Experimental Fault Injection based on the Prototyping of an AES Cryptosystem

Jean-Baptiste Rigaud
Jean-Max Dutertre
Michel Agoyan
Bruno Robisson
Assia Tria

Département SAS
Équipe mixte CEA-LETI/ENSME
Site Georges Charpak
Centre Microélectronique de Provence
880, route de Mimet
13541 Gardanne
Introduction.
  - Course Overview

Attacked Circuit: AES Cryptosystem.
  - Algorithm and design
  - Prototyping of AES on SPARTAN-3

Design and use of an FPGA-based attack platform.
  - Fault Injection Principle (digital IC timing constraints)
  - Experimental Results

Conclusion.
Master students in Microelectronics Design

- Cryptography
- Secured Circuits

Application of academics courses

- VHDL
- Design Methodology
- FPGA Prototyping

Two parts

- 128 bit AES design : Spartan 3
- Injection fault platform : Virtex 5
Why Cryptography?

- Confidentiality
- Authentication
- Integrity
- Non repudiation

Tools for cryptography

- Secret key scheme
- Public/private keys scheme
- Hash function

Applications

- Credit card, mobile phone, pay TV, secured internet, etc.
Advanced Encryption Standard

- NIST 2001
- Key length: 128 bits

A good example for teaching IC design

- Data path and key expander synchronization
- Sbox modeling

© Enrique Zabala - Universidad ORT/Montevideo/Uruguay
Timing constraints

- Nominal clock frequency: 100 MHz
- 11 clock cycles / ciphering

Area

- 20 S-boxes

External control

- Clock pin
- Start signal
Xilinx Spartan 3 evaluation board

- Serial Link
- Simple control commands
- Automatic Test Generation (Perl)
- On the fly comparison of expected result (Open ssl' AES library)

Why FPGA target?
- Education purposes
- Faster integration
- Easier fault injection due to long interconnection delays
Design and Use of an FPGA-based Attack Platform

- **Theoretical work.**
  - Short overview of Differential Fault Attacks.
  - Digital IC timing constraints (as a fault injection means).
  - A Delay Locked Loop based attack platform.

- **Laboratory work.**
  - Synthesis of the attack platform.
  - Experimental results.
Disturb the ciphering process through unusual environmental conditions.

Differential Fault Attack = comparison between correct and faulty cipher texts

→ retrieve information on the encryption process (i.e. information leakage)

Strong requirements:

• control of the fault size (bit or byte level),
• target a given round (and only it).
**Synchronous IC principle (reminder)**

Data are captured on the clock’s rising edge.

Time between two rising edges (i.e. clock period) depends on the propagation delay.
Digital IC timing constraint

Violating this timing constraint results in fault injection.
Fault location - Propagation delay

outputs = $f$ (inputs)

f logical function
each $D_i$ had its own propagation delay

Fault location :
where $delay_{D_i} > T_{clk} - setup$ time

Propagation times depend on :

• the logical states (0 / 1)
  → the propagation delay changes with the inputs

• the power supply voltage

• the temperature

allow to change the fault location
Fault injection - Over clocking

A well known approach decreasing the clock period until faults appear by setup time violation

\[ T_{\text{clk}} \]

\[ T_{\text{clk fault}} \]

propagation delay + setup time

drawback: faults are injected at each clock cycle
no timing control
• Fault injection - Local over clocking

Setup time violation by modifying one clock cycle

\[ T_{\text{clk}} \]

\[ D_{p_{\text{Max}}} + \delta_{su} \]

\[ T_{\text{clk}} - \Delta \]

\[ \delta_t \text{ variation step} = 35 \text{ ps} \]

Experiment \( T_{\text{clk}} = 10 \text{ ns} \)

\( \approx 300 \text{ steps} @ 100 \text{ MHz} \)
Fault injection - Local over clocking (cont’d)

clk’ generation: use of an on chip Delay Locked Loop (Xilinx Virtex-5).
- Fault injection - Local over clocking (cont’d)

clk’ generation: use of an on-chip Delay Locked Loop (Xilinx Virtex-5).
Fault injection - Local over clocking (cont’d)

clk’ generation: use of an on chip Delay Locked Loop (Xilinx Virtex-5).

All digital, easy to implement.
Synthesis of the attack platform

\[ \Delta = 0 \]
Synthesis of the attack platform

\[ \Delta = 20 \times 35 \text{ ps} \]
Synthesis of the attack platform

\[ \Delta = 40 \times 35 \text{ ps} \]
Synthesis of the attack platform

\[ \Delta = 60 \times 35 \text{ ps} \]
Synthesis of the attack platform

\[ \Delta = 80 \times 35 \text{ ps} \]
Δ = 100 × 35 ps

Synthesis of the attack platform
Fault injection experiments

- Experimental setup

![Experiment setup diagram](image)
Controllability of faults’ nature and location.

Targeting the final round of the AES

\[ \Delta \leftarrow 0 \]

\[ \text{direct reading of the injected faults (by XORing a correct and faulty ciphertext)} \]

Test campaign pseudo-code:

```
send the key $K$ and the plaintext $T$ to the test chip
\[ \Delta \leftarrow 0 \]

\textbf{while} (clock\_period > $\Delta$) \textbf{do}

encrypt and retrieve the ciphertext

\[ \Delta \leftarrow \Delta + \delta_t \]

\textbf{end while}
```

Note that faults are located in the encryption data path (longest propagation delay).
Experimental results

Target: final round ($f_{clk, nom} = 100$ MHz)

Step by step $T_{clk}$ decrease ($\delta_t = 35$ ps)
**Location control: plaintext variation**

- **Byte nb. 13**
  - No fault
  - One-bit fault
  - Two-bits fault
  - Other fault

- **Byte nb. 3**
  - No fault
  - One-bit fault
  - Two-bits fault
  - Other fault

*Same key*

*Different plaintext*
- Fault injection based on power supply decrease.
  (at nominal frequency)

\[
V_{DD} \downarrow \quad \Rightarrow \quad D_{pMax} \quad \uparrow \quad (D_{clk \rightarrow Q}, \delta_{su}, |T_{skew}| \quad \uparrow)
\]

\[
T_{clk} < D_{clk \rightarrow Q} + D_{pMax} - T_{skew} + \delta_{su}
\]
Fault injection based on power supply decrease.

Critical time as a function of $V_{DD}$

1st fault at 1.07 V
Temperature increase (at nominal frequency)
Temperature increase (at nominal frequency)

\[ D_{p_{\text{Max}}} \uparrow \ (D_{\text{clk}\rightarrow Q}, \delta_{su}, |T_{\text{skew}}| \uparrow ) \]

Experimental results

1st fault at 210 °C
Conclusion

An ambitious two in one course (Master or PhD students).

Achievements:

- Design methodology on a concrete programmable device,
- Development of a complete test environment (serial interface, command scripts),
- Implementation of the AES standard,
- Review of timing constraints and critical path issues,
- Design of a DLL-based attack platform,
- Practice of fault attacks,
- Awareness of hardware security.

FPGA : a well suited target.