2015 Workshop on Fault Diagnosis and Tolerance in Cryptography

An Efficient One-Bit Model for Differential Fault Analysis on Simon Family

Juan del Carmen Grados Vásquez*, Fábio Borges†, Renato Portugal*, and Pedro Lara‡
*Laboratório Nacional de Computação Científica
Petrópolis, Brasil
juancgv@lncc.br, portugal@lncc.br
†Technische Universität Darmstadt, CASED – Telecooperation Lab
DE-64289 Darmstadt, Germany
fabio.borges@cased.de
‡CEFET-RJ
Petrópolis, Brasil
pedro.lara@cefet-rj.br

***Abstract*—In this paper, we describe a family of symmetric cryptographic algorithms and present its cryptanalysis. Specifically, we use differential fault analysis to show a fault attack threat to the block cipher family named Simon. In addition, we present the improvement of a fault attack based on a differential attack method. Moreover, we are the first to show how to extract the entire secret key using only one round. This property is important because an attacker has to control the hardware to inject faults. However, if the attacker has control of only few hardware components and they compute only one round, previous attacks are not able to recover the entire key. With this side-channel analysis, an attacker can inject faults in one round of Simon with block of 96 or 128 bits to recover the respective entire key of 96 or 128 bits without using SAT solver neither computing Gröbner bases. The key can be recovered using only differential fault analysis.**

***Keywords*-Simon family, side-channel analysis, cryptanalysis, fault attack, symmetric cryptography, algorithms.**

I. Introduction

Simon is a family of lightweight block ciphers based upon Feistel structure. This family was designed by the National Security Agency (NSA) with the goal of having a better performance for both hardware and software in comparison to other ciphers, which are currently available in the field. Its design provides optimal performance on resource-constrained devices. To provide implementation on a wide range of devices, Simon supports 5 block sizes of 32, 48, 64, 96, 128 bits and up to 3 key sizes for each block size.

In hostile environment, device-embedded cryptographic algorithms are susceptible to the so-called physical attacks, namely Side Channel Attack (SCA). To recover the key, SCA makes use of the physical leakages emanating from a device (power consumption, electromagnetic radiations, etc.). Differential Fault Analysis (DFA) is a type of SCA, proposed by Biham and Shamir in [1]. The principle is to induce environmental conditions with unexpected faults into cryptographic implementations to reveal their internal states.

DFA on block ciphers has two major phases. In the fault injection phase, adversaries inject faults to the selected positions. In the fault analysis phase, adversaries analyze the differences between the correct and faulty outputs to extract the secret key. Throughout this paper, the term fault position refers to the round where faults are injected. In contrast, the term fault location refers to the bit or byte index for the injected fault in a specified round. Examples of applications of DFA on block ciphers have been proposed [2]–[6].

In [7], Tupsamudre *et al.* presented the first DFA on the Simon family. Their authors proposed two models to retrieve the last-round key. The first one is a one-bit-flip model and the second is a random one-byte model. In both models an adversary is able to inject a fault, flipping one bit, or one byte, in an intermediate position of Simon. After several fault injections these models retrieve the last-round key. In a following work, [8] improved the results of [7] in two points: (1) [8] performed a detailed analysis when half of the bits of the block size in an intermediate position are randomly changed by fault injection instead of a bit, or a random byte, (2) [8] obtained the entire key of the Simon members. Hereafter, we refer to the model presented in [8] as the random n-bit model.

A. Our Contributions

In this paper, we present a novel modification of the DFA introduced in [7]. With this modification, it is possible to retrieve the entire key by using half of fault positions in comparison with [7], [8]. Our modification is explained in the Section III. For instance, to retrieve the entire key of Simon with block and key sizes 128, we need the half number of fault injections on average and a single fault position compared to the one-bit-flip model proposed in [7], which uses two fault positions. The same happens when the Simon has block and key sizes 96.

978-1-4673-7579-5/15 $31.00 © 2015 European Union
DOI 10.1109/FDTC.2015.18

B. High Level Explanation of our Attack

The goal of our modification is to recover the entire key of Simon by modifying the method presented in [7]. The modification consists in changing the intermediate position for fault injections. We will explain how the models presented in [7] can be used when the fault position is changed. In addition, we discuss what happen in Simon members when two intermediate fault positions are available for fault injections.

C. Summary of our Results

Section III describes how to retrieve the entire key of Simon with key sizes 96 and 128, and block sizes 96 and 128. There are two advantages in our approach compared to the attacks presented in [7] to retrieve the entire key. The first advantage is that our method needs a single fault position for fault injections compared to two fault positions needed in the one-bit-flip model in [7]. The second advantage is that we use the half number of fault injections on average compared to the one-bit-flip model presented in [7]. In comparison with random one-byte and n-bit models, we can retrieve the entire key using half of positions to retrieve the entire key of the Simon family. However, in counterpart, the average of number of injected faults is greater.

II. Background

Notation

- T: total number of rounds in the cipher. For instance, $T = 32$ for Simon 32/64.
- $\left(L^{i-1}, R^{i-1}\right)$: $2n$-bit input of the i^{th} round of the cipher, $i \in \{0, \cdots, T-1\}$.
- $\left(L^{i+1}, R^{i+1}\right)$: $2n$-bit output of the i^{th} round of the cipher, $i \in \{0, \cdots, T-1\}$.
- L^{i^*},R^{i^*} wrong left half input and wrong right half input respectively at position i.
- P: plaintext.
- C: ciphertext.
- C^*: faulty ciphertext.
- K^i: n-bit round-key used in the i^{th} round of the cipher, $i \in \{0 \cdots T-1\}$.
- $x <<< a$: circular left rotation of x by a bits.
- x_l: l^{th} bit of the bit string x.
- $\oplus$: logical operator xor.
- $\odot$: logical operator and.
- $a\%b : a \mod b$

A. The Simon Family Cipher

The design of Simon is a classical Feistel scheme. Each round of the Simon cipher operates on two n-bit halves, thus the general round block size is $2n$. In the remainder of this paper, we use n to refer to half of the block size.

Each round of Simon applies a non-linear, non-bijective hence no invertible function

$$\begin{aligned} F : \mathbb{F}_2^n &\to \mathbb{F}_2^n \\ x &\mapsto ((x <<< 8) \odot (x <<< 1)) \oplus (x <<< 2) \end{aligned} \tag{1}$$

to the left half input L^{i-1}. Then, the operation $\oplus$ is computed using the output of F, the right half input R^{i-1}, and the round key K^i. The last operation in a round is to swap the half inputs. The output of the last round is a ciphertext C.

Variants of Simon exist for different parameters of key size, block size, and number of rounds. The names of each Simon variant with its parameters are presented in Table I.

1) Key Schedule: The key schedule of Simon is described as a recursive function that uses two, three or four n-bit registers. The number of registers is denoted by m. Depending on m the recursive function can be:

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

where $c = (2^n - 1) \oplus 3$ is a constant value, $(z_j)_{i-m}$ denotes the i^{th} bit of z_j, and $i-m$ is taken module 62. The values of z_j are five constant sequences $z_0, \ldots, z_4$ depicted in Table II, and j is a parameter of the cipher, *cf.* Table I.

B. Previous DFA attack models

The authors of [7] noted that there is information leakage when the AND operation is used in function F. They retrieved the last-round key using DFA on the leakage. They proposed two models: (1) one-bit-flip model and (2) random one-byte model.

It can be seen from the application of F on L^{i-1} that the j^{th} bit of L^{i-1} affects possibly 3 distinct bits $(j+1)\,\%n$, $(j+2)\,\%n$, and $(j+8)\,\%n$ of $F\left(L^{i-1}\right)$, i.e., the bits:

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

where $j \in \{0, \cdots, n-1\}$. Since $L^i = R^{i-1} \oplus F\left(L^{i-1}\right) \oplus K^{i-1}$ the same bit positions of L^i are also affected by the j^{th} bit of L^{i-1}.

The key of their differential fault analysis attack is to recover L^{T-2} and then retrieve K^{T-1}. In [7], Tupsamudre

Table I
MEMBERS OF THE SIMON FAMILY WITH THEIR PARAMETERS.

Cipher	Block size $2n$	Key words m	Key size mn	Rounds T	Index to z j
Simon 32/64	32	4	64	32	0
Simon 48/72	48	3	72	36	0
Simon 48/96	48	4	96	36	1
Simon 64/96	64	3	96	42	2
Simon 64/128	64	4	128	44	3
Simon 96/92	96	2	92	52	2
Simon 96/144	96	3	144	54	3
Simon 128/128	128	2	128	68	2
Simon 128/192	128	3	192	69	3
Simon 128/256	128	4	256	72	4

Table II
THE z_j VECTORS USED IN THE SIMON KEY SCHEDULE.

j	z_j
0	11111010001001010110000111001101111101000100101011000011100110
1	10001110111110010011000010110101000111011111001001100001011010
2	10101111011100000001101001001100010100001000111111001011011 0011
3	11011011101011000110010111100000010010001010011100110100001111
4	11010001111001101011011000100000010111000011001010010011101111

et al. presented helpful formulas. The formula to retrieve K^{T-1} is

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

Suppose a fault e is induced in the intermediate result L^{T-2}. Let $\left(L^T, R^T\right)$ be the resulting faulty ciphertext, then the fault e can be found using the next formula:

$$e = L^T \oplus L^{T^*} \oplus F\left(R^T\right) \oplus F\left(R^{T^*}\right). \tag{5}$$

Since the output of correct and faulty computation is known, it is possible to deduce the value and location of the fault e injected in L^{T-2} and, hence, it is possible to determine the bits that are flipped in L^{T-2}.

1) Bit-Flip Fault Attack on Simon at round $T-2$: In the computation of $F(L^{T-2})$, the authors of [7] observed 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 R^T. Therefore, it is possible to deduce the bit L^{T-2} and consequently to retrieve the bit of K^{T-1} using the equation (4). Let us explore this point in more detail.

Suppose a fault flips j^{th} bit of the intermediate result L^{T-2} resulting in a faulty ciphertext $\left(L^{T^*}, R^{T^*}\right)$. Then $R^{T^*} = L^{T-1^*} = R^{T-2} \oplus F\left(L^{(T-2)^*}\right) \oplus K^{T-2}$. The XOR of the right half of the correct and faulty ciphertext is written as

$$R^T \oplus R^{T^*} = F\left(L^{T-2}\right) \oplus F\left(L^{(T-2)^*}\right).$$

Because the j^{th} bit of L^{T-2} affects possibly 3 distinct bits of $F\left(L^{(T-2)}\right)$, the correct computation of R^T differs from its faulty computation in at most 3 distinct positions:

$$\begin{aligned}
\left(R^T \oplus R^{T^*}\right)_{(j+1)\%n} &= \left(L_j^{T-2} \odot L_{(j-7)\%n}^{T-2}\right) \\
&\quad \oplus \left(\left(L_j^{T-2} \oplus 1\right) \odot L_{(j-7)\%n}^{T-2}\right) \\
\left(R^T \oplus R^{T^*}\right)_{(j+8)\%n} &= \left(L_{j+7}^{T-2} \odot L_j^{T-2}\right) \\
&\quad \oplus \left(L_{j+7}^{T-2} \odot \left(L_j^{T-2} \oplus 1\right)\right) \\
\left(R^T \oplus R^{T^*}\right)_{(j+2)\%n} &= 1
\end{aligned} \tag{6}$$

According the truth Tables III and IV, for the first two formulas, we can deduce the value of $L_{(j-7)\%n}^{T-2}$ and $L_{(j+7)\%n}^{T-2}$ respectively. For example, from Table III it can be that if the value of $L^T \oplus L_{(j+1)\%n}^{T^*}$ is 0, then irrespective of the bit value L_j^{T-2}, the value of $L_{(j-7)\%n}^{T-2}$ is also 0, otherwise it is 1. The value of $L_{(j-7)\%n}^{T-2}$ can be deduce in similar way.

Table III
TRUTH TABLE FOR $\left(R^T \oplus R^{T^*}\right)_{(j+1)\%n}$.

L_j^{T-2}	$L_{(j-7)\%n}^{T-2}$	$\left(R^T \oplus R^{T^*}\right)_{(j+1)\%n}$
0	0	0
1	0	0
0	1	1
1	1	1

Now that the values of $L_{(j-7)\%n}^{T-2}$ and $L_{(j+7)\%n}^{T-2}$ are known, it is possible to retrieve the corresponding bits of

Table IV
TRUTH TABLE FOR $\left(R^T \oplus R^{T^*}\right)_{(j+8)\%n}$.

L_j^{T-2}	$L_{(j+7)\%n}^{T-2}$	$\left(R^T \oplus R^{T^*}\right)_{(j+8)\%n}$
0	0	0
1	0	0
0	1	1
1	1	1

K^{T-1} using equation (4), i.e.,

$$K_{(j-7)\%n}^{T-2} = L_{(j-7)\%n}^{T-2} \oplus F\left(R^T\right)_{(j-7)\%n} \oplus L_{(j-7)\%n}^{T}$$
$$K_{(j+7)\%n}^{T-2} = L_{(j+7)\%n}^{T-2} \oplus F\left(R^T\right)_{(j+7)\%n} \oplus L_{(j+7)\%n}^{T}. \quad (7)$$

The authors of [7] state that to retrieve the n-bit round key is necessary $n/2$ faulty ciphertexts if there is a control over the location of the flipped bit in the injected fault. In addition, if there is no control, they calculate the average of faulty encryptions in an experimental way, see Table V.

Table V
BIT-FLIP FAULT ATTACK ON SIMON ASSUMING NO CONTROL OVER THE FAULT POSITION.

n bits	Avg. No. of Faulty Encryptions
16	25
24	43
32	62
48	104
64	150

2) Random Byte Fault Attack on Simon at round $T-2$: The most practical attack might affect a byte of L^{T-2}. The authors of [7] showed that it is possible to use the same working principle of the one-bit-flip model attack to retrieve K^{T-1} injecting a fault in one byte (except for two cases). They experimentally found the average of random fault injections that are required to retrieve K^{T-1}. This average is shown in Table VI. They also showed that when there is control over the location for fault injections, the number of fault injections to retrieve K^{T-1} is $n/8$.

They mentioned two exceptions: (1) the least and most significant bits of the induced byte fault are one, and (2) a byte fault flips two adjacent bits. In the first exception the adversary can retrieve only one last-round key bit, and in the second the adversary can retrieve four key bits but modifying slightly the formulas to deduce the bits.

3) n-bit Fault Attack on Simon : Similar to the last two attacks, the authors of [8] analyzed the input and output differences in the AND operation when applying random fault injections on n bits in a intermediate position. They have precisely calculated the average number of fault injections to obtain a round key by examining the relationships between the bits obtained through multiple fault injections. Their analysis reduce significantly the average number of fault injections to retrieve L^{T-2}. For instance, in Simon 128/128 the n-bit model uses 3.91 faults instead of 8 used by the one-byte model. Also, they showed specifically the fault positions, to obtain m round keys based on the key word size m. That is, they repeated the analysis on the round-key extractions, changing the fault positions. Based on key schedule, they calculated the round keys one after another from the m deduced round keys.

Table VI
AVERAGE OF FAULTY ENCRYPTIONS FOR THE RANDOM BYTE FAULT ATTACK ON SIMON AT ROUND $T-2$.

n bits	Avg. No. of Faulty Encryptions
16	6
24	9
32	13
48	21
64	30

III. ONE-BIT-FLIP FAULT ATTACK ON SIMON AT ROUND $T-3$

We observed that if an adversary is able to flip a single round, then we can retrieve information of two rounds. This is new compared with the one-bit-flip, one-byte and n-bit models presented in Section II-B which retrieve information of only one round. Specifically, we will show in this section that is possible obtain all bits of L^{T-2} and L^{T-3} injecting faults in L^{T-3}. Also, we will show how to retrieve m round keys using our modification.

Before explaining how to retrieve information of L^{T-2} and L^{T-3} using the single round $T-3$, we need to understand what are the probabilities of affecting 1, 2 or 3 bits of $F\left(L^{T-3}\right)$ when an error occurred in L^{T-3}. Remember that we know the number of possible affected bits are 1, 2 or 3 by the equation (3).

Without loss of generality, let L be a n-bit string. Let F be a non-linear function defined in (1). If $F\left(L^*\right)$ is the application of F on L when its j^{th} bit is flipped, then the probability of flipping 1, 2, or 3 bits of $F\left(L^*\right)$ is $1/4$, $1/2$ and $1/4$ respectively.

To proof this statement, we have to look at the 3 distinct bits of $F\left(L^*\right)$ affected by the j^{th} flipped bit of L^*:

$$F\left(L^*\right)_{j+1} = \left(\left(L_j \oplus 1\right) \odot L_{(j-7)\%n}\right) \oplus L_{j-1}$$
$$F\left(L^*\right)_{j+2} = \left(L_{j+6} \odot L_{j+1}\right) \oplus L_j \oplus 1 \quad (8)$$
$$F\left(L^*\right)_{j+8} = \left(L_{j+7} \odot \left(L_j \oplus 1\right)\right) \oplus L_{j+6}$$

First, we calculate the individual probabilities of $F\left(L^*\right)_{j+1}$, $F\left(L^*\right)_{j+2}$ and $F\left(L^*\right)_{j+8}$ be affected by the j^{th} flipped bit in L. Because $F\left(L^*\right)_{j+2} \oplus F\left(L\right)_{j+2}$ always is 1, the probability of the $(j+2)^{th}$ bit of $F\left(L^*\right)$ be flipped is 1. The $(j+1)^{th}$ bit of $F\left(L^*\right)$ flips only if $L_j = 1$ and $L_{(j-7)\%n} = 1$, or if $L_j = 0$ and $L_{(j-7)\%n} = 1$. In other words, when $1/2$ of the possibilities happens. In similar

way, the $(j+8)^{th}$ bit of $F(L^*)$ flips only if $L_j = 1$ and $L_{(j+7)\%n} = 1$, or if $L_j = 0$ and $L_{(j+7)\%n} = 1$. Again, when $1/2$ of the possibilities happens. Then the probability that no error occurs in the $(j+1)^{th}$ and $(j+8)^{th}$ bits of $F(L^*)$ is equal to $1/2$ for both cases. Thus the probability P_1 of only one flipped bit occur in $F(L^*)$ is $\mathrm{P}_1 = 1 \cdot 1/2 \cdot 1/2 = 1/4$. The probability of a flipped bit occur at the $(j+1)^{th}$ position but not in the $(j+8)^{th}$ is $1/4$. The probability of a flipped bit occur at the $(j+8)^{th}$ position but not in the $(j+1)^{th}$ is also $1/4$. Then the probability P_2 that two bits are flipped is equal to the sum of the probabilities of each occurrence, that is $\mathrm{P}_2 = 1/4 + 1/4 = 1/2$. The probability P_3 of 3 flipped bits occur in $F(L^*)$ is equal to the probability of both $(j+1)^{th}$ and $(j+8)^{th}$ are flipped, that is $\mathrm{P}_3 = 1/2 \cdot 1/2 = 1/4$.

The one-bit-flip and one-byte models can not be performed without knowing the positions of the flipped bits in the left half input L^{T-2}. To retrieve these positions, equation (5) is necessary. Similarly, for our modification we cannot perform our attack if we do not know the position of the flipped bit in the left half input L^{T-3}, and the positions of flipped bits in L^{T-2} affected by $F\left(L^{(T-3)^*}\right)$. In the following discussion, we show how to retrieve the location of the j^{th} flipped bit in L^{T-3}, and then how to retrieve the locations of the flipped bits in L^{T-2}.

A. Deducing j

Since the j^{th} flipped bit in $L^{(T-3)^*}$ affects the 3 possibles positions $a = (j+1)\%n$, $b = (j+2)\%n$ and $c = (j+8)\%n$, the idea for deducing j arises from the fact that theses locations are fixed. In fact, if 3 flipped bits appear in $F\left(L^{(T-3)^*}\right)$, then to deduce j it is only necessary to calculate a, and then j must be $a-1$. But because a, b, and c are unknown we need to find them. In the next section, we explain how to retrieve those affected bits. For now suppose that these bits are represented with ones in the a^{th}, b^{th} and c^{th} positions of a n-bit string e'. For example, suppose the bit at the location $j = 2$ was flipped resulting in $L^{(T-3)^*}$. Also, suppose this flipped bit affects 3 bits of $F\left(L^{(T-3)^*}\right)$. Then

$$e' = 0\underset{j}{0}01100000100\cdots 0.$$

The case with 3-bit flips in e' does not always happen. In fact, we have showed that the probability of flipping 1, 2 or 3 bits is $1/4$, $1/2$, and $1/4$ respectively. To capture all those cases we have developed Algorithm 1. Here, LSB$(\cdot)$ and MSB$(\cdot)$ are functions to obtain the least and the most significant bits respectively. The function wt$(\cdot)$ calculates the Hamming weight, and abs$(\cdot)$ returns the absolute value.

How to deduce j when e' has Hamming weight 2 or 3 is shown in steps 4 and 11 respectively. The idea to retrieve j for the case $\mathrm{wt}(e') = 3$ is to find the two adjacent flipped bits, i.e., $(j+1)\%n$ and $(j+2)\%n$, in e' and then $j = \mathrm{LSB}(e') - 1$. The idea in the case $\mathrm{wt}(e') = 2$ is to know whether e' has two adjacent bits, which is stablished by functions LSB$(\cdot)$ and MSB$(\cdot)$ in the steps 29 and 13 respectively. Note that the two-adjacent bit case is the same in step 4 except when the difference between the least and the most significant bits is equal to $n-1$. This case is considered as two separate bits and the expression to calculate j is showed in step 27. The case of two separate bits has 4 subcases. The first two happen when the $(j+1)\%n < (j+8)\%n$, or $(j+2)\%n < (j+8)\%n$. In those subcases the expressions to retrieve j are shown in steps 15 and 18 respectively. The last two subcases occur when $(j+1)\%n > (j+8)\%n$, or $(j+2)\%n > (j+8)\%n$. In those subcases the expressions to retrieve j are shown in steps 21 and 24 respectively.

It is not possible to deduce j when e' has Hamming weight 1. In the next section, we will show that this case is not necessary in our attack.

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

Suppose that a fault flips the j^{th} bit of the intermediate result L^{T-3} resulting in a faulty ciphertext $\left(L^{T^*}, R^{T^*}\right)$. As we can see in Section III-A, one flipped bit at the j^{th} location of L^{T-3} possibly affects 3 distinct bits of $F\left(L^{(T-3)^*}\right)$. Those bits are $F\left(L^{T-3}\right)_{(j+1)}$, $F\left(L^{T-3}\right)_{(j+2)}$, and $F\left(L^{T-3}\right)_{(j+8)}$, and again they may affect 3 distinct bits of L^{T-2} because $L^{T-2} = F\left(L^{T-3}\right) \oplus R^{T-3} \oplus K^{T-3}$.

It is possible that the j^{th} flipped bit in L^{T-3} flips only one bit of L^{T-2}, which flips only one bit of $F\left(L^{(T-2)^*}\right)$. Since $L^{T-3} = R^{T-2}$, one might think that the i^{th} affected bit in $F\left(L^{(T-2)^*}\right)$ might coincide with the j^{th} flipped bit in R^{T-2}, and then no error would occur in the output of the last round of Simon. However, from the discussion at the beginning of Section III, this case does not happen because we know that if only one bit is affected in $F\left(L^{(T-2)^*}\right)$, then the bit location is $i = (j+2)\%n$ and for all cases of Simon $j \neq i \neq (j+2)\%n$. If one flipped bit occurs in L^{T-3}, then the left half input L^T will always have at least one flipped bit.

By (5), we know how to find the fault e that flipped the bits in L^{T-2}. Using Algorithm 1 with input e, it is possible to deduce the j^{th} bit flipped in L^{T-3} and consequently in R^{T-2}. Since the bits L^{T-2}_{j+1}, L^{T-2}_{j+2} and L^{T-2}_{j+8} are inside a byte then it is possible to apply the random one-byte model presented in Section II-B2 to retrieve information about L^{T-2} and K^{T-1}. But the formulas of that section has to be re-written because the j^{th} fault location of R^{T-2} may coincide with some fault locations of each 3 possible distinct bits affected by L^{T-2}_{j+1}, L^{T-2}_{j+2} or L^{T-2}_{j+8}. These new formulas are in Appendix A.

Because the output of F on L^{T-2} is XORed with R^{T-2}, in these new formulas we need adding the difference $\tilde{R}^{T-2}_x$ between R^{T-2}_x and $R^{(T-2)^*}_x$ where x corresponds with the fault location of each bit affected

Algorithm 1 Deducing j

Input: bit string e' of size n
Output: deducing j

$lsb \leftarrow \text{LSB}(e')$
$msb \leftarrow \text{MSB}(e')$
$j \leftarrow -1$
if $\text{wt}(e') = 3$ **then**
  for $i = 0$ **to** $n-1$ **do**
    if $e'[i\%n] = 1$ and $e'[(i+1)\%n] = 1$ **then**
      $j \leftarrow i - 1$
    end if
  end for
end if
if $\text{wt}(e') = 2$ **then**
  $d \leftarrow \text{abs}(lsb - msb)$
  if $d > 1$ **then**
    if $d = 7$ **then**
      $j \leftarrow (lsb - 1)\%n$
    end if
    if $d = 6$ **then**
      $j \leftarrow (lsb - 2)\%n$
    end if
    if $d = n - 7 + 1$ **then**
      $j \leftarrow (msb - 2)\%n$
    end if
    if $d = n - 7$ **then**
      $j \leftarrow (msb - 1)\%n$
    end if
    if $d = n - 1$ **then**
      $j \leftarrow n - 2$
    end if
  else
    for $i = 0$ **to** $n-1$ **do**
      if $e'[i\%n] = 1$ and $e'[(i+1)\%n] = 1$ **then**
        $j \leftarrow i - 1$
      end if
    end for
  end if
end if
return $j\%n$

by the $(j+1)^{th}$, $(j+2)^{th}$ and $(j+8)^{th}$ of L^{T-2}. If only one bit is flipped in the output of $F\left(L^{(T-2)^*}\right)$ it is at the location $((j+2)\%n)^{th}$. This flipped bit affects 3 bits of $\left(R^T \oplus R^T\right)$, namely $\left(R^T \oplus R^T\right)_{(j+3)\%n}$, $\left(R^T \oplus R^T\right)_{(j+4)\%n}$ and $\left(R^T \oplus R^T\right)_{(j+10)\%n}$. In this case note that the terms $\tilde{R}^{T-2}_{j+3}$,$\tilde{R}^{T-2}_{j+4}$ and $\tilde{R}^{T-2}_{j+10}$ for these formula, is always zero. This explain why is not necessary to deduce j when L^{T-2} has Hamming weight 1.

Table VII shows the bits of L^{T-2} that are retrievable using those formulas when L^{T-2}_{j+1}, L^{T-2}_{j+2} or L^{T-2}_{j+8} were affected. The first column shows the affected bits by L^{T-2}_{j+1}, L^{T-2}_{j+2} or L^{T-2}_{j+8}. The second column shows the conditions, because multiple bits are flipped. The third column shows the deduced values using the truth tables of the formulas in Appendix A.

For example, suppose the j^{th} bit in L^{T-3} was flipped and this bit affects the $(j+2)^{th}$ bit in L^{T-2}. Also, suppose that the $(j+2)^{th}$ flipped in L^{T-2} affects the $(j+3)^{th}$ and $(j+4)^{th}$ bits in L^{T-1}. Using the $((j+3)\%n)^{th}$ bit of $\left(R^T \oplus R^{T^*}\right)$, and supossing that the $((j+1)\%n)^{th}$ of L^{T-2} was not flipped, we can get the following formula:

$$\begin{aligned}\left(R^T \oplus R^{T^*}\right)_{(j+3)\%n} = &\left(L^{T-2}_{(j+2)\%n} \odot L^{T-2}_{(j-5)\%n}\right) \\ &\oplus \left(\left(L^{T-2}_{(j+2)\%n} \oplus 1\right) \odot L^{T-2}_{(j-5)\%n}\right) \\ &\oplus \tilde{R}^{T-2}_{(j+3)\%n}.\end{aligned} \tag{9}$$

Here, $\tilde{R}^{T-2}_{(j+3)\%n} = \left(R^{T-2} \oplus R^{(T-2)^*}\right)_{(j+3)\%n}$. After constructing the truth table for that formula, we can deduce the $((j-5)\%n)^{th}$ bit of L^{T-2}. This example is highlighted in gray in Table VII.

Table VII
DEDUCING BITS OF L^{T-2}.

affected bits by L^{T-2}_{j+1}	conditions	deduce value
$\left(R^T \oplus R^{T^*}\right)_{(j+2)\%n}$		$L^{T-2}_{(j-6)\%n}$
$\left(R^T \oplus R^{T^*}\right)_{(j+3)\%n}$	$\tilde{L}^{T-2}_{(j+2)\%n} = 1$	$L^{T-2}_{(j-5)\%n}$
	$\tilde{L}^{T-2}_{(j+2)\%n} = 0$	
$\left(R^T \oplus R^{T^*}\right)_{(j+9)\%n}$	$\tilde{L}^{T-2}_{(j+8)\%n} = 1$	
	$\tilde{L}^{T-2}_{(j+8)\%n} = 0$	$L^{T-2}_{(j+8)\%n}$
affected bits by L^{T-2}_{j+2}	conditions	deduce value
$\left(R^T \oplus R^{T^*}\right)_{(j+3)\%n}$	$\tilde{L}^{T-2}_{(j+1)\%n} = 1$	$L^{T-2}_{(j-5)\%n}$
	$\tilde{L}^{T-2}_{(j+1)\%n} = 0$	$L^{T-2}_{(j-5)\%n}$
$\left(R^T \oplus R^{T^*}\right)_{(j+4)\%n}$		
$\left(R^T \oplus R^{T^*}\right)_{(j+10)\%n}$	$\tilde{L}^{T-2}_{(j+8)\%n} = 1$	$L^{T-2}_{(j+9)\%n}$
	$\tilde{L}^{T-2}_{(j+8)\%n} = 0$	$L^{T-2}_{(j+9)\%n}$
affected bits by L^{T-2}_{j+8}	conditions	deduce value
$\left(R^T \oplus R^{T^*}\right)_{(j+9)\%n}$	$\tilde{L}^{T-2}_{(j+1)\%n} = 1$	
	$\tilde{L}^{T-2}_{(j+1)\%n} = 0$	$L^{T-2}_{(j+1)\%n}$
$\left(R^T \oplus R^{T^*}\right)_{(j+10)\%n}$	$\tilde{L}^{T-2}_{(j+2)\%n} = 1$	$L^{T-2}_{(j+9)\%n}$
	$\tilde{L}^{T-2}_{(j+2)\%n} = 0$	
$\left(R^T \oplus R^{T^*}\right)_{(j+16)\%n}$		$L^{T-2}_{(j+15)\%n}$

C. Retrieving L^{T-3} and K^{T-2}

To retrieve L^{T-3} and K^{T-2} we use the one-bit-flip model attack idea described in Section II-B1, but using only $T-1$ rounds of Simon, as indicated by the dashed rectangle in Figure 1. This is possible now because we know the output $\left(L^{T-1}, R^{T-1}\right)$ of the round $T-2$. We will be calling this ouput by "correct intermediate text", and since K^{T-1} is known, the elements of this output are $L^{T-1} = R^T$ and

$R^{T-1} = F\left(Y^T\right) \oplus L^T \oplus K^{T-1}$. When an intermediate error occur in L^{T-3}, we will be calling the output of the round $T-2$ by "faulty intermediate text". We could construct the "faulty intermediate text" by adding new errors, but instead we reuse the errors generated for each faulty error injected in L^{T-3} explained in the last section, and then we use the idea of the one-bit model at round $T-2$ to retrieve L^{T-3}. That is, we repeat the random fault injections in L^{T-3} but using the "faulty intermediate text" which can be calculated in the following way: $Y^{T^*} = L^{(T-1)^*}$ and $R^{(T-1)^*} = R^{T-1} \oplus e$.

Following the example of the Section III-B, we can retrieve the $\left((j-7)\,\%n\right)^{th}$ and $\left((j+7)\,\%n\right)^{th}$ bits of L^{T-3}. The flipped bits of this example are highlighted in red, and the retrieve bits are highlighted in green in Figure 1

D. Retrieving the entire key of Simon 96/96, Simon 128/128 and other members

The idea of our modification is to use less fault positions compared with [7], [8]. In this context, to retrieve the entire secret key of Simon 96/96 and Simon 128/128, we need to obtain 2 round keys based on the key word size $m = 2$. Then we use the previous analysis of sections III-B and III-C. From the key expansion of Simon, we can calculate the round keys one after another from the deduced $m = 2$ round keys. That is, for $k = i-2$ and $m = 2$ of (2) we can obtain

$$K^k = K^{k+2} \oplus \left(K^{k+1} >>> 3\right) \oplus \left(K^{k+1} >>> 4\right) \oplus c \oplus \left(z_j\right)_{j-m+2}. \tag{10}$$

Since K^{T-1} and K^{T-2} can be found using the method presented in Section III-C, we can feed these values into (10) and after $T-2$ rounds we can find the entire key of the Simon 96/96 or the Simon 128/128.

For the cases $m = 3$ and $m = 4$ the above explanation can be applied if m adjacent round-keys are known and changing the rounds for fault injections. That is, the adversary needs to select one more fault position. For the case $m = 3$, the adversary has two options. The first one is inject faults to the left half input L^{T-3} and to retrieve the round-keys K^{T-1} and K^{T-2} and the output $\left(L^{T-2}, R^{T-2}\right)$ of round $T-3$. This output can be retrieved with the method presented in Section III-C. To retrieve K^{T-3} the adversary must inject faults to the left half input L^{T-5} and to retrieve K^{T-3} and K^{T-4} the adversary must use the same method and the output $\left(L^{T-2}, R^{T-2}\right)$, which is known. The second option is: after retrieving $\left(L^{T-2}, R^{T-2}\right)$ the adversary injects faults at round $T-4$ and one of the methods presented in Section II-B2. For the case $m = 4$, the adversary must follow the first option of the case $m = 3$.

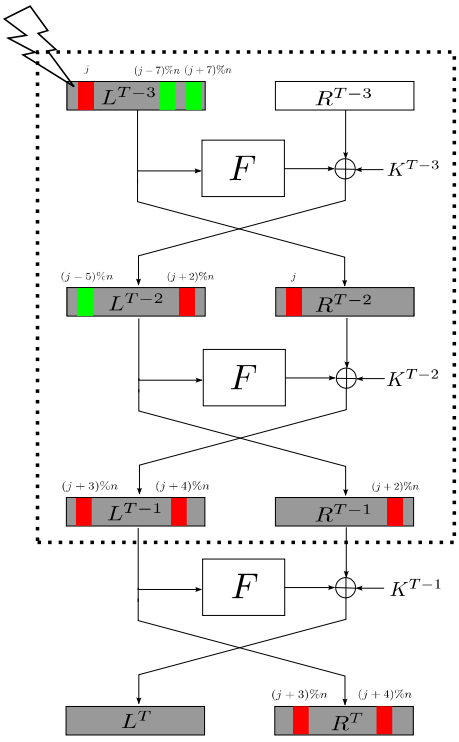

Figure 1. Fault propagation when L^{T-3} is randomly corrupted in the j^{th} bit. Gray denotes the faulty intermediate states and the heavy lines shows the rounds that is necessary to analyze for retrieve K^{T-2}.

E. Average Number of Fault Injections

The expected number of K^{T-1} bits recovered by a single bit-flip in L^{T-3} is

$$2\text{P}(1 \text{ flip in } L^{T-2}) + 4\text{P}(2 \text{ flips in } L^{T-2}) + 4\text{P}(3 \text{ flips in } L^{T-2}) = 2\frac{1}{4} + 4\frac{1}{2} + 4\frac{1}{4} = 3.5 \text{ bits}.$$

In addition, using the same pair of correct and faulty ciphertexts 2 bits of K^{T-2} can be recovered. Thus one bit flip in $L^{(T-3)}$ recovers 3.5 bits of K^{T-1} and 2 bits of K^{T-2} on an average. The current bit-flip attack is therefore more efficient than the bit-flip attack of [7] which recovers only 2 bits of K^{T-1}.

The random process to recover the K^{T-1} and K^{T-2} can be modeled using the cover time of a random walk. According to [9], the average number of faulty injections needed to recover K^{T-1} and K^{T-2} using our one-bit-flip model is

$$R = \frac{(n-1)}{4}\left(\Psi\left(\frac{n+1}{2}\right) + 2\left(\gamma + \ln\left(2\right)\right)\right) + \frac{(2\gamma - 1) + n\Psi\left(\frac{1}{2}\right)}{4} \tag{11}$$

where Ψ is a digamma function and γ is the Euler number.

Table VIII
COMPARISON OF RESULTS OF DFA ON SIMON FAMILY.

Block Size	Key Size	Key Words(m)	Fault Location	Avg. [7] One-byte	Avg. [7] One-bit-flip	Avg. [8] n-bit	Fault Location	Avg. One-bit-flip
32	64	4	$L^{27}, L^{28}, L^{29}, L^{30}$	24	101.72	12.20	L^{27}, L^{29}	50.85
48	72	3	L^{32}, L^{33}, L^{34}	27	130.78	9.91	L^{32}, L^{33}	87.19
48	96	4	$L^{31}, L^{32}, L^{33}, L^{34}$	36	174.37	13.22	L^{31}, L^{33}	87.19
64	96	3	L^{38}, L^{39}, L^{40}	39	189.44	10.45	L^{38}, L^{39}	126.29
64	128	4	$L^{39}, L^{40}, L^{41}, L^{42}$	52	252.58	13.93	L^{39}, L^{41}	126.29
96	96	2	L^{49}, L^{50}	42	210.24	7.46	L^{49}	105.12
96	144	3	L^{50}, L^{51}, L^{52}	63	315.36	11.19	L^{50}, L^{51}	210.24
128	128	2	L^{65}, L^{66}	60	299.68	7.82	L^{65}	149.84
128	192	3	L^{65}, L^{66}, L^{67}	90	449.52	11.73	L^{65}, L^{66}	299.68
128	256	4	$L^{67}, L^{68}, L^{69}, L^{70}$	120	599.36	15.64	L^{67}, L^{69}	299.68

Table IX shows the experimental average number of fault injections to obtain the two round keys K^{T-1} and K^{T-2}. Specifically, the second column shows the average of fault injections to obtain K^{T-1} and K^{T-2}. These values confirm the theoretical average number of fault injections R for each member of Simon. For an instance when $n = 16$ $R = 25.43$. Also, these values confirm the previous study of the average number of fault injection presented in Table V, but in our modification, we retrieve two round keys and not only one. The third column of the Table IX shows the average number of reused fault injections to retrieve K^{T-1}. For example, for Simon 32/64 this is approximately 15. Note that the average number of reused fault injections is always limited by the average number of fault injection used in the one-bit-flip model. This also confirm that it is possible to retrieve L^{T-2} and L^{T-3} using our modification.

Table IX
AVERAGE OF NUMBER OF FAULT ENCRYPTIONS TO OBTAIN L^{T-3} AND L^{T-2}.

n	Avg. No. Fault Injections L^{T-3} and L^{T-2}	Avg. No. Fault Injections reused L^{T-2}	Time (s)
16	24.81	15.26	10.17
24	42.74	29.70	15.49
32	61.75	44.19	36.84
48	103.42	77.02	57.85
64	147.57	110.81	151.82

IV. SIMULATIONS

In order to verify the proposed attack and to evaluate the average number of fault injections, we implement the attack in Python and execute the code on an Intel Core i5 2.6 GHz processor using Ubuntu 12.04 (64 bits) OS. In the simulation, we assume that L^{T-3} is randomly corrupted with one bit fault. The plaintexts and the secret keys are randomly chosen.

As mentioned in Section III-E, the first column of Table IX shows the average number of fault injections to obtain all bits in L^{T-3} and L^{T-2} when $n = 16, 24, 32, 48, 64$. The third column of Table IX shows the average number of fault injections that will be reused to obtain only L^{T-2}. The number of samples is 100,000 in Table IX. From the results, the average number of fault injections to obtain all 128 bits of L^{T-2} and L^{T-3} is 150.

V. COMPARISON WITH RELATED WORK

Table VIII gives a summary of our method applied on SIMON family. We also show a comparison between this study and previous work [7], [8]. The number of samples is 100,000 in Table VIII. The 5^{th}, 6^{th},7^{th} and 9^{th} columns of Table VIII shows the average number of fault injections required to retrieve the entire secret key for all members of Simon. The 5^{th} column shows the average number of fault injections using the one-byte model. The 6^{th} column shows the average number of fault injections using the one-bit-flip model presented in [7]. The 7^{th} column shows the average number of fault injections using the n-bit model. And, the 9^{th} column shows the average number of fault injections using our one-bit model.

Note that when our modification is compared with the one-bit-flip model in the cases $m = 2$ and $m = 4$, our modification needs in average half of fault injections and locations to retrieve the entire key. When our modification is compared with both the one-byte model and n-bit model, our modification needs half of fault positions. Obviously, for the last two comparisons the average of number of fault injections is greater.

Different from [7], and equal to [8], our attack method achieves the entire key and not only a round key. Besides, our modification works with all original rounds of each member of the Simon in contrast with the linear cryptanalysis proposed in [10]–[12], which work with reduced round version of Simon.

VI. CONCLUSION

In this paper, we have described a DFA on Simon family inspired on the ideas presented in [7]. As we show in Section III-A, besides using the information leaked by the AND operation, we exploit the pseudo invertibility of the round function F when a single fault injection happens in the input.

We show how to extract the secret key using a random one-bit fault model for all parameters in Simon family. As an example for Simon 128/128, we extract the entire secret key using 149.84 fault injections on average. We believe that this pseudo invertibility contributes to the study of Fault Analysis on other cryptographic primitives.

Because the average number of fault injections in our modification is greater than the random one-byte and n-bit models, but less in number of fault positions, our modification is useful when the adversary pays a high price to inject the faults in more than one round.

In the future, we will investigate if it is possible to extend our method using random-byte fault model or the n-bit model.

ACKNOWLEDGMENT

We are most grateful to Mohamed Saied for his comments during the preparation of this work.

REFERENCES

[1] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology, ser. CRYPTO '90. London, UK, UK: Springer-Verlag, 1991, pp. 2–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=646755.705229

[2] N. Bagheri, R. Ebrahimpour, and N. Ghaedi, "New differential fault analysis on PRESENT," EURASIP Journal on Advances in Signal Processing, vol. 2013, no. 1, 2013. [Online]. Available: http://dx.doi.org/10.1186/1687-6180-2013-145

[3] H. Chen, W. Wu, and D. Feng, "Differential Fault Analysis on Clefia," in Information and Communications Security, ser. Lecture Notes in Computer Science, S. Qing, H. Imai, and G. Wang, Eds. Springer Berlin Heidelberg, 2007, vol. 4861, pp. 284–295. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77048-0_22

[4] G. Piret and J.-J. Quisquater, "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad," in Cryptographic Hardware and Embedded Systems - CHES 2003, ser. Lecture Notes in Computer Science, C. Walter, Ç. Koç, and C. Paar, Eds. Springer Berlin Heidelberg, 2003, vol. 2779, pp. 77–88. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-45238-6_7

[5] L. Hemme, "A Differential Fault Attack Against Early Rounds of (Triple-)DES," in Cryptographic Hardware and Embedded Systems - CHES 2004, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds. Springer Berlin Heidelberg, 2004, vol. 3156, pp. 254–267. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-28632-5_19

[6] N. Courtois, K. Jackson, and D. Ware, "Fault-Algebraic Attacks on Inner Rounds of DES," Proceedings of the IEEE, Sept 2010.

[7] H. Tupsamudre, S. Bisht, and D. Mukhopadhyay, "Differential fault analysis on the families of simon and speck ciphers," in Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on, Sept 2014, pp. 40–48.

[8] J. Takahashi and T. Fukunaga, "Fault Analysis on SIMON Family of Lightweight Block Ciphers," in Information Security and Cryptology - ICISC 2014, ser. Lecture Notes in Computer Science, J. Lee and J. Kim, Eds. Springer International Publishing, 2015, vol. 8949, pp. 175–189. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15943-0_11

[9] L. Lovsz, "Random walks on graphs: A survey," 1993.

[10] J. Alizadeh, H. A. Alkhzaimi, M. R. Aref, N. Bagheri, and M. M. Lauridsen, "Improved Linear Cryptanalysis of Round Reduced SIMON." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.469.7749&rank=4

[11] J. Alizadeh, N. Bagheri, P. Gauravaram, A. Kumar, , and S. K. Sanadhya, "Linear Cryptanalysis of Round Reduced Variants of SIMON." [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.400.5362&rep=rep1&type=pdf

[12] Q. Wang, Z. Liu, K. Varici, Y. Sasaki, V. Rijmen, and Y. Todo, "Cryptanalysis of Reduced-Round SIMON32 and SIMON48," in Progress in Cryptology – INDOCRYPT 2014, ser. Lecture Notes in Computer Science, W. Meier and D. Mukhopadhyay, Eds. Springer International Publishing, 2014, pp. 143–160. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13039-2_9

APPENDIX

A flip in $L^{T-2}_{(j+1)\%n}$ affects 3 bits:

$$\left(R^T \oplus R^{T^*}\right)_{(j+2)\%n} = \left(L^{T-2}_{(j+1)\%n} \odot L^{T-2}_{(j-6)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+1)\%n} \oplus 1\right) \odot L^{T-2}_{(j-6)\%n}\right) \oplus \tilde{R}^{T-2}_{(j+2)\%n}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+3)\%n} = \begin{cases} \left(L^{T-2}_{(j+2)\%n} \odot L^{T-2}_{(j-5)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+2)\%n} \oplus 1\right) \odot L^{T-2}_{(j-5)\%n}\right) \oplus 1 \oplus \tilde{R}^{T-2}_{(j+3)\%n} \\ \text{if } L_{(j+2)\%n} \text{ was affected} \\ 1 \oplus \tilde{R}^{T-2}_{(j+3)\%n} \\ \text{if otherwise} \end{cases}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+9)\%n} = \begin{cases} \left(L^{T-2}_{(j+8)\%n} \odot L^{T-2}_{(j+1)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+8)\%n} \oplus 1\right) \odot \left(L^{T-2}_{(j+1)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+9)\%n} \\ \text{if } L_{(j+8)\%n} \text{ was affected} \\ \left(L^{T-2}_{(j+8)\%n} \odot L^{T-2}_{(j+1)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+8)\%n}\right) \odot \left(L^{T-2}_{(j+1)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+9)\%n} \\ \text{if otherwise} \end{cases}$$

A flip in $L^{T-2}_{(j+2)\%n}$ affects 3 bits:

$$\left(R^T \oplus R^{T^*}\right)_{(j+3)\%n} = \begin{cases} \left(L^{T-2}_{(j+2)\%n} \odot L^{T-2}_{(j-5)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+2)\%n} \oplus 1\right) \odot L^{T-2}_{(j-5)\%n}\right) \oplus 1 \oplus \tilde{R}^{T-2}_{(j+3)\%n} \\ \text{if } L_{(j+1)\%n} \text{ was affected} \\ \left(L^{T-2}_{(j+2)\%n} \odot L^{T-2}_{(j-5)\%n}\right) \oplus \left(\left(L^{T-2}_{(j+2)\%n} \oplus 1\right) \odot L^{T-2}_{(j-5)\%n}\right) \oplus \tilde{R}^{T-2}_{(j+3)\%n} \\ \text{if otherwise} \end{cases}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+4)\%n} = L^{T-2}_{(j+2)\%n} \oplus (L^{T-2}_{(j+2)\%n} \oplus 1) \oplus \tilde{R}^{T-2}_{(j+4)\%n} = 1 \oplus \tilde{E}^{T-2}_{(j+4)\%n}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+10)\%n} = \begin{cases} \left(L^{T-2}_{(j+9)\%n} \odot L^{T-2}_{(j+2)\%n}\right) \oplus \left(L^{T-2}_{(j+9)\%n} \odot \left(L^{T-2}_{(j+2)\%n} \oplus 1\right)\right) \oplus 1 \oplus \tilde{R}^{T-2}_{(j+10)\%n} \\ \text{if } L_{(j+8)\%n} \text{ was affected} \\ \left(L^{T-2}_{(j+9)\%n} \odot L^{T-2}_{(j+2)\%n}\right) \oplus \left(L^{T-2}_{(j+9)\%n} \odot \left(L^{T-2}_{(j+2)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+10)\%n} \\ \text{if otherwise} \end{cases}$$

A flip in $L^{T-2}_{(j+8)\%n}$ affects 3 bits:

$$\left(R^T \oplus R^{T^*}\right)_{(j+9)\%n} = \begin{cases} \neg\left(L^{T-2}_{(j+8)\%n} \oplus L^{T-2}_{(j+1)\%n}\right) \oplus \tilde{R}^{T-2}_{(j+9)\%n} \\ \text{if } L_{(j+1)\%n} \text{ was affected} \\ \left(L^{T-2}_{(j+8)\%n} \odot L^{T-2}_{(j+1)\%n}\right) \oplus \left(L^{T-2}_{(j+1)\%n} \odot \left(L^{T-2}_{(j+8)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+9)\%n} \\ \text{if otherwise} \end{cases}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+10)\%n} = \begin{cases} \left(L^{T-2}_{(j+9)\%n} \odot L^{T-2}_{(j+2)\%n}\right) \oplus \left(L^{T-2}_{(j+9)\%n} \odot \left(L^{T-2}_{(j+2)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+10)\%n} \\ \text{if } L_{(j+2)\%n} \text{ was affected} \\ 1 \\ \text{if otherwise} \end{cases}$$

$$\left(R^T \oplus R^{T^*}\right)_{(j+16)\%n} = \left(L^{T-2}_{(j+15)\%n} \odot L^{T-2}_{(j+8)\%n}\right) \oplus \left(L^{T-2}_{(j+15)\%n} \odot \left(L^{T-2}_{(j+8)\%n} \oplus 1\right)\right) \oplus \tilde{R}^{T-2}_{(j+16)\%n}$$