INDIAN INSTITUTE OF TECHNOLOGY INDORE

UNDERGRADUATE THESIS

Pinpointing Fault Analysis Attacks on Simon and Simeck Family of Lightweight Ciphers

---

Author:
NAMAN SINGHAL
ID No. 160001038

Supervisors:
Dr. BODHISATWA MAZUMDAR

Thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Technology

in the

Department of Computer Science and Engineering

November 28, 2019

Declaration of Authorship

I, NAMAN SINGHAL declare that this thesis titled, "Pinpointing Fault Analysis Attacks on Simon and Simeck Family of Lightweight Ciphers" and the work presented in it is my own. I confirm that:

- This work was done wholly or mainly while in candidature for the BTP project at IIT Indore.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Certificate

This is to certify that the thesis entitled, "*Pinpointing Fault Analysis Attacks on Simon and Simeck Family of Lightweight Ciphers*" and submitted by Naman Singhal ID No 160001038 in partial fulfillment of the requirements of CS 493 B.Tech Project embodies the work done by him under my supervision.

Supervisor

Dr.BODHISATWA MAZUMDAR
Associate Professor,
Indian Institute of Technology Indore
Date:

INDIAN INSTITUTE OF TECHNOLOGY INDORE

Abstract

Department of Computer Science and Engineering

Bachelor of Technology

Pinpointing Fault Analysis Attacks on Simon and Simeck Family of Lightweight Ciphers

In this paper, we pinpoint hotspots in Simon block cipher that pose vulnerability against Fault Analysis Attacks. SIMON is a lightweight block cipher which was presented by the NSA in 2013. We are able to present a Differential Fault Attack (DFA) with reduced number of faults for all variation of SIMON. We restrict the DFA to a single last register of the cipher. We are further able to reveal hotspots for attack on SIMON in higher rounds than before. Moreover, we are able to extend the attack to Simeck with equally good results.

Acknowledgements

I would like to thank my B.Tech Project supervisors **Dr. Bodhisatwa Mazumdar** for his guidance and constant support in structuring the project and his valuable feedback throughout the course of this project. He provided the prefect framework to work in an efficient manner and come up with innovative ideas. Him overseeing the project meant there was a lot that I learnt while working on it. I thank him for his time and efforts.

I am really grateful to the Institute for the opportunity to be exposed to systemic research. Lastly, I offer my sincere thanks to everyone who helped me complete this project, whose name I might I have forgotten to mention.

Contents

List of Tables

Chapter 1

Introduction

1.1 About

In 2013, the US National Security Agency have published lightweight block cipher families, SIMON and SPECK, that cater towards optimal performance metrics of speed and area in resource-constrained environment. Subsequently, a family of lightweight ciphers called SIMECK was published in CHES 2015 [1], wherein the designers merged good features of both SIMON and SPECK to yield an efficient and compact cipher for hardware implementation. Both SIMON and SIMECK are based on Feistel construction with similar round functions except that they employ different rotational constants ($(1, 8, 2)$ for SIMON, and $(0, 5, 1)$ for SIMECK). The requirements of lightweight ciphers, such as resource count in implementation less than 3k gate equivalents, meet stringent timing requirements of ISO 18000-63 that cannot be met by the traditional block ciphers such as AES.

Both SIMON and SPECK have a Feistel structure, and provides implementation and security support for for five block sizes of 32, 48, 64, 96, and 128 bits with key sizes as show in Table 1.1. Although the cipher conform to the system requirements of lightweight resource-constrained platforms, the security analysis was not stringently done. However, researchers performed extensive scrutiny over the last few years [2, 3, 4, 5]. The initial work on Simon focused on linear ad differential cryptanalysis [3, 5, 6]. Based on Matsui's algorithm, optimal differential trails for 12, 16, 19, 28, and 37 rounds of SIMON 32/48/64/96/128 were demonstrated in [7]. The work computed the best differential distinguishers that can be computed on SIMON cipher.

The works in [4, 8] used the concept of *partial difference distribution table* (pDDT) to search differential trails in Simon-like ciphers. Although the work determined improved differential trails for SIMON and SPECK, optimal differential trails may not always be found as they employ heuristics to compute high probability differential trails, . For SIMON round function, a formula to compute differential probability was given by [9]. The authors employed SAT/SMT solvers to determine optimal differential trails for SIMON, and demonstrated provably optimal differential trails for SIMON 32, SIMON 48, and 16-round optimal differential trail with probability 2^{-54} for SIMON64.

Our Contributions. This paper investigates the problem of pinpointing fault analysis in SIMON. The main contributions of the paper can be summarized as follows:

1) Proposed a better attack in SIMON with much fewer faults than existing literature (especially for $N = 48, 64$), with low time complexity.

2) Restricted attack to a single register improving ease of attack.

3) Exposed Vulnerability in Higher Rounds of SIMON giving attacker more options to mount an attack.

4) Found optimal round to attack for all version of SIMON.

5) Extended the same attack to Simeck giving a better attack than the existing literature.

1.2 Background

SIMON is a family of lightweight cipher comprising a Feistel structure that operates on a $2n$-bit state, where n is the word size, $n = 16, 24, 48$, and 64. The key comprises m n-bit wwords where $m = 2, 3, 4$, thus SIMON cipher of block size of $2n$ bits and key size mn bits is referred to as SIMON$2n/mn$.

Block Size (2n)	Key Size (mn)	Word Size (n)	Key words (m)	Rounds
32	64	16	4	32
48	72	24	3	36
	96		4	
64	96	32	3	42
	128		4	44
96	96	48	2	52
	144		3	54
128	128	64	2	68
	192		3	69
	256		4	72

TABLE 1.1: Different Version Of SIMON

Chapter 2

Existing Work

The fault analysis on SIMON was first analyzed in [10]. The work demonstrated that an adversary, in the threat model of random single bit flip, can recover the n-bit last round key of SIMON and SPECK by injecting $\frac{n}{2}$ and $\frac{n}{3}$ faults, respectively, into the left input of $(T-2)^{th}$ round. The work further extends demonstrate the attack on a random byte fault model, wherein multiple bits of last round key were recovered depending on the Hamming weight of the induced byte fault. However, owing to key scheduling algorithm, the last m-round key values need to be recovered to recover the entire mn-bit key. In other words, the adversary needs to inject faults in $(T-3)^{th}, \ldots, (T-m-1)^{th}$ rounds, to recover the entire secret key, which implies larger controllability and hence cost of the adversary to inject faults in m round locations. In an improvement on the average number of required faults, the work in [11] required 3.05 faults on average, with the requirement of injecting faults in $(T-2)^{th}, \ldots, (T-m-1)^{th}$ rounds.

Vasquez et al. proposed a new attack in 2015 [12], injecting faults in $(T-3)^{th}$ round instead in $(T-2)^{t}h$ round, reducing the number of faults needed. Single round needed to be controlled for $M=2$, but 2 rounds had to be controlled for $N=3/4$. Further Ravi Anand et al. presented an attack in 2018 [13], attacking $(T-5)^{th}$ round and needing to inject faults in only one round, with very few faults. However $SIMON96/96$, $SIMON96/144$, $SIMON128/128$, $SIMON96/192$ and $SIMON128/256$ were not tacked and the attack sacrificed in time complexity.

Chapter 3

Proposed Attack

3.1 Analysing Simon's Architecture

3.1.1 SIMON's Architecture

SIMON2n/nm is defined by the following round function:

$$F(x) = (x \lll 1) \wedge (x \lll 8) \oplus (x \lll 2)$$
$$L_{i+1} = R_i \oplus F(L_i) \oplus K_i$$
$$R_{i+1} = L_i$$

The cipher starts with the plaintext (L_0, R_0). Round i takes (L_i, Ri) as input and results in $(L_{i+1}, Ri + 1)$ as the output. The cipher repeats the process T times giving (L_T, R_T) as the cipthertext.

3.1.2 Fault Model

We follow the 1 bit flip DFA model for the attack. The attacker need to have control over the L register of a particular round. Each fault will be a single bit flip at a random location in register L leading to a faulty ciphertext. The fault injected does not have any permanent impact on the cipher and is reset after each fault injection, implying that each fault injection is independent. We keep the plaintext and key constant throughout the attack. Throughout the text we assume that the Fault is injected in register L_r and bit $l_{r,j}$ has flipped.

3.1.3 Fault Impact and Propagation

After fault injection, (L_i, R_i) becomes (L_i^*, R_i^*).

$$\begin{aligned}
\Delta R_{i+1} = R_{i+1} \oplus R_{i+1}^* &= L_i \oplus L_i^* = \Delta L_i \\
\Delta L_{i+1} = L_{i+1} \oplus L_{i+1}^* &= F(L_i) \oplus F(L_i^*) \oplus \Delta R_i \\
&= \Delta F(L_i) \oplus \Delta L_{i-1} \\
\Delta F(L_i) &= [(L_i \lll 1) \wedge (L_i \lll 8) \oplus (L_i \lll 2)] \\
&\quad \oplus [(L_i^* \lll 1) \wedge (L_i^* \lll 8) \oplus (L_i^* \lll 2)]
\end{aligned}$$

We can calculate of fault propagation in the next round using the equation:

$$\Delta l_{i+1,j} = (l_{i,j-1} \wedge l_{i,j-8}) \oplus [(l_{i,j-1} \oplus \Delta l_{i,j-1}) \wedge (l_{i,j-8} \oplus \Delta l_{i,j-8})] \\ \oplus \Delta l_{i,j-2} \oplus \Delta l_{i-1,j}$$

This equation shows that $\Delta l_{i+1,j}$ is impacted only by $\Delta l_{i,j-1}$, $\Delta l_{i,j-8}$, $\Delta l_{i,j-2}$, $\Delta l_{i-1,j}$. Using this equation we can trace the impact of the single bit fault in L_r

$$\begin{aligned} \Delta l_{r-1,i} &= 0 \; \forall \; i \\ \Delta l_{r,j} &= 1 \\ \Delta l_{r,i} &= 0 \; \forall \; i \neq j \\ \Delta l_{r+1,j+1} &= l_{r,j-7} \\ \Delta l_{r+1,j+8} &= l_{r,j+7} \\ \Delta l_{r+1,j+2} &= 1 \\ \Delta l_{r+1,i} &= 0, \textit{For Rest Bits} \end{aligned}$$

Solving this for further rounds, we can compute which Deltas are 0, 1 and *unknown*.

Round	1	*Unknown*
r	j	-
$r+1$	$j+2$	$j+1, j+8$
$r+2$	$j, j+4$	$j+2, j+3, j+9, j+10, j+16$
$r+3$	$j+6$	$j+1, j+3 to j+5, j+8, j+10, j+11, j+12, j+17, j+18, j+24$
$r+4$	j	$j+2$ *to* $j+14, j+16, j+18, j+19, j+20, j+25, j+26, j+32$
$r+5$	$j+2$	$j+1, j+3$ *to* $j+22, j+24$ *to* $j+28, j+33, j+34, j+40$
$r+6$	j	$j+2$ *to* $j+30, j+32$ *to* $j+36, j+41, j+42, j+48$
$r+7$	$j+2$	$j+1, j+3$ *to* $j+38, j+40$ *to* $j+44, j+49, j+50, j+56$
$r+8$	j	$j+2$ *to* $j+46, j+48$ *to* $j+52, j+57, j+58, j+64$

TABLE 3.1: Single Bit Fault Propagation

Note that depending on the value of N, fault propagation will overlap and change, example: For $N = 16, round\ r+3$ we have $(j+17)\%16 => j+1$, $(j+18)\%16 => j+2$ and $(j+24)\%16 => j+8$.

3.1.4 Important Equations

We can calculate the $\Delta(L_i)$ using L_{i+1} and L_{i+2}

$$\Delta L_i = [F(L_{i+1}) \oplus F(L^*_{i+1})] \oplus [L_{i+2} \oplus L^*_{i+2}] \tag{3.1}$$

Using $\Delta(L_{i+1})$, $\Delta(L_i)$ and $\Delta(L_{i-1})$ we can make questions in (L_i).

$$\Delta l_{i+1,j} = (l_{i,j-1} \wedge l_{i,j-8}) \oplus [(l_{i,j-1} \oplus \Delta l_{i,j-1}) \wedge (l_{i,j-8} \oplus \Delta l_{i,j-8})] \\ \oplus \Delta l_{i,j-2} \oplus \Delta l_{i-1,j} \tag{3.2}$$

3.2 Pinpointing Fault Location

Identifying the Fault Location accurately is the first part of the attack. As we have already seen in table (3.1), any single bit has it's own propagation signature which is independent of the Key value. We will use this property to pinpoint the fault location using a modified method as seen in [13]

Suppose any text $A = \{a_0, a_1, ..., a_{n-1}\}$. The plain text is (L_0, R_0) representing the L and R registers and key K. Cipher text obtained is (L_T, R_T) where T is the total number of Rounds. Now the experiment is repeated but a 1-bit fault is injected in the L register of r^{th} round in SIMON at bit (γ) we get new cipher text $(L_T^{(\gamma)}, R_T^{(\gamma)})$. The r^{th} round is fixed throughout the attack. The objective here is to find (γ) using (L_T, R_T) and $(L_T^{(\gamma)}, R_T^{(\gamma)})$.

We will use 2 phases "Making the Correlation Matrix (offline)" and "Computing the Fault Location (online)".

3.2.1 Correlation Matrix

S represents the correlation matrix. This is how we calculate it

$$
\begin{aligned}
L_{T-1} &= R_T \\
L_{T-1}^{(\gamma)} &= R_T^{(\gamma)} \\
\Delta L_T^{(\gamma)} &= L_T \oplus L_T^{(\gamma)} \\
\Delta L_{T-1}^{(\gamma)} &= L_{T-1} \oplus L_{T-1}^{(\gamma)} \\
\Delta L_{T-2}^{(\gamma)} &= [F(L_{T-1}) \oplus F(L_{T-1}^{(\gamma)})] \oplus [L_T \oplus L_T^*] \\
\Delta L_{T-2}^{(\gamma)} &= [F(L_{T-1}) \oplus F(L_{T-1}^{(\gamma)})] \oplus [\Delta L_T^{(\gamma)}]
\end{aligned}
$$

$$
\begin{aligned}
\theta^{(\gamma)} &= \{\psi_0^{(\gamma)}, \psi_1^{(\gamma)}, ..., \psi_{3n-1}^{(\gamma)}\} \\
\theta^{(\gamma)} &= \Delta L_T^{(\gamma)} + \Delta L_{T-1}^{(\gamma)} + \Delta L_{T-2}^{(\gamma)} \\
S^{(\gamma)} &= \{s_0^{(\gamma)}, s_1^{(\gamma)}, ..., s_{3n-1}^{(\gamma)}\} \\
s_i^{(\gamma)} &= 1/2 - Pr(\psi_i^{(\gamma)} = 1)
\end{aligned}
$$

The probability $(Pr(\psi_i^{(\gamma)} = 1))$ is calculated over a sufficient number of trials, where for each trial we consider a random plaintext (L_0, R_0) and random key (K). We store $S^{(0)}$, $S^{(1)}$, ... $S^{(n-1)}$ as the correlation matrix S.

3.2.2 Computing the fault location

For a unknown plaintext (L_0, R_0) and unknown key (K) we obtain a ciphertext (L_T, R_T). Now, we obtain a faulty ciphertext $(L_T^{(\gamma)}, R_T^{(\gamma)})$ by injecting a 1-bit fault in an unknown position (γ) is the L register of r^{th} round. We will calculate the trail T using $\Delta L_T^{(\gamma)}$, $\Delta L_{T-1}^{(\gamma)}$ and $\Delta L_{T-2}^{(\gamma)}$ as above.

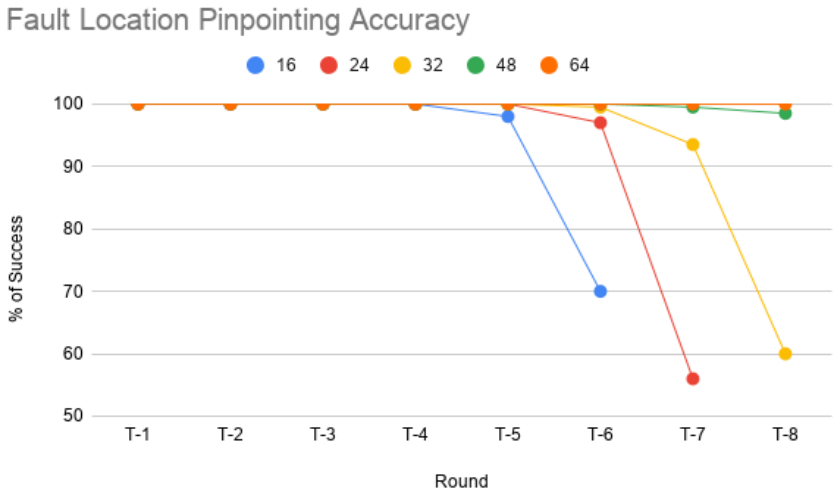
$$\theta^{(\gamma)} = \{\psi_0^{(\gamma)}, \psi_1^{(\gamma)}, ..., \psi_{3n-1}^{(\gamma)}\}$$
$$\theta^{(\gamma)} = \Delta L_T^{(\gamma)} + \Delta L_{T-1}^{(\gamma)} + \Delta L_{T-2}^{(\gamma)}$$
$$T^{(\gamma)} = \{\tau_0^{(\gamma)}, \tau_1^{(\gamma)}, ..., \tau_{3n-1}^{(\gamma)}\}$$
$$\tau_i^{(\gamma)} = 1/2 - (\psi_i^{(\gamma)})$$

Now we identify (γ), by computing the $S^{(j)}$ which has the highest correlation with $T^{(\gamma)}$. For this, a modified version of Pearson's correlation coefficient $\mu(X, Y)$ is used as shown in [13]

$\mu(X, Y) = -1$ in case of a mismatch, i.e. if $(x_i = 1/2, y_i = -1/2)$ or $(x_i = -1/2, y_i = 1/2)$ holds true any i. Otherwise it is calculated as:

$$\mu(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

3.2.3 Results and Accuracy



N	*Round*							
	$T-1$	$T-2$	$T-3$	$T-4$	$T-5$	$T-6$	$T-7$	$T-8$
16	100%	100%	100%	100%	98%	70%	–	–
24	100%	100%	100%	100%	100%	97%	56%	–
32	100%	100%	100%	100%	100%	99.5%	93.5%	60%
48	100%	100%	100%	100%	100%	100%	99.5%	98.5%
64	100%	100%	100%	100%	100%	100%	100%	100%

TABLE 3.2: Fault Prediction Accuracy

3.3 The Attack

3.3.1 Finding the secret key

We know for $SIMON2N/NM$ we need to find M secret keys, $K_{T-1}, K_{T-2}...K_{T-M}$. Once we know the last M keys we can unroll the whole key schedule and find the original secret key.

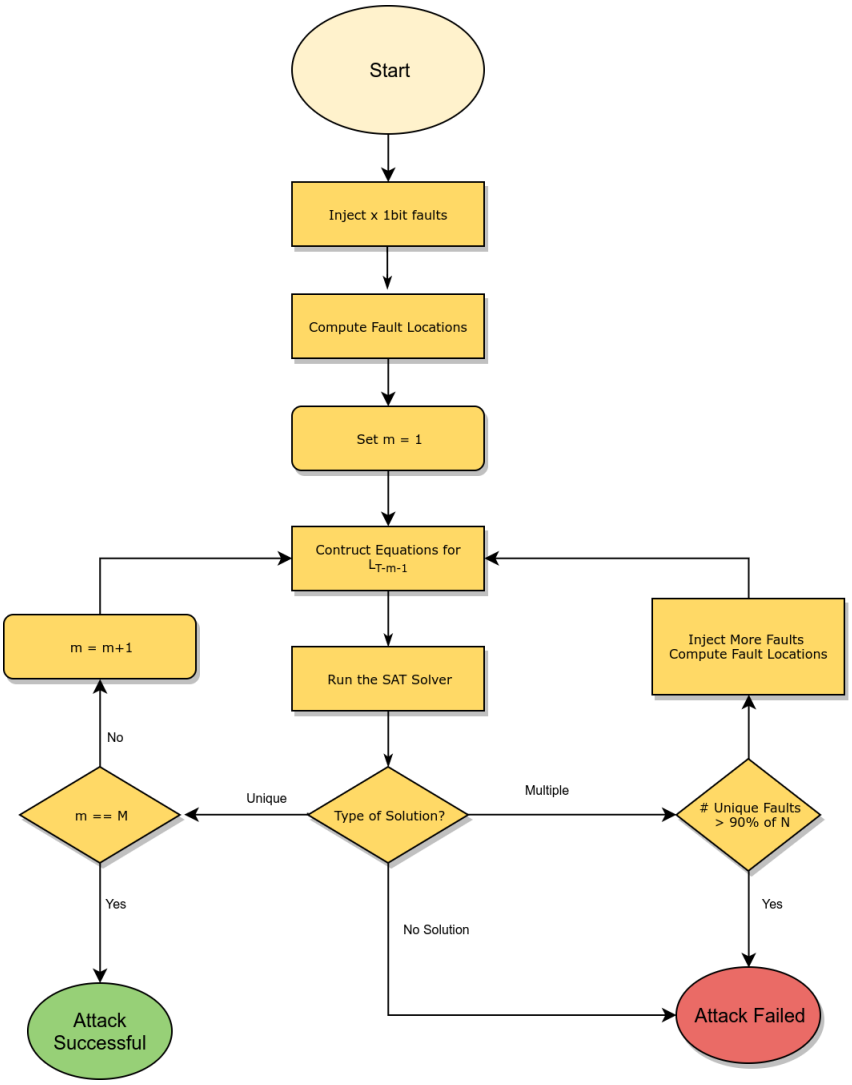
We can write SIMON's round structure as :

$$\begin{aligned} K_{T-1} &= L_T \oplus L_{T-2} \oplus F(L_{T-1}) \\ K_{T-2} &= L_{T-1} \oplus L_{T-3} \oplus F(L_{T-2}) \\ &... \\ K_{T-M} &= L_{T-M+1} \oplus L_{T-M-1} \oplus F(L_{T-M}) \end{aligned}$$

Since we already have L_T, L_{T-1} as the cipher text. To find the Secret Key the AIM is to find $\{L_{T-2}$, L_{T-3} ... $L_{T-M-1}\}$; M left registers

3.3.2 Attack Flow

The attack starts with injecting x number of faults in the L register of the chosen round r. Using fault pinpointing we compute the exact location where the faults was injected for each faulty text. Since we need to find M Left registers we first make equations to compute L_{T-2}, then move up till L_{T-M-1}. To compute L_i we make equations using all the faulty ciphertext obtained till now. Once we make all the equations, we run the SAT solver. If SAT solver gives a unique solution, we found the complete L_i register. If SAT solver gives multiple solutions we need to increase the number of faults and retry. If SAT solver gives No Solution, the attack has failed because of poor fault prediction. We repeat the process till we obtain M left registers. Using $\{L_{T-2}$, L_{T-3} ... $L_{T-M-1}\}$ we easily calculate the secret key.



3.4 Making the Equations

Suppose we have to make equations for computing L_{T-i}. We assume that we know all the registers { L_T, L_{T-1}, ... L_{T-i+1}}. Using equation [3.1], for each faulty text we can compute the ΔL_{T-i} and ΔL_{T-i+1}. For all $j = 1$ *to* N, we use equation [3.2].

$$\Delta l_{T-i+1,j} = (l_{T-i,j-1} \wedge l_{T-i,j-8}) \oplus [(l_{T-i,j-1} \oplus \Delta l_{T-i,j-1}) \wedge (l_{T-i,j-8} \oplus \Delta l_{T-i,j-8})] \\ \oplus \Delta l_{T-i,j-2} \oplus \Delta l_{T-i-1,j}$$

But we don't know ΔL_{T-i-1}. We use the information that we know the round and location of the fault, so skip all j for which the value of $\Delta l_{T-i-1,j}$ is unknown according to table (3.1). Now we have the equation only in terms of L_{T-i}, we repeat the process for all the faulty texts.

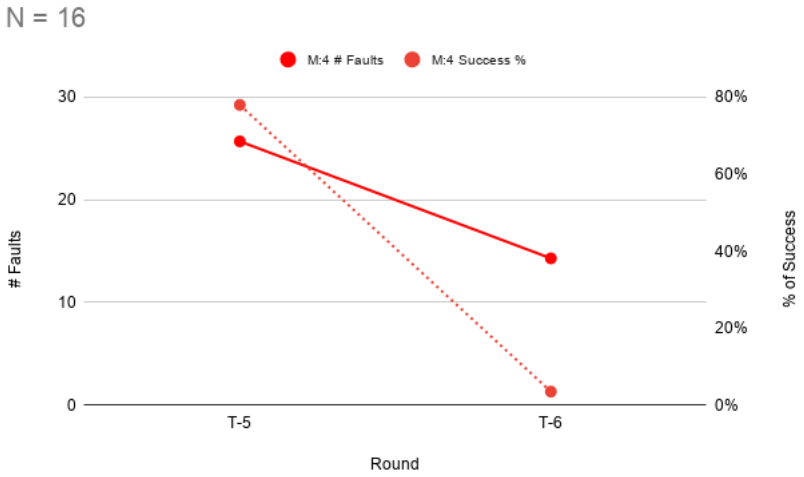
Chapter 4

Results

We generate these results using our implementation in C++ and using Cryptominisat as the SAT Solver. At any point if the number of unique faults become greater than 90% of N, we consider it a failed attempt.

4.1 Experimental Results

4.1.1 *SIMON*32/64

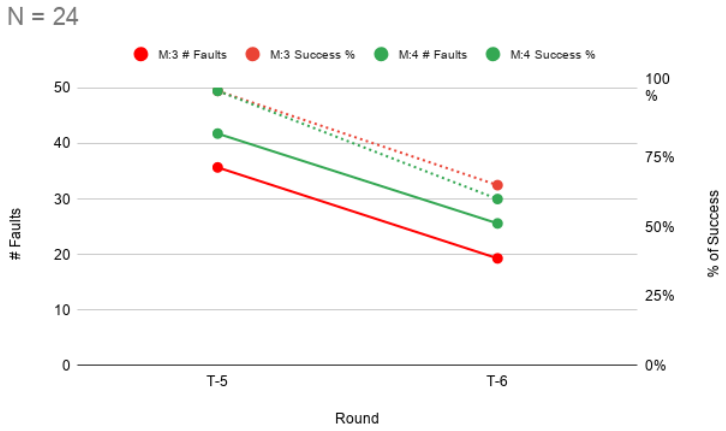


Here we can see that going from round $T-5$ to $T-6$ will reduce the number of faults required but the poor fault pinpointing accuracy gives poor success rate, restriction us the $T-5$.

M	Round	Total Faults	Total Unique Faults	Success %
4	$T-5$	25.7	12.9	78%
4	$T-6$	14.5	9.7	3/4%

TABLE 4.1: SIMON Attack Results $N = 16$

4.1.2 *SIMON*48/72 and *SIMON*48/96

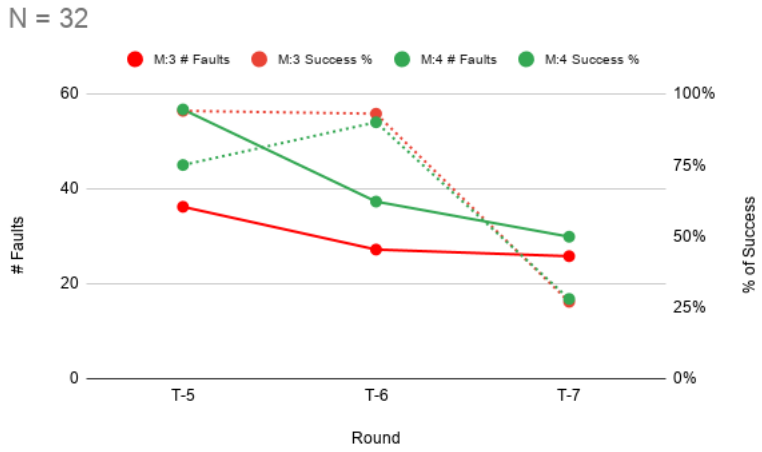


Better Fault Pinpointing accuracy for $N = 24$ lets us attack $T - 6$ with good accuracy. But getting any results for $T - 7$ becomes impossible due to poor pinpointing.

M	Round	Total Faults	Total Unique Faults	Success %
3	$T - 5$	35.7	15.5	99%
3	$T - 6$	19.3	13.1	65%
4	$T - 5$	41.8	19.9	99%
4	$T - 6$	25.6	15.6	60%

TABLE 4.2: SIMON Attack Results $N = 24$

4.1.3 *SIMON*64/96 and *SIMON*64/128

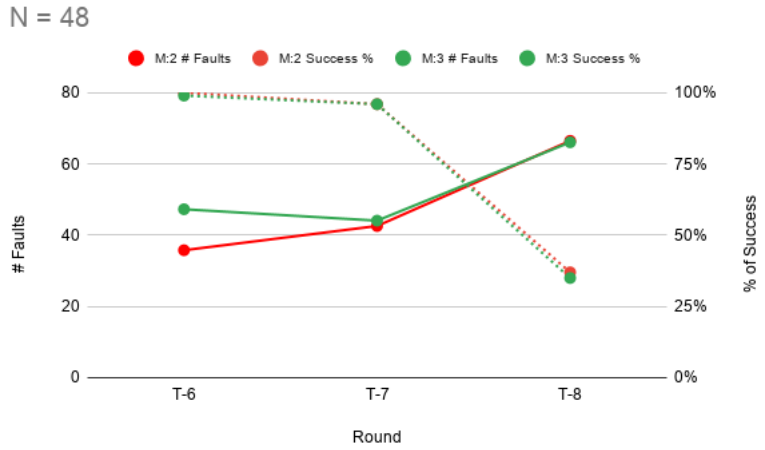


We get low success rate at $T - 5$ due to very high number of faults needed. $T - 6$ becomes the optimal round to attack with high success rate. $T - 7$ needs same number of faults for $M = 3$ and gives better results for $M = 4$ but with poor success rate because of low pinpointing accuracy.

M	Round	Total Faults	Total Unique Faults	Success %
3	$T-5$	36.2	21.5	94%
3	$T-6$	27.2	17.9	93%
3	$T-7$	25.8	17.4	27%
4	$T-5$	56.7	26.7	75%
4	$T-6$	37.3	21.6	90%
4	$T-7$	29.9	19.27	28%

TABLE 4.3: SIMON Attack Results $N = 32$

4.1.4 *SIMON96/96* and *SIMON96/144*

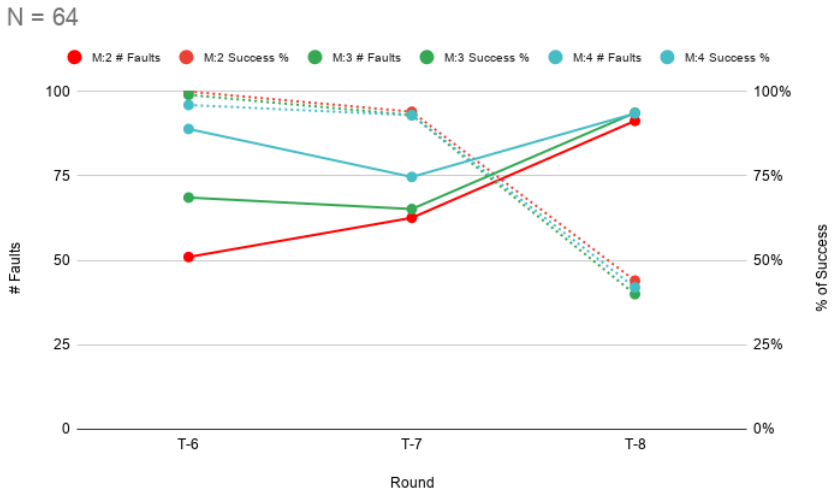


$T-6$ is the optimal round to attack for $M = 2$ and $T-7$ for $M = 3$. $T-8$ shows an increase in faults needed giving a limit to highest round we should attack.

M	Round	Total Faults	Total Unique Faults	Success %
2	$T-6$	35.8	24.6	100%
2	$T-7$	42.6	27.2	96%
2	$T-8$	66.5	34.6	37%
3	$T-6$	47.3	29.6	99%
3	$T-7$	44.1	28.1	96%
3	$T-8$	66.1	34.5	35%

TABLE 4.4: SIMON Attack Results $N = 48$

4.1.5 *SIMON*128/128, *SIMON*128/192 and *SIMON*128/256



$T-6$ is the optimal round to attack for $M=2$ and $T-7$ for $M=3,4$. $T-8$ again shows an increase in faults needed.

M	Round	Total Faults	Total Unique Faults	Success %
2	$T-6$	51.0	34.5	100%
2	$T-7$	62.6	38.5	93%
2	$T-8$	91.3	47.4	44%
3	$T-6$	68.6	41.0	99%
3	$T-7$	65.2	39.8	93%
3	$T-8$	93.7	48.1	40%
4	$T-6$	88.9	47.5	96%
4	$T-7$	74.7	43.3	93%
4	$T-8$	93.5	48.4	42%

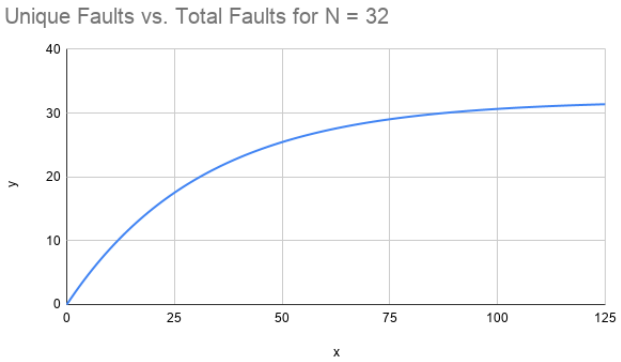
TABLE 4.5: SIMON Attack Results $N=64$

4.2 Theoretical Analysis

4.2.1 Relation between total faults and unique faults

Let x be the total number of faults injected and y be total number of unique faults. For size of block N, they relate with the following relation:

$$y = N\left(1-\left(\frac{N-1}{N}\right)^{x}\right)$$



This shows how injecting another unique fault becomes increasingly difficult as the number of Unique Fault location come closer to N.

4.2.2 Optimal Attack Round and Attack Accuracy

We can notice that while making equations, the information is revealed because of the unknown register bits. When forming equations for L_i, the more number of usable unknown registers in ΔL_{i+1} less number of faults are required.

N	*Round*								
	r	$r+1$	$r+2$	$r+3$	$r+4$	$r+5$	$r+6$	$r+7$	$r+8$
16	0	2	5	7	8	6	2	0	0
24	0	2	5	9	12	12	7	1	0
32	0	2	5	9	15	16	11	4	1
48	0	2	5	9	15	18	17	14	10

TABLE 4.6: Fault Propagation : Number of Usable Unknowns

The above table shows the number of useful equations we are able to compute when fault is injected in L_r, assuming 100% fault propagation. The number of unknowns will be lower than the value in the table as the fault doesn't propagate perfectly to lower rounds.

Lets consider $N = 16, M = 4$, when we inject fault in $T-5$. We construct equations for $\{L_{T-5}, L_{T-4}, L_{T-3}, L_{T-2}\}$. Computing L_{T-5} will require most faults because ΔL_{T-6} as only 2 unknowns i.e. 2 equations, conputing L_{T-2} will be the easiest with 8 equations for each fault. The total faults required is defined be the maximum of faults required for each L_i. So going to $T-6$ will reduce the number faults needed as we go from using [2,5,7,8] equations to [5,7,8,6] equations. But going to $T-7$ will not improve results.

The same relation between the above experimental results and this theoretical analysis continue for all variations of SIMON. For $N = 48, 64$ injecting faults at $T-8$ does not give better results then $T-7$ because the fault doesn't propagate perfectly to lower rounds reducing the number of expected equation.

4.3 Comparison Of Results

Comparing our results against existing DFA attacks following Random Bit flip attack model with complete Key-Space Reduction. We show that our results are better for all versions of SIMON.

*SIMON*2*n*/*mn*	Key Words	# Faults Injected		
		[10]	[12]	Proposed Attack
*SIMON*32/64	4	101.7	50.8	25.7
*SIMON*48/72	3	130.8	87.2	19.3
*SIMON*48/96	4	174.4	87.2	25.6
*SIMON*64/96	3	189.4	126.3	27.2
*SIMON*64/128	4	252.6	126.3	37.3
*SIMON*96/96	2	210.2	105.1	35.8
*SIMON*96/144	3	315.4	210.2	44.1
*SIMON*128/128	2	299.7	149.8	51.0
*SIMON*128/192	3	449.5	299.7	65.6
*SIMON*128/256	4	599.4	299.7	74.7

TABLE 4.7: # Faults Comparison with Existing Work

Paper [13] shows an attack with lesser number of faults than our results but with much poor time complexity. Also that attack couldn't be mounted for higher values of N (48, 64), due to the exponential increase in time complexity.

*SIMON*2*n*/*mn*	Key Words	Fault Injections			
		[13] # faults	[13] Time	Proposed Attack # faults	Proposed Attack Time
*SIMON*32/64	4	4	191.23*s*	25.7	0.1*s*
*SIMON*48/96	4	6	290.99*s*	25.6	0.2*s*
*SIMON*64/128	4	9	404.03*s*	37.3	0.4*s*

TABLE 4.8: # Faults Comparison with Existing Work

We restrict our attack to just a single Left register of a round. This makes our attack much easier to mount relative to existing work. We also mount our attack in much higher rounds exposing more vulnerability.

*SIMON*2*n*/*mn*	Key Words	Rounds Attacked			
		[10]	[12]	[13]	Proposed Attack
*SIMON*32/64	4	$L_{27}, L_{28}, L_{29}, L_{30}$	L_{27}, L_{29}	L_{27}, R_{27}	L_{27}
*SIMON*48/72	3	L_{32}, L_{33}, L_{34}	L_{32}, L_{33}	–	L_{30}
*SIMON*48/96	4	$L_{31}, L_{32}, L_{33}, L_{34}$	L_{31}, L_{33}	L_{31}, R_{31}	L_{30}
*SIMON*64/96	3	L_{40}, L_{41}, L_{42}	L_{32}, L_{33}	–	L_{36}
*SIMON*64/128	4	$L_{39}, L_{40}, L_{41}, L_{42}$	L_{31}, L_{39}	L_{39}, R_{39}	L_{38}
*SIMON*96/96	2	L_{49}, L_{50}	L_{49}	–	L_{46}
*SIMON*96/144	3	L_{50}, L_{51}, L_{52}	L_{50}, L_{51}	–	L_{47}
*SIMON*128/128	2	L_{65}, L_{66}	L_{65}	–	L_{62}
*SIMON*128/192	3	L_{65}, L_{66}, L_{67}	L_{65}, L_{66}	–	L_{62}
*SIMON*128/256	4	$L_{67}, L_{68}, L_{69}, L_{70}$	L_{67}, L_{69}	–	L_{65}

TABLE 4.9: Round Comparison with Existing work

Chapter 5

SIMECK Case Study

5.1 Background

Simeck is a lightweight cipher comprising of the Best Features from SIMON and SPECK as shown in [1]. It changes the round function of SIMON from $(1, 8, 2) \rightarrow$ Simeck $(0, 5, 1)$. It reuses the same architecture and Key Schedule. We extend the above proposed attack to Simeck and post the results below.

Block Size (2n)	Key Size (mn)	Word Size (n)	Key words (m)	Rounds
32	64	16	4	32
48	96	24	4	36
64	128	32	4	44

TABLE 5.1: Simeck Versions

5.2 Simeck Propogation Table

Round	1	*Unknown*
r	j	-
$r+1$	$j+1$	$j, j+5$
$r+2$	$j+2$	$j, j+1, j+5, j+6, j+10$
$r+3$	$j+3$	$j, j+1, j+2, j+5, j+6, j+7, j+10, j+11, j+15$
$r+4$	$j+4$	j *to* $j+3, j+5$ *to* $j+8, j+10, j+11, j+12, j+15, j+16, j+20$
$r+5$	–	j *to* $j+13, j+15$ *to* $j+17, j+20, j+21, j+25$
$r+6$	–	j *to* $j+18, j+20$ *to* $j+22, j+25, j+26, j+30$
$r+7$	–	j *to* $j+23, j+25$ *to* $j+27, j+30, j+31, j+35$
$r+8$	–	j *to* $j+28, j+30$ *to* $j+32, j+35, j+36, j+40$

TABLE 5.2: Single Bit Fault Propagation

5.3 Simeck Fault Pinpointing

N	*Round*							
	$T-1$	$T-2$	$T-3$	$T-4$	$T-5$	$T-6$	$T-7$	$T-8$
16	100%	100%	100%	100%	99.5%	81.5%	–	–
24	100%	100%	100%	100%	100%	99.5%	92%	58%
32	100%	100%	100%	100%	100%	99.5%	99.5%	95%

TABLE 5.3: Fault Prediction Accuracy

5.4 Simeck Results

Round	Total Faults	Total Unique Faults	Success %
$T-5$	25.6	12.8	95%
$T-6$	14.2	9.4	13%

TABLE 5.4: Attack Results *Simeck*32/64

Round	Total Faults	Total Unique Faults	Success %
$T-5$	41.5	19.9	93%
$T-6$	27.1	16.1	91%
$T-7$	25.2	15.7	24%

TABLE 5.5: Attack Results *Simeck*48/96

Round	Total Faults	Total Unique Faults	Success %
$T-5$	56.7	26.7	98%
$T-6$	39.6	22.4	86%
$T-7$	39.1	22.3	72%
$T-8$	44.4	22.5	8%

TABLE 5.6: Attack Results *Simeck*64/128

5.5 Comparisons

*Simeck*2n/mn	# Faults Injected	
	[14]	Proposed Attack
*Simeck*32/64	113.3	25.6
*Simeck*48/96	165.7	27.1
*Simeck*64/128	228.2	39.6

TABLE 5.7: Faults Comparison with Existing Work

*Simeck*2*n*/*mn*	Rounds Attacked	
	[14]	Proposed Attack
*Simeck*32/64	$L_{30}, L_{29}, L_{28}, L_{27}$	L_{27}
*Simeck*48/96	$L_{34}, L_{33}, L_{32}, L_{31}$	L_{30}
*Simeck*64/128	$L_{42}, L_{41}, L_{40}, L_{39}$	L_{38}

TABLE 5.8: Rounds Comparison with Existing Work

Chapter 6

Conclusion

In this work, we have presented a DFA on SIMON in which we can recover the complete secret key with few faults and low time complexity. We were able to restrict the attack to the left register of a single round and mounted the attack on multiple round. This exposed vulnerability in more rounds and also computes the optimal round to attack for each version of the cipher. We also present theoretical analysis into finding the optimal attack round. We further successfully extended the attack to Simeck and presented similar results with fewer faults and low time complexity.

Bibliography

[1] Gangqiang Yang et al. "The simeck family of lightweight block ciphers". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2015, pp. 307–329.

[2] Javad Alizadeh et al. "Cryptanalysis of SIMON variants with connections". In: *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer. 2015, pp. 90–107.

[3] Farzaneh Abed et al. "Differential cryptanalysis of round-reduced Simon and Speck". In: *International Workshop on Fast Software Encryption*. Springer. 2014, pp. 525–545.

[4] Alex Biryukov, Arnab Roy, and Vesselin Velichkov. "Differential analysis of block ciphers SIMON and SPECK". In: *International Workshop on Fast Software Encryption*. Springer. 2014, pp. 546–570.

[5] Hoda AlKhzaimi and Martin M Lauridsen. "Cryptanalysis of the SIMON Family of Block Ciphers." In: *IACR Cryptology ePrint Archive* 2013 (2013), p. 543.

[6] Mohamed Ahmed Abdelraheem et al. "Improved linear cryptanalysis of reduced-round SIMON-32 and SIMON-48". In: *International Conference on Cryptology in India*. Springer. 2015, pp. 153–179.

[7] Zhengbin Liu, Yongqiang Li, and Mingsheng Wang. "Optimal differential trails in simon-like ciphers". In: *IACR Transactions on Symmetric Cryptology* (2017), pp. 358–379.

[8] Alex Biryukov and Vesselin Velichkov. "Automatic search for differential trails in ARX ciphers". In: *Cryptographers' Track at the RSA Conference*. Springer. 2014, pp. 227–250.

[9] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. "Observations on the SIMON block cipher family". In: *Annual Cryptology Conference*. Springer. 2015, pp. 161–185.

[10] Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay. "Differential fault analysis on the families of SIMON and SPECK ciphers". In: *2014 Workshop on Fault Diagnosis and Tolerance in Cryptography*. IEEE. 2014, pp. 40–48.

[11] Junko Takahashi and Toshinori Fukunaga. "Fault analysis on SIMON family of lightweight block ciphers". In: *International Conference on Information Security and Cryptology*. Springer. 2014, pp. 175–189.

[12] J. d. C. G. Vasquez et al. "An Efficient One-Bit Model for Differential Fault Analysis on Simon Family". In: *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. 2015, pp. 61–70. DOI: 10.1109/FDTC.2015.18.

[13] Ravi Anand et al. "Differential Fault Attack on SIMON with Very Few Faults". In: *Progress in Cryptology – INDOCRYPT 2018*. Ed. by Debrup Chakraborty and Tetsu Iwata. Cham: Springer International Publishing, 2018, pp. 107–119.

[14] Venu Nalla, Rajeev Anand Sahu, and Vishal Saraswat. "Differential Fault Attack on SIMECK". In: *Proceedings of the Third Workshop on Cryptography and Security in Computing Systems*. CS2 '16. Prague, Czech Republic, New York, NY, USA: ACM, 2016, pp. 45–48. ISBN: 978-1-4503-4065-6. DOI: 10.1145/2858930.2858939. URL: http://doi.acm.org/10.1145/2858930.2858939.