Fault Sensitivity Analysis Against Elliptic Curve Cryptosystems

The University of Electro-Communications:
Hikaru Sakamoto, Yang Li, Kazuo Ohta, and Kazuo Sakiyama

FDTC2011  Nara Japan  2011/09/28
Contents

- Introduction
  - Fault Sensitivity Analysis
  - Fault injection technique
  - Montgomery Powering Ladder
- Proposed attack
- Experiments and results
- Difference between FSA and DPA
- Conclusion and future work
Introduction

Propose attack using Fault Sensitivity Analysis (FSA) against public key (PK) implementation

<table>
<thead>
<tr>
<th></th>
<th>Previous FA</th>
<th>FSA</th>
<th>In Previous FA, use the value of the faulty output</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>✔️ [BA97]</td>
<td>✔️ [LSG+10]</td>
<td>In FSA, do not use the value of the faulty output</td>
</tr>
<tr>
<td>PK (ECC)</td>
<td>✔️ [BMM00]</td>
<td>New</td>
<td></td>
</tr>
</tbody>
</table>

Contribution

- Successful attack against PK using FSA for the first time
- In case study, we attack against ECC in LSI on SASEBO-R
Fault Sensitivity Analysis (FSA)

Fault injection (Fault intensity is $F$)

Input $\rightarrow$ Device $\rightarrow$ Correct output

- $F$ is low
- $F$ is high

The borderline of $F$ which can induce fault (Fault sensitivity (FS) information)

Identify the secret key

Depends on input value

Input $\rightarrow$ Device $\rightarrow$ Faulty output
Fault injection technique

By supplying an illegal clock, the setup time violation is induced to devices.

Illegal clock

Period: \( T \) \( \rightarrow \) \( T' \) (\(< T\))

Clock frequency is high \( \rightarrow \) Fault intensity \((F)\) is high

Clock frequency is low \( \rightarrow \) Fault intensity \((F)\) is low
Montgomery Powering Ladder (MPL)

- MPL is one of the scalar multiplication algorithm
- Point addition and doubling are performed in calculating 1 bit
  - Dummy operations do not exist in MPL

(Ex) Input: \( P, d = 19 = (10011)_2 \)
Output: \( Q(=19P) \)

\[
\begin{array}{cccccc}
  d_i & 1 & 0 & 0 & 1 & 1 \\
  P_1 & P & 2P & 4P & 9P & 19P \\
  \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
  P_2 & 2P & 3P & 5P & 10P & 20P \\
\end{array}
\]

Result of MPL

Initial doubling  Point doubling  Point addition
Main idea of our attack

In FSA,

FS is specific information on input of calculations

Input: \( P \) → calculation → \( FS_P \)

Input: \( 2P \) → calculation → \( FS_{2P} \)

Input: \( a \) → calculation → \( FS_a \)

Input: \( b \) → calculation → \( FS_b \)
Main idea of our attack (cont.)

In point doubling of MPL,

\[ d_i = 0 : \quad 2 \times P_1 \]
\[ d_i = 1 : \quad 2 \times P_2 \]
Template and Attack procedure

\[
\begin{array}{c|cc}
\mathbf{d}_i & 1 & 0 \\
\mathbf{P}_1 & \mathbf{P} & 2\mathbf{P} \\
\mathbf{P}_2 & 2\mathbf{P} & 3\mathbf{P}
\end{array}
\]

Initial Doubling: \( 2 \times \mathbf{P} = 2\mathbf{P} \)

Template

Point doubling performed for the first time (Initial doubling)

\[ \mathbf{d}_i \] is a template to detect the secret key.

\[ \mathbf{P}_1, \mathbf{P}_2 \] are points used in the attack.

\[ \mathbf{P} \] is the point doubling target.

\[ 2\mathbf{P} \] is the result of point doubling.

\[ 3\mathbf{P} \] is the result of point doubling the previous result.

\[ 0 \] indicates no point doubling was performed for this entry.

\[ 1 \] indicates point doubling was performed for this entry.

**Attack procedure**

- Make template
- Measure attack target of point doubling
- Calculate correlation of the point doubling and the template

A key corresponding to template where correlation is larger is correct secret key.
How to make template

ex) Template $2P \rightarrow 4P$

1. Input $2P$ to device
   - Input: $2P \rightarrow$ Device

2. The device performs initial doubling
   - $P_1 \quad P \quad P_2 \rightarrow 2P$

3. Measure fault sensitivity (FS) information
   - Measure $\rightarrow$ FS information
How to identify the key bit (2\textsuperscript{nd} MSB)

(1) Guess the value of $d_2$

\begin{align*}
\text{① } d_2 &= 0 : \quad P \rightarrow 2P \\
\text{② } d_2 &= 1 : \quad 2P \rightarrow 4P
\end{align*}

(2) Make templates

- Initial doubling
- Measure

(3) Measure performed point doubling

\begin{align*}
&\begin{array}{c|c}
    d_i & 1 \quad d_2 \\
    P_1 & P \\
    P_2 & 2P \\
\end{array} \\
&M\text{easure}
\end{align*}

(4) Identify the value of $d_2$

\begin{align*}
&M\text{easurement data} = Template \ ① \\
&\begin{array}{c}
    d_2 \\
\end{array} = 0 \\
&M\text{easurement data} = Template \ ② \\
&\begin{array}{c}
    d_2 \\
\end{array} = 1
\end{align*}
How to identify the key bit (3<sup>rd</sup> MSB)

(1) Guess the value of $d_3$

① $d_3 = 0 : 2P \rightarrow 4P$
② $d_3 = 1 : 3P \rightarrow 6P$

(2) Make templates

(3) Measure performed point doubling

<table>
<thead>
<tr>
<th>$d_i$</th>
<th>$P$</th>
<th>$2P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_2$</td>
<td>$2P$</td>
<td>$3P$</td>
</tr>
</tbody>
</table>

Measure

(4) Identify the value of $d_3$

Measurement data $=$ Template ①

$d_3 = 0$

Measurement data $=$ Template ②

$d_3 = 1$
Case study: Attack for ECC implementation in Cryptographic LSI on SASEBO-R

- Using elliptic curve over extended binary field
- Using López-Dahab algorithm [LD99] as scalar multiplication algorithm

*SASEBO: Side channel Attack Standard Evaluation BOard*
López-Dahab algorithm [LD99]

Point addition and doubling using
X and Z coordinates as projective coordinates

Point doubling by
López-Dahab algorithm

Input: \( P_1 = (X_1, Z_1) \).
Output: \( P_1 = 2P_1 \).

1: \( t_1 = X_1 X_1 \)
2: \( t_2 = Z_1 Z_1 \)
3: \( Z_1 = t_1 t_2 \)
4: \( t_1 = t_1 t_1 \)
5: \( t_2 = t_2 t_2 \)
6: \( t_3 = bt_2 \)
7: \( X_1 = t_3 + t_1 \)
8: return \( P_1 \)

Measure these steps in the attack

It is difficult to induce a fault in modular addition over \( \text{GF}(2^m) \)
for (fault injection position) from (step 1) to (step 6) do
  repeat
    while correct results are generated do
      increase the clock frequency;
    end while
    record the clock frequency;
  until several times
  calculate average of these recorded clock frequencies
end for

Decrease measurement noise
Experimental results (2^{nd} MSB)

$P_1 \quad 1 \quad 0 \text{ or } 1 \quad 2P \quad (2 \times P = 2P)$

Attacker can identify the secret key

Fault injection step

Template with correct guess
Correlation coefficient: 0.9392

Template with wrong guess
Correlation coefficient: 0.3083

FDTC2011  Nara Japan  2011/9/28
Experimental results (3rd MSB)

\[ d_i \begin{array}{c} 1 \ 0 \ 0 \quad 0 \quad \text{or} \quad 1 \\ P_1 \begin{array}{c} P \ 2P \ ightarrow \ 4P \\ P_2 \begin{array}{c} 2P \ 3P \ ightarrow \ 6P \end{array} \end{array} \end{array} \]

**Point doubling of attack target**

\( 2 \times 2P = 4P \)

**Template with correct guess**

Correlation coefficient: 0.9016

**Template with wrong guess**

Correlation coefficient: 0.3085

By repeating this procedure, the attacker can identify all the key bits.
Attack condition

The attacker must be able to
• Make any templates using initial doubling
  • Input the initial point from the outside
• Guess performed point doubling correctly

Our attack cannot work on the implementation with
• randomized input point
• randomized the secret key
Difference between FSA and DPA

FSA is a new side-channel attack using FS information
We use the FS as the side-channel leakage to identify the secret key

We expect lower measurement noise for the FS-based attack than power-based one
Conclusion and Future work

- **Conclusion**
  - Successful attack for a public key implementation using FSA for the first time
    - Make templates to distinguish point doubling using initial doubling
  - As a case study, we success to attack for ECC implementation in LSI on SASEBO-R

- **Future work**
  - We will study
    - possible attacks on an implementation with randomized input point or secret key
    - further differences between FSA and DPA
Thank you for your attention
References


