Impossible Differential Cryptanalysis of Reduced Round SIMON

Zhan Chen¹, Ning Wang²,³, and Xiaoyun Wang²,³,⁴*

¹ Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
z-chen14@mails.tsinghua.edu.cn
² Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China
wangning.2014@hotmail.com
³ School of Mathematics, Shandong University, Jinan 250100, China
xiaoyunwang@mail.tsinghua.edu.cn
⁴ Institute of Advanced Study, Tsinghua University, Beijing 100084, China

Abstract

Impossible differential is a useful method for cryptanalysis. SIMON is a light weight block cipher that has attracted lots of attention ever since its publication in 2013. In this paper we propose impossible differential attack on five versions of SIMON, using bit conditions to minimize key bits guessed. We calculate keybits and give the exact attack results.

Keywords: SIMON, impossible differential, bit condition

1 Introduction

SIMON [1] is a family of block ciphers designed by the U.S. National Security Agency (NSA) in 2013. It is designed to have excellent performance on both hardware and software [2]. It has a feistel structure and 5 different block sizes with different key lengths. Ever since its publication, SIMON has attracted much cryptanalysis such as differential analysis [3] [4] [5] [6], linear cryptanalysis [7] [8], impossible differential and zero-correlation linear hull cryptanalysis [7] [9]. Impossible differential attacks were independently introduced by Knudsen [10] and Biham et al. [11], the aim of impossible differential cryptanalysis is to use differentials that never occur to eliminate wrong key candidates that result in such a differential.

This paper is organized as follows. We give a brief description of SIMON and some notation- s in section 2. In section 3, we express a useful property of SIMON concerning bit conditions

*Corresponding author
which is used in our attack. In section 4 we give a 19-round impossible differential attack on SIMON32. In section 5 we mount 20-round impossible differential attack on SIMON48/72. Section 6 concludes the paper.

2 A brief description of Simon

SIMON is a feistel structure block cipher with block size $2n$ where $n \in \{16, 24, 32, 48, 64\}$, and key size $mn$ where $m \in \{2, 3, 4\}$, usually denoted as SIMON$2n/mn$. We list some notations as follows:

- $X_i[n, \ldots, 2n-1, 0, \ldots, n-1]$ : the input of the $i$-th round
- $L_i[n, \ldots, 2n-1]$ : the left half of the $i$-th round input
- $R_i[0, \ldots, n-1]$ : the right half of the $i$-th round input
- $\Delta X_i$ : the difference of two inputs $X_i$ and $X'_i$
- $k_i[0, \ldots, n-1]$ : the subkey of the $i$-th round
- $X \ll r$ : the left rotation of $X$ by $r$ bits
- $\oplus$ : bitwise exclusive OR
- $\cap$ : bitwise AND
- $\%$ : modular operation

All versions of SIMON with corresponding numbers of rounds are listed in Table 1.

<table>
<thead>
<tr>
<th>Block size ($2n$)</th>
<th>Key size ($mn$)</th>
<th>Number of rounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>48</td>
<td>72</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>36</td>
</tr>
<tr>
<td>64</td>
<td>96</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>44</td>
</tr>
<tr>
<td>96</td>
<td>96</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>144</td>
<td>54</td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>192</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>72</td>
</tr>
</tbody>
</table>

Simon uses a simple round function $F(x) = (x \ll 1) \cap (x \ll 8) \oplus (x \ll 2)$. The plaintext is $(L_0, R_0)$. After round $i$, $(L_i, R_i)$ are updated to $(L_{i+1}, R_{i+1})$ as follows:

$$L_{i+1} = F(L_i) \oplus R_i \oplus k_i$$
$$R_{i+1} = L_i$$
The output of the last round \((L_{N_r}, R_{N_r})\) yields the ciphertext.

The key schedules generate a sequence of \(N_r\) subkeys \(\{k_0, \ldots, k_{N_r-1}\}\). The procedure differs, depending on the value \(m\). The first \(m\) subkeys are initialized by the master key. For \(i = m, \ldots, N_r-1\),

\[
k_i = c \oplus (z_j)_{i-m} \oplus k_{i-m} \oplus Y_{i-m} \oplus (Y_{i-m} \ll 1),
\]

where

\[
Y_{i-m} = \begin{cases} 
  k_{i-m+1} \ll 3, & \text{if } m = 2, \\
  k_{i-m+2} \ll 3, & \text{if } m = 3, \\
  k_{i-m+3} \ll 3 \oplus k_{i-m+1}, & \text{if } m = 4.
\end{cases}
\]

Here \(c = 2^n - 4\) and \(z_j\) is a version-dependent constant sequence. For more details refer to [2].

### 3 Some Observations of Bit Property

**Observation 1 (from [12])** Let \(\Delta x = x \oplus x', \Delta y = y \oplus y'\), then

\[
(x \land y) \oplus (x' \land y) = \Delta x \land y \\
(x \land y) \oplus (x \land y') = x \land \Delta y \\
(x \land y) \oplus (x' \land y') = (x \land \Delta y) \oplus (\Delta x \land y) \oplus (\Delta x \land \Delta y)
\]

**Observation 2 (from [3])** Given two inputs \(X_i\) and \(X'_i\) of the \(i\)-th round, the difference of output \(\Delta X_{i+1}\) can be computed without any information of subkeys. Each bit of the difference \(\Delta X_{i+1}\) can be computed with no more than one key bits, depending on two bits of \(\Delta X_{i+1}\).

We know from the round function that

\[
X_{i+1}[j + n] = X_i[(j + 1)\% n + n] \cap X_i[(j + 8)\% n + n] \oplus X_i[(j + 2)\% n + n] \oplus X_i[j] \oplus K_i[j],
\]

so

\[
\Delta X_{i+1}[j + n] = (\Delta X_i[(j + 1)\% n + n] \cap X_i[(j + 8)\% n + n]) \\
\quad \oplus (X_i[(j + 1)\% n + n] \cap \Delta X_i[(j + 8)\% n + n]) \\
\quad \oplus (\Delta X_i[(j + 1)\% n + n] \cap \Delta X_i[(j + 8)\% n + n]) \\
\quad \oplus \Delta X_i[(j + 2)\% n + n] \oplus \Delta X_i[j].
\]

When computing \(\Delta X_{i+1}[j + n]\) we need the value \((\Delta X_{i+1}[(j + 1)\% n + n]), \Delta X_{i+1}[(j + 8)\% n + n])\).

If \((\Delta X_{i+1}[(j + 1)\% n + n]), \Delta X_{i+1}[(j + 8)\% n + n]) = (0, 0)\), then \(\Delta X_{i+1}[j + n]\) can be computed without any keybits.

If \((\Delta X_{i+1}[(j + 1)\% n + n]), \Delta X_{i+1}[(j + 8)\% n + n]) = (0, 1)\), then only \(X_{i+1}[(j + 1)\% n + n]\) is needed, we only have to guess \(k_{i+1}[(j + 1)\% n]\).

If \((\Delta X_{i+1}[(j + 1)\% n + n]), \Delta X_{i+1}[(j + 8)\% n + n]) = (1, 0)\), then only \(X_{i+1}[(j + 8)\% n + n]\) is needed, we only have to guess \(k_{i+1}[(j + 8)\% n]\).

If \((\Delta X_{i+1}[(j + 1)\% n + n]), \Delta X_{i+1}[(j + 8)\% n + n]) = (1, 1)\), then we only have to guess \(k_{i+1}[(j + 1)\% n] \oplus k_{i+1}[(j + 8)\% n]\).

This bit condition reduces the key bits guessed greatly. We will use this property in our attack.
4 Impossible Differential Attack on SIMON32/64

We use a impossible differential characteristic of 11 rounds. This path is that, given input difference $\Delta X = [0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000]$, after 11 SIMON32/64 rounds, the output difference cannot be like this: $[0000, 0000, 1000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000]$ or this: $[0000, 0010, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000, 0000]$. Here we should note that, the input difference or output difference must have at least one non-zero bit. Since only zero difference propagates into zero difference, and the plaintext pairs or ciphertext pairs we use all have non-zero difference. Using a potential zero input difference or output difference may result in better attack than it should be.

We add four rounds on top and four rounds at the bottom of the first impossible differential path, and present attack on 19 rounds SIMON32/64. The state of each round is listed in Table 2:

<table>
<thead>
<tr>
<th>Table 2: States of SIMON32/64</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Delta L_0 = (000?, ??0?, 01??, ??0?)$</td>
</tr>
<tr>
<td>$\Delta L_1 = (0000, 0??0, 0001, ??0?)$</td>
</tr>
<tr>
<td>$\Delta L_2 = (0000, 000?, 0000, 01?)$</td>
</tr>
<tr>
<td>$\Delta L_3 = (0000, 0000, 0000, 0001)$</td>
</tr>
<tr>
<td>$\Delta L_4 = (0000, 0000, 0000, 0000)$</td>
</tr>
<tr>
<td>$\Delta L_5 = (0000, 0000, 1000, 0000)$</td>
</tr>
<tr>
<td>$\Delta L_6 = (?000, 001?, 0000, 0000)$</td>
</tr>
<tr>
<td>$\Delta L_7 = (0000, 1??0, 0000, 0000)$</td>
</tr>
<tr>
<td>$\Delta L_8 = (?01?, ???, 0000, 0000)$</td>
</tr>
<tr>
<td>$\Delta L_9 = (1???, ??0?, 0000, 0000)$</td>
</tr>
</tbody>
</table>

4.1 Procedure of the attack

**Step 1** We build structures as follows: there are 10 necessary conditions on the plaintext, we divide all the 232 plaintexts into 210 parts, with 10 bits $X_0[0, 7, 14, 16, 17, 18, 22, 24, 25, 31]$ fixed for each part and other 22 bits traversing. Because $\Delta X_1$ can be computed without any key bits, we can use this property for data collection. There are 11 conditions on $\Delta X_1$, only 8 of them need to be considered. The other 3 are certain to hold. By round function definition, we build 8 equations $X_1[j + n] = X_0[(j + 1)\%n + n] \cap X_0[(j + 8)\%n + n] \oplus X_0[(j + 2)\%n + n] \oplus X_0[j]$ for $X_1[17, 18, 19, 20, 24, 25, 26, 27]$ and solve the equation system. So for each of the 210 parts, we obtain 28 structures with 10 bits of $X_0$ and 8 bits of $X_1$ fixed and other 14 bits traversing. In total we get 218 structures.

**Step 2** Two structures with three different bits $X_0[25, 7], X_1[27]$ can form 228 pairs. The 232 plaintexts can form 217+1 structures which is 217+28 = 245 pairs. Encrypt these pairs and choose pairs whose ciphertexts have zero difference at $\Delta X_1[23, 25, 1, 8, 9, 10, 11, 15]$ and non-zero difference at $\Delta X_1[16, 2]$, the expected number of pairs left is $245−10 = 235$.

**Step 3** Compute 8 bits difference $\Delta X_1[1, 2, 3, 4, 10, 11, 12, 13]$ for the remaining pairs, and select the pairs that satisfy the required difference. The expected number of pairs left is
2^{35} \times 2^{-8} = 2^{27}.

**Step 4** We want $\triangle L_2 = (0000, \underline{000?}, 0000, \underline{01?0})$.


2. From property 2, $\triangle X_2[20] = (X_1[21] \cap \triangle X_1[28]) \oplus (\triangle X_1[21] \cap X_1[28]) \oplus (\triangle X_1[21] \cap \triangle X_1[28]) \oplus \triangle X_1[22] \oplus \triangle X_1[4]$.

   If $(\triangle X_1[21], \triangle X_1[28]) = (0, 0)$, then discard these pairs. If $\triangle X_1[22] \oplus \triangle X_1[4] = 0$, then $k_0[5, 12]$ has 4 values.

   If $(\triangle X_1[21], \triangle X_1[28]) = (0, 1)$ then $\triangle X_2[20] = X_1[21] \oplus \triangle X_1[22] \oplus \triangle X_1[4]$ where $X_1[21] = X_0[22] \cap X_0[29] \oplus X_0[23] \oplus X_0[5] \oplus k_0[5]$, then $k_0[5]$ has 1 value.

   If $(\triangle X_1[21], \triangle X_1[28]) = (1, 0)$, then $\triangle X_2[20] = X_1[28] \oplus \triangle X_1[22] \oplus \triangle X_1[4]$ where $X_1[28] = X_0[29] \cap X_0[20] \oplus X_0[30] \oplus X_0[12] \oplus k_0[12]$, then $k_0[12]$ has 1 value.

   If $(\triangle X_1[21], \triangle X_1[28]) = (1, 1)$, then $\triangle X_2[20] = X_1[21] \oplus X_1[28] \oplus 1 \oplus \triangle X_1[22] \oplus \triangle X_1[4]$, so $k_0[5] \oplus k_0[12]$ has 1 value.

   For all the above circumstances, discard the pairs that do not meet requirement, there are $2^{27} \times (1 - \frac{1}{8})$ left. $k_0[5, 12]$ has $\frac{16}{7}$ values.

3. From property 2, $\triangle X_2[21] = (X_1[22] \cap \triangle X_1[29]) \oplus (\triangle X_1[22] \cap X_1[29]) \oplus (\triangle X_1[22] \cap \triangle X_1[29]) \oplus \triangle X_1[4]$.

   Similarly, discard unnecessary pairs there are $2^{27} \times (1 - \frac{1}{8})^2$ left. $k_0[6, 13]$ has $\frac{16}{7}$ values.


5. From property 2, if $\triangle X_1[28] = 1$ then $\triangle X_2[27] = X_1[19] \oplus \triangle X_1[29] \oplus \triangle X_1[11]$, $k_0[3]$ has 1 value.

   If $\triangle X_1[28] = 0$ then $\triangle X_2[27] = \triangle X_1[29] \oplus \triangle X_1[11]$. If $\triangle X_1[29] \oplus \triangle X_1[11] = 0$ then discard these pairs. If $\triangle X_1[29] \oplus \triangle X_1[11] = 0$, then $k_0[3]$ has 2 values.

   After discarding unnecessary pairs there are $2^{27} \times (1 - \frac{1}{8})^2 \times (1 - \frac{1}{7})$ left. $k_0[3]$ has $\frac{3}{8}$ values.

6. Similarly about $\triangle X_2[28]$, after discarding unnecessary pairs there are $2^{27} \times (1 - \frac{1}{8})^2 \times (1 - \frac{1}{7})^2$ left. $k_0[3]$ has $\frac{3}{8}$ values.

7. Similarly about $\triangle X_2[29]$, after discarding unnecessary pairs there are $2^{27} \times (1 - \frac{1}{8})^2 \times (1 - \frac{1}{7})^2 = 2^{25.37}$ left. $k_0[14]$ has $\frac{4}{7}$ values.

   In this step, 8 key bits have in total $1 \times \frac{16}{7} \times \frac{16}{7} \times 1 \times \frac{4}{3} = \frac{4048}{441}$ values.

**Step 5** We want $\triangle R_{17} = (?000, \underline{001?}, 0000, \underline{0000})$.

As is done in step 4, we discard pairs that do not help in our attack, and calculate keybits. There are $2^{25.37} \times (1 - \frac{1}{8})^3 \times (1 - \frac{1}{7})^2 = 2^{24.15}$ pairs left and 8 key bits have in total $1 \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} \times 1 \times \frac{16}{7} \times \frac{16}{7} = \frac{16384}{123}$ values.

**Step 6** As for $\triangle L_3 = (0000, \underline{0000}, 0000, \underline{0001})$, there are $2^{24.15} \times (1 - \frac{1}{7}) \times (1 - \frac{1}{7})^2 = 2^{23.127}$ pairs left and the 10 key bits have in total $3 \times \frac{36}{7} \times 1 \times \frac{4}{3} \times \frac{8}{3} = \frac{384}{7}$ values.
Step 7 As for $R_{16} = (0000, 0000, 1000, 0000)$, there are $2^2 \times (1 - \frac{1}{3})^2 \times (1 - \frac{1}{8})$ pairs left and the 10 key bits have in total $2 \times \frac{4}{3} \times \frac{16}{3} \times 2 \times \frac{16}{7} = \frac{4096}{63}$ values.

Step 8 As for $R_{15} = (0000, 0000, 0000, 0000)$, 6 key bits have in total $2 \times 4^3 \times 16^3 \times 2 \times 16^7 = 4096^{63}$ values.

Step 9 As for $\triangle L_4 = (0000, 0000, 0000, 0000)$, 6 key bits have in total $2 \times 4^3 \times 16^3 \times 2 \times 16^7 = 4096^{63}$ values.

Step 10 In the end we have $2^{27} \times (1 - \frac{1}{3})^{10} \times (1 - \frac{1}{8})^6 = 2^{21.693}$ pairs left. The 10 key bits have in total $2^{27} \times (1 - \frac{1}{3})^{10} \times (1 - \frac{1}{8})^6$ values.

That is to say each pair can sieve out $2^{28.646}$ wrong key bits. There remains $2^{49} \times (1 - \frac{28.646}{249})^{21.693} = 2^{49} \times 2^{-3.65} = 2^{45.35}$ key candidates.

Step 11 Above is the sieving results of the first impossible differential path. We add four rounds on top and bottom of the second impossible differential path. The number of key bits involved is also 49. 44 of them appeared for the first path. So the $2^{45.35}$ 44-bit keys together with the other 5-bit keys of the second path is the remaining key candidates.

Similarly we sieve the key candidates using the second path, in the end, the number of remaining keys is $2^{45.35+5} \times (1 - \frac{28.646}{249})^{21.693} = 2^{50.35-1.431} = 2^{48.919}$. That is to say, the total 54-bit keys are left with $2^{48.919}$.

The right key is definitely included in the remaining keys. We have guessed 54 bits, only have to traverse 10 bit equivalent bits to compute the master key, and test on 2 pairs of plaintexts.

4.2 Complexity analysis

The data complexity is $2^{32}$ known plaintexts. The memory complexity is the storing of remaining key candidates in step 11, which is $2^{48.919} \times 54/32 = 2^{49.674}$ states. The time complexity is also dominated by Step 11. In step 11, the time complexity is $2^{48.919} \times 2^{10} = 2^{58.919}$ 19-round SIMON32.

5 Impossible Differential Attacks on SIMON48/72

The 12-round impossible differential path we use is that, given input difference $[(0000, 0000, 0000, 0000, 0000, 0000), (1000, 0000, 0000, 0000, 0000, 0000)]$, after 12 round SIMON48 the output difference cannot be: $[(0100, 0000, 0000, 0000, 0000, 0000), (0000, 0000, 0000, 0000, 0000, 0000)]$.

We add four rounds on top and four rounds at the bottom, and present attack on 20-round SIMON48/72. The state of each round is listed in the following Table 3:

5.1 Procedure of the attack

Step 1 Same as done for SIMON32/64, we build $2^{32}$ structures and form $2^{63}$ pairs. Choose pairs that satisfy the differences of $\triangle X_{20}$ and $\triangle X_{19}$, there remains $2^{31}$ pairs.
The data complexity is 2^{5.2}. Complexity analysis

Step 3

The right key is definitely included in the remaining keys. We have guessed 64 bits, and have to traverse 8 equivalent key bits to compute master key and test on two pairs of plaintexts.

5.2 Complexity analysis

The data complexity is 2^{48} known plaintexts. The memory complexity is the storing of remaining key candidates in step 2, which is 2^{63.278} \times 64/48 = 2^{63.393} states. The time complexity is 2^{63.278} \times 2^8 = 2^{71.278} 20-round SIMON48.

6 Conclusion

For SIMON48/96, SIMON64/96 and SIMON64/128, only one impossible differential path can be used for the attack. We present impossible differential attacks on SIMON32, SIMON48 AND SIMON96 with bit-wise precision in this paper. Table 4 is a comparison of previous results and our new results. We have done a detailed calculation of key bits involved, this means no better attack can be obtained other than this.

References

Table 4: Summary of impossible differential attacks on SIMON

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Full rounds</th>
<th>Attacked rounds</th>
<th>Complexity</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time</td>
<td>Data</td>
</tr>
<tr>
<td>SIMON32/64</td>
<td>32</td>
<td>13</td>
<td>(2^{30.1})</td>
<td>(2^{33.291})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14</td>
<td>(2^{44.183})</td>
<td>(2^{33.291})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>(2^{61.14})</td>
<td>(2^{32})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>(2^{62.56})</td>
<td>(2^{32})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>(2^{58.919})</td>
<td>(2^{32})</td>
</tr>
<tr>
<td>SIMON48/72</td>
<td>36</td>
<td>15</td>
<td>(2^{59.079})</td>
<td>(2^{50.262})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>(2^{61.87})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>(2^{71.278})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>(2^{70.69})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td>SIMON48/96</td>
<td>36</td>
<td>15</td>
<td>(2^{63})</td>
<td>(2^{38})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
<td>(2^{69.079})</td>
<td>(2^{50.262})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>(2^{85.82})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>(2^{94.73})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>(2^{94.556})</td>
<td>(2^{48})</td>
</tr>
<tr>
<td>SIMON64/96</td>
<td>42</td>
<td>16</td>
<td>(2^{91.986})</td>
<td>(2^{65.248})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>(2^{95.279})</td>
<td>(2^{64})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>21</td>
<td>(2^{94.56})</td>
<td>(2^{64})</td>
</tr>
<tr>
<td>SIMON64/128</td>
<td>44</td>
<td>16</td>
<td>(2^{91.986})</td>
<td>(2^{65.248})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22</td>
<td>(2^{126.56})</td>
<td>(2^{64})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22</td>
<td>(2^{125.115})</td>
<td>(2^{64})</td>
</tr>
</tbody>
</table>


