Fault Injection Resilience

Sylvain GUILLEY, Laurent SAUVAGE, Jean-Luc DANGER, Nidhal SELMANE.

Institut TELECOM / TELECOM-ParisTech
CNRS – LTCI (UMR 5141)

FDTC (Santa Barbara, CA, USA), Saturday August 21st, 2010.
Presentation Outline

1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
Fault Attacks in Cryptography

Harmfulness

- 1 fault suffices to break unprotected CRT-RSA
- 1 fault suffices to break unprotected AES-128

Fault Injection Techniques

- **Global faults:**
  - are low cost,
  - often asymmetrical,
  - can break real-world implementations.

- **Local faults:**
  - require sample preparation,
  - can be symmetrical (arbitrary),
  - can break real-world implementations.
Observation attacks are easily thwarted by masking:
\[ \forall r_1, r_2, \left( M^{d+r_1} \times \phi(N) \mod r_2 \times N \right) \mod N = M^d \mod N, \]
hence multiple degrees of freedom to mask cryptographic parameters.

Perturbation attacks are fought thanks to similar properties:
- Randomness can also be injected within the algorithm, so as to enable verifications afterwards [BHT09].

This paper by Jean-Sébastien CORON (@ AsiaCrypt 2009) [CM09] proves that RSA with PSS is provably secure against random fault injection attacks in the random oracle model.
Algorithm 1: RSA implementation protected against SCA and FIA.

Input : $x \in \mathbb{G}$, $d = (d_{n-1}, \ldots, d_0)_{\mathbb{Z}}$
Output: $x^d \in \mathbb{G}$ or “Error”

1. Generate a random $r \in \mathbb{G}^*$
2. $R[0] \leftarrow r$
3. $R[1] \leftarrow r^{-1}$
4. $R[2] \leftarrow x$
5. for $i \in [0, n - 1]$ do
6. \hspace{1em} $R[1 - d_i] \leftarrow R[1 - d_i] \cdot R[2]$
8. end
9. if $R[0] \cdot R[1] \cdot x = R[2]$ then
10. \hspace{1em} return $r^{-1} \cdot R[0]$
11. else
12. \hspace{1em} return “Error”
13. end
All this is very challenged by recent attacks:

- Attack of Berzati et al. at CHES’2010 [BCDG10]
- Attack on Vigilant’s CT-RSA of CHES’2008 to come...
In symmetric cryptography:

Detection seems to be the only research effort. It involves redundancy, such as:

- Space
- Time
- Information
- Algorithm

Objective of this talk:

We tackle other ways to protect symmetric cryptography against fault injection attacks.
Presentation Outline

1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
Reminder about the characteristics of detection

- **Undetected faults**: fatal... and inexistent in resilience
- **Unnecessary detections**: inconvenience (*that does not exist in side-channel resilience*)
Virtues of Resilience also against Perturbation Attacks

Detection scheme:

- Stress: no stress → heavy stress → no stress
- Detection: nominal → alert → nominal
- Device's state: functional → non-functional (locked state)

Resilience scheme:

- Stress: no stress → heavy stress → no stress
- Results: correct → incorrect → correct
- Device's state: functional → non-functional → functional
Presentation Outline

1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
Algorithm 2: Probabilistic Encryption Algorithm built on top of AES, non-protected against FIAs.

**Input**: A plaintext $x$ to be encrypted with the key $k$.

**Output**: A ciphertext along with a random number.

1. Determine a random number $r$ of the same size as $x$; /* This number will whiten $x$ */.
2. Return the couple $(y = \text{AES}_k(x \oplus r), r)$.

Algorithm 3: Deterministic Decryption Algorithm matching algorithm (2).

**Input**: A ciphertext under the form $(y = \text{AES}_k(x \oplus r), r)$ to be decrypted by the AES key $k$.

**Output**: The plaintext $x$.

1. Decrypt $y$ with key $k$:
   $z = \text{AES}_{k}^{-1}(y)$.
2. Return the demasked input:
   $z \oplus r = x$. 

S. Guilley, < sylvain.guilley@TELECOM-ParisTech.fr >
Suggestion of resolution for the asymmetry encryption/decryption

Easy to protect: $\Rightarrow$ Algorithm (3)

Difficult to protect: $\Rightarrow$ Algorithm (2)

Deterministic decryption, in a tamper-proof and tamper-evident reader

Probabilistic encryption, with blinding at the input & at the output
Cryptography is the most demanding resource

Susceptible organs of a smartcard in two representative sensitive operations (EXTERNAL and INTERNAL AUTHENTICATE). Typically, the cryptography will be either **RSA** or **3DES**.
Blinding inputs ... and outputs!

The initialization vector (IV) is:

- a random number for the input, and
- a secretly exchanged nonce for the output.

\[ \text{MGF} = \text{Mask Generation Function}. \]
Presentation Outline

1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
When we cannot trust the external TRNG

<table>
<thead>
<tr>
<th>Multi-valued logic</th>
<th>Redundant logic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case #1</strong></td>
<td><strong>Case #1</strong></td>
</tr>
<tr>
<td>![1/2, 1, 1/2, 0]</td>
<td>![0, 0, 0, 0]</td>
</tr>
<tr>
<td><strong>Case #2</strong></td>
<td><strong>Case #2</strong></td>
</tr>
<tr>
<td>![1, * 1, 1/2, 0]</td>
<td>![11, 00, 10, 10]</td>
</tr>
</tbody>
</table>

Two kinds of faults (in red), namely \( \{0, 1\} \rightarrow 1/2 \) for 3-valued logic and \( \{01, 10\} \rightarrow \{00, 11\} \), i.e. \{VALID0, VALID1\} for DPL, after which the initial value (in green) has been forgotten.

S. Guilley, <sylvain.guilley@TELECOM-ParisTech.fr>
Vocabulary

- **DPL**: Dual-rail with Precharge Logic
- **EPE**: Early Propagation (in evaluation or in precharge) Effect

- **DPL w/ EPE**: \( \exists a \ \text{VALID}, f(a, \text{NULL}) = \text{VALID}; \)
- **DPL w/o EPE**: \( \forall a \ \text{VALID}, f(a, \text{NULL}) = \text{NULL}. \)
**DPL w/ EPE** is Protected against Multiple Asymmetrical Faults [SBG⁺09]

<table>
<thead>
<tr>
<th></th>
<th>VALID0</th>
<th>VALID1</th>
<th>NULL0</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALID0</td>
<td>VALID0</td>
<td>VALID0</td>
<td>VALID0 (EPE)</td>
</tr>
<tr>
<td>VALID1</td>
<td>VALID0</td>
<td>VALID1</td>
<td>NULL0</td>
</tr>
<tr>
<td>NULL0</td>
<td>VALID0 (EPE)</td>
<td>NULL0</td>
<td>NULL0</td>
</tr>
</tbody>
</table>


where the tokens \{VALID0, VALID1, NULL0\} implement respectively the items \{'0', '1', 'U'\}. 

**S. Guilley, < sylvain.guilley@TELECOM-ParisTech.fr >**
DPL w/o EPE is Protected in front of Multiple Symmetric Faults [BDF+09]

<table>
<thead>
<tr>
<th>(b)</th>
<th>(a)</th>
<th>VALID0</th>
<th>VALID1</th>
<th>NULL0</th>
<th>NULL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALID0</td>
<td>VALID0</td>
<td>VALID0</td>
<td>NULL0</td>
<td>NULL1</td>
<td></td>
</tr>
<tr>
<td>VALID1</td>
<td>VALID0</td>
<td>VALID1</td>
<td>NULL0</td>
<td>NULL1</td>
<td></td>
</tr>
<tr>
<td>NULL0</td>
<td>NULL0</td>
<td>NULL0</td>
<td>NULL0</td>
<td>NULL1</td>
<td></td>
</tr>
<tr>
<td>NULL1</td>
<td>NULL1</td>
<td>NULL1</td>
<td>NULL0</td>
<td>NULL1</td>
<td></td>
</tr>
</tbody>
</table>

Remark that if we call: ‘0’: VALID0, ’1’: VALID1, ’X’: NULL = \{NULL0, NULL1\}, then we have the same behavior (i.e. “propagate always”) as VHDL. This is illustrated below:

<table>
<thead>
<tr>
<th>(b)</th>
<th>(a)</th>
<th>’0’</th>
<th>’1’</th>
<th>’X’</th>
</tr>
</thead>
<tbody>
<tr>
<td>’0’</td>
<td>’0’</td>
<td>’0’</td>
<td>’X’</td>
<td></td>
</tr>
<tr>
<td>’1’</td>
<td>’0’</td>
<td>’1’</td>
<td>’X’</td>
<td></td>
</tr>
<tr>
<td>’X’</td>
<td>’X’</td>
<td>’X’</td>
<td>’X’</td>
<td></td>
</tr>
</tbody>
</table>
Against Asymmetric Faults

Against Symmetric Faults

Combinatorial block (e.g. one sbox, such as AES SubBytes) implemented in DPL w/o EPE style

Multiple faults, where the false valid is not completely hidden by the ’X’ wave. The ’X’ avalanche absorbs most, if not all, the valid faults.
Performance overhead of different SCA+FIA countermeasures.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Detection + DPL</th>
<th>Resilience = DPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Countermeasure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[KKT04] + [THH+05]</td>
<td>DRSL [CZ06]</td>
<td>IWDDL [MMMT09]</td>
</tr>
<tr>
<td>Area</td>
<td>5.49 ×</td>
<td>2.56 ×</td>
</tr>
<tr>
<td>Throughput</td>
<td>4.49 ×</td>
<td>2.00 ×</td>
</tr>
</tbody>
</table>

S. Guilley, <sylvain.guilley@TELECOM-ParisTech.fr>
Presentation Outline

1. Context
   - Fault Injection Attacks
   - State-of-the-Art in Protections for Asymmetric Cryptography
   - State-of-the-Art in Protections for Symmetric Cryptography

2. Detection versus Fault Injection Resilience (FIR)
   - Features of Detection
   - Features of FIR

3. Case-Study #1: Protocol-Level Resilience
   - Against Faults
   - Against Leakage

4. Case-Study #2: Netlist-Level Resilience
   - Against Asymmetric Faults
   - Against Symmetric Faults

5. Conclusions
Conclusions

- Asymmetric crypto *seems* easier to protect than symmetric crypto.
- We demonstrate both a **protocol-level** and an **implementation-level** fault injection resilience (FIR) scheme.
- Those techniques combine nicely with leakage resistance techniques.

- **FIPS-140** requires detection schemes...
- whereas **CC** are open to any kind of countermeasures.


WDDL is Protected Against Setup Time Violation Attacks.


In conjunction with CHES’09, Lausanne, Switzerland. DOI: 10.1109/FDTC.2009.40; Online version: http://hal.archives-ouvertes.fr/hal-00410135/en/.

[THH⁺05] Kris Tiri, David Hwang, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede.

A side-channel leakage free coprocessor IC in 0.18 μm CMOS for Embedded AES-based Cryptographic and Biometric Processing.


San Diego, CA, USA.