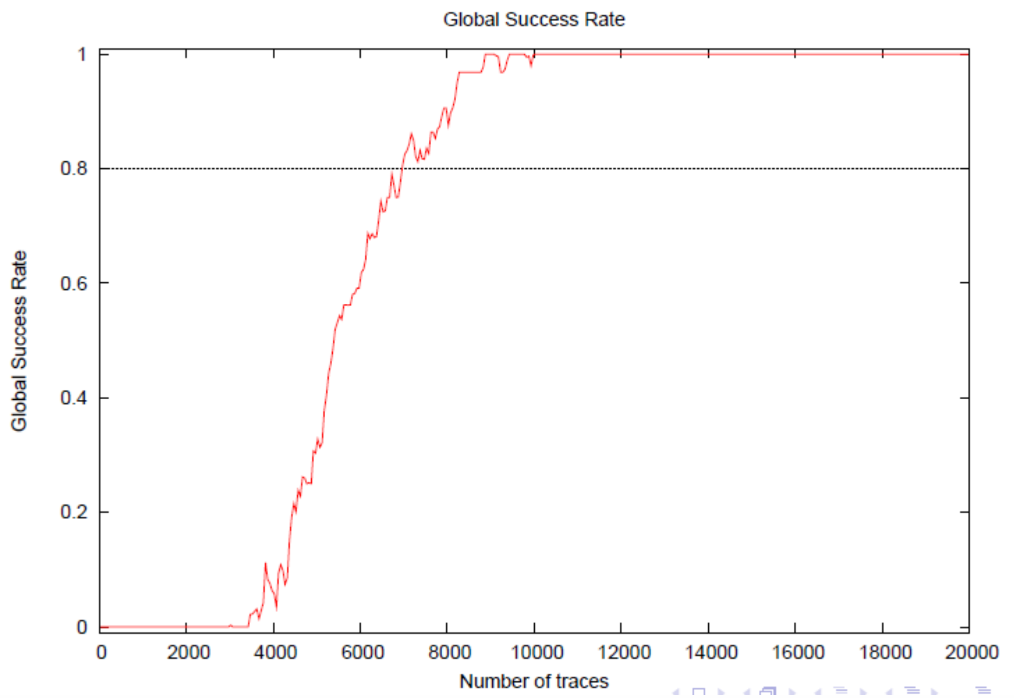


Advanced metrics

Success rate

- Percentage of correct subkeys found according to the number of traces processed
- Based on the analysis of several independent sets of traces

- Example



Countermeasures

Objective

- Remove the link between intermediate values and consumption

Masking

- A random mask obscures the intermediate values
- Can be at different levels (algorithmic -> gates)

Hiding

- Make consumption independent of intermediate values
- Special logic, addition of hazards, reduction of SNR

Software countermeasures

Temporal contingencies: operations are shifted in time

- Using NOP
- Adding random delays
- Use of "false" variables and operations (sequence scrambling)
- Data balancing (redundancy to keep the HW constant)

Swapping instructions

- Changing the order of execution of SBOXes

Masking

- Xor

Harware countermeasures

Adding noise

- HW generator using an RNG
  - Overall consumption is increased (problem?)

Consumption filtering

- RLC filters
- Use of active components
- Isolated power supply

New logics

- Balanced logic
- dual rail, triple rail

Real life examples

Source: Philips

{* SECURITY *}

IoT worm can hack Philips Hue lightbulbs, spread across cities

Easy chain reaction hack would spread across Paris, boffins say

Darren Pauli Thu 10 Nov 2016 // 06:02 UTC

This NXP side-channel attack can clone Google Titan 2FA keys

Charlie Osborne 12 January 2021 at 13:28 UTC
Updated: 12 January 2021 at 14:49 UTC

Google Hardware Authentication

Source: Google store

Questions?

Florent Bruguier

Contact : florent.bruguier@lirmm.fr