Differential Fault Analysis on the families of SIMON and SPECK ciphers

Harshal Tupsamudre *, Shikha Bisht **, Debdeep Mukhopadhyay ***

Indian Institute of Technology, Kharagpur

Abstract. In 2013, the US National Security Agency proposed two new families of lightweight block ciphers: SIMON and SPECK. However, no security analysis was provided for these ciphers. Currently, linear and differential cryptanalytic results for SIMON are available in the literature, but no fault attacks on these two cipher families have been reported so far. In this paper, we present the first fault attack on the families of SIMON and SPECK ciphers. The attack assumes a fault model that can flip only one bit of the intermediate result. Using this attack, the n-bit last round key used in SIMON can be recovered using $(n/2)$ bit faults on an average while the n-bit last round key used in SPECK can be recovered using $(n/3)$ bit faults. Furthermore, we demonstrate a more practical attack on SIMON that employs a random byte fault model. This attack retrieves multiple bits of the key depending upon the Hamming weight of the byte fault. The average number of byte faults required to retrieve all n bits of the last round key is $(n/8)$.

Keywords: Differential Fault Analysis, Fault Attack, Lightweight Block Ciphers, SIMON, SPECK

1 Introduction

Simon and Speck are two families of lightweight block ciphers based upon Feistel structure, designed to provide optimal performance on resource constrained devices. While the Simon family provides the best performance in the hardware environments the Speck family is designed to work well in the software environments. The design requirements and performance analysis of the Simon and Speck family were published by the US National Security Agency in 2013, [1], but no security assessments were provided. Initial results of linear and differential cryptanalysis of the Simon family are available in [2], [3] and [4]. However, it is also important to analyse the security of these block ciphers against the very well known family of side channel attacks, which exploit the information leakage from the physical implementation of the cipher.

In this paper, we present the first fault attack on SIMON and SPECK families of cipher. We show that these ciphers are insecure against an adversary who can flip one bit in the intermediate state of the cipher to produce erroneous ciphertexts. In this case, we can retrieve the n-bit last round key of SIMON and SPECK using $(n/2)$ and $(n/3)$ bit faults respectively. We refer to the fault model used in this attack as the bit-flip fault model. We further show that the SIMON is vulnerable

* harshal.coep@gmail.com
** s.bisht09@gmail.com
*** debdeep.mukhopadhyay@gmail.com

against the fault attack that employs a random byte fault model. In this case, multiple bits of the last round key can be deduced depending upon the Hamming weight of the induced byte fault. The average number of byte faults required to retrieve all n bits of the last round key is $(n/8)$.

Notations. We have used the following notations for both SIMON and SPECK families of cipher.
T : Total number of rounds in the cipher.
K : mn bit secret key used in the cipher.
$(\mathbf{x^i}, \mathbf{y^i})$: The $2n$ bit output of i^{th} RoundFunction of the cipher, $i \in \{1, \ldots, T\}$. Input to the cipher is denoted by (x^0, y^0).
$(\mathbf{x^{i^*}}, \mathbf{y^{i^*}})$: The $2n$ bit **faulty** output of i^{th} RoundFunction of the cipher, $i \in \{1, \ldots, T\}$.
$\mathbf{k^i}$: The n bit round key used in the i^{th} round of the cipher.
$\boldsymbol{S^{-\alpha}(w)}$: Circular right rotation of a n bit word w by α bits.
$\boldsymbol{S^{\beta}(w)}$: Circular left rotation of a n bit word w by β bits.
Further, we denote a bitwise logical AND operation by &, a bitwise logical OR operation by |, a bitwise logical NOT operation by $\neg$ and a bitwise logical XOR operation by $\oplus$. Addition in modulo 2^n is denoted by $+$. We represent the n bits of a word w by $w_{n-1}w_{n-2}\ldots w_1w_0$, where w_0 is the least significant bit and w_{n-1} is the most significant bit of a word w.

Organization. The rest of the paper is organized as follows. We first describe a fault attack on SIMON cipher, that assumes a bit-flip fault model. Then we demonstrate a more realistic attack on SIMON, which employs a random byte fault model. Finally, we describe an attack on the SPECK cipher that also uses a bit-flip fault model.

2 Fault Attack on SIMON

Before explaining the fault attack mechanism, we describe the characteristics of the round function used in the SIMON that enables us to mount the attack.

2.1 Round Function of SIMON

Fig.1. shows a single round transformation of SIMON. A round in the SIMON is a function $R_k : GF(2^n) \times GF(2^n) \rightarrow GF(2^n) \times GF(2^n)$ defined as:

$$R_{k^i}(x^i, y^i) = (x^{i+1}, y^{i+1}) = (y^i \oplus f(x^i) \oplus k^i, x^i) \tag{1}$$

where $i \in \{0, \ldots, T-1\}$ and $f(x^i) = (S^1(x^i)\ \&\ S^8(x^i)) \oplus S^2(x^i)$.

The l^{th} bit of $f(x^i)$ is computed using 3 distinct bits of x^i.

$$f(x^i)_l = (x^i_{(l-1)\%n}\ \&\ x^i_{(l-8)\%n}) \oplus x^i_{(l-2)\%n} \tag{2}$$

where, $l \in \{0 \ldots n-1\}$. Furthermore, the j^{th} bit of x^i affects 3 distinct bits $(j+1)\%n$, $(j+2)\%n$ and $(j+8)\%n$ of $f(x^i)$ as follows:

$$\begin{aligned} f(x^i)_{(j+1)\%n} &= (x^i_j\ \&\ x^i_{(j-7)\%n}) \oplus x^i_{(j-1)\%n} \\ f(x^i)_{(j+2)\%n} &= (x^i_{(j+1)\%n}\ \&\ x^i_{(j-6)\%n}) \oplus x^i_j \\ f(x^i)_{(j+8)\%n} &= (x^i_{(j+7)\%n}\ \&\ x^i_j) \oplus x^i_{(j+6)\%n} \end{aligned} \tag{3}$$

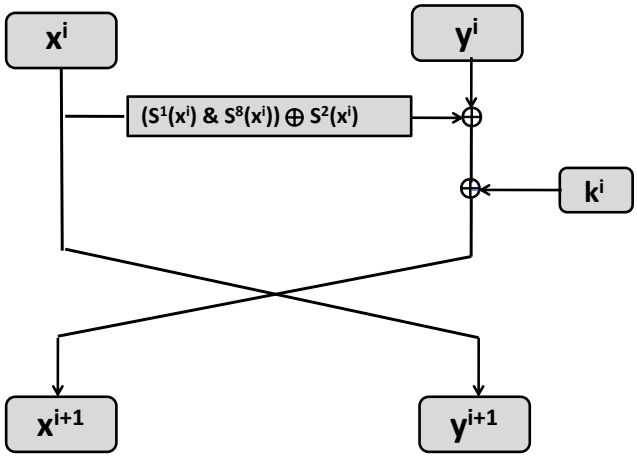

Fig. 1. i^{th} Round of SIMON

where $j \in \{0 \dots n-1\}$. And since $x^{i+1} = y^i \oplus f(x^i) \oplus k^i$, the same bit positions of x^{i+1} are also affected by the j^{th} bit of x^i.

$$\begin{aligned} x^{i+1}_{(j+1)\%n} &= y^i_{(j+1)\%n} \oplus f(x^i)_{(j+1)\%n} \oplus k^i_{(j+1)\%n} \\ x^{i+1}_{(j+2)\%n} &= y^i_{(j+2)\%n} \oplus f(x^i)_{(j+2)\%n} \oplus k^i_{(j+2)\%n} \\ x^{i+1}_{(j+8)\%n} &= y^i_{(j+8)\%n} \oplus f(x^i)_{(j+8)\%n} \oplus k^i_{(j+8)\%n} \end{aligned} \tag{4}$$

2.2 Equation of the Last Round Key

The output of SIMON is denoted as (x^T, y^T), where

$$\begin{aligned} x^T &= y^{T-1} \oplus f(x^{T-1}) \oplus k^{T-1} \\ y^T &= x^{T-1} \end{aligned} \tag{5}$$

Therefore, we can express the last round key k^{T-1} as:

$$k^{T-1} = y^{T-1} \oplus f(x^{T-1}) \oplus x^T \tag{6}$$

Since $y^T = x^{T-1}$ and $y^{T-1} = x^{T-2}$

$$k^{T-1} = x^{T-2} \oplus f(y^T) \oplus x^T \tag{7}$$

From the above equation, it can be seen that the last round key k^{T-1} can be retrieved if the value of x^{T-2} is known. In the following discussion, we describe fault attacks that target and retrieve x^{T-2} in order to recover k^{T-1}.

2.3 Determining the Fault Position and Value

Suppose a fault e is induced in the intermediate result x^{T-2}. Let the resulting faulty ciphertext be (x^{T^*}, y^{T^*}). Since $y^i = x^{i-1}$, $i \in \{1, \ldots, T\}$ we can write:

$$\begin{aligned} x^T \oplus x^{T^*} &= y^{T-1} \oplus f(x^{T-1}) \oplus y^{(T-1)^*} \oplus f(x^{(T-1)^*}) \\ &= y^{T-1} \oplus f(y^T) \oplus y^{(T-1)^*} \oplus f(y^{T^*}) \\ &= x^{(T-2)} \oplus f(y^T) \oplus x^{(T-2)^*} \oplus f(y^{T^*}) \\ &= x^{T-2} \oplus f(y^T) \oplus x^{T-2} \oplus e \oplus f(y^{T^*}) \\ &= f(y^T) \oplus e \oplus f(y^{T^*}) \\ \therefore e &= x^T \oplus x^{T^*} \oplus f(y^T) \oplus f(y^{T^*}) \end{aligned} \tag{8}$$

Since we know the output of correct and faulty computation, we can deduce the value of fault e injected in x^{T-2} and hence, we can determine the bits that are flipped in x^{T-2}.

2.4 Bit-Flip Fault Attack on SIMON

The attack procedure is as follows:

1. Suppose a fault flips j^{th} bit of the intermediate result x^{T-2} resulting in a faulty ciphertext (x_T^*, y_T^*).

   $$y^{T^*} = x^{T-1} = y^{T-2} \oplus f(x^{(T-2)^*}) \oplus k^{T-2} \tag{9}$$

   The xor of correct and faulty computation of y^T can be written as:

   $$y^T \oplus y^{T^*} = f(x^{T-2}) \oplus f(x^{(T-2)^*}) \tag{10}$$

   Since the j^{th} bit of x^{T-2} affects 3 distinct bits of $f(x^{T-2})$, the correct computation of y_T differs from its faulty computation in at most 3 positions:

   $$\begin{aligned} (y^T \oplus y^{T^*})_{(j+1)\%n} &= (x_j^{T-2} \,\&\, x_{(j-7)\%n}^{T-2}) \oplus ((x_j^{T-2} \oplus 1) \,\&\, x_{(j-7)\%n}^{T-2}) \\ (y^T \oplus y^{T^*})_{(j+8)\%n} &= (x_{(j+7)\%n}^{T-2} \,\&\, x_j^{T-2}) \oplus (x_{(j+7)\%n}^{T-2} \,\&\, (x_j^{T-2} \oplus 1)) \\ (y^T \oplus y^{T^*})_{(j+2)\%n} &= x_j^{T-2} \oplus x_j^{T-2} \oplus 1 = 1 \end{aligned} \tag{11}$$

2. From Table 1 it can be seen that if the value of $(y^T \oplus y^{T^*})_{(j+1)\%n}$ is 0, then irrespective of the bit value x_j^{T-2} the value of the bit $x_{(j-7)\%n}^{T-2}$ is 0, otherwise it is 1. We can also deduce from Table 2 that if the value of bit $(y^T \oplus y^{T^*})_{(j+8)\%n}$ is 0 then the value of the bit $x_{(j+7)\%n}^{T-2}$ is 0, otherwise it is 1.

   $$\begin{aligned} x_{(j-7)\%n}^{T-2} &= (y^T \oplus y^{T^*})_{(j+1)\%n} \\ x_{(j+7)\%n}^{T-2} &= (y^T \oplus y^{T^*})_{(j+8)\%n} \end{aligned} \tag{12}$$

3. Since we now know the values of the bits $x_{(j-7)\%n}^{T-2}$ and $x_{(j+7)\%n}^{T-2}$, we can retrieve the corresponding bits of k^{T-1} using equation (7):

   $$\begin{aligned} k_{(j-7)\%n}^{T-1} &= (x_{(j-7)\%n}^{T-2} \oplus f(y_{(j-7)\%n}^T) \oplus x_{(j-7)\%n}^T) \\ k_{(j+7)\%n}^{T-1} &= (x_{(j-7)\%n}^{T-2} \oplus f(y_{(j-7)\%n}^T) \oplus x_{(j-7)\%n}^T) \end{aligned} \tag{13}$$

   Thus, using a bit fault in x^{T-2}, we can recover 2 bits of k^{T-1}. Consequently, for retrieving the n bit key, we require n/2 faulty ciphertexts.

Table 1. Deducing the value of bit $x^{T-2}_{(j-7)\%n}$ from $(y^T \oplus y^{T^*})_{(j+1)\%n}$

x^{T-2}_j	$x^{T-2}_j \oplus 1$	$x^{T-2}_{(j-7)\%n}$	$(y^T \oplus y^{T^*})_{(j+1)\%n}$
0	1	0	0
1	0	0	0
0	1	1	1
1	0	1	1

Table 2. Deducing the value of bit $x^{T-2}_{(j+7)\%n}$ from $(y^T \oplus y^{T^*})_{(j+8)\%n}$

x^{T-2}_j	$x^{T-2}_j \oplus 1$	$x^{T-2}_{(j+7)\%n}$	$(y^T \oplus y^{T^*})_{(j+8)\%n}$
0	1	0	0
1	0	0	0
0	1	1	1
1	0	1	1

2.5 Random Byte Fault Attack on SIMON

In this section we describe a more practical fault attack, where we assume that the attacker can affect a byte of x^{T-2} with a random fault. The working principle of this attack is the same as that of the bit-flip fault attack except for the following two cases:

1. Every flipped bit of x^{T-2} retrieves two key bits of k^{T-1}. However if the least and the most significant bits of the induced fault are 1 then each of these bits can retrieve only one key bit.
   Suppose a fault flips the bits x^{T-2}_j and x^{T-2}_{j-7}. A flip in x^{T-2}_j affects 3 bits of y_T so that we get:

$$\begin{aligned}
(y^T \oplus y^{T^*})_{(j+1)\%n} &= (x^{T-2}_j \,\&\, x^{T-2}_{(j-7)\%n}) \oplus ((x^{T-2}_j \oplus 1) \,\&(x^{T-2}_{(j-7)\%n} \oplus 1)) \\
(y^T \oplus y^{T^*})_{(j+8)\%n} &= (x^{T-2}_{(j+7)\%n} \,\&\, x^{T-2}_j) \oplus (x^{T-2}_{(j+7)\%n} \,\&\, (x^{T-2}_j \oplus 1)) \\
(y^T \oplus y^{T^*})_{(j+2)\%n} &= x^{T-2}_j \oplus x^{T-2}_j \oplus 1 = 1
\end{aligned} \tag{14}$$

   And a flip in x^{T-2}_{j-7} also affects 3 bits of y_T, so that we have:

$$\begin{aligned}
(y^T \oplus y^{T^*})_{(j-6)\%n} &= (x^{T-2}_{(j-7)\%n} \,\&\, x^{T-2}_{(j-14)\%n}) \oplus ((x^{T-2}_{(j-7)\%n} \oplus 1) \,\&x^{T-2}_{(j-14)\%n}) \\
(y^T \oplus y^{T^*})_{(j+1)\%n} &= (x^{T-2}_j \,\&\, x^{T-2}_{(j-7)\%n)}) \oplus ((x^{T-2}_j \oplus 1) \,\&\, (x^{T-2}_{(j-7)\%n} \oplus 1)) \\
(y^T \oplus y^{T^*})_{(j-5)\%n} &= x^{T-2}_{(j-7)\%n} \oplus x^{T-2}_{(j-7)\%n} \oplus 1 = 1
\end{aligned} \tag{15}$$

   Similar to the bit fault attack, we expect to retrieve the two bits $x^{T-2}_{(j-7)\%n}$ and $x^{T-2}_{(j+7)\%n}$ from the equation set (14), and the two bits $x^{T-2}_{(j-14)\%n}$ and x^{T-2}_j from the equation set (15), however one can see from Table 3 that using the value of $(y^T \oplus y^{T^*})_{(j+1)\%n}$, the j^{th} and $((j-7)\%n)^{th}$ bit cannot be retrieved. We can only determine whether the bits x^{T-2}_j and $x^{T-2}_{(j-7)\%n}$ are complement

Table 3. Relation between the bits $x^{T-2}_{(j)\%n}$ and $x^{T-2}_{(j-7)\%n}$

x^{T-2}_j	$x^{T-2}_j \oplus 1$	$x^{T-2}_{(j-7)\%n}$	$x^{T-2}_{(j-7)\%n} \oplus 1$	$(y^T \oplus y^{T^*})_{(j+1)\%n}$
0	1	1	0	0
1	0	0	1	0
0	1	0	1	1
1	0	1	0	1

of each other or they have the same value. The actual value of either x^{T-2}_j or $x^{T-2}_{(j-7)\%n}$ cannot be known. Thus, in this case, only two bits: $x^{T-2}_{(j+7)\%n}$ from equation set (14) and $x^{T-2}_{(j-14)\%n}$ from equation set (15) can be retrieved. In all the other cases, the number of key bits that can be retrieved using a byte fault is twice the Hamming weight of the fault, as every flipped bit reveals two bits of the last round key.

2. The attack procedure also differs slightly when a byte fault flips two contiguous bits x^{T-2}_j and x^{T-2}_{j-1}. In this case, a flip in x^{T-2}_j affects 3 bits of y^T so that we get:

$$\begin{aligned}
(y^T \oplus y^{T^*})_{(j+1)\%n} &= (x^{T-2}_j \,\&\, x^{T-2}_{(j-7)\%n}) \oplus x^{T-2}_{(j-1)\%n} \\
&\quad \oplus ((x^{T-2}_j \oplus 1) \,\&\, x^{T-2}_{(j-7)\%n}) \oplus x^{T-2}_{(j-1)\%n} \oplus 1 \\
&= (x^{T-2}_j \,\&\, x^{T-2}_{(j-7)\%n}) \oplus ((x^{T-2}_j \oplus 1) \,\&\, x^{T-2}_{(j-7)\%n}) \oplus 1 \quad (16) \\
(y^T \oplus y^{T^*})_{(j+8)\%n} &= (x^{T-2}_{(j+7)\%n} \,\&\, x^{T-2}_j) \oplus (x^{T-2}_{(j+7)\%n} \,\&\, (x^{T-2}_j \oplus 1)) \\
(y^T \oplus y^{T^*})_{(j+2)\%n} &= x^{T-2}_j \oplus x^{T-2}_j \oplus 1 = 1
\end{aligned}$$

And a flip in x^{T-2}_{j-1} also affects 3 bits of y_T, so that we have:

$$\begin{aligned}
(y^T \oplus y^{T^*})_{(j)\%n} &= (x^{T-2}_{j-1} \,\&\, x^{T-2}_{(j-8)\%n}) \oplus ((x^{T-2}_{j-1} \oplus 1) \,\&\, x^{T-2}_{(j-8)\%n}) \\
(y^T \oplus y^{T^*})_{(j+7)\%n} &= (x^{T-2}_{(j+6)\%n} \,\&\, x^{T-2}_{j-1}) \oplus (x^{T-2}_{(j+6)\%n} \,\&\, (x^{T-2}_{j-1} \oplus 1)) \quad (17) \\
(y^T \oplus y^{T^*})_{(j+1)\%n} &= (x^{T-2}_j \,\&\, x^{T-2}_{(j-7)\%n}) \oplus ((x^{T-2}_j \oplus 1) \,\&\, x^{T-2}_{(j-7)\%n}) \oplus 1
\end{aligned}$$

From Table 4 it can be seen that if the value of $(y^T \oplus y^{T^*})_{(j+1)\%n}$ is 0, then irrespective of the bit value x^{T-2}_j the value of the bit $x^{T-2}_{(j-7)\%n}$ is 1, otherwise it is 0.

$$x^{T-2}_{(j-7)\%n} = \neg(y^T \oplus y^{T^*})_{(j+1)\%n} \quad (18)$$

The bit $x^{T-2}_{(j+7)\%n}$ from equation set (16) and bits $x^{T-2}_{(j-8)\%n}$ and $x^{T-2}_{(j+6)\%n}$ from equation set (17) are obtained in the same way as described previously in bit-flip fault attack.

Table 4. Deducing the value of bit $x^{T-2}_{(j-7)\%n}$ from $(y^T \oplus y^{T^*})_{(j+1)\%n}$

x^{T-2}_j	$x^{T-2}_j \oplus 1$	$x^{T-2}_{(j-7)\%n}$	$(y^T \oplus y^{T^*})_{(j+1)\%n}$
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0

Attack Complexity. A byte fault of Hamming weight z in x^{T-2} retrieves $2z$ bits of the last round key k^{T-1}. The number of possible byte faults having Hamming weight z is $\binom{8}{z}$. Therefore, the expected number of key bits that can be retrieved by a random byte fault is:

$$\sum_{z=1}^{8} z * Pr[z] = \sum_{z=1}^{8} 2z * \binom{8}{z} * \frac{1}{255} \approx 8 \tag{19}$$

And hence the average number of byte faults required to recover all the n bits of k^{T-1} is $(n/8)$.

3 Fault Attack on SPECK

Similar to the attack description of SIMON, we begin by describing the characteristics of the round function used in the SPECK which enable us to mount the attack.

3.1 Round function of SPECK

Fig.2. shows a single round transformation of SPECK. A round in the SPECK is a function $R_k : GF(2^n) \times GF(2^n) \rightarrow GF(2^n) \times GF(2^n)$ defined as

$$R_{k^i}(x^i, y^i) = (x^{i+1}, y^{i+1}) = (f(x^i, y^i) \oplus k^i, S^\beta y^i \oplus f(x^i, y^i) \oplus k^i) \tag{20}$$

where $i \in \{0, \ldots, T-1\}$ and $f(x^i, y^i) = S^{-\alpha}(x^i) + y^i$. The addition in function f is performed modulo 2^n. The j^{th} bit of $f(x^i, y^i)$ is computed as

$$f(x^i, y^i)_j = x^i_{(j-\alpha)\%n} \oplus y^i_j \oplus c_j \tag{21}$$

where the carry bit $c_j = (x^i_{(j-1-\alpha)\%n} \& y^i_{j-1}) \mid (y^i_{j-1} \& c_{j-1}) \mid (x^i_{(j-1-\alpha)\%n} \& c_{j-1})$ and $j \in \{0, \ldots, n-1\}$ and $c_0 = 0$.

3.2 Equation of the last round key

The output of SPECK is denoted by (x^T, y^T), where

$$\begin{aligned} x^T &= (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus k^{T-1} \\ y^T &= (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus k^{T-1} \oplus S^\beta(y^{T-1}) \\ &= x^T \oplus S^\beta(y^{T-1}) \end{aligned} \tag{22}$$

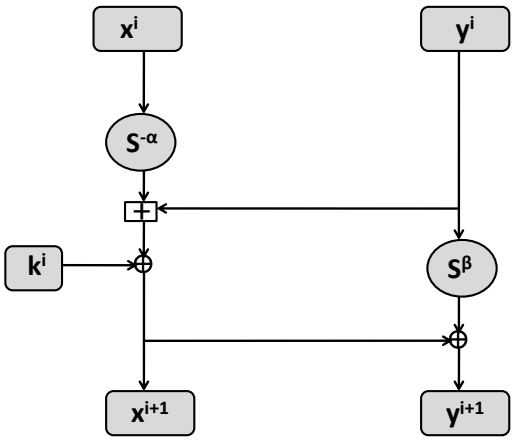

Fig. 2. i^{th} Round of SPECK

We can express the last round key k^{T-1} as follows:

$$k^{T-1} = (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus x^T \tag{23}$$

Since $y^{T-1} = S^{-\beta}(y^T \oplus x^T)$,

$$\therefore k^{T-1} = (S^{-\alpha}(x^{T-1}) + S^{-\beta}(y^T \oplus x^T)) \oplus x^T \tag{24}$$

Consider the j^{th} bit of k^{T-1}

$$k_j^{T-1} = (\ x^{T-1}_{(j-\alpha)\%n} \oplus (y^T \oplus x^T)_{(j-\beta)\%n} \oplus c_j\ ) \oplus x_j^T \tag{25}$$

From the above equation, it can be seen that the j^{th} bit of last round key k^{T-1} can be retrieved if the value of bit $x^{T-1}_{(j-\alpha)\%n}$ and carry bit c_j is known. In the following discussion, we describe a fault attack that targets y^{T-1} and retrieves x^{T-1} in order to recover k^{T-1}.

3.3 Determining the Fault Position and Value

Suppose a fault e is induced in the intermediate result y^{T-1}. Let the resulting faulty ciphertext be (x^{T^*}, y^{T^*}).

$$\begin{aligned} y^T \oplus y^{T^*} &= x^T \oplus S^\beta(y^{T-1}) \oplus x^{T^*} \oplus S^\beta(y^{(T-1)^*}) \\ \therefore y^{T-1} \oplus y^{(T-1)^*} &= S^{-\beta}(y^T \oplus y^{T^*} \oplus x^T \oplus x^{T^*}) \\ \therefore e &= S^{-\beta}(y^T \oplus y^{T^*} \oplus x^T \oplus x^{T^*}) \end{aligned} \tag{26}$$

Since we know the output of correct and faulty computation, we can deduce the value of fault e injected in y^{T-1} and therefore, we can determine the bits that are flipped in y^{T-1}.

3.4 Bit-Flip Fault Attack on SPECK

Consider the j^{th} bit of last round key,

$$k_j^{T-1} = (\ x_{(j-\alpha)\%n}^{T-1} \oplus (y^T \oplus x^T)_{(j-\beta)\%n} \oplus c_j\ ) \oplus x_j^T \tag{27}$$

In order to retrieve the j^{th} bit of k^{T-1}, we require the value of the bit $x_{(j-\alpha)\%n}^{T-1}$ as well as the carry bit c_j. Since we know that the initial carry $c_0 = 0$, we flip the bit of y^{T-1} starting from position 0. By doing so we get the value of $x_{(0-\alpha)\%n}^{T-1}$ and hence the value of c_1. Subsequently we flip the bit of y^{T-1} in position 1 and retrieve next bit. In this way we start from the least significant bit of y^{T-1} and proceed to the most significant bit until all bits of the last round key are recovered. The attack procedure is as follows:

1. Flip the j^{th} bit in the input of y^{T-1} so that it results in a faulty ciphertext (x_T^*, y_T^*). Initially $j = 0$. The xor of correct and faulty computation of the output x_T can be written as:

$$x^T \oplus x^{T^*} = (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus (S^{-\alpha}(x^{T-1}) + y^{(T-1)^*}) \tag{28}$$

   We can write the j^{th} bit of xor as:

$$\begin{aligned} (x^T \oplus x^{T^*})_j &= (x_{(j-\alpha)\%n}^{T-1} \oplus y_j^{T-1} \oplus c_j) \oplus (x_{(j-\alpha)\%n}^{T-1} \oplus (y_j^{T-1} \oplus 1) \oplus c_j) \\ \therefore (x^T \oplus x^{T^*})_j &= 1 \end{aligned} \tag{29}$$

   It should be emphasized here, that a flip in the bit y_j^{T-1} not only changes the j^{th} bit in the output of modular addition but can also affect the carry-out bit c_{j+1}. Let us denote the carry-out bit by c_{j+1}^* when the bit y_j^{T-1} is flipped. If $c_{j+1} \neq c_{j+1}^*$, then due to rippling effect of carry the next l bits in x^{T^*} are also affected. In general, we can write:

$$(x^T \oplus x^{T^*})_m = \begin{cases} 1, & (m = j) \text{ or } (m > j \text{ and } c_m \neq c_m^*) \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

   where $m \in \{j, \ldots, j + l\}$.

2. Now, based on this observation and the number of differences $\#1(x^T \oplus x^{T^*})$ in the xor of x_T and x_T^*, we can derive the corresponding value of bit $x_{(j-\alpha)\%n}^{T-1}$ as shown in Table 5. It can be observed that,

$$x_{(j-\alpha)\%n}^{T-1} = \begin{cases} c_j, & \#1(x^T \oplus x^{T^*}) = 1 \\ \neg c_j, & \text{otherwise} \end{cases} \tag{31}$$

   Since the value of carry-in bit c_j is known, the value of the bit $x_{(j-\alpha)\%n}^{T-1}$ can be deduced. Now, as we know the values of c_j, $x_{(j-\alpha)\%n}^{T-1}$ and y_j^{T-1}, the value of carry-out bit c_{j+1} can be found which is used for retrieving $x_{(j+1-\alpha)\%n}^{T-1}$ in

Table 5. Deducing the value of bit $x^{T-1}_{(j-\alpha)\%n}$ and carry-in bit c_j from $\#1(x^T \oplus x^{T^*})$ and bit y^{T-1}_j.

$\#1(x^T \oplus x^{T^*})$	$x^{T-1}_{(j-\alpha)\%n}$	y^{T-1}_j	$y^{T-1}_j \oplus 1$	c_j
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	0	1
>1	1	0	1	0
>1	1	1	0	0
>1	0	0	1	1
>1	0	1	0	1

the next iteration of the attack.
Also, if $l > 1$, we can write:

$$\begin{aligned} &(x^T \oplus x^{T^*})_p = 1 \\ \therefore\ &(x^T \oplus x^{T^*})_p = (x^{T-1}_{(p-\alpha)\%n} \oplus y^{T-1}_p \oplus c_p) \oplus (x^{T-1}_{(p-\alpha)\%n} \oplus y^{T-1}_p \oplus c_j) \oplus 1 \\ \therefore\ &(x^T \oplus x^{T^*})_p = (x^{T-1}_{(p-\alpha)\%n} \oplus y^{T-1}_p \oplus c_p) \oplus (x^{T-1}_{(p-\alpha)\%n} \oplus (y^{T-1}_p \oplus 1) \oplus c_j) \end{aligned} \tag{32}$$

where $p \in \{j+1, \ldots, j+l\}$. This equation is similar to equation (29) given in step 1 of this attack procedure. Therefore, we can repeat the step 2 for l more times and retrieve l more bits of x^{T-1} *viz.*, $x^{T-1}_{(j+1-\alpha)\%n}$ to $x^{T-1}_{(j+l-\alpha)\%n}$.

3. Now we can use equation (27) to retrieve the bits of last round key k^{T-1}.
   if $(l = 0)$
   $\quad j^{th}$ bit of k^{T-1} can be recovered
   else
   $\quad l+1$ bits k^{T-1}_j to k^{T-1}_{j+l} can be recovered

4. $j = j\ +\ l$
   if$(j = n)$ break
   goto step 1

Attack Complexity. A single bit-flip in the intermediate state y^{T-1} reveals at least one bit of x^{T-1} and therefore at least one bit of k^{T-1}. However, as explained above, more than one bit of x^{T-1} can be retrieved depending upon the number of carry bits which are flipped due to the faulty bit. The probability of the carry bit being flipped is (1/2) and therefore the probability of obtaining one more bit of x^{T-1} is also (1/2). In general, the probability of obtaining l more bits of x^{T-1} is equal to the probability of l carry bits getting flipped due to a single bit flip in y^{T-1}. This is because for l^{th} carry bit to be flipped all the lower $(l-1)$ carry bits should be flipped. Therefore the expected number of bits of last round key that can be retrieved using a single bit-flip is:

$$1 + \sum_{z=1}^{l} z * Pr[z] = 1 + \sum_{z=1}^{l} z * \frac{1}{2^z} \approx 3 \tag{33}$$

Thus a bit-flip recovers three bits of last round key. Therefore the average number of bit faults required to recover all the n bits of last round key k^{T-1} is $(n/3)$.

4 Conclusion

In this paper, we have described the first fault attack on the families of SIMON and SPECK ciphers. In SIMON, we have exploited the information leaked by the AND operation used during the computation of x^{T-1} (which is equal to y^T). We observed that if a bit u is flipped in the input x^{T-2}(which is equal to y^{T-1}), then the bit v of x^{T-2} which is ANDed with u, can be retrieved. Since the flipped bit u is ANDed in two different positions in the computation of x^{T-1}, we can deduce two bits of x^{T-2}, which are used to derive the corresponding bits of the last round key k^{T-1}. This principle is used to mount both the bit-flip and random byte fault attack. The average number of bit faults to retrieve the n-bits of k^{T-1} is $(n/2)$. If a random byte fault model is used, the number of byte faults required is $(n/8)$.

In SPECK, we have exploited the information leaked by the modular addition used during the computation of x^T. Here, we flip the bits of y^{T-1}, beginning from its least significant bit. We observed that if the faulty computation differs from the correct computation in l bits, then we can deduce l bits of the last round key k^{T-1}. Therefore the average number of bit faults to retrieve the n-bits of k^{T-1} is $(n/3)$.

References

1. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. Available at http://eprint.iacr.org/
2. H. A. Alkhzaimi and M. M. Lauridsen. Cryptanalysis of the SIMON Family of Block Ciphers. Cryptology ePrint Archive, Report 2013/543, 2013. Avaliable at http://eprint.iacr.org/
3. F. Abed, E. List, S. Lucks, and J. Wenzel. Differential Cryptanalysis of Reduced-Round Simon. Cryptology ePrint Archive, Report 2013/526, 2013. Available at http://eprint.iacr.org/.
4. Javad Alizadeh, Nasour Bagheri, Praveen Gauravaram, Abhishek Kumar and Somitra Kumar Sanadhya. Linear Cryptanalysis of Round Reduced SIMON. IACR Cryptology eprint Archive, Report 2013/663, 2013. Available at http://eprint.iacr.org/2013/663