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. Currently, linear and differential cryptanalytic results for SIMON are available in the literature but no fault attacks have been reported so far on these two cipher families. In this paper, we show that these families of ciphers are vulnerable to differential fault attacks. Specifically, we demonstrate two fault attacks on SIMON and one fault attack on SPECK. The first attack on SIMON assumes a bit-flip fault model and recovers the $\boldsymbol{n}$-bit last round key of SIMON using $\boldsymbol{n/2}$ bit faults. The second attack on SIMON uses a more practical, random byte fault model and requires $\boldsymbol{n/8}$ faults on average to retrieve the last round key. The attack presented on SPECK also assumes a bit-flip fault model and recovers the $\boldsymbol{n}$-bit last round key of SPECK using $\boldsymbol{n/3}$ bit faults on average.

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 optimally in the software environments. In order to provide implementation on a wide range of devices, both SIMON and SPECK support 5 block sizes of 32, 48, 64, 96 and 128 bits and upto 3 key sizes for each block size. The design requirements and performance analysis of the SIMON and SPECK family were published by the US National Security Agency (NSA) in 2013 [1], but security assessment of these ciphers was not provided. Though, the initial results of linear and differential cryptanalysis of the SIMON family are now available in [2], [3] and [4], 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.

Contribution. 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

[⋆] harshal.coep@gmail.com

[⋆⋆] s.bisht09@gmail.com

[⋆⋆⋆] debdeep.mukhopadhyay@gmail.com

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. Further, we show that SIMON is also vulnerable to a 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 the 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.
$\mathbf{T}$: Total number of rounds in the cipher.
$(\mathbf{x^{i+1}}, \mathbf{y^{i+1}})$: The $2n$ bit output of the i^{th} round of the cipher, $i \in \{0, \ldots, T-1\}$. Input to the cipher is denoted by (x^0, y^0).
$(\mathbf{x^{(i+1)^*}}, \mathbf{y^{(i+1)^*}})$: The $2n$ bit **faulty** output of the i^{th} round of the cipher, $i \in \{0, \ldots, T-1\}$.
$\mathbf{k^i}$: The n bit round key used in the i^{th} round of the cipher, $i \in \{0, \ldots, T-1\}$.
$\boldsymbol{S^{-\alpha}(w)}$: 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 $w_{n-1}w_{n-2}\ldots w_1 w_0$, 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. First, we describe a fault attack on SIMON, 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 SPECK 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 of 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 SIMON is a function $R_k : GF(2^n) \times GF(2^n) \rightarrow GF(2^n) \times GF(2^n)$ defined as:

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

where $i \in \{0, \ldots, T-1\}$ and $f(x^i) = (S^1(x^i) \text{ \& } S^8(x^i)) \oplus S^2(x^i)$.
It can be seen from the definition of $f(x^i)$ that its l^{th} bit is computed using 3 distinct bits of x^i.

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

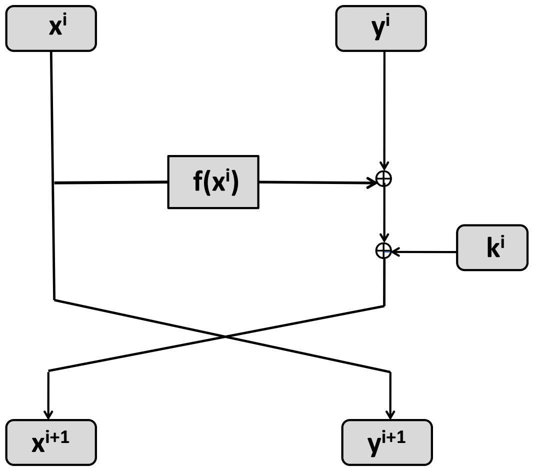

Fig. 1. i^{th} Round of SIMON

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 $f(x^i)$.

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

where $j \in \{0 \ldots n-1\}$. 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.

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

The fault attacks that we describe later, exploit the information leaked by the AND operation used in the computation of a round function.
In the following discussion, we show that the secrecy of the last round key k^{T-1} relies completely upon the secrecy of the left half input x^{T-2} of the penultimate round.

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 discussions, we describe the 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 $y^{i+1} = x^i$, $i \in \{0, \ldots, T-1\}$ 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 and position of the 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

In the attack, we exploit the information leaked from the use of AND operation in the computation of $f(x^{T-2})$. We observe that if one of the input bits of the AND operation is 0 then flipping the other input bit does not affect the output bit of y^T. Therefore, we can deduce the bit of x^{T-2} and consequently retrieve the bit of k^{T-1} using equation (7). The attack details are given below:

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.

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

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

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

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

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

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):

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

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.

2.5 Simulation Results

Using C, we simulated the bit-flip fault attack on different members of the SIMON family. We assumed that the attacker is able to identify and target the left half input x^{T-2} of the penultimate round using a side-channel, but has no control over the fault position. However, the attacker can deduce the value and position of the induced fault using equation (8) and retrieve the bits of the last round key k^{T-1} as described in the attack. In the experiment, we obtained faulty encryptions until all the n bits of k^{T-1} are retrieved. For every value of n, we repeated the experiment 1000 times and obtained the average number of faulty encryptions required to recover k^{T-1}. Table 3 shows the number of faulty encryptions required to recover the n-bit last round key k^{T-1}. Since no control over the bit-flip position is assumed, the faults can affect the same bit position more than once. Hence, the number of faulty encryptions required to obtain k^{T-1} is more than $n/2$. However, if a precise control over the fault position is assumed then the number of faulty encryptions required is close to the estimated value $n/2$.

Table 3. Bit-flip Fault Attack on SIMON Assuming no Control Over the Fault Position.

n bits	k^{T-1}	Avg. No. of Faulty Encryptions
16	0xfa 0x24	25
24	0x26 0x53 0xaf	43
32	0x87 0x46 0x09 0x1a	62
48	0x22 0x4d 0xe9 0xcf 0x51 0xdd	104
64	0x19 0x26 0x5a 0xc7 0x4f 0xf2 0x90 0x01	150

2.6 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 byte fault are 1 then each of these bits can retrieve only one key bit as shown below: Suppose, a fault flips the bits x_j^{T-2} and x_{j-7}^{T-2}. A flip in x_j^{T-2} affects 3 bits of y_T:

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

A flip in x_{j-7}^{T-2} also affects 3 bits of y_T:

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

Similar to the bit fault attack, we expect to retrieve the two bits $x_{(j-7)\%n}^{T-2}$ and $x_{(j+7)\%n}^{T-2}$ from the equation set (14), and the two bits $x_{(j-14)\%n}^{T-2}$ and x_j^{T-2} from the equation set (15). However, one can see from Table 4 that using the value of $(y^T \oplus y^{T^*})_{(j+1)\%n}$, the j^{th} and $((j-7)\%n)^{th}$ bits cannot be retrieved. This is because in this case, both the input bits x_j^{T-2} and $x_{(j-7)\%n}^{T-2}$ of the AND operation used in the computation of $y_{(j+1)\%n}^{T^*}$ are flipped. We can only determine whether the bits x_j^{T-2} and $x_{(j-7)\%n}^{T-2}$ are complement of each other or they have the same value. The actual value of either x_j^{T-2} or $x_{(j-7)\%n}^{T-2}$ cannot be known. Thus in this case, only two bits: $x_{(j+7)\%n}^{T-2}$ from equation set (14) and $x_{(j-14)\%n}^{T-2}$ 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 of x^{T-2} reveals two bits of the last round key.

Table 4. Relation between the bits x_j^{T-2} and $x_{(j-7)\%n}^{T-2}$.

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

2. The attack procedure also differs slightly when a byte fault flips two contiguous bits x_j^{T-2} and x_{j-1}^{T-2}. In this case, a flip in x_j^{T-2} affects 3 bits of y^T:

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

And, a flip in x_{j-1}^{T-2} also affects 3 bits of y_T:

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

Table 5. Deducing the value of bit $x_{(j-7)\%n}^{T-2}$ from $(y^T \oplus y^{T^*})_{(j+1)\%n}$.

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

From Table 5 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 1, otherwise it is 0. This is because in this case, the input bit $x^{T-2}_{(j-1)\%n}$ of the xor operation used in the computation of $y^{T^*}_{(j+1)\%n}$ is also flipped.

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

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 the bit-flip fault attack.

2.7 Attack Complexity

A byte fault of Hamming weight z in x^{T-2} retrieves $2z$ bits of the last round key k^{T-1} except for the case where the most and the least significant bits of the byte fault are 1. In this case, $2z-2$ bits of the last round key are retrieved. The number of possible byte faults having Hamming weight z is $\binom{8}{z}$ and there are 64 byte faults where the least and the most significant bits are 1. If we assume that every byte fault of Hamming weight z retrieves $2z$ bits then the expected number of key bits that can be retrieved by a random byte fault is:

$$\sum_{z=1}^{8} 2z * Pr[z] = \frac{1}{255} * \left(\sum_{z=1}^{8} 2z * \binom{8}{z} \right)$$

But, if the least and most significant bits of the byte fault having Hamming weight z are 1, then $2z-2$ key bits are retrieved. Since there are 64 such faults, we subtract 2*64 from the above expression to obtain:

$$\frac{1}{255} * \left(\left(\sum_{z=1}^{8} 2z * \binom{8}{z} \right) - 128 \right) \approx 8 \tag{19}$$

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

2.8 Simulation Results

We also simulated the random byte fault attack on different members of the SIMON family. As done in the previous simulation, we assumed that the attacker is able to identify and target the left half input x^{T-2} of the penultimate round using a side-channel, but has no control over the fault position. However, the attacker can deduce the value and position of the induced fault using equation (8) and retrieve the bits of the last round key k^{T-1} as described in the attack. In the experiment, we obtained faulty encryptions until all the n bits of k^{T-1} are retrieved. For every value of n, we repeated the experiment 1000 times and obtained the average number of faulty encryptions required to recover k^{T-1}. Table 6 shows the number of faulty encryptions required to recover the n-bit last round key k^{T-1}. Since no control over the fault position is assumed, the faults can affect the same bit position more than once. Hence, the number of faulty encryptions is more than $n/8$. However, if a precise control over the fault position is assumed then the number of faulty encryptions required is close to the estimated value $n/8$.

Table 6. Random Byte Fault Attack on SIMON Assuming no Control Over the Fault Position.

n bits	k^{T-1}	Avg. No. of Faulty Encryptions
16	0xfa 0x24	6
24	0x26 0x53 0xaf	9
32	0x87 0x46 0x09 0x1a	13
48	0x22 0x4d 0xe9 0xcf 0x51 0xdd	21
64	0x19 0x26 0x5a 0xc7 0x4f 0xf2 0x90 0x01	30

3 Fault Attack on SPECK

Similar to the attack description of SIMON, we begin by describing the characteristics of the round function of 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 SPECK is a

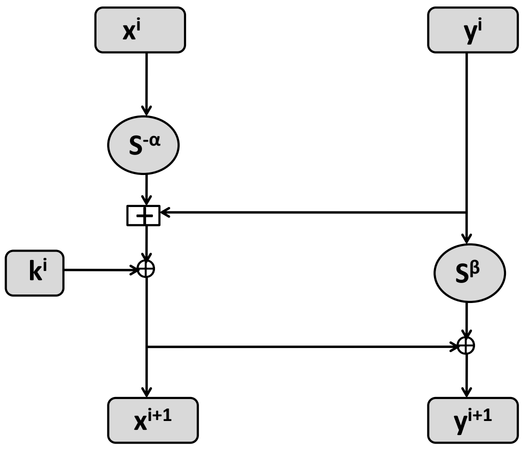

Fig. 2. i^{th} Round of SPECK

function $R_k : GF(2^n) \times GF(2^n) \rightarrow GF(2^n) \times GF(2^n)$ defined as

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

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 $c_j = (x^i_{(j-1+\alpha)\%n} \ \& \ y^i_{j-1}) \ |(y^i_{j-1} \ \& \ c_{j-1}) \ | \ (x^i_{(j-1+\alpha)\%n} \ \& \ c_{j-1})$, $c_0 = 0$ and $j \in \{0, \ldots, n-1\}$.
The fault attack that we describe later, exploits the information leaked by the modular addition operation used in the computation of a round function.
In the following discussion, we show that the secrecy of the last round key relies completely upon the secrecy of the left half input x^{T-1} of the final round.

3.2 Equation of the last round key

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

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

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}

$$k^{T-1}_j = (\ x^{T-1}_{(j+\alpha)\%n} \oplus (y^T \oplus x^T)_{(j+\beta)\%n} \oplus c_j \ ) \oplus x^T_j \tag{25}$$

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^*}).

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

Since we know the output of correct and faulty computation, we can deduce the value and position of the 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,

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

From the above equation it can be seen that the key bit k_j^{T-1} can be recovered if the values of the bit $x_{(j+\alpha)\%n}^{T-1}$ and carry bit c_j are known. Since the value of initial carry $c_0 = 0$ is known, we show that a flip in the bit of y^{T-1} at position 0 reveals the value of $x_{(0+\alpha)\%n}^{T-1}$ and therefore the bit k_0^{T-1} can be retrieved. Now, as we have the value of bits $x_{(0+\alpha)\%n}^{T-1}$, c_0 and y_0^{T-1}, we can also deduce the value of carry-out c_1. Subsequently, we flip the next bit of y^{T-1}, *i.e.* y_1^{T-1}, to retrieve the key bit k_1^{T-1} and repeat this process, until all bits of the last round key are recovered. The attack details are given below:

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:

   $$x^T \oplus x^{T^*} = (S^{-\alpha}(x^{T-1}) + y^{T-1}) \oplus (S^{-\alpha}(x^{T-1}) + y^{(T-1)^*}) \quad (27)$$

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

   $$\begin{aligned}&(x^T \oplus x^{T^*})_j = (x_{(j+\alpha)\%n}^{T-1} \oplus y_j^{T-1} \oplus c_j) \oplus (x_{(j+\alpha)\%n}^{T-1} \oplus (y_j^{T-1} \oplus 1) \oplus c_j)\\ &\therefore (x^T \oplus x^{T^*})_j = 1\end{aligned} \quad (28)$$

   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 the rippling effect of carry the next l bits in x^{T^*} are also affected. In general, we can write:

   $$(x^T \oplus x^{T^*})_m = \begin{cases} 1, & (m = j) \text{ or } (m > j \text{ and } c_m \neq c_m^*) \\ 0, & \text{otherwise} \end{cases} \quad (29)$$

   where $m \in \{j, \ldots, j+l\}$ and $0 \leq l < n$.

2. Now, based on 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 $x_{(j+\alpha)\%n}^{T-1}$ from Table 7. There are two cases which can be observed from this table:
   (a) If $\#1(x^T \oplus x^{T^*}) = 1$, it implies that the carry- out bit is not flipped, *i.e.* $c_{j+1} = c_{j+1}^*$. In this case, irrespective of the value of bit y_j^{T-1}, $x_{(j+\alpha)\%n}^{T-1} = c_j$.
   (b) If $\#1(x^T \oplus x^{T^*}) > 1$, it implies that the carry-out bit is also flipped, *i.e.* $c_{j+1} \neq c_{j+1}^*$. In this case, irrespective of the value of bit y_j^{T-1}, $x_{(j+\alpha)\%n}^{T-1} = \neg c_j$.

Table 7. Deducing the value of bit $x^{T-1}_{(j+\alpha)\%n}$ from $\#1(x^T \oplus x^{T^*})$ and carry-in bit c_j .

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

Therefore, we can write:

$$x^{T-1}_{(j+\alpha)\%n} = \begin{cases} c_j, & \#1(x^T \oplus x^{T^*}) = 1 \\ \neg c_j, & \text{otherwise} \end{cases} \tag{30}$$

Since the value of carry-in bit c_j is known, the value of the bit $x^{T-1}_{(j+\alpha)\%n}$ can be deduced. Now, as we know the values of c_j, $x^{T-1}_{(j+\alpha)\%n}$ and y_j^{T-1}, the value of carry-out bit c_{j+1} can be found which is used for retrieving $x^{T-1}_{(j+1+\alpha)\%n}$ in the next iteration of the attack.
Also, if $l > 0$, we can write:

$$\begin{aligned} &(x^T \oplus x^{T^*})_p = 1 \\ \therefore\ &(x^T \oplus x^{T^*})_p = (x^{T-1}_{(p+\alpha)\%n} \oplus y_p^{T-1} \oplus c_p) \oplus (x^{T-1}_{(p+\alpha)\%n} \oplus y_p^{T-1} \oplus c_p) \oplus 1 \\ \therefore\ &(x^T \oplus x^{T^*})_p = (x^{T-1}_{(p+\alpha)\%n} \oplus y_p^{T-1} \oplus c_p) \oplus (x^{T-1}_{(p+\alpha)\%n} \oplus (y_p^{T-1} \oplus 1) \oplus c_p) \end{aligned} \tag{31}$$

where $p \in \{j+1, \ldots, j+l\}$. This equation is similar to equation (28) 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 $x^{T-1}_{(j+l+\alpha)\%n}$. Hence, it is not required to perform the bit-flip fault attack on the next l bits of y^{T-1}.

3. Now we can use equation (25) to retrieve the bits of last round key k^{T-1}.
   if $(l = 0)$
      j^{th} bit of k^{T-1} can be recovered
   else
      $l + 1$ bits k_j^{T-1} to k_{j+l}^{T-1} can be recovered

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

3.5 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}. For l^{th} carry bit to be flipped, all the lower $(l-1)$ carry bits should also be flipped. The probability of this event is $1/2^l$. Therefore, the expected number of bits of last round key that can be retrieved using a single bit-flip is:

$$1 + \sum_{z=1}^{l} z * Pr[z] = 1 + \sum_{z=1}^{l} z * \frac{1}{2^z} \approx 3 \tag{32}$$

Thus, a bit-flip recovers three bits of the 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 $(n/3)$.

3.6 Simulation Results

We simulated the bit-flip fault attack on different members of the SPECK family using C. We assumed that the attacker is able to identify and target the right half input y^{T-1} of the final round using a side-channel, but has no control over the fault position. However, the attacker can deduce the value and position of the induced fault using equation (26).

In the attack described above, we observed that using the carry-in bit $c_0 = 0$, a flip in the least significant bit of y^{T-1} can be used to obtain the bit of x_α and subsequently the higher bits of y^{T-1} can be flipped in a sequential manner. But in the experiment, we assumed no control over the bit flip position and therefore, we continued to obtain the faulty encryptions until all the n bits of k^{T-1} are retrieved. For every value of n, we repeated the experiment 1000 times and obtained the average number of faulty encryptions required to recover k^{T-1}. Table 8 shows the number of faulty encryptions required to recover the n-bit last round key k^{T-1}. Since no control over the bit-flip position is assumed, the faults can affect the same bit-position more than once. Hence, the number of faulty encryptions required to obtain k^{T-1} is more than $n/3$. However, if a precise control over the fault position is assumed then the number of faulty encryptions required is close to the estimated value $n/3$.

Table 8. Bit-flip Fault Attack on SPECK Assuming no Control Over the Fault Position.

n bits	k^{T-1}	Avg. No. of Faulty Encryptions
16	0xfa 0x24	18
24	0x26 0x53 0xaf	25
32	0x87 0x46 0x09 0x1a	44
48	0x22 0x4d 0xe9 0xcf 0x51 0xdd	85
64	0x19 0x26 0x5a 0xc7 0x4f 0xf2 0x90 0x01	114

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 $f(x^{T-2})$. This information can be observed through the right half y^T of the final output. We observed that if a bit u is flipped in the input x^{T-2}, 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 then 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 required to retrieve the n bits of k^{T-1} is $(n/2)$. If a random byte fault is induced, then the number of key bits which are retrieved, depends upon the Hamming weight of the induced fault. In this case, the average number of byte faults required to retrieve the n-bit last round key 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, since we require the value of carry-in bit. We observed that if the faulty output x^{T^*} differs from the correct output x^T 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. 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. http://eprint.iacr.org/.
3. F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential and Linear Cryptanalysis of Reduced-Round SIMON." Cryptology ePrint Archive, Report 2013/526, 2013. http://eprint.iacr.org/.
4. J. Alizadeh, N. Bagheri, P. Gauravaram, A. Kumar, and S. K. Sanadhya, "Linear Cryptanalysis of Round Reduced SIMON." Cryptology ePrint Archive, Report 2013/663, 2013. http://eprint.iacr.org/.