Improved Fault Analysis on SIMON Block Cipher Family

Hua Chen  Jingyi Feng  Vincent Rijmen  Yunwen Liu
Limin Fan  Wei Li

Presenter: Svetla Nikova

ESAT/COSIC, KU Leuven, and iMinds, Belgium

FDTC 2016, 16 Aug, 2016
Motivation

- **Simon** is a lightweight block cipher family proposed in 2013.
- It employs a Feistel-type structure with $2n$-bit block size and $mn$-bit key size.
## Motivations

Parameter list for the instances of Simon family

<table>
<thead>
<tr>
<th>block size $2n$</th>
<th>key size $mn$</th>
<th>word size $n$</th>
<th>key words $m$</th>
<th>rounds $T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>64</td>
<td>16</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>48</td>
<td>72</td>
<td>24</td>
<td>3</td>
<td>36</td>
</tr>
<tr>
<td>48</td>
<td>96</td>
<td>24</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>64</td>
<td>96</td>
<td>32</td>
<td>3</td>
<td>42</td>
</tr>
<tr>
<td>64</td>
<td>128</td>
<td>32</td>
<td>4</td>
<td>44</td>
</tr>
<tr>
<td>96</td>
<td>96</td>
<td>48</td>
<td>2</td>
<td>52</td>
</tr>
<tr>
<td>96</td>
<td>144</td>
<td>48</td>
<td>3</td>
<td>54</td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>64</td>
<td>2</td>
<td>68</td>
</tr>
<tr>
<td>128</td>
<td>192</td>
<td>64</td>
<td>3</td>
<td>69</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>64</td>
<td>4</td>
<td>72</td>
</tr>
</tbody>
</table>
Motivation

- Since **Simon** is presented, its implementation security has also caught attention, such as Fault Attack.
- In FDTC 2014, the first Fault Attack against **Simon** was presented.
  - Byte and bit injection fault model are both adopted.
  - For the keysize $mn$, the input of $T$-2-th, $T$-3-th, $T$-4-th, ..., $T$-m-1-th round is required to be injected faults respectively.
  - The average number of faults for the byte and bit injection model is respectively $mn/8$ or $mn/2$ if the injection position can be controlled.
  - When the injection position can be selected randomly, the theoretical estimation of injection numbers was not given.
Motivation

- In ICISC 2014, the second Fault Attack against Simon was presented.
  - Instead of byte or bit fault model, \( n \)-bit fault model is adopted. (Each bit of a \( n \)-bit word is flipped with the probability 0.5)
  - For the keysize \( mn \), the input of \( T-2 \)-th, \( T-3 \)-th, \( T-4 \)-th, \( \ldots \), \( T-m-1 \)-th round is still required to be injected faults respectively.
  - A theoretical estimation of average injection numbers was given.
Motivation

- In FDTC 2015, the third Fault Attack against Simon was proposed.
  - Bit fault model is adopted.
  - For the keysize $mn$, the first injected round is $T-3$-th round instead of $T-2$-th round and the total number of injected rounds is reduced half.
  - A theoretical estimation of average injection numbers was given.
Motivation

Related work of fault attacks on **Simon**:

<table>
<thead>
<tr>
<th>Related work</th>
<th>Fault model</th>
<th>Number of injected rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDTC 2014</td>
<td>Random byte/bit model</td>
<td>$m$</td>
</tr>
<tr>
<td>ICISC 2014</td>
<td>Random $n$-bit model</td>
<td>$m$</td>
</tr>
<tr>
<td>FDTC 2015</td>
<td>Random bit model</td>
<td>$\lceil m/2 \rceil$</td>
</tr>
</tbody>
</table>

**Our goal:**

- Number of injected rounds: 1
- Reduce the injection numbers
- Give the theoretical estimation of injection numbers under random byte fault model, which is not given in former work.
Some properties of Simon

Property 1 Given a \( t(1 \leq t \leq n) \)-bit difference \( e = e_0e_1e_2, \ldots e_{t-1} \), if it is induced into \( L^0 \) from the \((s-t+1)\)-th to the \( s \)-th bit position \((0 \leq s \leq n-1)\), (that is, \( \Delta L^0_{s-t+1} \Delta L^0_{s-t+2}, \ldots, \Delta L^0_s = e \)), then for \( 1 \leq j \leq T/2 \), after the encryption of \( r \) rounds, \( \Delta L^r \) satisfies:

When \( r = 2j-1 \),

\[
\Delta L^r_{i} = 0, \quad s \leq i \leq s + (n - t - 16j + 8) \tag{1}
\]

When \( r = 2j \),

\[
\begin{cases}
\Delta L^r_{i} = 0, & s + 1 \leq i \leq s + (n - t - 16j) \\
\Delta L^r_{i} = e_{t-1}, & i = s, \quad j < (n - t)/16
\end{cases} \tag{2}
\]
Some properties of **Simon**

Property 1 gives a kind of differential propagation path.
Some properties of Simon

Property 1 gives a kind of differential propagation path.

<table>
<thead>
<tr>
<th>Rounds $r$</th>
<th>$\Delta L$</th>
<th>$\Delta R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$000000....00e_1e_2e_3...e_t00$</td>
<td>$000000...0000...00000000$</td>
</tr>
<tr>
<td>1</td>
<td>$000..0..***********000$</td>
<td>$000000....00e_1e_2e_3...e_t00$</td>
</tr>
<tr>
<td>2</td>
<td>$000..***********e_t00$</td>
<td>$000..0..***********000$</td>
</tr>
<tr>
<td>3</td>
<td>$00...***********000$</td>
<td>$000..***********e_t00$</td>
</tr>
<tr>
<td>4</td>
<td>$00..***********e_t00$</td>
<td>$00...***********000$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

The differential propagation path shows:

- If the rightmost bit position of $e$ is $s$, then before $e$ is fully diffused, the $s$-th bit difference value of $\Delta L$ remains unchanged after even rounds’ encryption.

- At the same time, $e_t$ is followed by a number of consecutive 0s.
Some properties of **Simon**

**Property 2** For two \(n\)-bit differences \(X = x_0x_1, \ldots, x_{n-1}\) and \(\Delta X = \Delta x_0\Delta x_1, \ldots, \Delta x_{n-1}\), let \(\Delta Y = \Delta y_0\Delta y_1, \ldots, \Delta y_{n-1} = F(X) \oplus F(X \oplus \Delta X)\), then some bits of \(X = x_0x_1x_2, \ldots, x_{n-1}\) can be deduced through some bit relations between \(\Delta X\).

\[
\begin{array}{cccccc}
\Delta x_{i+1} & \Delta x_{i+8} & x_{i+1} & x_{i+8} \\
0 & 0 & ? & ? \\
0 & 1 & \Delta y_i \oplus \Delta x_{i+2} & ? \\
1 & 0 & ? & \Delta y_i \oplus \Delta x_{i+2} \\
1 & 1 & ? & ? \\
\end{array}
\]

Property 2 can help to recover some bits of intermediate values, which can further reveal some bits of round keys.
Fault Attack on \textbf{Simon}

- Fault model: random byte fault
- Fault injection location: $L^{T-m-1}$ (m=2,3 or 4 depending on the key size)

\[ \Delta L^0 = \ldots 0 \Delta L_{s-7}^0 \Delta L_{s-6}^0 \ldots \Delta L_{s-5}^0 \ldots \Delta L_s^0 \ldots \]

\[ \Delta R^0 = 0 \]
Fault Attack on Simon

Attack procedure:

1. Select a plaintext and encrypt it correctly.
2. Inject a byte fault in $LT_{-m-1}$.
3. $\Delta LT_{-1}$ and $\Delta RT_{-1}$ can be easily obtained from the structure of Feistel.
4. By using property 1, the attacker can determine the rightmost bit injection position with the value 1. (e.g., if $\Delta LT_0 = 1$, then $s$ can be determined).
Fault Attack on **Simon**

Attack procedure:

1. Select a plaintext and encrypt it correctly.

\[ \Delta L^0 = \ldots \Delta L_{s-7}^0 \Delta L_{s-6}^0 \ldots \Delta L_{s}^0 \]

\[ \Delta R^0 = 0 \]
Fault Attack on SIMON

Attack procedure:

1. Select a plaintext and encrypt it correctly.
2. Inject a byte fault in $L^{T-m-1}$.
Fault Attack on Simon

Attack procedure:

1. Select a plaintext and encrypt it correctly.
2. Inject a byte fault in $LT_{m-1}$.
3. $\Delta L^{T-1}$ and $\Delta R^{T-1}$ can be easily obtained from the structure of Feistel.
Fault Attack on Simon

Attack procedure:

1. Select a plaintext and encrypt it correctly.
2. Inject a byte fault in $L^{T-m-1}$.
3. $\Delta L^{T-1}$ and $\Delta R^{T-1}$ can be easily obtained from the structure of Feistel.
4. By using property 1, the attacker can determine the rightmost bit injection position with the value 1. (e.g., if $\Delta L^0_s = 1$, then $s$ can be determined).
Fault Attack on Simon

Attack procedure:

5 Compute $\Delta L_{T-2}$ and $\Delta L_{T-1} \oplus \Delta R_{T-2}$. \(\Delta L_{T-2}\) and \(\Delta L_{T-1}\) can be easily obtained. The whole value of \(\Delta R_{T-2}\) is unknown, but some bits are 0s according to property 1. So \(\Delta L_{T-1} \oplus \Delta R_{T-2}\) can be partially deduced.
Fault Attack on Simon

Attack procedure:

5 Compute $\Delta L^{T-2}$ and $\Delta L^{T-1} \oplus \Delta R^{T-2}$. $\Delta L^{T-2}$, $\Delta L^{T-1}$ can be easily obtained. The whole value of $\Delta R^{T-2}$ is unknown, but some bits are 0s according to property 1. So $\Delta L^{T-1} \oplus \Delta R^{T-2}$ can be partially deduced.

6 By using property 2, some bits of $L^{T-2}$ can be recovered, which can directly deduce some bits of $K^{T-1}$. 
Fault Attack on Simon

Attack procedure:

5 Compute $\Delta L^{T-2}$ and $\Delta L^{T-1} \oplus \Delta R^{T-2}$. $\Delta L^{T-2}$, $\Delta L^{T-1}$ can be easily obtained. The whole value of $\Delta R^{T-2}$ is unknown, but some bits are 0s according to property 1. So $\Delta L^{T-1} \oplus \Delta R^{T-2}$ can be partially deduced.

6 By using property 2, some bits of $L^{T-2}$ can be recovered, which can directly deduce some bits of $K^{T-1}$.

7 By repeating Step 1 to Step 6, the whole value of $K^{T-1}$ can be extracted gradually.
Fault Attack on Simon

Attack procedure:

8 To recover the whole master key, $K^{T-2}$ also requires to be recovered when $m = 2$. By partially decrypting the ciphertexts with $K^{T-1}$, $L^{T-1}$ and $R^{T-1}$ can be obtained.
Fault Attack on Simon

Attack procedure:

8 To recover the whole master key, $K^{T-2}$ also requires to be recovered when $m = 2$. By partially decrypting the ciphertexts with $K^{T-1}$, $L^{T-1}$ and $R^{T-1}$ can be obtained.

9 By executing the similar steps as Step 2 to Step 7, $K^{T-2}$ can be recovered.
Attack procedure:

8 To recover the whole master key, $K^{T-2}$ also requires to be recovered when $m = 2$. By partially decrypting the ciphertexts with $K^{T-1}$, $L^{T-1}$ and $R^{T-1}$ can be obtained.

9 By executing the similar steps as Step 2 to Step 7, $K^{T-2}$ can be recovered.

10 For $m = 3$ or $m = 4$, additional round keys require to be recovered, and they can be revealed by the similar steps as Step 8 to Step 9.
How many faults are required to recover $L^{T-2}$?

Calculation procedure:

1. Calculate the probability that $\Delta L_{i}^{T-2} = 1$ with the fault value $e$ injected from the $(s - 7)$-th to $s$-th bit.
Data Complexity Analysis

How many faults are required to recover $L^{T-2}$?

Calculation procedure:

1. Calculate the probability that $\Delta L_i^{T-2} = 1$ with the fault value $e$ injected from the $(s - 7)$-th to $s$-th bit.

2. According to property 2, calculate the probability that $L_i^{T-2}$ can be recovered after the fault injection. (Denoted by $U_{i,s,e}$.)
Data Complexity Analysis

3 Calculate the number of the fault injections required to recover all the bits of $L^{T-2}$ (Denoted by $f_n$)
3 Calculate the number of the fault injections required to recover all the bits of $L^{T-2}$ (Denoted by $f_n$)

- Denote by $q_i$ the probability that $L_{i}^{T-2}$ is recovered considering all the $(s, e)$ combinations.

$$q_i = \frac{1}{255n} \sum_{s=0}^{n-1} \sum_{e=1}^{255} U_{i,s,e}$$
3 Calculate the number of the fault injections required to recover all the bits of $L_{T-2}^T$ (Denoted by $f_n$)

- Denote by $q_i$ the probability that $L_{i-2}^T$ is recovered considering all the $(s, e)$ combinations.

$$q_i = \frac{1}{255n} \sum_{s=0}^{n-1} \sum_{e=1}^{255} U_{i,s,e}$$

- $q_{i}^{l}$ represents the probability that $L_{i-2}^T$ is recovered after $l$ fault injections.

$$q_{i}^{l} = 1 - (1 - q_i)^l$$
3 Calculate the number of the fault injections required to recover all the bits of $L^{T-2}$ (Denoted by $f_n$)

- Denote by $q_i$ the probability that $L_i^{T-2}$ is recovered considering all the $(s, e)$ combinations.

$$q_i = \frac{1}{255n} \sum_{s=0}^{255} \sum_{e=1}^{n-1} U_{i,s,e}$$

- $q_i^l$ represents the probability that $L_i^{T-2}$ is recovered after $l$ fault injections.

$$q_i^l = 1 - (1 - q_i)^l$$

- Finally,

$$f_n = \sum_{l=1}^{\infty} (Q^l - Q^{l-1})l, \quad Q^0 = 1$$
Data Complexity Analysis

4 After $L^{T-2}$ is recovered, $K^{T-1}$ can be deduced directly. In addition, the same correct and faulty ciphertexts to recover $L^{T-2}$ are also used to recover $L^{T-3},...,L^{T-m-1}$, which corresponds to $K^{T-2},...,K^{T-m}$ respectively. So the total number of the fault injections to extract the master key is about $f_n$. 

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Data Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2n$</td>
<td>$f_n$</td>
</tr>
<tr>
<td>$64/96$</td>
<td>$27.97$</td>
</tr>
<tr>
<td>$96/96$</td>
<td>$33.57$</td>
</tr>
<tr>
<td>$96/144$</td>
<td>$46.93$</td>
</tr>
<tr>
<td>$128/128$</td>
<td>$48.23$</td>
</tr>
<tr>
<td>$128/192$</td>
<td>$67.18$</td>
</tr>
<tr>
<td>$128/256$</td>
<td>$89.21$</td>
</tr>
</tbody>
</table>
After $L^{T-2}$ is recovered, $K^{T-1}$ can be deduced directly. In addition, the same correct and faulty ciphertexts to recover $L^{T-2}$ are also used to recover $L^{T-3}, \ldots, L^{T-m-1}$, which corresponds to $K^{T-2}, \ldots, K^{T-m}$ respectively. So the total number of the fault injections to extract the master key is about $f_n$.

<table>
<thead>
<tr>
<th>SIMON $2n/mn$</th>
<th>$f_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMON $64/96$</td>
<td>27.97</td>
</tr>
<tr>
<td>SIMON $96/96$</td>
<td>33.57</td>
</tr>
<tr>
<td>SIMON $96/144$</td>
<td>46.93</td>
</tr>
<tr>
<td>SIMON $128/128$</td>
<td>48.23</td>
</tr>
<tr>
<td>SIMON $128/192$</td>
<td>67.18</td>
</tr>
<tr>
<td>SIMON $128/256$</td>
<td>89.21</td>
</tr>
</tbody>
</table>
Applicability and Extendibility Analysis

- For $\text{Simon}$ with $n = 96$ or $128$, our attack also works when faults are injected in the location earlier than the $(T - m - 1)$-th round.
- Besides random byte fault model, our attack is also applicable to random $t$-bit fault model with the similar attack procedure.
Applicability and Extendibility Analysis

- For **Simon** with $n = 96$ or 128, our attack also works when faults are injected in the location earlier than the $(T - m - 1)$-th round.

- For **Simon32/64**, **Simon48/72**, **Simon48/96** and **Simon64/128**, our attack can not extract the whole master key with a fault injected into only one intermediate round.
For Simon with $n = 96$ or $128$, our attack also works when faults are injected in the location earlier than the $(T - m - 1)$-th round.

For Simon32/64, Simon48/72, Simon48/96 and Simon64/128, our attack can not extract the whole master key with a fault injected into only one intermediate round.

Besides random byte fault model, our attack is also applicable to random $t$-bit fault model with the similar attack procedure.
PC verification
PC verification

- Experimental number of the fault injections

<table>
<thead>
<tr>
<th>SIMON2n/mn</th>
<th>Random n-bit model</th>
<th>Random bit model</th>
<th>Random byte model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICISC 2014</td>
<td>FDTC 2014</td>
<td>FDTC 2015</td>
</tr>
<tr>
<td>SIMON64/96</td>
<td>10.45</td>
<td>189.44</td>
<td>126.29</td>
</tr>
<tr>
<td>SIMON96/96</td>
<td>7.46</td>
<td>210.24</td>
<td>105.12</td>
</tr>
<tr>
<td>SIMON96/144</td>
<td>11.19</td>
<td>315.36</td>
<td>210.24</td>
</tr>
<tr>
<td>SIMON128/128</td>
<td>7.82</td>
<td>299.68</td>
<td>149.84</td>
</tr>
<tr>
<td>SIMON128/192</td>
<td>11.73</td>
<td>449.52</td>
<td>299.68</td>
</tr>
<tr>
<td>SIMON128/256</td>
<td>15.64</td>
<td>599.36</td>
<td>299.68</td>
</tr>
</tbody>
</table>
PC verification

- **Experimental number of the fault injections**

<table>
<thead>
<tr>
<th>SIMON2n/mn</th>
<th>Random n-bit model</th>
<th>Random bit model</th>
<th>Random byte model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICISC 2014</td>
<td>FDTC 2014</td>
<td>FDTC 2015</td>
</tr>
<tr>
<td>SIMON64/96</td>
<td>10.45</td>
<td>189.44</td>
<td>126.29</td>
</tr>
<tr>
<td>SIMON96/96</td>
<td>7.46</td>
<td>210.24</td>
<td>105.12</td>
</tr>
<tr>
<td>SIMON96/144</td>
<td>11.19</td>
<td>315.36</td>
<td>210.24</td>
</tr>
<tr>
<td>SIMON128/128</td>
<td>7.82</td>
<td>299.68</td>
<td>149.84</td>
</tr>
<tr>
<td>SIMON128/192</td>
<td>11.73</td>
<td>449.52</td>
<td>299.68</td>
</tr>
<tr>
<td>SIMON128/256</td>
<td>15.64</td>
<td>599.36</td>
<td>299.68</td>
</tr>
</tbody>
</table>

- **Round locations of the fault injections**

<table>
<thead>
<tr>
<th>SIMON2n/mn</th>
<th>Random n-bit model</th>
<th>Random bit model</th>
<th>Random byte model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICISC 2014</td>
<td>FDTC 2014</td>
<td>FDTC 2015</td>
</tr>
<tr>
<td>SIMON64/96</td>
<td>$L^{38}, L^{39}, L^{40}$</td>
<td>$L^{38}, L^{39}, L^{40}$</td>
<td>$L^{38}, L^{39}$</td>
</tr>
<tr>
<td>SIMON96/96</td>
<td>$L^{49}, L^{50}$</td>
<td>$L^{49}, L^{50}$</td>
<td>$L^{49}$</td>
</tr>
<tr>
<td>SIMON96/144</td>
<td>$L^{50}, L^{51}, L^{52}$</td>
<td>$L^{50}, L^{51}, L^{52}$</td>
<td>$L^{50}, L^{51}$</td>
</tr>
<tr>
<td>SIMON128/128</td>
<td>$L^{65}, L^{66}$</td>
<td>$L^{65}, L^{66}$</td>
<td>$L^{65}$</td>
</tr>
</tbody>
</table>
Summary

- Compared with the previous work, our attack successfully reduces the number of injected round locations to 1 for six instances of Simon.
Summary

- Compared with the previous work, our attack successfully reduces the number of injected round locations to 1 for six instances of Simon.
- We also give a theoretical estimation of data complexity, which shows less fault injections are required in our attack compared with other attacks under the same fault model.
Summary

- Compared with the previous work, our attack successfully reduces the number of injected round locations to 1 for six instances of Simon.
- We also give a theoretical estimation of data complexity, which shows less fault injections are required in our attack compared with other attacks under the same fault model.
- Our method can also be extended to the random $t$-bit model.
Thank you!