Introduction to Side Channel Attacks

Arnaud Tisserand

CNRS, Lab-STICC

ARCHI'19, Lorient



Summary

Introduction

Applications with Security Requirements

- medical devices, *e*-health
- { home | building | factory } automation
- *e*-commerce
- transports
- communications: cell. phones, Internet, industrial networks, ...
- IOT, WSN, RFID...
- embedded systems
- { cloud | fog | edge | ... } computing
- smart { grids | cars | cities | buildings | ... }
- defense
- digital administration
- *etc.*

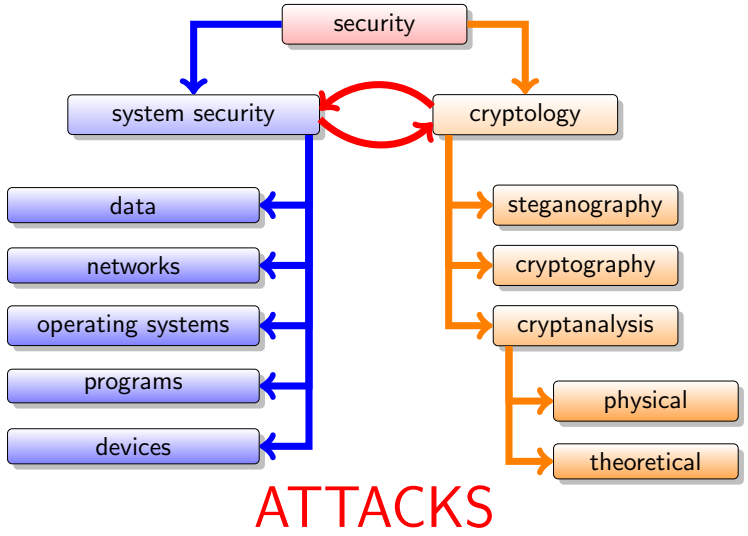
Security and Embedded Systems

Integrated circuits perform security tasks, somewhere in the system...

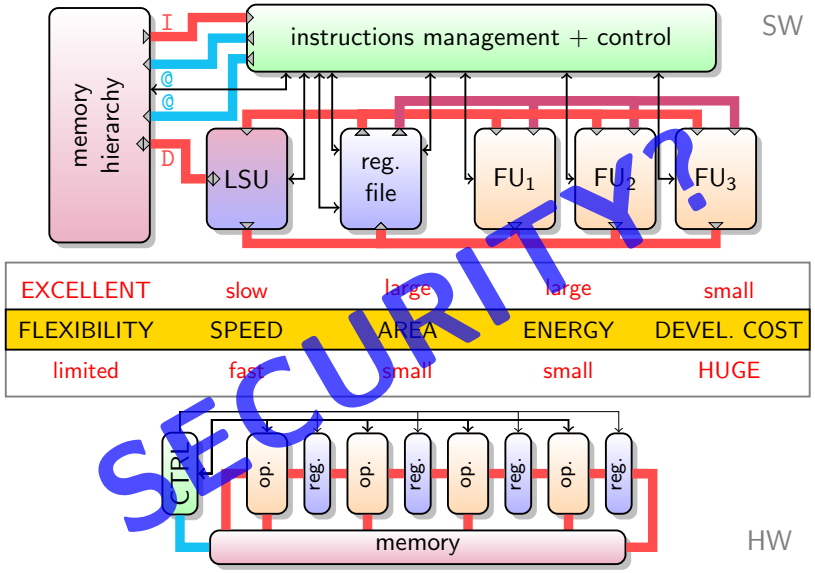
Examples where a close access is difficult:

Examples where a close access can be possible:

Security Aspects



Software vs Hardware Support



Crytographic Background

Cryptographic Features

Objectives:

- Confidentiality
- Integrity
- Authenticity
- Non-repudiation
- ...

Cryptographic primitives:

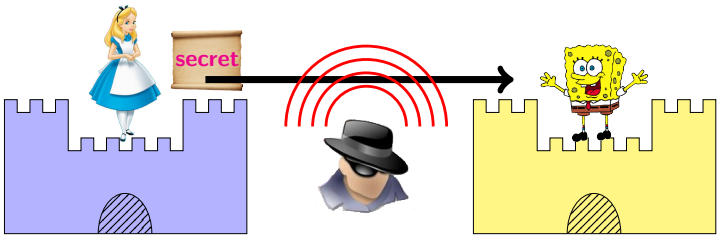
- Encryption
- Digital signature
- Hash function
- Random numbers generation
- ...

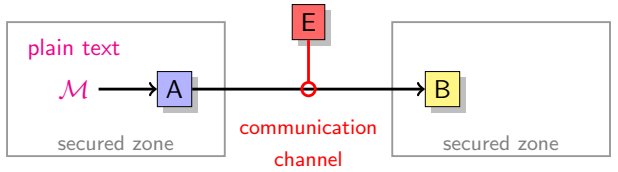
Implementation issues:

- Performances: speed, delay, throughput, latency
- Cost: device (memory, size, weight), low power/energy consumption, design
- Security: protection against attacks

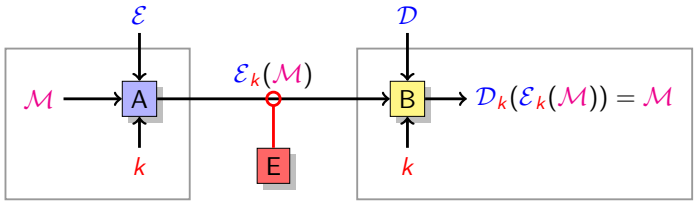
Basic Cyphering

Alice wants to secretly send a message to Bob in such a way Eve (eavesdropper/spy) **does not** get any information



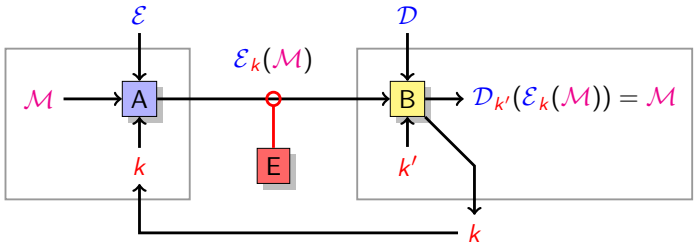


Symmetric / Private-Key Cryptography



- A: Alice, B: Bob
- $\mathcal{M}$: plain text/message
- $\mathcal{E}$: encryption/ciphering algorithm, $\mathcal{D}$: decryption/deciphering algorithm
- k: secret key to be shared by A and B
- $\mathcal{E}_k(\mathcal{M})$: encrypted text
- $\mathcal{D}_k(\mathcal{E}_k(\mathcal{M}))$: decrypted text
- E: eavesdropper/spy

Asymmetric / Public-Key Cryptography



- k: B's public key (known to everyone including E)
- $\mathcal{E}_k(\mathcal{M})$: ciphered text
- k': B's private key (must be kept secret)
- $\mathcal{D}_{k'}(\mathcal{E}_k(\mathcal{M}))$: deciphered text

Symmetric or Asymmetric Cryptography?

Private-key or symmetric cryptography:

- simple algorithms
  - → fast computation
  - → limited cost (silicon area, energy)
- ☹ requires a key exchange
- ☹ key distribution problem for n persons

Public-key or asymmetric cryptography:

- no key exchange required
- only 2 keys per person (1 private, 1 public)
- allows digital signature
- ☹ more complex algorithms
  - → slower computation
  - → higher cost

Advanced Encryption Standard (AES)

Established by NIST in 2001

Symmetric encryption

Block size: 128 bits

key length	#round
128	10
192	12
256	14

Based on substitution-permutation network

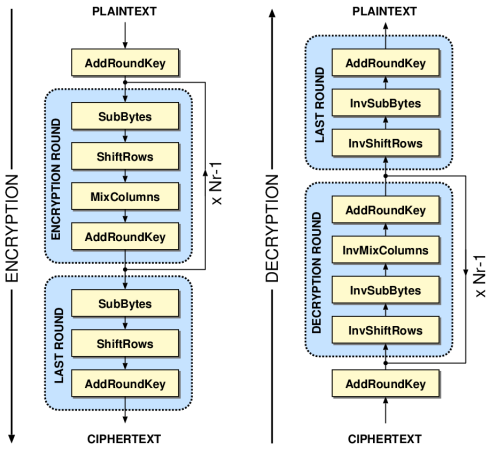
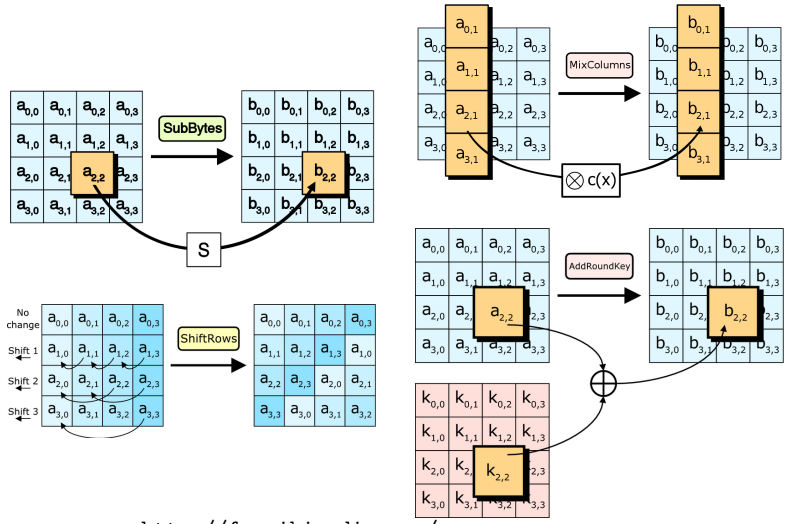


Image source: `http://fr.wikipedia.org/`

NIST: National Institute of Standards and Technology

AES Round Operations



Images source: `http://fr.wikipedia.org/`

RSA Asymmetric Cryptosystem (1/2)

Published in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman [12]

Key generation (Alice side)

- Choose two large prime integers p and q
- Compute the modulus $n = pq$
- Compute $\varphi(n) = (p-1)(q-1)$
- Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$
- Compute $d = e^{-1} \bmod \varphi(n)$
- Private key (kept secret by Alice): d and also $p, q, \varphi(n)$
- Public key (published): (n, e)

RSA Asymmetric Cryptosystem (2/2)

Private key (Alice): d Public key (all): (n, e)

Encryption (Bob side):

- convert the message `M` to an integer m $(1 < m < n$ and $\gcd(m, n) = 1)$
- compute the cipher text $c = m^e \bmod n$

Decryption (Alice side):

- compute $m = c^d \bmod n$
- convert the integer m to the message `M`

Theoretical security: integer factorization, *i.e.* computing (p, q) knowing n, is not possible when n is large enough

Modular Exponentiation

Computation of operations such as : $a^b \bmod n$

$$a^b = \underbrace{a \times a \times a \times a \times \ldots \times a \times a \times a}_{a \text{ appears } b \text{ times}}$$

Order of magnitude of exponents: $2^{\text{size of exponent}} \rightsquigarrow 2^{1024} \ldots 2^{2048} \ldots 2^{4096}$

Fast exponentiation principle:

$$\begin{aligned} a^b &= (a^2)^{\frac{b}{2}} && \text{when } b \text{ is even} \\ &= a \times (a^2)^{\frac{b-1}{2}} && \text{when } b \text{ is odd} \end{aligned}$$

Least significant bit of the exponent: `bit` $= 0 \rightsquigarrow$ even and `bit` $= 1 \rightsquigarrow$ odd

Square and Multiply Algorithm

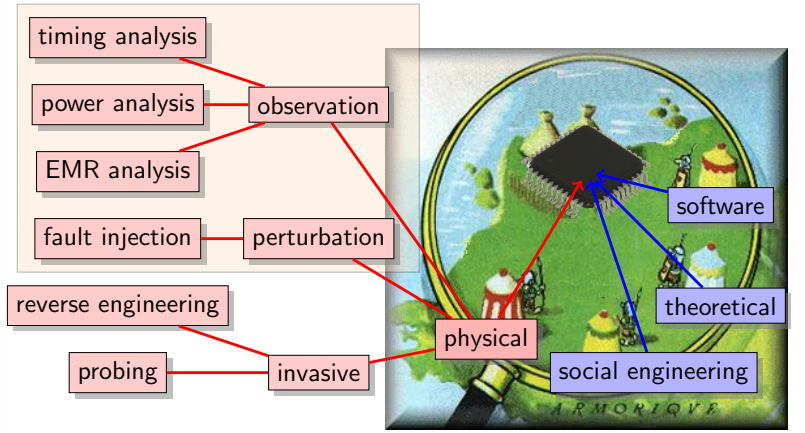
input: a, b, n where $b = (b_{t-1} b_{t-2} \ldots b_1 b_0)_2$
output: $a^b \bmod n$

```
r = 1
for i from 0 to t-1 do
    if b_i = 1 then
        r = r · a  mod n
    endif
    a = a^2  mod n
endfor
return r
```

This is the right to left version (there exists a left to right one)

Side Channel Attacks

Main Types of Attacks



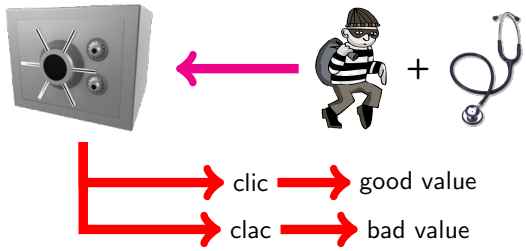
EMR = Electromagnetic radiation

Side Channel Attacks (SCAs) (1/2)

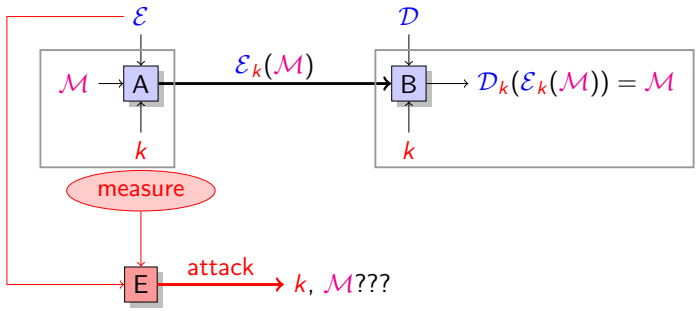
Attack: attempt to find, without any knowledge about the secret:

- the message (or parts of the message)
- informations on the message
- the secret (or parts of the secret)

"Old style" side channel attacks:



Side Channel Attacks (SCAs) (2/2)



General principle: measure external parameter(s) on a running device in order to deduce internal (secret) informations

What Should be Measured?

Answer: everything that can "enter" and/or "get out" in/from the device

- time
- power consumption
- electromagnetic radiation
- temperature
- sound
- number of cache misses
- number and type of error messages
- ...

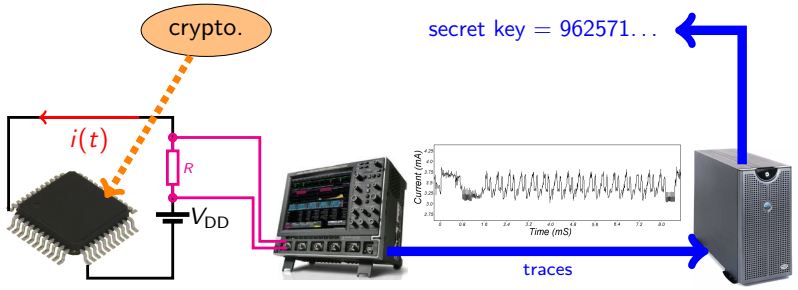
The measured parameters may provide informations on:

- global behavior (temperature, power, sound...)
- local behavior (EMR, # cache misses...)

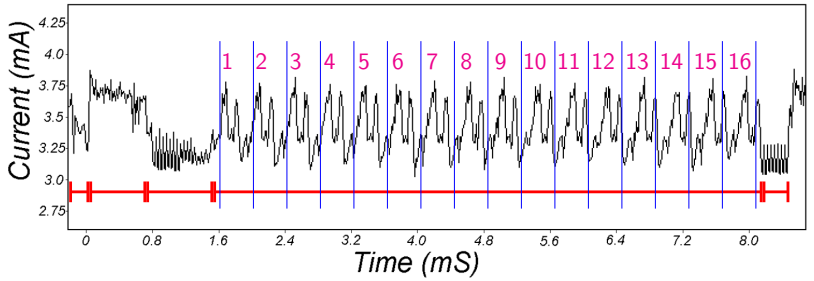
Power Consumption Analysis

General principle:

1. measure the current $i(t)$ in the cryptosystem
2. use those measurements to "deduce" secret informations



"Read" the Traces



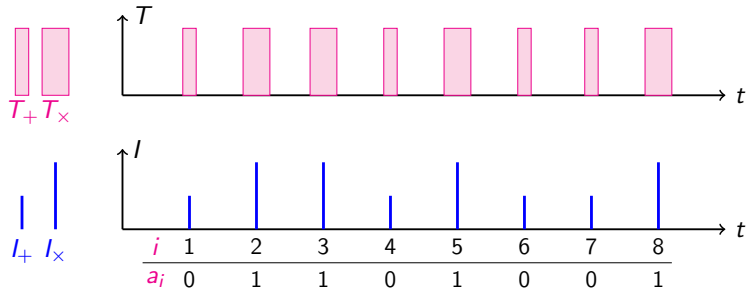
- algorithm → decomposition into steps
- detect loops
  - constant time for the loop iterations
  - non-constant time for the loop iterations

Source: [6] Kocher, Jaffe and Jun. **Differential Power Analysis**, Crypto99

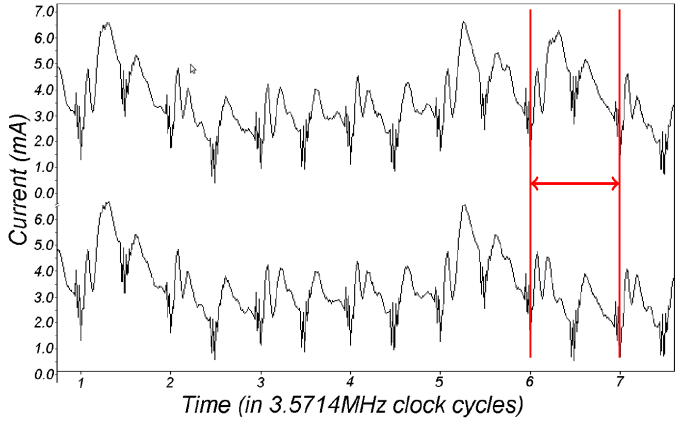
Differences & External Signature

An algorithm has a current signature and a time signature:

```
r = c0
for i from 1 to n do
    if ai = 0 then
        r = r+c1
    else
        r = r×c2
```



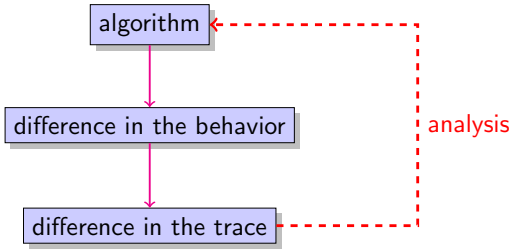
Simple Power Analysis (SPA)



Source: [6]

SPA in Practice

General principle:

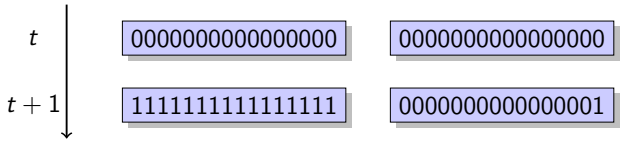


Methods: interpretation of the differences in

- control signals
- computation time
- operand values
- ...

Limits of the SPA

Example of behavior difference: (activity into a register)

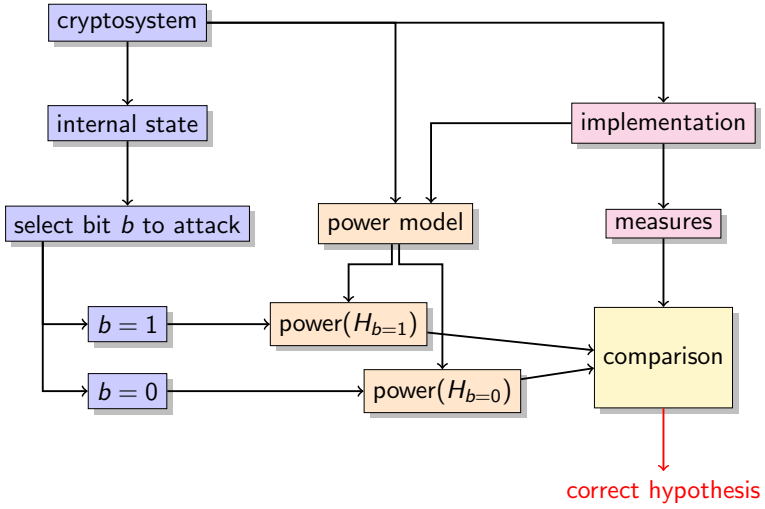


Important: a small difference may be evaluated has a noise during the measurement → traces cannot be distinguished

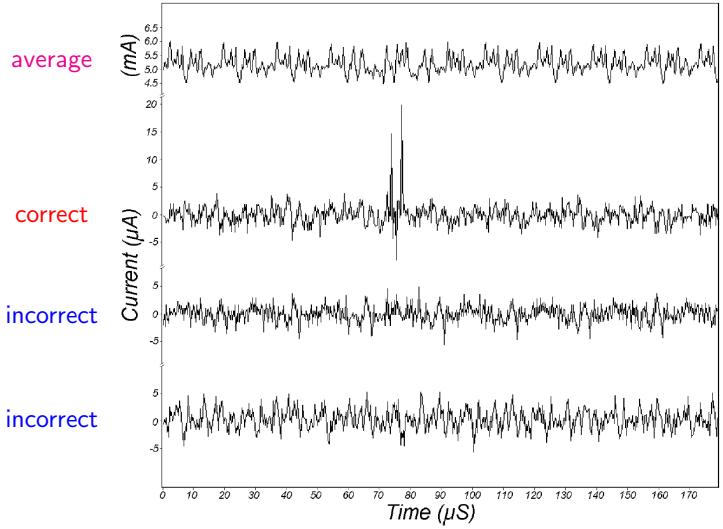
Question: what can be done when differences are too small?

Answer: use statistics over several traces

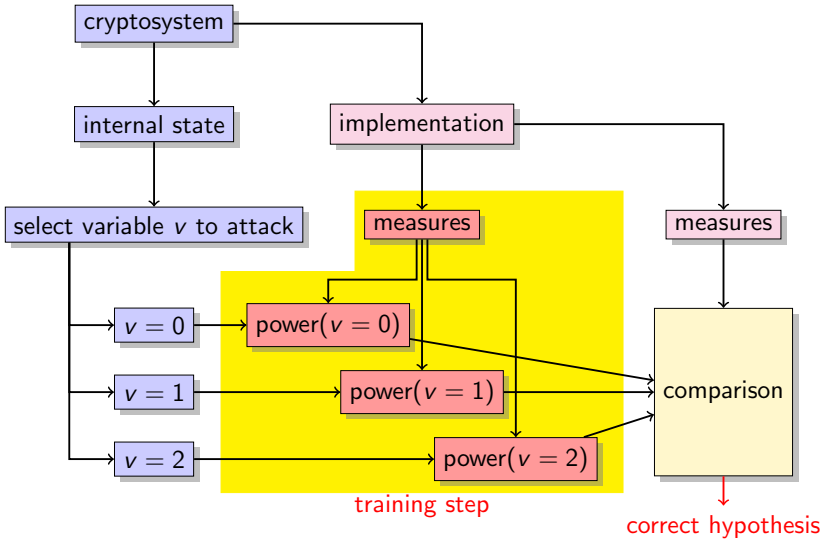
Differential Power Analysis (DPA)



Differential Power Analysis (DPA) Example

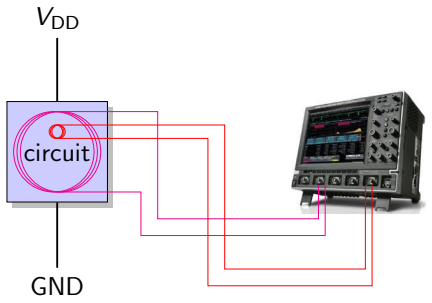


Template Attack



Electromagnetic Radiation Analysis (1/2)

General principle: use a probe to measure the EMR



EMR measurement:

- global EMR with a large probe
- local EMR with a micro-probe

Electromagnetic Radiation Analysis (2/2)

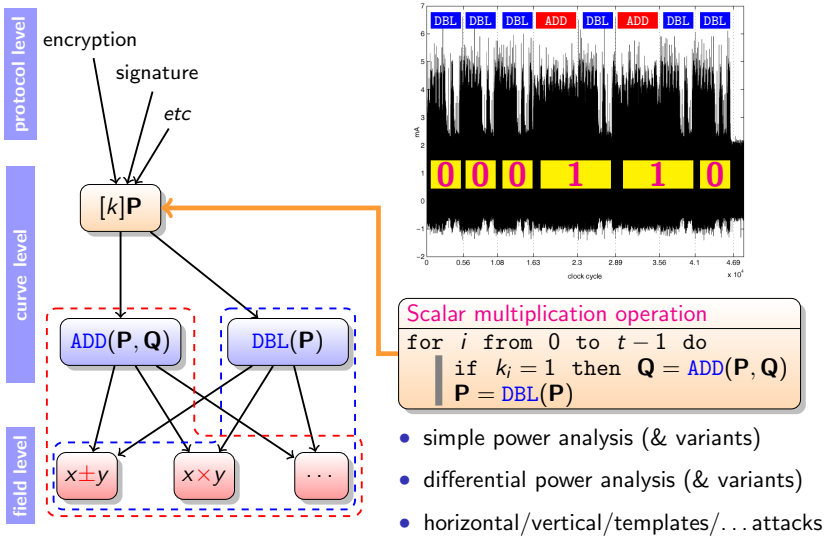
EMR analysis methods:

- simple electromagnetic analysis: SEMA
- differential electromagnetic analysis: DEMA

Local EMR analysis may be used to determine internal architecture details, and then select weak parts of the circuit for the attack

→ X-Y table

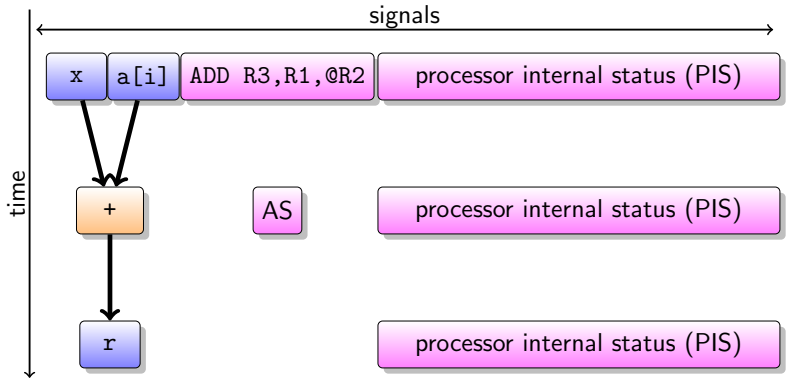
Side Channel Attack on Elliptic Curve Crypto



- simple power analysis (& variants)
- differential power analysis (& variants)
- horizontal/vertical/templates/... attacks

Activity in a Processor

Operation to be executed: `r ← x + a[i]`



- AS: ALU status
- PIS: pipeline management, bypasses, memory hierarchy, branch predictor, monitoring, etc)

Protections

Protections, Countermeasures

Principles for preventing attacks:

- embed additional protection blocks
- modify the original circuit into a secured version
- application levels: circuit, architecture, algorithm, protocol...

Countermeasures:

- electrical shielding
- detectors, estimators, decoupling
- use uniform computation durations and power consumption
- use detection/correction codes (for fault injection attacks)
- provide a random behavior (algorithms, representation, operations...)
- add noise (e.g. masking, useless instructions/computations)
- circuit reconfiguration (algorithms, block location, representation of values...)

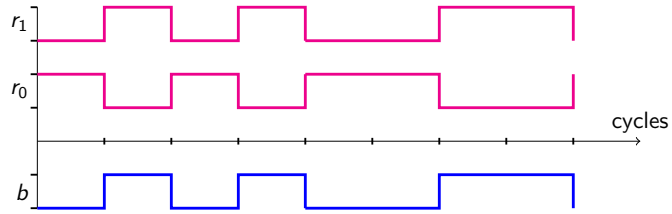
Low-Level Coding and Circuit Activity

Assumptions:

- b is a bit (i.e. $b \in \{0,1\}$, logical or mathematical value)
- electrical states for a wire ▬ : V_{DD} (logical 1) or GND (logical 0)

Low-level codings of a bit:

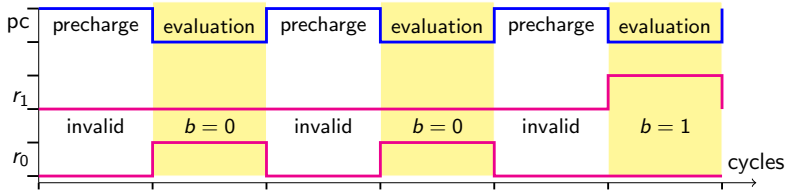
	$b=0$	$b=1$
standard	▬ GND	▬ V_{DD}
dual rail	r_0 = V_{DD}, r_1 = GND ⎤ $(1,0)_{DR}$	r_0 = GND, r_1 = V_{DD} ⎤ $(0,1)_{DR}$



Circuit Logic Styles

Countermeasure principles: uniformize circuit activity and exclusive coding

Solution based on precharge logic and dual-rail coding:



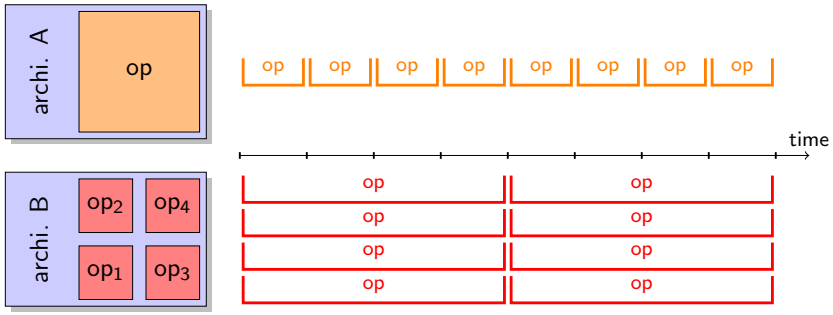
Solution based on validity line and dual-rail coding:

Important overhead: silicon area and local storage (registers)

Countermeasure: Architecture

Increase internal parallelism:

- replace one fast but big operator
- by several instances of a small but slow one



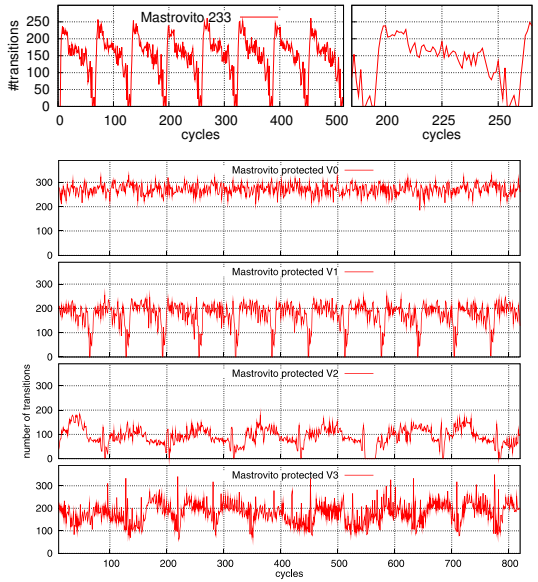
Protected Multipliers

Unprotected

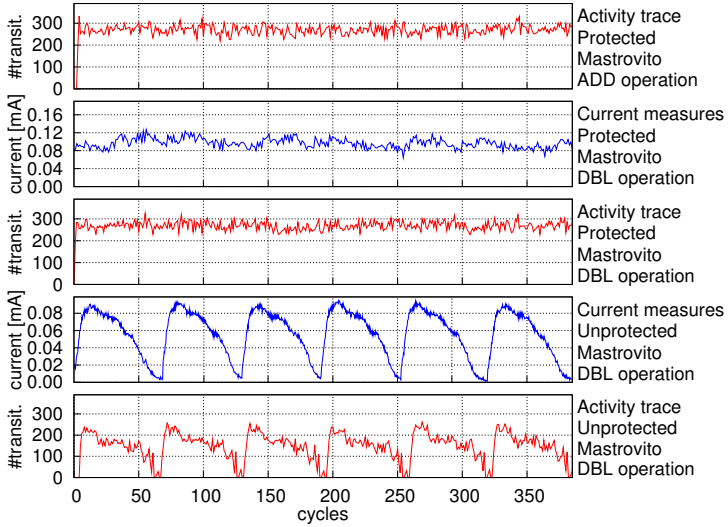
Protected

Overhead:
Area/time $< 10\,\%$

References:
PhD D. Pamula [8]
Articles: [11], [10], [9]



Protected (Old) Accelerator

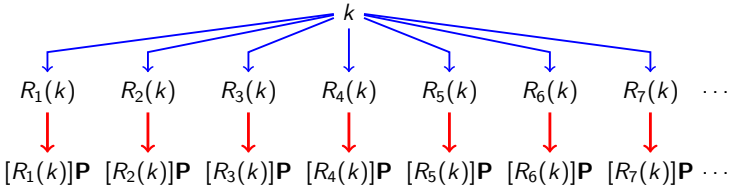


Warning: old dedicated accelerator (similar behavior is expected for our new one)

Arithmetic Level Countermeasures

Redundant number system =

- a way to improve the performance of some operations
- a way to represent a value with different representations



Important property: $\forall i \quad [R_i(k)]\mathbf{P} = [k]\mathbf{P}$

Proposed solution: use random redundant representations of k

Double-Base Number System

Standard radix-2 representation:

$$k = \sum_{i=0}^{t-1} k_i 2^i =$$

2^{t-1}	2^{t-2}	...	2^2	2^1	2^0	implicit weights
k_{t-1}	k_{t-2}	$\cdots$	k_2	k_1	k_0	t explicit digits

Digits: $k_i \in \{0, 1\}$, typical size: $t \in \{160, \ldots, 600\}$

Double-Base Number System (DBNS):

$$k = \sum_{j=0}^{n-1} k_j 2^{a_j} 3^{b_j} =$$

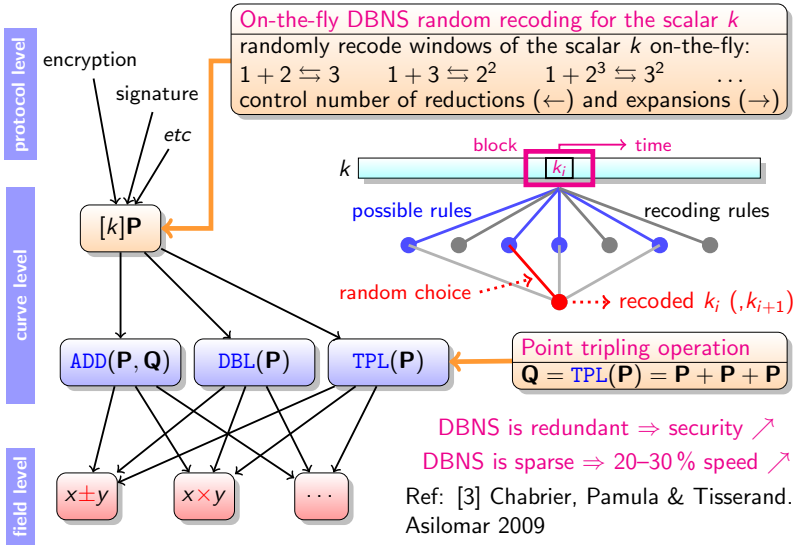
k_{n-1}	$\cdots$	k_1	k_0	n $(2,3)-$terms
a_{n-1}	$\cdots$	a_1	a_0	explicit "digits"
b_{n-1}	$\cdots$	b_1	b_0	explicit ranks

$a_j, b_j \in \mathbb{N}$, $k_j \in \{1\}$ or $k_j \in \{-1, 1\}$, size $n \approx \log t$

DBNS is a very redundant and sparse representation: $1701 = (11010100101)_2$

$$\begin{array}{rcccl} 1701 & = & 243 + 1458 & = & 2^0 3^5 + 2^1 3^6 \quad = \quad (1,0,5),(1,1,6) \\ & = & 1728 - 27 & = & 2^6 3^3 - 2^0 3^3 \quad = \quad (1,6,3),(-1,0,3) \\ & = & 729 + 972 & = & 2^0 3^6 + 2^2 3^5 \quad = \quad (1,0,6),(1,2,5) \\ & \ldots & & & \end{array}$$

Randomized DBNS Recoding of the Scalar k



DBNS is redundant ⇒ security ↗

DBNS is sparse ⇒ 20–30 % speed ↗

Ref: [3] Chabrier, Pamula & Tisserand. Asilomar 2009

Conclusion

Conclusion

- Physical attacks are very serious threats
- Attacks are more and more efficient (many variants, AI, DL)
- Security analysis and integration is mandatory at all levels (specification, algorithm, operation, implementation, test, ...)
- Security = *function*(secret value, attacker capabilities)
- Security = trade-off between performances, robustness and cost
- Security = computer science + microelectronics + mathematics

Current works examples:

- Secure processors and accelerators
- Hardware operators/accelerators with reduced activity variations
- Representation of numbers with error detection/correction features
- Circuit reconfiguration (representations, algorithms)
- Design space exploration with security objectives/metrics
- Methods/tools for automating security analysis
- CAD tools with security improvement capabilities

Resources: Conferences, Workshops, Journals, etc

- International Association for Cryptologic Research (IACR) Eprint Archives
- ACM Special Interest Group on Security, Audit and Control (SIGSAC)
- IEEE Computer Society's Technical Committee on Security and Privacy (TCSP)
- French national working group on Code & Crypto (C2) of the GDR IM
- French national working group on Security of Embedded Systems of the GDR SoC
- Conferences, workshops: CHES, FDTC, COSADE, CARDIS, CryptArchi ...
- Journals: TCHES, Journal of Cryptographic Engineering, IEEE Trans. on Computers, Circuits and Systems, VLSI Systems, ...
- `http://www.crypto.ruhr-uni-bochum.de/en_sclounge.html`
- `http://www.schneier.com/`

References I

[1] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan.
The sorcerer's apprentice guide to fault attacks.
Proceedings of the IEEE, 94(2):370–382, February 2006.

[2] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache.
Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures.
Proceedings of the IEEE, 100(11):3056–3076, November 2012.

[3] T. Chabrier, D. Pamula, and A. Tisserand.
Hardware implementation of DBNS recoding for ECC processor.
In *Proc. 44rd Asilomar Conference on Signals, Systems and Computers*, pages 1129–1133, Pacific Grove, California, U.S.A., November 2010. IEEE.

[4] D. Karaklajic, J.-M. Schmidt, and I. Verbauwhede.
Hardware designer's guide to fault attacks.
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 21(12):2295–2306, December 2013.

[5] P. C. Kocher.
Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems.
In *Proc. Advances in Cryptology (CRYPTO)*, volume 1109 of *LNCS*, pages 104–113. Springer, August 1996.

[6] P. C. Kocher, J. Jaffe, and B. Jun.
Differential power analysis.
In *Proc. Advances in Cryptology (CRYPTO)*, volume 1666 of *LNCS*, pages 388–397. Springer, August 1999.

[7] F. Koeune and F.-X. Standaert.
A tutorial on physical security and side-channel attacks.
In *5th International School on Foundations of Security Analysis and Design (FOSAD)*, volume 3655 of *LNCS*, pages 78–108. Springer-Verlag, 2005.

[8] D. Pamula.
Arithmetic Operators on GF(2^m) for Cryptographic Applications: Performance - Power Consumption - Security Tradeoffs.
Phd thesis, University of Rennes 1 and Silesian University of Technology, December 2012.

References II

[9] D. Pamula, E. Hrynkiewicz, and A. Tisserand.
Analysis of GF(2^{233}) multipliers regarding elliptic curve cryptosystem applications.
In *11th IFAC/IEEE International Conference on Programmable Devices and Embedded Systems (PDeS)*, pages 271–276, Brno, Czech Republic, May 2012.

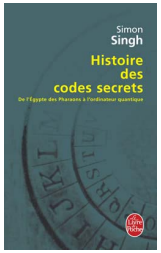
[10] D. Pamula and A. Tisserand.
GF(2^m) finite-field multipliers with reduced activity variations.
In *4th International Workshop on the Arithmetic of Finite Fields*, volume 7369 of *LNCS*, pages 152–167, Bochum, Germany, July 2012. Springer.

[11] D. Pamula and A. Tisserand.
Fast and secure finite field multipliers.
In *Proc. 18th Euromicro Conference on Digital System Design (DSD)*, pages 653–660, Madeira, Portugal, August 2015.

[12] R. L. Rivest, A. Shamir, and L. Adleman.
A method for obtaining digital signatures and public-key cryptosystems.
Communications of the ACM, 21(2):120–126, February 1978.

Good Books (in French)

Histoire des codes secrets
Simon Singh
1999
Livre de poche



Mathématiques, espionnage et piratage informatique
Joan Gomez
2010
Le monde est mathématique, RBA

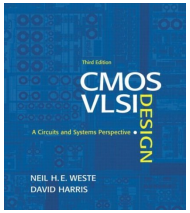
Good Books (in French)

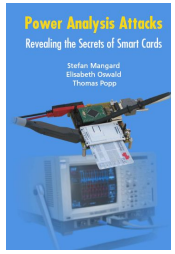
Cryptographie appliquée
Bruce Schneier
1997, 2ème édition
Wiley
ISBN: 2–84180–036–9

Micro et nano-électronique
Bases, Composants, Circuits
Hervé Fanet
2006
Dunod
ISBN: 2–10–049141–5

Good Books (in English)

CMOS VLSI Design
A Circuits and Systems Perspective
Neil Weste and David Harris
3rd edition, 2004
Addison Wesley
ISBN: 0–321–14901–7

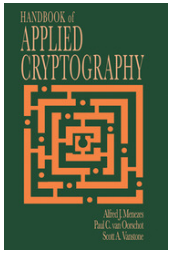




Power Analysis Attacks
Revealing the Secrets of Smart Cards
Stefan Mangard, Elisabeth Oswald and Thomas Popp
2007
Springer
ISBN:978-0-387-30857-9

Good Books (in English)

Handbook of Applied Cryptography
Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone
2001
CRC Press
ISBN:0-8493-8523-7
Web: `http://cacr.uwaterloo.ca/hac/`



The end, questions ?

Contact:

- mailto:arnaud.tisserand@univ-ubs.fr
- http://www-labsticc.univ-ubs.fr/~tisseran
- CNRS
  Lab-STICC, Centre Recherche UBS
  Rue St Maudé. BP 92116. 56321 Lorient cedex, France

Thank you