Fault-based Cryptanalysis on Block Ciphers

ASK 2015

Victor LOMNE

ANSSI (French Network and Information Security Agency)

Friday, October 2^{nd}, 2015 - Singapore

Agenda

1 Introduction
  a. Physical Cryptanalysis
  b. Fault-based Cryptanalysis

2 Fault Injection Means
  a. Global Faults
  b. Local Faults
  c. Other Tools

3 Cryptanalysis methods
  a. Fault Model
  b. Safe Error Attack
  c. DFA
  d. Statistical Fault Attack

4 Countermeasures
  a. Analog Level
  b. Digital Level
  c. Application to Crypto

5 Conclusion

Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Context

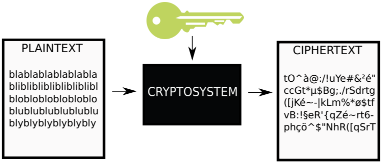
- Since the 90’s, increasing use of secure embedded devices
  - 9G smartcard ICs sold in 2013 (SIM cards, credit cards ...)

- Strong cryptography from a mathematical point of view used to manage sensitive data
  - 3DES, AES, RSA, ECC, SHA-2-3 ...

Classical Cryptanalysis

- Black-Box Model assumed in classical cryptanalysis:
  - key(s) stored in the device
  - cryptographic operations computed inside the device

black-box model



PLAINTEXT

blablablablablabla
bliblibliblibliblibl
blobloblobloblobloblo
blublublublublublu
blyblyblyblyblybly

CRYPTOSYSTEM

CIPHERTEXT

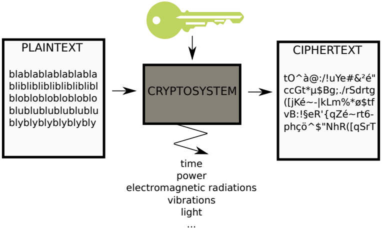
tO^à@:/!uYe#&²é"
ccGt*µ$Bg;./rSdrtg
([jKé~-|kLm%*ø$tf
vB:!§eR'{qZé~rt6-
phçö^$"NhR([qSrT

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

Secure Cipher - Unsecure Implementation (1/2)

- $[Kocher + 1996] \Rightarrow$ exploitation of physical leakages
  - cryptosystems integrated in CMOS technology
  - physical leakages correlated with computed data

gray-box model
(spy the computation)

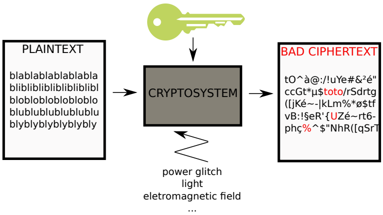


- The attacker has also access to physical leakages
- New class of attacks $\Rightarrow$ Side-Channel Attacks (SCA)

Secure Cipher - Unsecure Implementation (2/2)

- $[Boneh + 1997] \Rightarrow$ exploitation of faulty encryptions
  - the attacker can generate faulty encryptions

gray-box model
(perturbate the computation)



PLAINTEXT

blablablablablabla
blibliblibliblibliblibl
blobloblobloblo
blublublublublublu
blyblyblyblyblybly

CRYPTOSYSTEM

BAD CIPHERTEXT

tO^à@:/!uYe#&²é"
ccGt*µ$toto/rSdrtg
([jKé~-|kLm%*ø$tf
vB:!§eR'{UZé~rt6-
phç%^$"NhR([qSrT

power glitch
light
eletromagnetic field
...

- the attacker has access to correct & faulty ciphertexts
- New class of attacks $\Rightarrow$ Fault Attacks (FA)

Agenda

1 Introduction
- a. Physical Cryptanalysis
- b. Fault-based Cryptanalysis

2 Fault Injection Means
- a. Global Faults
- b. Local Faults
- c. Other Tools

3 Cryptanalysis methods
- a. Fault Model
- b. Safe Error Attack
- c. DFA
- d. Statistical Fault Attack

4 Countermeasures
- a. Analog Level
- b. Digital Level
- c. Application to Crypto

5 Conclusion

Fault based Cryptanalysis

- FA consist in perturbing the execution of the cryptographic operation in order to get faulty results leaking information on the secret

- Hypotheses are made on:
  - the targeted intermediate value
  - the effect of the injection on the intermediate value

- The attacker can then apply algorithmic methods to extract the secret from the obtained (correct and/or faulty) results

Fault Zoology (1/2)

- Different ways to generate a fault:
  - electrical glitch on pins (VCC, CLK, I/O, ...)
  - electrical glitch on the die (FBBI)
  - light injection
  - ElectroMagnetic (EM) field injection
- The duration of the fault can be:
  - transient
  - permanent

Fault Zoology (2/2)

- Different effects:
  - modification of operation flow
  - modification of operands
- Different goals:
  - Bypassing a security mechanism
    e.g. PIN verification, file access right control, secure bootchain, ...
  - Generating faulty encryptions/signatures
    ⇒ fault-based cryptanalysis
  - Combined Attacks
    JavaCard based, FA + SCA

Agenda

1 Introduction
- a. Physical Cryptanalysis
- b. Fault-based Cryptanalysis

2 Fault Injection Means
- a. Global Faults
- b. Local Faults
- c. Other Tools

3 Cryptanalysis methods
- a. Fault Model
- b. Safe Error Attack
- c. DFA
- d. Statistical Fault Attack

4 Countermeasures
- a. Analog Level
- b. Digital Level
- c. Application to Crypto

5 Conclusion

Electrical glitch on Power Supply (1/3)

- Principle:
  under/over-power a device during a very short time

- Over-powering cause unexpected electrical phenomenoms inside the IC
  e.g. local shortcuts, ...

- Under-powering slows down the processing of the IC
  e.g. bad memory read/write, ...

- Low/medium-cost attack
  ex. of equipment: custom electronic board, pulse generator, ...

Electrical glitch on Power Supply (2/3)

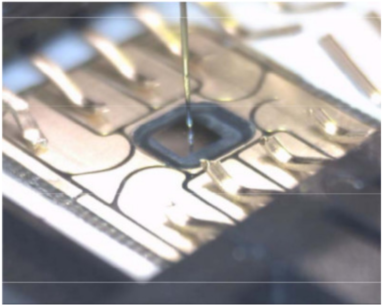
- Adversary can control:
  - Amplitude of the glitch
  - Duration of the glitch
  - Shape of the glitch

- Generally no control of the fault precision:
  - On a microcontroller running code, modification of the current executed opcode and/or operand(s)
  - On a hardware coprocessor, modification of (some of) the current processed words (e.g. registers)

Electrical glitch on Power Supply (3/3)

- Recent variant [Tobich+ 2012]:
  FBBI: Forward Body Bias Injection

- Consist in putting a needle in contact with the IC silicon through its backside



Tamper the clock (1/2)

- Principle:
  reduce one or several clock period(s)

- slows down the processing of the IC
  e.g. DFF sampling before correct computation of current instruction/combinational logic ...

- Low/medium-cost attack
  ex. of equipment: custom electronic board, signal generator, ...

Tamper the clock (2/2)

- Adversary can control:
  - Duration of the reduced clock period
  - Number of reduced clock period(s)

- Generally no control of the fault precision:
  - On a microcontroller running code, modification of the current executed opcode and/or operand(s)
  - On a hardware coprocessor, modification of (some of) the current processed words (e.g. registers)

Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Light attacks (1/2)

- Principle:
  inject a light beam into the device to disturb it

- Old school setups were using flash lamp

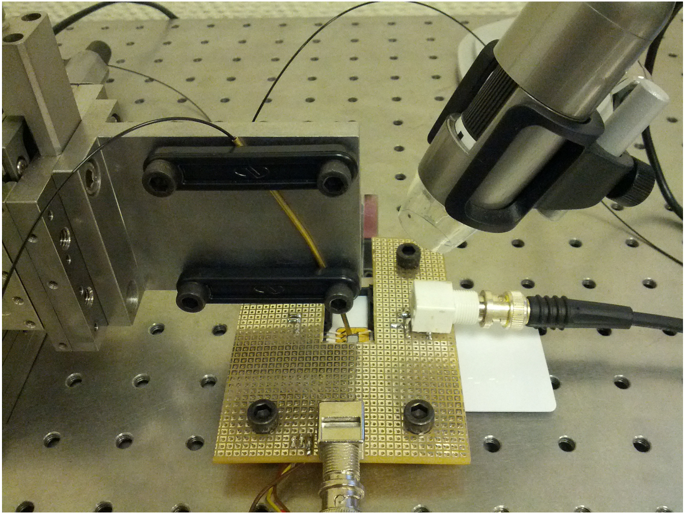
- Modern setups are based on laser modules

- It requires to open the package of the IC in order the light beam can be injected into the frontside or the backside of the die

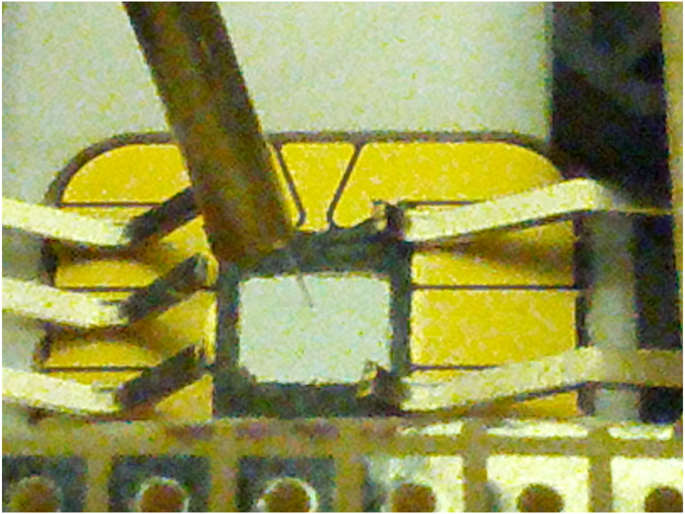
Light attacks (2/2)

- A photoelectric phenomenom transforms light energy into electrical energy, provoking unexpected behaviour of transistors
- On complex ICs with many metal layers, or on secure ICs with a shield, it can be difficult to inject light on the frontside of the IC
- As silicon is transparent to infrared light, backside light injection uses infrared light
  e.g. NIR laser diodes
- Medium/high cost attack

Laser Setup example 1 (1/2)



Laser Setup example 1 (2/2)



EMI attacks

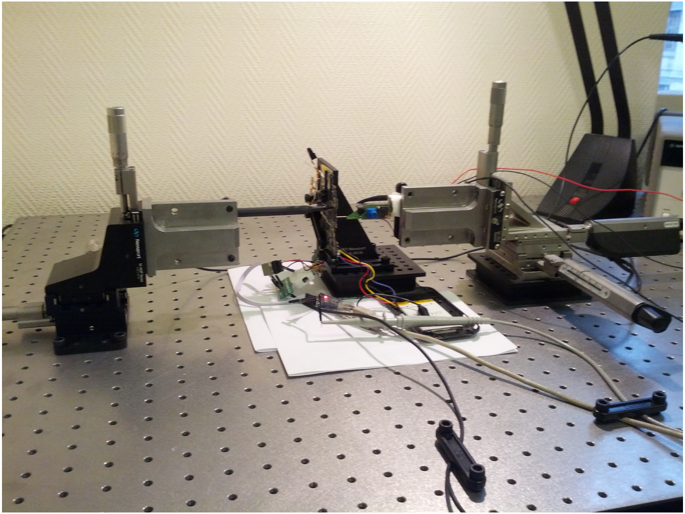
- Principle:
  inject an electromagnetic field inside the device to disturb it

- Can be done without removing the package of the IC

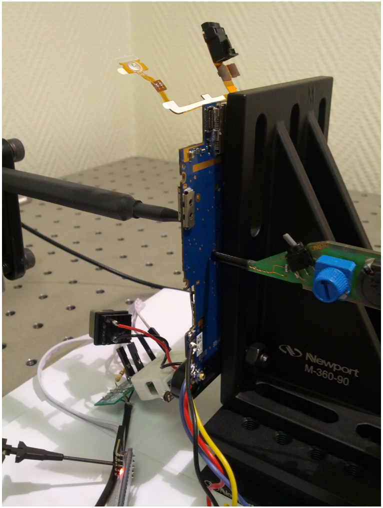
- In practice, a glitch of high power is injected into an EM sensor put above the IC
  ex. of equipment: high power pulse generator + EM sensor

- Medium/high cost attack

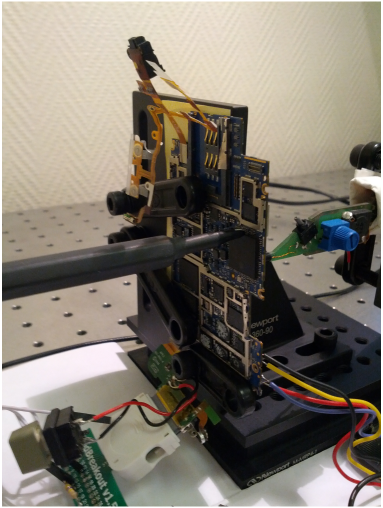
ElectroMagnetic Injection Setup example



ElectroMagnetic Injection Setup example



ElectroMagnetic Injection Setup example



Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Synchronization Mean

- In many cases, need of a synchronization mean to trig the fault at the right instant

- A classical way consists in monitoring the power consumption/EM activity of the IC such that finding the side-channel signature of the event one wants disturb

- Several solutions:
  - Using the triggering capabilities of oscilloscopes
  - Using a custom synchronization board, with real-time pattern matching mechanism

Agenda

1 Introduction
  a. Physical Cryptanalysis
  b. Fault-based Cryptanalysis

2 Fault Injection Means
  a. Global Faults
  b. Local Faults
  c. Other Tools

3 Cryptanalysis methods
  a. Fault Model
  b. Safe Error Attack
  c. DFA
  d. Statistical Fault Attack

4 Countermeasures
  a. Analog Level
  b. Digital Level
  c. Application to Crypto

5 Conclusion

Classification of Fault Models

- One can define a Fault Model as a function f such that:

$$f : x \to x \star e \tag{1}$$

x target variable, e fault logical effect and $\star$ a logical operation

- Any Fault-based Cryptanalysis requires an Invariant $\Rightarrow$ new classification of FA based on the Invariant:
    - FA based on a Fixed Fault Diffusion Pattern
      Differential Fault Analysis [Biham+ 1997], [Piret+ 2003]
    - FA based on a Fixed Fault Logical Effect
      Safe Error Attacks [Biham+ 1997],
      Statistical Fault Attacks [Fuhr+ 2013]

Agenda

1 Introduction
- a. Physical Cryptanalysis
- b. Fault-based Cryptanalysis

2 Fault Injection Means
- a. Global Faults
- b. Local Faults
- c. Other Tools

3 Cryptanalysis methods
- a. Fault Model
- b. Safe Error Attack
- c. DFA
- d. Statistical Fault Attack

4 Countermeasures
- a. Analog Level
- b. Digital Level
- c. Application to Crypto

5 Conclusion

Safe Error Attack (SEA) [Biham+ 1997]

- SEA requires two copies of the target device:
  - a first copy that the adversary can fully control
  - a second copy set at an unknown secret
- SEA requires the ability to encrypt several times the same plaintext
- SEA does not require any faulty ciphertext
- SEA requires two phases:
  - a profiling phase
  - an attack phase

Safe Error Attack (SEA) - Sketch

1. Profiling phase
   - Use the device the adversary can fully control
   - For every bit of the master key, find the fault parameters allowing to reset this bit
2. Attack phase
   - Use the device set at an unknown secret
   - Encrypt a plaintext and keep the ciphertext
   - For every bit of the key, encrypt once again the same plaintext, while injecting a fault with parameters of profiling phase for the current bit
   - If both ciphertexts are equal, the current bit is equal to 0, otherwise equal to 1

Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Differential Fault Analysis (DFA) [Piret+ 2003]

- DFA requires the ability to encrypt two times the same plaintext

- DFA requires to have one or several pairs of correct and wrong ciphertexts corresponding to the same plaintext
  $P_1 \rightarrow (C_1, \widetilde{C_1})$
  $P_2 \rightarrow (C_2, \widetilde{C_2})$
  ...
  $P_N \rightarrow (C_N, \widetilde{C_N})$

- DFA requires to be able to fault only a part of the **State** at a particular position in the encryption
  e.g. one byte of the AES **State** before the last **MixColumns**

Differential Fault Analysis (DFA) - Sketch (1/2)

1. Assuming a one byte difference between the two States before the last MixColumns, compute the list D of the 16×255 possible differences after last MixColumns

2. Consider two pairs of correct and faulty ciphertexts $(C_1, \widetilde{C_1})$ and $(C_2, \widetilde{C_2})$

3. Make an hypothesis on the 2 left most bytes of K, Kh^1, Kh^2. For each of the 2^{16} candidates, compute:
   $$\delta_{C_1} = S^{-1}(C_1^1 \oplus Kh^1, C_1^2 \oplus Kh^2) \oplus S^{-1}(\widetilde{C_1^1} \oplus Kh^1, \widetilde{C_1^2} \oplus Kh^2)$$
   $$\delta_{C_2} = S^{-1}(C_2^1 \oplus Kh^1, C_2^2 \oplus Kh^2) \oplus S^{-1}(\widetilde{C_2^1} \oplus Kh^1, \widetilde{C_2^2} \oplus Kh^2)$$

Differential Fault Analysis (DFA) - Sketch (2/2)

4. Compare the results with the 2 left-most bytes of the differences in D. The (Kh^1, Kh^2) for which a match is found for both ciphertext pairs are stored in a list L

5. For each candidate of L, try to extend it by one byte (computing both differences to check)

6. Keep extending candidates in L until they are 16-bytes long. At this stage, only the right key is remaining

Agenda

1 Introduction
   a. Physical Cryptanalysis
   b. Fault-based Cryptanalysis

2 Fault Injection Means
   a. Global Faults
   b. Local Faults
   c. Other Tools

3 Cryptanalysis methods
   a. Fault Model
   b. Safe Error Attack
   c. DFA
   d. Statistical Fault Attack

4 Countermeasures
   a. Analog Level
   b. Digital Level
   c. Application to Crypto

5 Conclusion

Statistical Fault Attack (SFA) [Fuhr+ 2013]

- SFA has the property to work even with a set of faulty ciphertexts corresponding to different unknown plaintexts
  $P_1 \rightarrow \widetilde{C_1}$
  $P_2 \rightarrow \widetilde{C_2}$
  $\ldots$
  $P_N \rightarrow \widetilde{C_N}$

- Nevertheless it requires a Fixed Fault Logical Effect
  e.g. stuck-at a fixed value a **State** byte with a good probability

- SFA cannot be thwarted at the protocol level !!!

Statistical Fault Attack (SFA) - Sketch (1/2)

1. Collect a set of faulty AES ciphertexts $\widetilde{C_1}, \widetilde{C_2}, \ldots, \widetilde{C_N}$, by injecting a fault on one byte of the **State** after the penultimate **AddRoundKey**. We assume that the fault has a stuck-at effect to an unknown value e:
$\widetilde{S_{ak}^1} \;=\; S_{ak}^1 \;\; AND \;\; e, \qquad e \in [0, 255]$

2. A collection of correct ciphertext bytes $C_1, C_2, \ldots, C_N$ would have an uniform distribution
Here, due to the stuck-at fault, the collection of faulted ciphertext bytes $\widetilde{C_1}, \widetilde{C_2}, \ldots, \widetilde{C_N}$ has a biaised distribution

Statistical Fault Attack (SFA) - Sketch (2/2)

3. We can express $\tilde{S}ak_9^i$ as a function of $\tilde{C}^i$ and an hypothesis on one byte of K_{10}:
$\tilde{S}ak_9^i = SB^{-1} \circ SR^{-1}(\tilde{C}^i \oplus K_{10})$

4. Use a distinguisher to discriminate the correct key hypothesis. For instance, use the Minimal mean Hamming weight:
$h(\hat{K}) = \frac{1}{n} \sum_{i=1}^{n} HW(\hat{S}ak_r^i)$.

Agenda

1 Introduction
  a. Physical Cryptanalysis
  b. Fault-based Cryptanalysis

2 Fault Injection Means
  a. Global Faults
  b. Local Faults
  c. Other Tools

3 Cryptanalysis methods
  a. Fault Model
  b. Safe Error Attack
  c. DFA
  d. Statistical Fault Attack

4 Countermeasures
  a. Analog Level
  b. Digital Level
  c. Application to Crypto

5 Conclusion

(De)synchronization

- A fault injection requires a precise timing to be effective

- Adding temporal randomness makes the timing of the fault harder to set

- Classical ways to add temporal randomness:
  - jittered clock
  - dummy instructions
  - randomize operation flow
  - ...

IC Package as Countermeasure

- Several kind of fault injection techniques require to expose the die of the IC to perform the attack
  FBBI, laser, ...

- Depending on the type of package, it can be more or less easy to expose the die:
  - smartcard packages are easy to open
  - metallic packages can be mechanically opened
  - epoxy packages require a chemical attack
  - Package-on-Package or 3D IC technology make the chip opening a nightmare

IC Package as Countermeasure: example 1

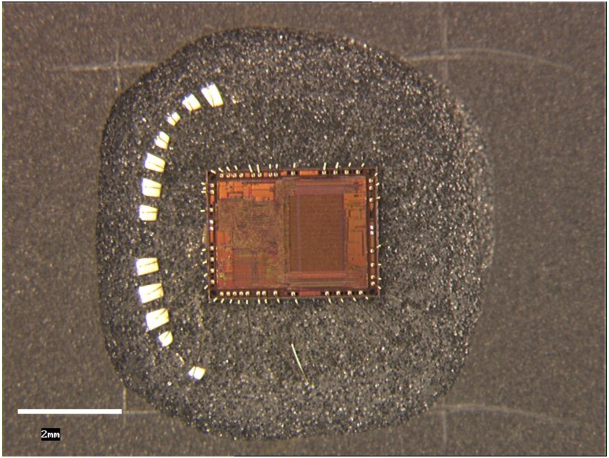


Figure : Epoxy package opened with fuming nitric acid

IC Package as Countermeasure: example 2

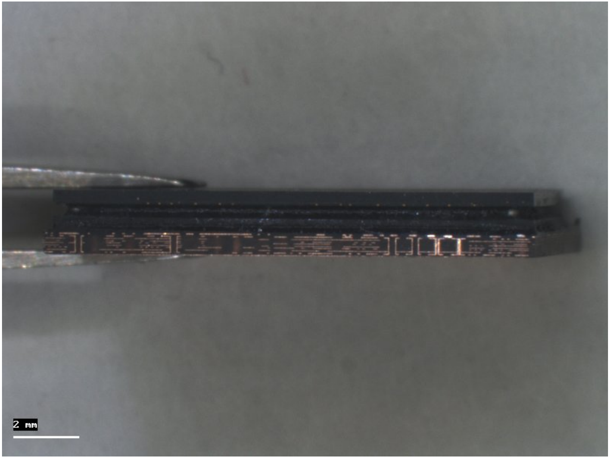


Figure : Application processor with RAM stacked above

IC Package as Countermeasure: example 2

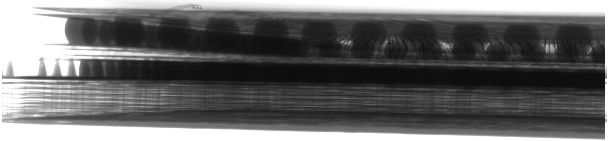


Figure : Application processor with RAM stacked above - X-ray view

Glitch Detectors

- The historical way to inject a fault in an IC is to under/over-power it during a short time

- Some IC manufacturers add glitch detectors after IC pads, checking that the current signal voltage stays in a defined range

- If a signal voltage goes outside from the defined range, a mechanism triggers an alarm
  e.g. flag set, interruption, reset, ...

Laser Detectors (1/2)

- Laser injection often requires to only disturb a small area of the IC

- It requires to perform a spatial cartography to find hot spots
  CPU/co-processor registers, memory cells or decoders, ...

- Laser detectors that are small dedicated blocks are placed among the other IC cells

Laser Detectors (2/2)

- Different kind of Laser detectors:
  - analog based laser detectors
    e.g. based on photodiodes
  - digital based laser detectors
    e.g. based on custom logic cells
- Laser detectors do not cover the whole suface of the IC, but make the job of the adversary harder

Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Redundancy

- Redundancy consists in:
  - performing two times an operation
  - comparing results of both operation executions
    ⇒ require a conditionnal test
- From a code theory point-of-view, it corresponds to the most obvious code one can construct
  ⇒ duplication code
- A variant consists in performing the operation and the inverse operation, then checking that the obtained result is equal to the initial data

Examples of Redundancy

- Redundancy can be used in different ways:
  - Sequential redundancy for a software function
  - Sequential or Parallel redundancy for a hardware function
  - Use of redundant logics (Dual Rail logic → SABL, WDDL, STTL, ...)
  - Securization of special registers by duplication or by storing a value and its inverse
    2 flip-flops are necessary to store one bit

Error Detection Codes

- Error Detection Codes are efficient tools to check the integrity of data

- ECC can protect linear operations (they are based on linear applications)

- ECC cannot protect non-linear operations
  in particular they are not well suited to protect cryptographic primitives

Examples of Error Detection Codes

- Error Correcting Codes can be used in different ways:
  - Ensure the integrity of a secret data stored in NVM
  - Protect a memory decoder
    → ensure the integrity of opcodes
  - Protect linear parts of cryptographic algorithms
  - ...

Infection

- Infection consists in mixing a diffusion scheme with the operation to protect such that:

    1. if the processed data are not modified by a fault, the diffusion scheme has no effect on the final result
    2. if the processed data are modified by a fault, the diffusion scheme expands the erroenous data such that the final result is no more exploitable by the adversary

Memory Protection Unit (MPU)

- Some microcontrollers have a Memory Protection Unit
  can be seen as a HW co-processor

- MPU works similarly to a MMU (Memory Management Unit):
  - For a given function to protect, the progammer defines a memory address range
  - The MPU ensures that the instructions of the function will be located in the defined memory address range
  - If a fault induces a code jump outside the defined memory address range, the MPU triggers an alarm

Code Signature

- Some microcontrollers have a Code Signature feature
  can be seen as a HW co-processor

- Code Signature works as follows:
  - For a given function to protect, the progammer computes a digest and stores it in NVM
  - Every time the function is executed, the code signature feature computes the current digest and compares it to the reference one
  - If they are different, an alarm is triggered

Agenda

1 Introduction
   a. Physical Cryptanalysis
   b. Fault-based Cryptanalysis
2 Fault Injection Means
   a. Global Faults
   b. Local Faults
   c. Other Tools
3 Cryptanalysis methods
   a. Fault Model
   b. Safe Error Attack
   c. DFA
   d. Statistical Fault Attack
4 Countermeasures
   a. Analog Level
   b. Digital Level
   c. Application to Crypto
5 Conclusion

Classical Detection Schemes For Block Ciphers

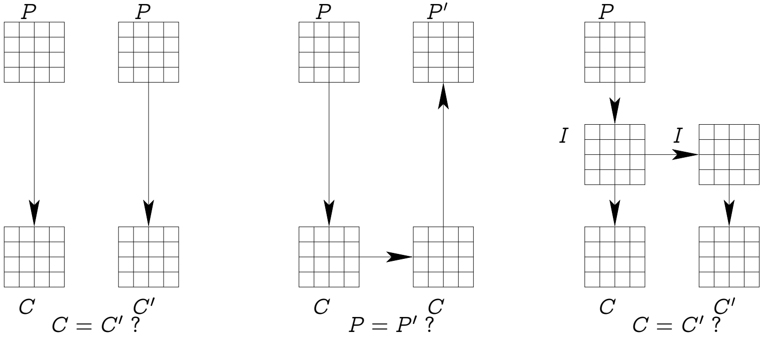
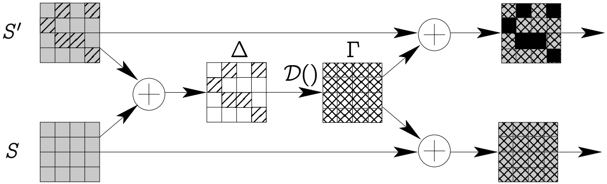


Figure : Three classical detection countermeasures. From left to right : Full Duplication, Encrypt/Decrypt, and Partial Duplication

Classical Infection Schemes For Block Ciphers

- Generic sketch exhibiting the Infection CM:
  - S, S' the two States
  - $\mathcal{D}$ the diffusion function (such as $\mathcal{D}(0) = 0$)



Agenda

1 Introduction

a. Physical Cryptanalysis

b. Fault-based Cryptanalysis

2 Fault Injection Means

a. Global Faults

b. Local Faults

c. Other Tools

3 Cryptanalysis methods

a. Fault Model

b. Safe Error Attack

c. DFA

d. Statistical Fault Attack

4 Countermeasures

a. Analog Level

b. Digital Level

c. Application to Crypto

5 Conclusion

Conclusion (1/2)

- Fault Attacks are a very powerful attack path:
  - they allow to modify the normal behaviour of a HW or SW function
  - they allow to extract cryptographic secrets
- Nevertheless FA require several skills:
  - knowledge of computer science, electronics, optics, ...
  - knowledge of IC architecture
  - knowledge of fault-based cryptanalysis

Conclusion (2/2)

- A lot of Fault Attack Countermeasures have been proposed in the litterature
- They are generally mixed to increase the security level of the product
  ⇒ principle of defense in depth
- No countermeasure is perfect !
- A developper has firstly to define the level of the adversary he wants to thwart, and then choose the adequate tradeoff between efficiency and security

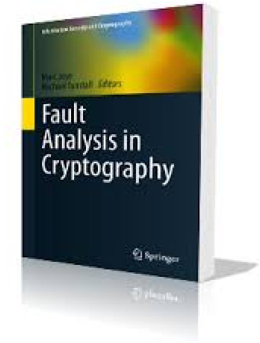
Certification Schemes

- Procedure to evaluate the security level of a product

- Three actors:
  the developper / the security lab / the scheme

- Some certification schemes:
  - Common Critera
  - EMVCo
  - ...

To go further



- book Fault Analysis in Cryptography
  Marc Joye and Michael Tunstall - SPRINGER

Questions ?

- contact: victor.lomne@ssi.gouv.fr

Bonus 1: Bug Attack

- Pentium FDIV bug was a bug in the Intel $P5$ Pentium floating point unit (FPU)
- Because of the bug, the processor would return incorrect results for many calculations
- Nevertheless, bug is hard to detect

  1 in 9 billion floating point divides with random parameters would produce inaccurate results
- Shamir proposed a modified version of the Bellcore attack which exploits this bug to retrieve a RSA private key
- More dangerous than a classical fault attack because can be perfomed remotely

Bonus 2: PS3 Hack

- George Hotz (a.k.a. Geohot) published in 2009 a hack of the Sony PS3
- The otherOS functionnality of the PS3 allowed to boot a Linux OS
- A bus glitch allowed him to gain control of the hypervisor
  ⇒ ring 0 access
  ⇒ full memory access
  ⇒ control gain of the OS bootchain
- In consequence Sony took George Hotz to court
- Sony and Hotz had settled the lawsuit out of court, on the condition that Hotz would never again resume any hacking work on Sony products