electronics

MDPI

Article

A Novel Differential Fault Analysis on the Key Schedule of SIMON Family

Jinbao Zhang [1], Ning Wu [1], Fang Zhou [1,*], Muhammad Rehan Yahya [1] and Jianhua Li [2]

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; zjb4050811@126.com (J.Z.); wunee@nuaa.edu.cn (N.W.); rehanyahya@yahoo.com (M.R.Y.)

[2] Institute of Computer Application, Taizhou University, Linhai 317000, China; ljh2007@tzc.edu.cn

* Correspondence: zfnuaa@nuaa.edu.cn; Tel.: +86-189-1395-7622

Received: 11 December 2018; Accepted: 10 January 2019; Published: 15 January 2019

Abstract: As a family of lightweight block ciphers, SIMON has attracted lots of research attention since its publication in 2013. Recent works show that SIMON is vulnerable to differential fault analysis (DFA) and existing DFAs on SIMON assume the location of induced faults are on the cipher states. In this paper, a novel DFA on SIMON is proposed where the key schedule is selected as the location of induced faults. Firstly, we assume a random one-bit fault is induced in the fourth round key K^{T-4} to the last. Then, by utilizing the key schedule propagation properties of SIMON, we determine the exact position of induced fault and demonstrate that the proposed DFA can retrieve 4 bits of the last round key K^{T-1} on average using one-bit fault. Till now this is the largest number of bits that can be cracked as compared to DFAs based on random bit fault model. Furthermore, by reusing the induced fault, we prove that 2 bits of the penultimate round key K^{T-2} could be retrieved. To the best of our knowledge, the proposed attack is the first one which extracts a key from SIMON based upon DFA on the key schedule. Finally, correctness and validity of our proposed attack is verified through detailed simulation and analysis.

Keywords: SIMON; fault attack (FA); lightweight block ciphers; differential fault analysis (DFA)

1. Introduction

In 2013, a family of lightweight block ciphers called SIMON was presented by the National Security Agency (NSA), based upon the Feistel structure. Compared with other ciphers, SIMON can provide a better performance for both hardware and software. The block size of SIMON is denoted as $2n$ (the n represents the word size) with n = 16, 24, 32, 48, or 64. For each block size, it supports 3 key sizes. Thus, SIMON can be implemented on a wide range of devices [1]. Since the publication of SIMON, many cryptanalysis papers about it have been presented, such as integral attack [2,3], differential attack [4–6], and linear attack [6,7]. In addition, other attacks, such as differential fault analysis (DFA), have also been proposed to retrieve the secret keys from SIMON [8].

As one of the typical fault attacks (FA) [9], DFA was first proposed by Biham and Shamir in 1997 to obtain the secret key from DES cryptosystem [10]. The idea of DFA is to make use of some erroneous calculations caused by inducing some unexpected faults to retrieve the secret keys of a cipher algorithm. DFA has been greatly developed and poses a serious threat to the security of many cipher algorithms, including block cipher algorithms [8,11–13].

In FDTC 2014, Tupsamudre et al. proposed DFA on SIMON family for the first time [8]. In this attack, the authors induced the faults into L^{T-2} (the left half input of the penultimate round) and proposed two fault models: random bit fault model and random byte fault model. Through theoretical analysis and experiments, they proved that it could retrieve 2 bits and one byte of the last

round key K^{T-1} by using one random bit-flip and one-byte fault, respectively. Later, Takahashi et al. proposed a random n-bit fault model against the SIMON family where the n represents the word size. They successfully retrieved the entire key of SIMON family through both, that is, theoretical computations and experimental simulations. For the data complexity (the data complexity refers to the number of the fault injections), they also presented a detail analysis [14].

After that, Vasquez et al. proposed an improved DFA on SIMON family [15]. Similarly to [8], they also assumed a random bit fault model. However, the location of induced fault is in L^{T-3}. Because more depth of induced fault lead to more efficient diffusion of the induced fault, this scheme can retrieve 3.5 bits of K^{T-1} on average by inducing one-bit fault. Furthermore, by reusing the induced one-bit fault, 2 bits of the penultimate round key K^{T-2} on average could be retrieved. As a result, they could break the entire key of SIMON96/96 and SIMON128/128 using only one round faults.

In a following work, another improved DFA on SIMON family was presented by Chen et al. in FDTC 2016 [16]. The authors injected faults into L^{T-m-1} based on a random byte fault model, where the m represents the key words of SIMON family. They presented a detail analysis about the data complexity in theory and shown that the entire key of SIMON could be recovered. For retrieving the entire secret key of SIMON, they successfully break 6 instances of SIMON by using only one round faults.

This paper proposes a novel DFA on SIMON family. Different from existing DFAs on SIMON family where faults are induced into the cipher state, we induce faults into the key schedule for the first time. Based on a random bit fault model, we prove that 4 bits of K^{T-1} and 2 bits of K^{T-2} could be retrieved on average when inducing only one-bit fault into the fourth round key K^{T-4} to the last. Compared to [8], which also uses random bit fault model, we can recover the entire key of SIMON family through half number of the fault locations. Compared to the previous works, our contributions in this paper are mainly as following:

1. Selection of the key schedule as the location of induced fault. Different from these existing DFAs on SIMON family ([8,14–16]) where all select the cipher state as the location of induced fault, our DFA on SIMON is the first one which selects the key schedule as the location of induced fault. Thus, we have provided a new train of thought and method for using DFA to crack keys of the SIMON family.
2. Compared with existing attacks based on the random bit fault model, our attack is more efficient. For the random bit fault model, paper [15] is the only one which could retrieve two round keys by using one induced round location. In other words, paper [15] can retrieve on average 3.5 bits of K^{T-1} and 2 bits of K^{T-2} by using one-bit fault induced into the $(T-3)$th round. Up to now, this is the most efficient method. However, selection of the key schedule especially K^{T-4} as the location of induced fault, our attack can retrieve 4 bits of K^{T-1} and 2 bits of K^{T-2} on average using one-bit fault.

The rest of this paper is arranged as follows. Section 2 presents some necessary notation and a brief introduction for SIMON. Then Section 3 proposes and discusses our DFA on SIMON key schedule. In this section, we present the assumption of the proposed attack, then discuss how to determine the position of the induced fault and retrieve K^{T-1} as well as K^{T-2}. Extended analysis includes the detailed data complexity assessment and scheme to crack the entire secret key of the SIMON family. Simulation results and comparisons are carried out in Section 4. Finally, concluding remarks are given in Section 5.

2. Preliminaries

2.1. Notation

- T: the round number of SIMON
- m: the key word size in SIMON
- n: the word size in SIMON

- P: plaintext
- C, C *: ciphertext and faulty ciphertext
- L^i, R^i: the left and right half input of the ith round, $i \in \{0, \ldots, T-1\}$, the output of the cipher is denoted by (L^T, R^T)
- L_j^i, R_j^i: the jth bit of L^i, R^i
- L^i *, R^i *: the left and right half faulty input of the ith round, $i \in \{0, \ldots, T-1\}$
- K^i: round -key used in ith round, $i \in \{0, \ldots, T-1\}$
- K^i *: faulty K^i when there is a fault in K^i
- $x <<< a$: left circular shift of x by a bits
- $y >>> b$: right circular shift of y by b bits
- &: bitwise AND
- $\oplus$: bitwise xor
- $a\%b$: a remainder b

2.2. Description for SIMON Family

As a lightweight block cipher, SIMON applies a Feistel structure with a n-bit word and a m-bit word key, which is denoted as SIMON $2n/mn$. In the SIMON family, n should be 16, 24, 32, 48, or 64, and $m = 2$, 3, or 4. The parameters of the SIMON family with different (n, m) combinations are described in Table 1.

Table 1. Parameters of the SIMON family.

Cipher SIMON$2n/mn$	Block Size $2n$	Key Size mn	Word Size n	Key Words m	Const Seq	Rounds T
SIMON 32/64	32	64	16	4	z_0	32
SIMON 48/72	48	72	24	3	z_0	36
SIMON 48/96		96		4	z_1	36
SIMON 64/96	64	96	32	3	z_2	42
SIMON 64/128		128		4	z_3	44
SIMON 96/96	96	96	48	2	z_2	52
SIMON 96/144		144		3	z_3	54
SIMON 128/128	128	128	64	2	z_2	68
SIMON 128/192		192		3	z_3	69
SIMON 128/256		256		4	z_4	72

2.2.1. Key Schedule Function

The SIMON key schedule generates a sequence of T key from an input key, where T is the round number. For SIMON 2 n/mn, the T key words $(K^0, \ldots, K^{T-1})$ depend on the value of m and it can be generated using the formulas (1), where c is a constant value and $c = 2^n - 4 = 0\text{xff}\ldots\text{fc}$. The z_j represents 5 constant sequences denoted as z_0, z_1, z_2, z_3 and z_4, respectively. More detailed descriptions about the key schedule function and z_j can be obtained in [1].

$$\begin{aligned} m = 2:\ K^i = {} & c \oplus (z_j)_{i-m} \oplus K^{i-2} \oplus \\ & (K^{i-1} >>> 3) \oplus (K^{i-1} >>> 4) \\ m = 3:\ K^i = {} & c \oplus (z_j)_{i-m} \oplus K^{i-3} \oplus \\ & (K^{i-1} >>> 3) \oplus (K^{i-1} >>> 4) \\ m = 4:\ K^i = {} & c \oplus (z_j)_{i-m} \oplus K^{i-4} \oplus K^{i-3} \oplus \\ & (K^{i-3} >>> 1) \oplus (K^{i-1} >>> 3) \\ & \oplus (K^{i-1} >>> 4) \end{aligned} \tag{1}$$

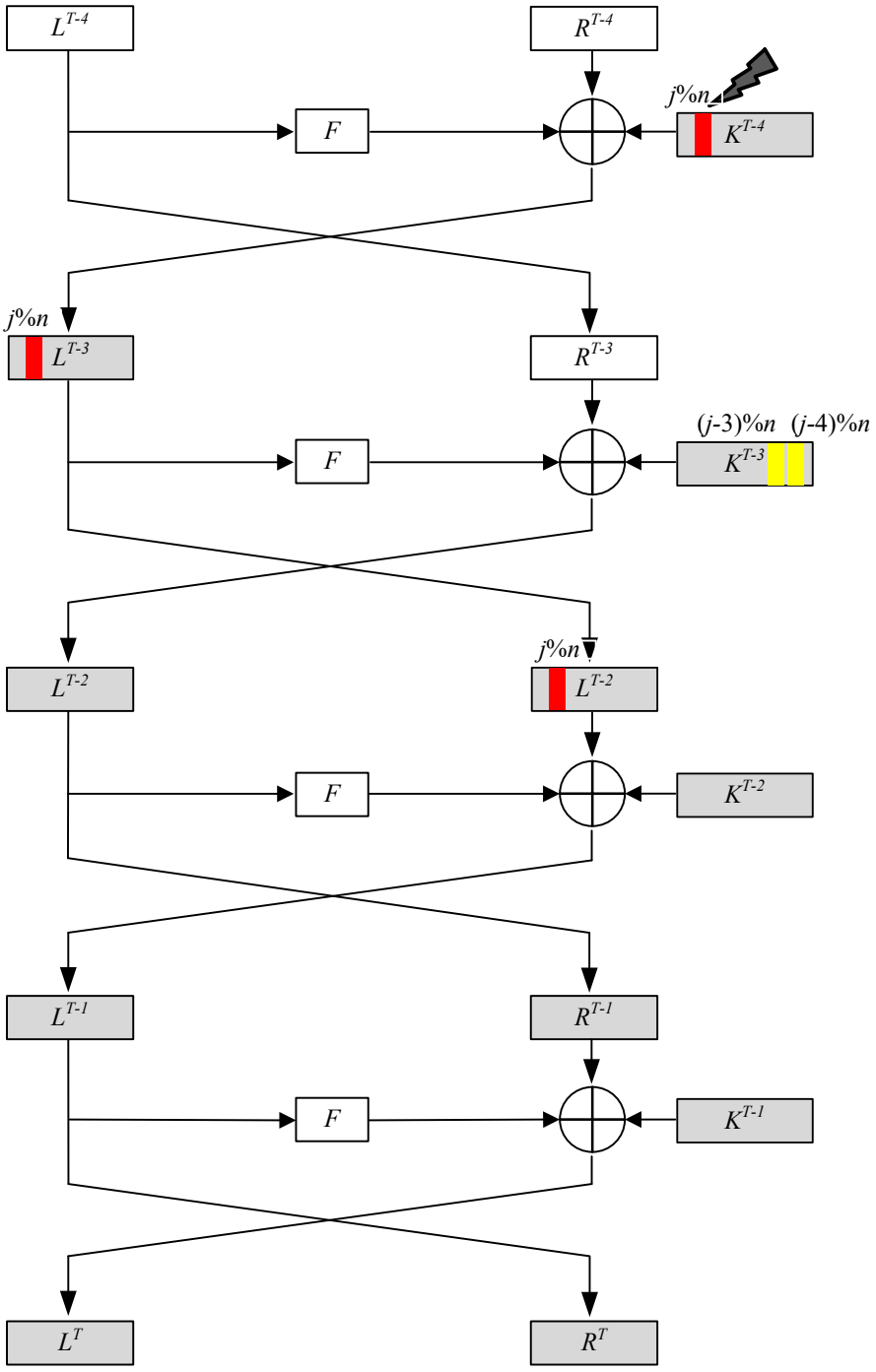

Figure 1. Fault propagation when the jth bit K^{T-4} is randomly corrupted.

2.2.2. Round Function

The SIMON round function $R_k(L^i, R^i)$: $\mathrm{GF}(2)^n \times \mathrm{GF}(2)^n \rightarrow \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$ is defined as:

$$R_k(L^i, R^i) = (L^{i+1}, R^{i+1}) = (R^i \oplus F(L^i) \oplus K^i, L^i)$$

where

$$F(L^i) = (L^i <<< 1) \& (L^i <<< 8) \oplus (L^i <<< 2) \quad (2)$$

for $i \in \{0, \ldots, T-1\}$. From (2), it can be known that the jth bit of L^i affects 3 distinct bits of $F(L^i)$:

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

3. The Proposed Attack on SIMON Key Schedule

3.1. Assumption of the Proposed Attack

Different from these existing DFAs on SIMON, we assume the adversary induces a random one-bit fault into the key schedule, and the exact position of the induced fault is in K^{T-4}. (L^T *, R^T *) is denoted as the faulty output when inducing fault. K^{T-4} is randomly corrupted by a random one-bit fault, the fault propagation process is as shown in Figure 1.

In Figure 1, the red thick line in K^{T-4} represents the induced $j\%n$ bit of K^{T-4}. Both the red thick line in L^{T-3} and R^{T-2} represent the corrupted bits. The two yellow thick lines in K^{T-3} represent the bit $(j-3)\%n$ and $(j-4)\%n$ *of* K^{T-3} respectively, which are all corrupted by the $j\%n$ bit of K^{T-4}. And the thick lines show the cipher states and round keys which are necessary to retrieve K^{T-1} and K^{T-2}. The gray in cipher states and round keys represent the faulty intermediate states and faulty round keys, respectively.

3.2. DFA on The (T−4) Round Key

In FDTC 2014, Tupsamudre et al. proposed the following formula to retrieve K^{T-1}:

$$K^{T-1} = L^{T-2} \oplus F(R^T) \oplus L^T \tag{4}$$

Thus, in order to make use of the induced random bit faults in K^{T-4} to retrieve K^{T-1}, we need to establish the relationship between induced faults and L^{T-2}. We suppose the position of induced fault is the jth bit of K^{T-4}. From Figure 1, it can be deduced that:

$$\begin{aligned} L^{T*} &= F(L^{T-1*}) \oplus K^{T-1*} \oplus R^{T-1*} \\ &= F(R^{T*}) \oplus K^{T-1*} \oplus L^{T-2*} \\ &= F(R^{T*}) \oplus K^{T-1*} \oplus K^{T-3*} \oplus F(L^{T-3*}) \oplus R^{T-3} \end{aligned} \tag{5}$$

Therefore, we can derive the following equation from the xor of L^T and L^T *:

$$\begin{aligned} L^T \oplus L^{T*} = \; & F(R^T) \oplus F(R^{T*}) \oplus K^{T-1} \oplus K^{T-1*} \\ & \oplus K^{T-3} \oplus K^{T-3*} \oplus F(L^{T-3}) \oplus F(L^{T-3*}) \end{aligned} \tag{6}$$

If we move $F(R^T) \oplus F(R$ *$)$ in the Equation (6) to the left side, the Equation (6) can be rewritten as:

$$\begin{aligned} L^T \oplus L^{T*} \; & \oplus \; F(R^T) \oplus F(R^{T*}) \\ & = K^{T-1} \oplus K^{T-1*} \oplus K^{T-3} \oplus K^{T-3*} \\ & \oplus F(L^{T-3}) \oplus F(L^{T-3*}) \end{aligned} \tag{7}$$

3.2.1. Determining the Position of Induced Fault

In this part, we will show how to determine the position of the induced fault based on Equation (7). From Figure 1, the induced bit fault in K^{T-4} that will corrupt the same position bit in L^{T-3} can be known, in other words, the jth bit of L^{T-3} is flipped. According to the Formula (3), we can identify that 3 distinct bits of $F(L^{T-3})$ may be affected: $(j+1)\%n$, $(j+2)\%n$ and $(j+8)\%n$. To further illustrate the

affected bits, the function $F(.)$ defined in Equation (2) needs further analysis. Assuming the jth bit of L^i is induced, according to the Formula (3), it can be deduced that:

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

From the Equation (8), we can know that once the jth bit of L^i is flipped, the $(j+2)\%n$ bit of $F(L^i)$ is also flipped. Due to the jth bit of L^{T-3} is flipped, it can be identified that the two bits $(j+1)\%n$ and $(j+8)\%n$ of $F(L^{T-3})$ may be affected, and the bit $(j+2)\%n$ of $F(L^{T-3})$ must be affected.

According to the Formula (1) and the principle of the SIMON key schedule, we can obtain the following equation no matter the value of m (that is $m = 2$, 3, or 4):

$$\begin{aligned} K^{T-3} \oplus K^{T-3*} &= (K^{T-4} >>> 3) \oplus (K^{T-4*} >>> 3) \\ &\oplus (K^{T-4} >>> 4) \oplus (K^{T-4*} >>> 4) \end{aligned} \tag{9}$$

Equation (9) shows that one fault bit of K^{T-4} will affect 2 distinct bits of K^{T-3}. In other words, the jth bit of K^{T-4} affects the bits $(j-3)\%n$ and $(j-4)\%n$ of K^{T-3}.

$$\begin{aligned} \left(K^{T-3} \oplus K^{T-3*}\right)_{(j-3)\%n} &= K^{T-4} \oplus (K^{T-4} \oplus 1) = 1 \\ \left(K^{T-3} \oplus K^{T-3*}\right)_{(j-4)\%n} &= K^{T-4} \oplus (K^{T-4} \oplus 1) = 1 \end{aligned} \tag{10}$$

Further, the bit $(j-3)\%n$ of K^{T-3} will affect the bits $(j-6)\%n$ and $(j-7)\%$ of K^{T-2}, the bit $(j-4)\%n$ of K^{T-3} will affect the bits $(j-7)\%n$ and $(j-8)\%$ of K^{T-2}. As a result, the bits $(j-3)\%n$ and $(j-4)\%n$ of K^{T-3} will affect the bits $(j-6)\%n$ and $(j-8)\%$ of K^{T-2}. Through similar analysis, we can deduce the affected bits in K^{T-1}. The jth bit of K^{T-4} affects the bits of K^{T-3}, K^{T-2} and K^{T-1} are given in Table 2: For simplicity, $(j-x)\%n$ writes as $j-x$, where $x \in \{0, \ldots, n\}$.

Table 2. The jth bit of K^{T-4} affects the bits of K^{T-3}, K^{T-2} and K^{T-1}.

The Position of Induced Fault	Key Words: m	Affected Bits
jth bit of K^{T-4}	4	K^{T-3}: $j-3, j-4$
		K^{T-2}: $j-6, j-8$
		K^{T-1}: $j, j-1, j-9, j-10, j-11, j-12$
	3	K^{T-3}: $j-3, j-4$
		K^{T-2}: $j-6, j-8$
		K^{T-1}: $j, j-9, j-10, j-11, j-12$
	2	K^{T-3}: $j-3, j-4$
		K^{T-2}: $j, j-6, j-8$
		K^{T-1}: $j-9, j-10, j-11, j-12$

By combining the Equation (7), Table 2 and the analysis above, it can be identified some bits value in $(L^T \oplus L^{T^*} \oplus F(R^T) \oplus F(RT^*))$. For convenience, we write $(L^T \oplus L^{T^*} \oplus F(R^T) \oplus F(RT^*))$ as "LFR", thus when key words $m = 4$ (only take $m = 4$ for example, when $m = 3$ or 2, the processes of discussion remain the same), we can get:

$$\begin{aligned}
&\mathrm{LFR}_{j\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-3)\%n} = K^{T-3} \oplus K^{T-3} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-4)\%n} = K^{T-3} \oplus K^{T-3} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-1)\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-9)\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-10)\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-11)\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j-12)\%n} = K^{T-1} \oplus K^{T-1} \oplus 1 = 1 \\
&\mathrm{LFR}_{(j+2)\%n} = L_j^{T-3} \oplus L_j^{T-3} \oplus 1 = 1,
\end{aligned} \tag{11}$$

as well as the following equations:

$$\begin{aligned}
\mathrm{LFR}_{(j+1)\%n} = \quad & (L_j^{T-3} \& L_{(j-7)\%n}^{T-3}) \\
& \oplus ((L_j^{T-3} \oplus 1) \& L_{(j-7)\%n}^{T-3}) \\
\mathrm{LFR}_{(j+8)\%n} = \quad & (L_{(j+7)\%n}^{T-3} \& L_j^{T-3}) \\
& \oplus (L_{(j+7)\%n}^{T-3} \& (L_j^{T-3} \oplus 1))
\end{aligned} \tag{12}$$

Through the Equation (11), it can be seen that 4 contiguous bits $(j-9)\%n$, $(j-10)\%n$, $(j-11)\%n$ and $(j-12)\%n$ of LFR are all 1. In fact, when $m = 3$, or 2, there are all 4 consecutive bits $(j-9)\%n$, $(j-10)\%n$, $(j-11)\%n$ and $(j-12)\%n$ be 1 in LFR. We present statistics on the value of bits in LFR under different conditions in Table 3.

Table 3. The jth bit of K^{T-4} affects the bits of LFR.

The Position of Induced Fault	Key Words: m	The Value of Bits in LFR(LFR = $L^T \oplus L^T$ *$\oplus F(R^T) \oplus F(RT^*)$)
jth bit of K^{T-4}	4	1: $j, j-1, j-3, j-4, j-9, j-10, j-11, j-12, j+2$
		may be 1: $j+1, j+8$
	3	1: $j, j-3, j-4, j-9, j-10, j-11, j-12, j+2$
		may be 1: $j+1, j+8$
	2	1: $j-3, j-4, j-9, j-10, j-11, j-12, j+2$
		may be 1: $j+1, j+8$

As can be seen in Table 3, there exists only one group of 4 contiguous 1 in LFR no matter if $m = 4$, 3 or 2, and this is a very important property. In fact, the idea for deducing the position j is based on this property.

To determine the position of induced fault, Algorithm 1 has been proposed. Here, the value of constant A depends on the word size n: (A, n) = {(F400, 16), (F40000, 24), ... }. $F(.)$ represents the non-linear function defined in Equation (2). The position of j can be determined by Algorithm 1, in other words, we can accurately determine the position of induced fault: jth bit of K^{T-4}.

Algorithm 1 Deducing the position of induced fault j in K^{T-4}

Input: (L^T, R^T), $(L^T *, R^T *)$, the word size n, constant A
Output: deducing j.

```
1. LFR ←L^T⊕L^T*⊕F(R^T)⊕F(RT*)
2. for i = 0: n − 1
3.     B ← circshift(A,[0,−i]) % right circular shift for binary
4.     C ← LFR & B
5.     if ( C == A)
6.         if (i <8)
7.             j%n ← n-8 + i
8.         else
9.             j%n ← i - 8
10.        end
11.    end
12. end
```

3.2.2. Retrieving L^{T-2} and K^{T-1}

According to the formula (4), it is known that if someone wants to retrieve K^{T-1}, she/he must obtain L^T, R^T and L^{T-2} first. Because (L^T, R^T) is the output of SIMON which could be obtained directly, so she/he only needs to obtain L^{T-2}. From Figure 1, it can be deduced that:

$$R^{T*} = F(L^{T-2*}) \oplus K^{T-2*} \oplus L^{T-3*} \tag{13}$$

Therefore, the following equation could be derived from the xor of R^T and $R^T *$:

$$\begin{aligned} R^T \oplus R^{T*} = F(L^{T-2}) \oplus F(L^{T-2*}) \oplus K^{T-2} \oplus K^{T-2*} \\ \oplus L^{T-3} \oplus L^{T-3*} \end{aligned} \tag{14}$$

From the discussion in Section 3.2.1, it is known that when the jth bit of K^{T-4} is flipped, the jth bit of L^{T-3} will also be flipped, so the bit $(j+2)\%n$ of $F(L^{T-3})$ is flipped. Because L^{T-2*} can be written as:

$$L^{T-2*} = F(L^{T-3}) \oplus K^{T-3*} \oplus R^{T-3}, \tag{15}$$

and the $(j-3)\%n$ and $(j-4)\%n$ of K^{T-3} are flipped (this conclusion can be obtained from Table 2), so there must be 3 bits which are flipped in L^{T-2}: $(j+2)\%n$, $(j-3)\%n$ and $(j-4)\%n$. Thus, combined the Equation (14) and the results from Table 2, the following equations could be obtained:

$$\begin{aligned} (R^T \oplus R^{T*})_{(j+4)\%n} &= (L^{T-2}_{(j+3)\%n} \,\&\, L^{T-2}_{(j-4)\%n}) \\ &\oplus (L^{T-2}_{(j+3)\%n} \,\&\, (L^{T-2}_{(j-4)\%n} \oplus 1)) \oplus 1 \\ (R^T \oplus R^{T*})_{(j+5)\%n} &= (L^{T-2}_{(j+4)\%n} \,\&\, L^{T-2}_{(j-3)\%n}) \\ &\oplus (L^{T-2}_{(j+4)\%n} \,\&\, (L^{T-2}_{(j-3)\%n} \oplus 1)) \\ (R^T \oplus R^{T*})_{(j-2)\%n} &= (L^{T-2}_{(j-3)\%n} \,\&\, L^{T-2}_{(j-10)\%n}) \\ &\oplus ((L^{T-2}_{(j-3)\%n} \oplus 1) \,\&\, L^{T-2}_{(j-10)\%n}) \oplus 1 \\ (R^T \oplus R^{T*})_{(j-3)\%n} &= (L^{T-2}_{(j-4)\%n} \,\&\, L^{T-2}_{(j-11)\%n}) \\ &\oplus ((L^{T-2}_{(j-4)\%n} \oplus 1) \,\&\, L^{T-2}_{(j-11)\%n}) \end{aligned} \tag{16}$$

Through the Equation (16), we have made truth tables to illustrate the relationships between the bits in L^{T-2} and the bits in $(R^T \oplus R^{T*})$.

From Table 4, it can be seen that if $(R^T \oplus R^{T*})_{(j+4)\%n} = 0$, then $L^{T-2}_{(j+3)\%n} = 1$, otherwise it is 0. This is independent of the value of $L^{T-2}_{(j-4)\%n}$. Besides, from Tables 5–7, it can be seen that if $(R^T \oplus R^{T*})_{(j+5)\%n} = 0$, then $L^{T-2}_{(j+4)\%n} = 0$, otherwise it is 1; If $(R^T \oplus R^{T*})_{(j-2)\%n} = 0$, then $L^{T-2}_{(j-10)\%n} = 1$, otherwise it is 0; If $(R^T \oplus R^{T*})_{(j-3)\%n} = 0$, then $L^{T-2}_{(j-11)\%n} = 0$, otherwise it is 1. As a result, we can obtain the following equations:

$$\begin{aligned} L^{T-2}_{(j+3)\%n} &= (R^T \oplus R^{T*})_{(j+4)\%n} \oplus 1 \\ L^{T-2}_{(j+4)\%n} &= (R^T \oplus R^{T*})_{(j+5)\%n} \\ L^{T-2}_{(j-10)\%n} &= (R^T \oplus R^{T*})_{(j-2)\%n} \oplus 1 \\ L^{T-2}_{(j-11)\%n} &= (R^T \oplus R^{T*})_{(j-3)\%n} \end{aligned} \tag{17}$$

Therefore, there are 4 bits $(j+3)\%n$, $(j+4)\%n$, $(j-10)\%n$ and $(j-11)\%n$ of K^{T-1} that could be recovered according to the Equations (17) and (4):

$$\begin{aligned} K^{T-1}_{(j+3)\%n} &= L^{T-2}_{(j+3)\%n} \oplus F(R^T)_{(j+3)\%n} \oplus L^T_{(j+3)\%n} \\ K^{T-1}_{(j+4)\%n} &= L^{T-2}_{(j+4)\%n} \oplus F(R^T)_{(j+4)\%n} \oplus L^T_{(j+4)\%n} \\ K^{T-1}_{(j-10)\%n} &= L^{T-2}_{(j-10)\%n} \oplus F(R^T)_{(j-10)\%n} \oplus L^T_{(j-10)\%n} \\ K^{T-1}_{(j-11)\%n} &= L^{T-2}_{(j-11)\%n} \oplus F(R^T)_{(j-11)\%n} \oplus L^T_{(j-11)\%n} \end{aligned} \tag{18}$$

Table 4. The relationship between the bit $L^{T-2}_{(j+3)\%n}$ and $(R^T \oplus R^{T*})_{(j+4)\%n}$.

$L^{T-2}_{(j+3)\%n}$	$L^{T-2}_{(j-4)\%n}$	$L^{T-2}_{(j-4)\%n} \oplus 1$	$(R^T \oplus R^{T*})_{(j+4)\%n}$
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0

Table 5. The relationship between the bit $L^{T-2}_{(j+4)\%n}$ and $(R^T \oplus R^{T*})_{(j+5)\%n}$.

$L^{T-2}_{(j+4)\%n}$	$L^{T-2}_{(j-3)\%n}$	$L^{T-2}_{(j-3)\%n} \oplus 1$	$(R^T \oplus R^{T*})_{(j+5)\%n}$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	1

Table 6. The relationship between the bit $L^{T-2}_{(j-10)\%n}$ and $(R^T \oplus R^{T*})_{(j-2)\%n}$.

$L^{T-2}_{(j-10)\%n}$	$L^{T-2}_{(j-3)\%n}$	$L^{T-2}_{(j-3)\%n} \oplus 1$	$(R^T \oplus R^{T*})_{(j-2)\%n}$
0	0	1	1
0	1	0	1
1	0	1	0
1	1	0	0

Table 7. The relationship between the bit $L^{T-2}_{(j-11)\%n}$ and $(R^T \oplus R^{T*})_{(j-3)\%n}$.

$L^{T-2}_{(j-11)\%n}$	$L^{T-2}_{(j-4)\%n}$	$L^{T-2}_{(j-4)\%n} \oplus 1$	$(R^T \oplus R^{T*})_{(j-3)\%n}$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	1

3.2.3. Retrieving K^{T-2}

After retrieving K^{T-1}, we can reuse L^{T-2} with similar operations to those described in Section 3.2.2 to retrieve K^{T-2}.

Now we obtain L^{T-2}, because $R^{T-1} = L^{T-2}$ and $L^{T-1} = R^T$, in other words the output of the $T-2$ round of SIMON: (L^{T-1}, R^{T-1}) is obtained. As shown in Figure 1, because $K^{T-2} = F(R^{T-1}) \oplus L^{T-1} \oplus L^{T-3}$, if L^{T-3} is known, then K^{T-2} could be recovered. Based on the Equation (12), there are 2 bits $(j-7)\%n$ and $(j+7)\%n$ of L^{T-3} could be deduced, therefore two bits of K^{T-2} can be recovered when inducing one-bit fault in K^{T-4}.

3.3. Extended Analysis

According to the Equation (18), we can obtain 4 bits of K^{T-1} which could be retrieved in theory by inducing one-bit fault in K^{T-4}. Furthermore, 2 bits of K^{T-2} could be retrieved on an average by reusing the induced one-bit fault. When considering the random bit fault model, paper [8] can recover only 2 bits of K^{T-1}. Although paper [15] can also recover 2 bits of K^{T-2}, but it can only recover 3.5 bits of K^{T-1} on average, Thus, our proposed one-bit fault attack for SIMON is more efficient.

As in the similar discussion in [15], when $m = 2$, the whole keys of SIMON (96/96 and 128/128) might also be retrieved by using only one round key faults. Indeed, for $m = 2$ and $k = i - 2$, according to the Formula (1), we can obtain:

$$\begin{aligned} m = 2: \; K^k = {} & c \oplus (z_j)_{k-m+2} \oplus K^{k+2} \oplus \\ & (K^{k+1} >>> 3) \oplus (K^{k+1} >>> 4) \end{aligned} \tag{19}$$

So, for $m = 2$, by using only continuous two round keys of SIMON, the entire key of SIMON could be retrieved. As discussed in Sections 3.2.2 and 3.2.3, K^{T-1} and K^{T-2} can be retrieved by inducing faults in K^{T-4}; thus we can retrieve the whole keys of SIMON (96/96 and 128/128) using only one round key faults. Similarly, considering $m = 3$ and $m = 4$, the following equations by the Formula (1) could be obtained:

$$\begin{aligned} m = 3: \; K^k = {} & c \oplus (z_j)_{i-3+m} \oplus K^{k+3} \oplus \\ & (K^{k+2} >>> 3) \oplus (K^{k+2} >>> 4) \\ m = 4: \; K^k = {} & c \oplus (z_j)_{i-4+m} \oplus K^{k+4} \oplus K^{k+1} \oplus \\ & (K^{k+1} >>> 1) \oplus (K^{k+3} >>> 3) \\ & \oplus (K^{k+3} >>> 4) \end{aligned} \tag{20}$$

From the Equation (20), for $m = 3$, we know that if we can obtain K^{T-2} and K^{T-3}, then the entire key of SIMON could be recovered. Therefore, we need to induce faults in two round keys: K^{T-4} and K^{T-6}. The first inducing faults in K^{T-4} are used to retrieve both K^{T-1} and K^{T-2}, the second inducing faults in K^{T-6} are used to only retrieve K^{T-3}. For $m = 4$, we also need to induce faults in two round keys: K^{T-4} and K^{T-6}. However, it is different from the case when $m = 3$, in this case, we need to induce faults in K^{T-6} to retrieve two round keys: K^{T-3} and K^{T-4}. The key retrieving for SIMON family and specific fault locations are shown in Tables 8–10.

Table 8. Experiment results for the average number of the fault inductions to retrieve K^{T-1}/L^{T-2}.

Word Size: n	Avg. No. of the Fault Inductions				At least No. of the Fault Inductions in Theory:			
	Random Byte Fault Model	Random Bit Fault Model			Random Byte Fault Model	Random Bit Fault Model		
	[8]	[8]	[15]	This paper	[8]: $n/8$	[8]: $n/2$	[15]: $n/3.5$	This Paper: $n/4$
16	6	25	15.26	11.12	2	8	4.57	4
24	9	43	29.70	19.82	3	12	6.86	6
32	13	62	44.19	29.25	4	16	9.14	8
48	21	104	77.02	49.67	6	24	13.71	12
64	30	150	110.81	71.61	8	32	18.29	16

Table 9. Comparison of the experimental results of the fault inductions.

SIMON 2n/mn	Avg. No. of the Fault Inductions					
	Random Byte Fault Model		Random n-Bit Fault Model	Random Bit Fault Model		
	[8]	[16]	[14]	[8]	[15]	This Paper
SIMON32/64	24		12.20	101.72	50.85	50.32
SIMON48/72	27		9.91	130.78	87.19	62.78
SIMON48/96	36		13.22	174.37	87.19	85.86
SIMON64/96	39	31.57	10.45	189.44	126.29	91.57
SIMON64/128	52		13.93	252.58	126.29	124.72
SIMON96/96	42	35.08	7.46	210.24	105.12	104.00
SIMON96/144	63	50.84	11.19	315.36	210.24	153.64
SIMON128/128	60	50.55	7.82	299.68	149.84	148.51
SIMON128/192	90	72.88	11.73	449.52	299.68	220.15
SIMON128/256	120	104.82	15.64	599.36	299.68	297.02

Table 10. Comparison of the fault locations of differential fault analysis (DFA) on SIMON family.

SIMON2 n/mn	Fault Locations of DFA on SIMON Family					
	Random Byte Fault Model		Random n-Bit Fault Model	Random Bit Fault Model		
	[8]	[16]	[14]	[8]	[15]	This Paper
SIMON32/64	$L^{27}, L^{28}, L^{29}, L^{30}$	L^{27}	$L^{27}, L^{28}, L^{29}, L^{30}$	$L^{27}, L^{28}, L^{29}, L^{30}$	L^{27}, L^{29}	K^{26}, K^{28}
SIMON48/72	L^{32}, L^{33}, L^{34}	L^{32}	L^{32}, L^{33}, L^{34}	L^{32}, L^{33}, L^{34}	L^{32}, L^{33}	K^{30}, K^{32}
SIMON48/96	$L^{31}, L^{32}, L^{33}, L^{34}$	L^{31}	$L^{31}, L^{32}, L^{33}, L^{34}$	$L^{31}, L^{32}, L^{33}, L^{34}$	L^{31}, L^{33}	K^{30}, K^{32}
SIMON64/96	L^{38}, L^{39}, L^{40}	L^{38}	L^{38}, L^{39}, L^{40}	L^{38}, L^{39}, L^{40}	L^{38}, L^{39}	K^{36}, K^{38}
SIMON64/128	$L^{39}, L^{40}, L^{41}, L^{42}$	L^{39}	$L^{39}, L^{40}, L^{41}, L^{42}$	$L^{39}, L^{40}, L^{41}, L^{42}$	L^{39}, L^{41}	K^{38}, K^{40}
SIMON96/96	L^{49}, L^{50}	L^{49}	L^{49}, L^{50}	L^{49}, L^{50}	L^{49}	K^{48}
SIMON96/144	L^{50}, L^{51}, L^{52}	L^{50}	L^{50}, L^{51}, L^{52}	L^{50}, L^{51}, L^{52}	L^{50}, L^{51}	K^{48}, K^{50}
SIMON128/128	L^{65}, L^{66}	L^{65}	L^{65}, L^{66}	L^{65}, L^{66}	L^{65}	K^{64}
SIMON128/192	L^{65}, L^{66}, L^{67}	L^{65}	L^{65}, L^{66}, L^{67}	L^{65}, L^{66}, L^{67}	L^{65}, L^{66}	K^{63}, K^{65}
SIMON128/256	$L^{67}, L^{68}, L^{69}, L^{70}$	L^{67}	$L^{67}, L^{68}, L^{69}, L^{70}$	$L^{67}, L^{68}, L^{69}, L^{70}$	L^{67}, L^{69}	K^{66}, K^{68}

4. Simulation Results and Comparisons

In this section, we performed simulations to verify the correctness and validity of our proposed attack using C on a personal computer with a HP Intel® Core i5-7300HQ. We assumed the adversary can induce a random one-bit fault in the round keys of SIMON.

Firstly, we randomly chose a set of plaintext and secret key to obtain the correct ciphertext (L^T, R^T). Then we induced random one-bit faults in K^{T-4} to obtain the faulty ciphertexts (L^T *, R^T *). Finally, we used the proposed attack to retrieve K^{T-1}. For different word size n = 16, 24, 32, 48, and 64, we carried out simulation experiments respectively. The experiment was repeated 100,000 times for every value of n same as in [14–16]. Experimental results show that the proposed attack can recover the n-bit K^{T-1} successfully. For different n, the average number of the fault inductions are described in Table 8. For retrieving the entire keys of the SIMON family, we make some comparisons with existing DFAs on SIMON family, as described in Tables 9 and 10.

As shown in Table 8, when considering the random bit fault model, our proposed attack needs the least average number of fault inductions. Compared with the number of fault inductions in theory, the average number of fault inductions is much more, this is because we assume the position of the induced fault is random, so the faults can affect the same position many times. However, if we control precisely the position of induced faults, then the average number of fault inductions is very close to the number in theory.

As shown in Tables 9 and 10, comparing the fault locations between the proposed attack and existing ones, our proposed attack is the only one which selects the round keys as the fault locations. When considering the random bit model, the proposed attack requires only half number of the fault locations compared with [8], and the numbers are as the same required in [15]. Except the random n-bit model [14] and the random byte model [16], the number of the fault inductions required in the proposed attack is least than required in [8,15]. Especially, when the key words $m = 3$, the number required in the proposed attack is much less than required in [15], this is because we induce faults in K^{T-6} to retrieve only K^{T-3} instead of retrieving both K^{T-3} and K^{T-4}, so our proposed attack is more efficient.

5. Conclusions

This paper proposes a novel DFA on SIMON family by exploiting the leaked information by the AND operation used in the $F(L^{T-3})$. We show how to retrieve 4 bits of K^{T-1} and 2 bits of K^{T-2} on average based on only one-bit fault induced in K^{T-4}. Furthermore, we have proved that the entire key of SIMON96/96 and SIMON128/128 could be retrieved by using only one round faults.

Compared with existing works, the proposed attack in this paper is the first one which selects the SIMON key schedule as the location of induced faults. Considering the random bit fault model, our attack is the most efficient one up to data. When considering the random n-bit model [14] and random byte model [16], our attack requires a higher average number of fault inductions; this is because the different fault models are selected.

In the future, we will try to crack more bits by conferring whether the bit $(j + 1)\%$ and $(j + 8)\%$ of L^{T-2} have been flipped. Further, we aim to explore the attack based upon different models such as the random n-bit model and random byte model so as to further reduce the required average number of fault inductions. Besides, how to apply our ideas on the block cipher SPECK will also be explored.

Author Contributions: Conceive and structure of the concept of this paper, J.Z.; Resources, F.Z. and N.W.; Supervision, N.W.; Writing-original draft, J.Z.; Writing-review and editing, M.R.Y. and J.L.

Funding: This work was supported by National Natural Science Foundation of China (Nos. 61774086, 61376025), Natural Science Foundation of Jiangsu Province (No. BK20160806), Funding of Jiangsu Innovation Program for Graduate Education (No. KYLX16_0373), Research project of education department of Zhejiang province (No. Y201840245).

Acknowledgments: The authors would like to thank Xiaoqiang Zhang and Qiangjia Bi for their beneficial suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Beaulieu, R.; Shors, D.; Smith, J.; Treatman-Clark, S.; Weeks, B.; Wingers, L. The SIMON and SPECK Families of Lightweight Block Ciphers. *Cryptology ePrint Archive*. Report 2013/404. 2013. Available online: http://eprint.iacr.org/ (accessed on 11 December 2018).
2. Wang, Q.; Liu, Z.; Varıcı, K.; Sasaki, Y.; Rijmen, V.; Todo, Y. Cryptanalysis of Reduced Round SIMON32 and SIMON48. In *Progress in Cryptology–INDOCRYPT 2014*; ser. LNCS; Meier, W., Mukhopadhyay, D., Eds.; Springer International Publishing: New York, NY, USA, 2014; Volume 8885, pp. 143–160.
3. Zhang, H.; Wu, W.; Wang, Y. Integral Attack Against Bit-Oriented Block Ciphers. In *Information Security and Cryptology-ICISC 2015*; ser. LNCS; Kwon, S., Yun, A., Eds.; Springer International Publishing: New York, NY, USA, 2015; Volume 9558, pp. 102–118.

4. Abed, F.; List, E.; Lucks, S.; Wenzel, J. Differential Cryptanalysis of Round-Reduced SIMON and SPECK. In *Fast Software Encryption*; ser. LNCS; Cid, C., Rechberger, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8540, pp. 525–545.
5. Wang, S. Related-key differential analysis of round-reduced Simeck. In Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud-I-SMAC, Palladam, India, 10–11 February 2017; pp. 834–838.
6. Abed, F.; List, E.; Lucks, S.; Wenzel, J. Differential and Linear Cryptanalysis of Reduced-Round SIMON. *Cryptology ePrint Archive*. Report 2013/526. 2013. Available online: http://eprint.iacr.org/ (accessed on 11 December 2018).
7. Alizadeh, J.; Bagheri, N.; Gauravaram, P.; Kumar, A.; Sanadhya, S.K. Linear Cryptanalysis of Round Reduced SIMON. *Cryptology ePrint Archive*. Report 2013/663. 2013. Available online: http://eprint.iacr.org/ (accessed on 11 December 2018).
8. Tupsamudre, H.; Bisht, S.; Mukhopadhyay, D. Differential Fault Analysis on the families of SIMON and SPECK ciphers. In Proceedings of the International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2014), Busan, Korea, 23 September 2014; IEEE Computer Society: Washington, DC, USA, 2014; pp. 40–48.
9. Karaklajić, D.; Schmidt, J.M.; Verbauwhede, I. Hardware designer's guide to fault attacks. *IEEE Trans. Very Large Scale Integr. Syst.* **2013**, *21*, 2295–2306. [CrossRef]
10. Biham, E.; Shamir, A. Differential Fault Analysis of Secret Key Cryptosystems. In *Advances in Cryptology-CRYPTO'97*; ser. LNCS; Kaliski, B.S., Jr., Ed.; Springer International Publishing: New York, NY, USA, 1997; Volume 1294, pp. 513–525.
11. Chong, H.K. Improved Differential Fault Analysis on AES Key Schedule. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 41–50.
12. Huo, Y.; Zhang, F.; Feng, X.; Wang, L. Improved Differential Fault Attack on the Block Cipher SPECK. In Proceedings of the International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2015), Saint Malo, France, 13 September 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 28–34.
13. Hemme, L. A Differential Fault Attack against Early Rounds of (triple-) DES. In *International Workshop on Cryptographic Hardware and Embedded Systems-(CHES 2004)*; ser. LNCS; Joye, M., Quisquater, J.-J., Eds.; Springer International Publishing: New York, NY, USA, 2004; Volume 3156, pp. 254–267.
14. Takahashi, J.; Fukunaga, T. Fault Analysis on SIMON Family of Lightweight Block Ciphers. In *Information Security and Cryptology-ICISC 2014*; ser. LNCS; Lee, J., Kim, J., Eds.; Springer International Publishing: New York, NY, USA, 2014; Volume 8949, pp. 175–189.
15. Vasquez, J.D.C.G.; Borges, F.; Portugal, R.; Lara, P. An Efficient One-Bit Model for Differential Fault Analysis on Simon Family. In Proceedings of the International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2015), Saint Malo, France, 13 September 2015; IEEE Computer Society: Washington, DC, USA, 2015; pp. 61–70.
16. Chen, H.; Feng, J.; Rijmen, V.; Liu, Y.; Fan, L.; Li, W. Improved Fault Analysis on SIMON Block Cipher Family. In Proceedings of the International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2016), Santa Barbara, CA, USA, 16 August 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 16–24.