Basics of Fault Attacks

Christophe Giraud
Oberthur Card Systems

Hugues Thiebeauld
Thales Microelectronics

Overview

- How to perform a fault attack?
- The different kinds of faults
- Fault attacks on symmetric and asymmetric cryptosystems
- About security
- Conclusion

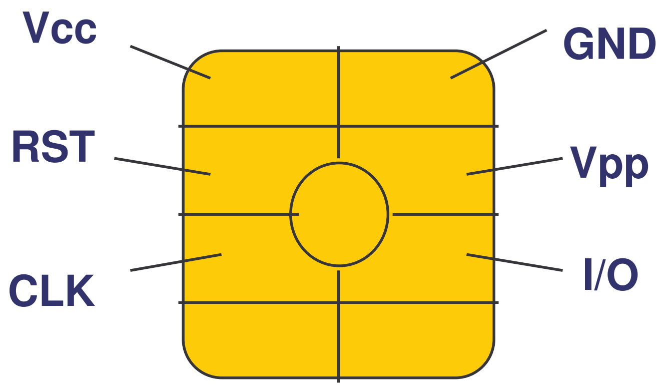
Smart Cards & Fault Induction

- Before 1996 :

  Smart cards security ≈ Mathematical security of cryptographic algorithms
- Side Channel Attacks :
  - 1996 : Timing Attacks
  - 1998 : Power consumption Analysis
  - 2001 : Electromagnetic Analysis
- Fault Attacks :
  - 1996 : Fault Induction Attack on the RSA CRT

How to perform a fault attack?

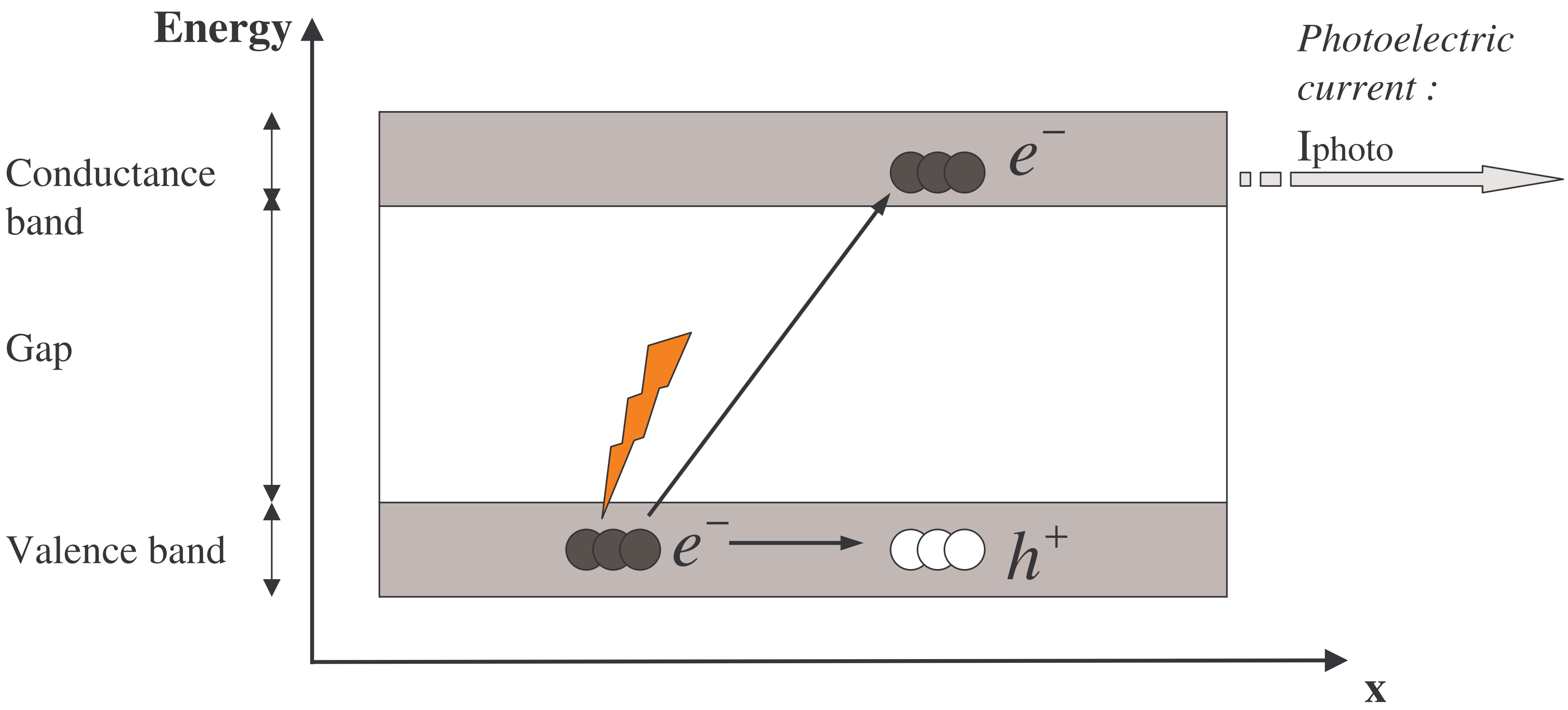
- **By using a glitch (power, clock)**
  - Non-invasive
  - Disturbs the whole component
  - Relatively easy to counteract



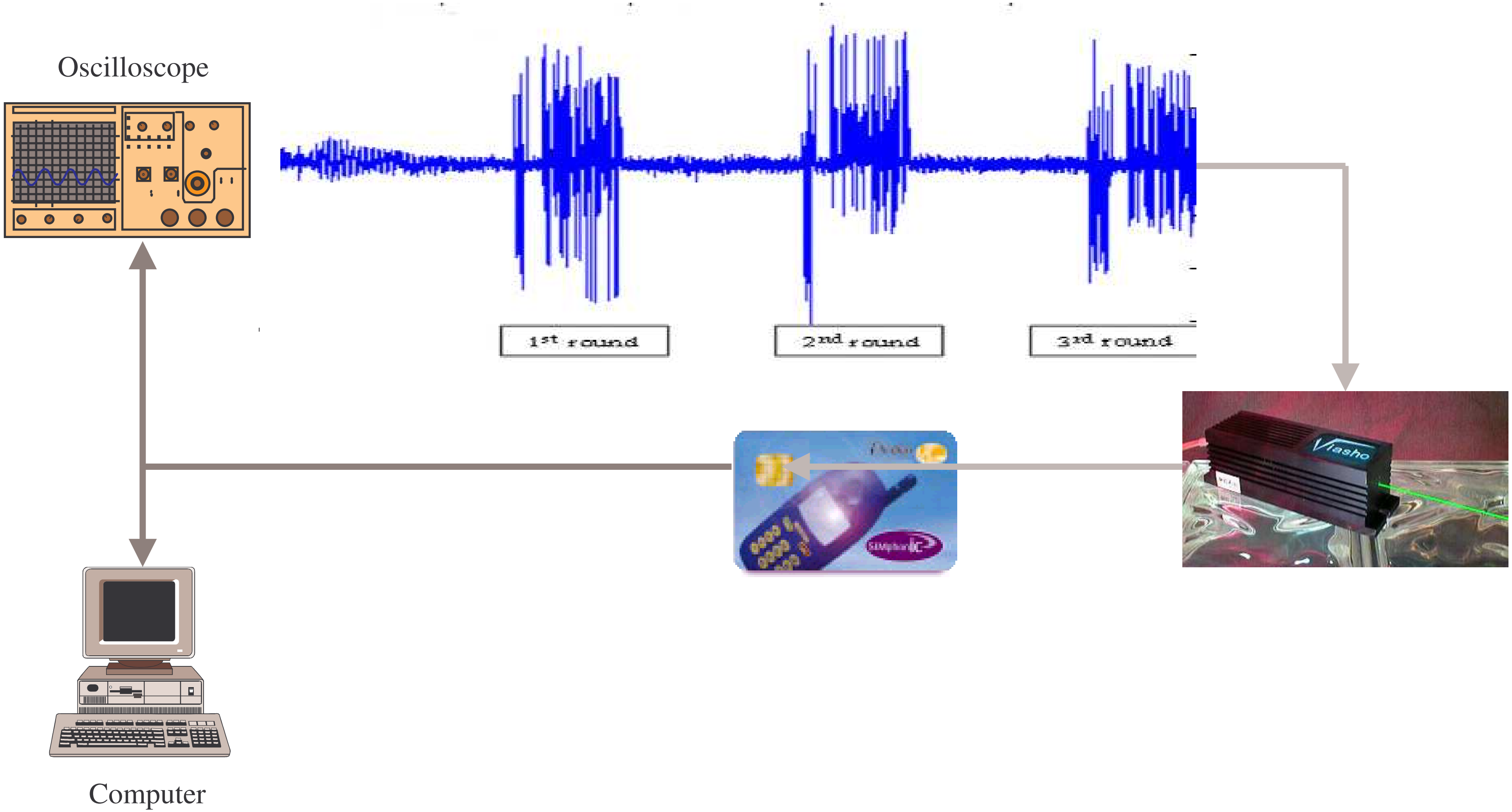
- **By using a light flash**
  - Semi-invasive
  - Equipment: a camera flash or a laser
  - A means of disturbing a precise part of the component

- **Other ways…**

Light attacks – Principle



Important parameter : the synchronisation



The different kinds of faults

- **Permanent faults**:
  - Modify definitively the value of a memory cell (DATA or EEPROM)
  - Very powerful if the data is related to sensitive objects (PIN, keys,…)
- **Transient faults**:
  - The most common
  - Disturbance of a code execution or of a computation
  - CPU and registers
  - Reading code / data

Fault attacks on asymmetric cryptosystems

- **RSA** (*Boneh, DeMillo & Lipton*)
  - Secret key : (d, p, q)
  - Public key : (e, N=p.q)
- RSA-CRT:
  - dp = d mod p-1   and   dq = d mod q-1
  - $S_p = m^{dp} \bmod p$ and $S_q = m^{dq} \bmod q$
  - $S = ((((S_q - S_p) \bmod q) \times (p^{-1} \bmod q)) \times p + S_p) \bmod N$
- Disturbance during the computation of S_p (or S_q) ➔ $\hat{S}$

$$q \text{ (or } p) = GCD(S - \hat{S}, N)$$

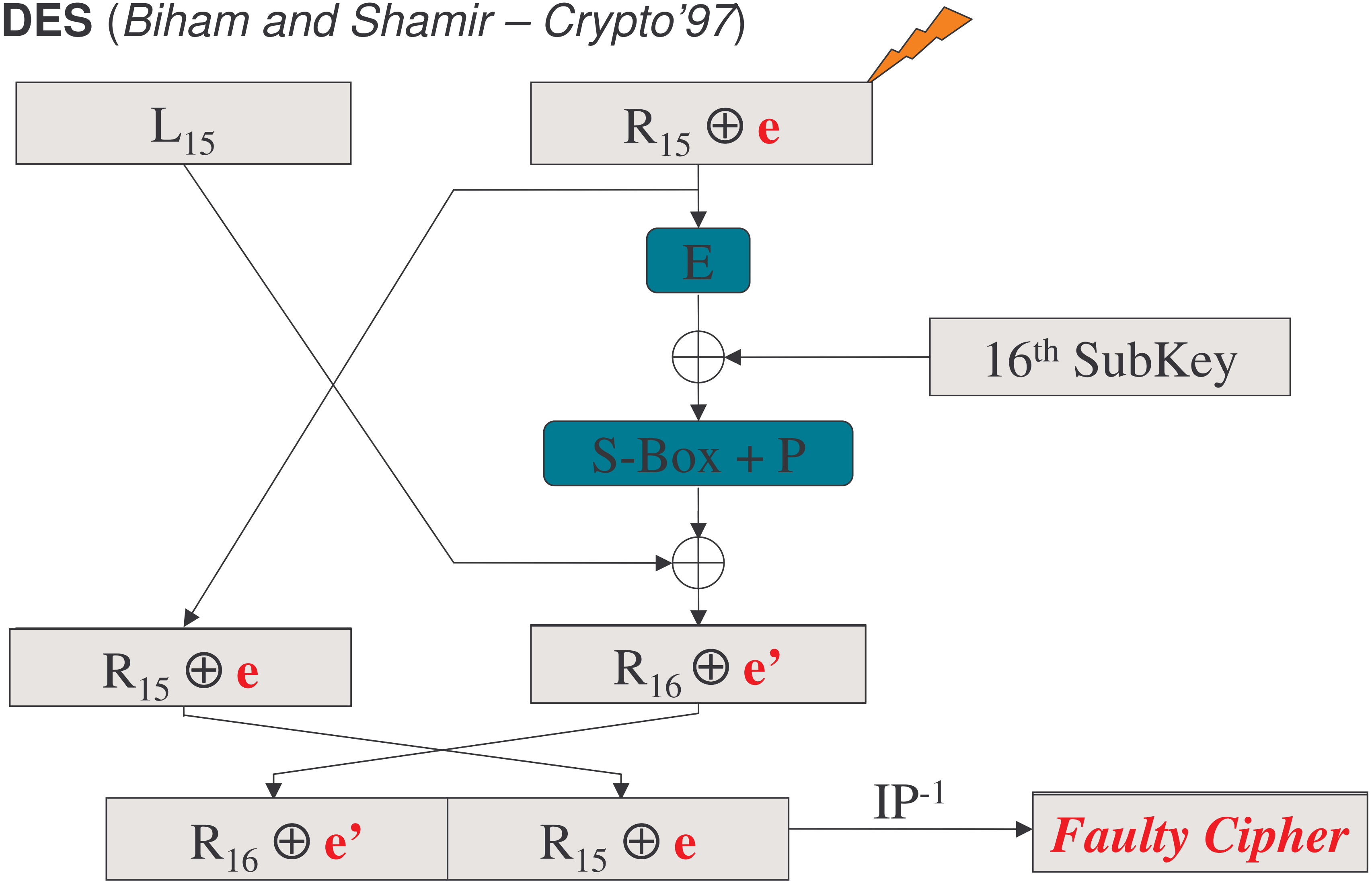
$$= GCD(m - \hat{S}^e, N) \qquad (\textit{Lenstra})$$

Oberthur Card Systems

THALES MICROELECTRONICS

Fault attacks on symmetric cryptosystems

- **DES** (*Biham and Shamir – Crypto'97*)



Fault attacks on symmetric cryptosystems

- **DES** (*Biham and Shamir – Crypto'97*)
  - $IP(C)_{32\text{-}63} = R_{15}$
  - $IP(C)_{32\text{-}63} \oplus IP(FC)_{32\text{-}63} = e$
  - $IP(C)_{0\text{-}31} \oplus IP(FC)_{0\text{-}31} = e'$
  - $IP(C)_{0\text{-}31} \oplus IP(FC)_{0\text{-}31} = PoS(\,E(R_{15}) \oplus K_{16}\,) \oplus PoS(\,E(R_{15} \oplus e)) \oplus K_{16}\,)$

→ Eliminate the K_{16}'s which do not satisfy:

$$e' = PoS(\,E(R_{15}) \oplus K_{16}\,) \oplus PoS(\,E(R_{15} \oplus e\,)) \oplus K_{16}\,)$$

- The secret key can be recovered by using 2 faulty ciphertexts.

Fault attacks on symmetric cryptosystems

- **AES** (*Piret and Quisquater – CHES 2003*)
  - Modification of 1 byte of the *MixColumns*' input has an impact on 4 bytes.
  - Modification of 1 byte between the *MixColumns* of the 7th round and the *MixColumns* of the 8th round.
  - The secret key can be recovered by using 2 faulty ciphertexts.

About Security

- Hardware countermeasures:
  - Sensors
  - Filters
  - Dual rails
  - Desynchronisation
- Software countermeasures:
  - Must be installed on each layer of an application
  - Very costly in terms of both memory space and timings
  - ➔ Choosing appropriate countermeasures:
    - Determine the attacker’s possibilities,
    - Select the objects and the functions to protect.

Conclusion

- Easy to set up…

  but requires technical experience to successfully put such attacks into practice.
- The threat exists so the risk has to be seriously considered.
- Efficient countermeasures are well-known…

  but must be implemented both carefully and sparingly !