Differential Fault Attack on SIMON with Very Few Faults

Ravi Anand[1], Akhilesh Siddhanti[2],
Subhamoy Maitra[3], Sourav Mukhopadhyay[4].

Indian Institute of Technology Kharagpur, Kharagpur

BITS Pilani, Goa Campus, Goa

Applied Statistics Unit, Indian Statistical Institute, Kolkata

Indian Institute of Technology Kharagpur, Kharagpur

INDOCRYPT 2018

Outline

Description of SIMON

SIMON is a family of lightweight block ciphers released by the NSA in 2013.
The cipher has a feistel structure and the state is updated as:

$$F(x) = (x \lll 1)\&(x \lll 8) \bigoplus (x \lll 2)$$

$$L_{i+1} = R_i \bigoplus F(L_i) \bigoplus k_i$$

$$R_{i+1} = L_i$$

where k_i is the round key which is generated by the key scheduling algorithm:

$$k_{i+4} = c \oplus (z_j)_i \oplus k_i \oplus (I \oplus S^{-1})(S^{-3}k_{i+3} \oplus k_{i+1})$$

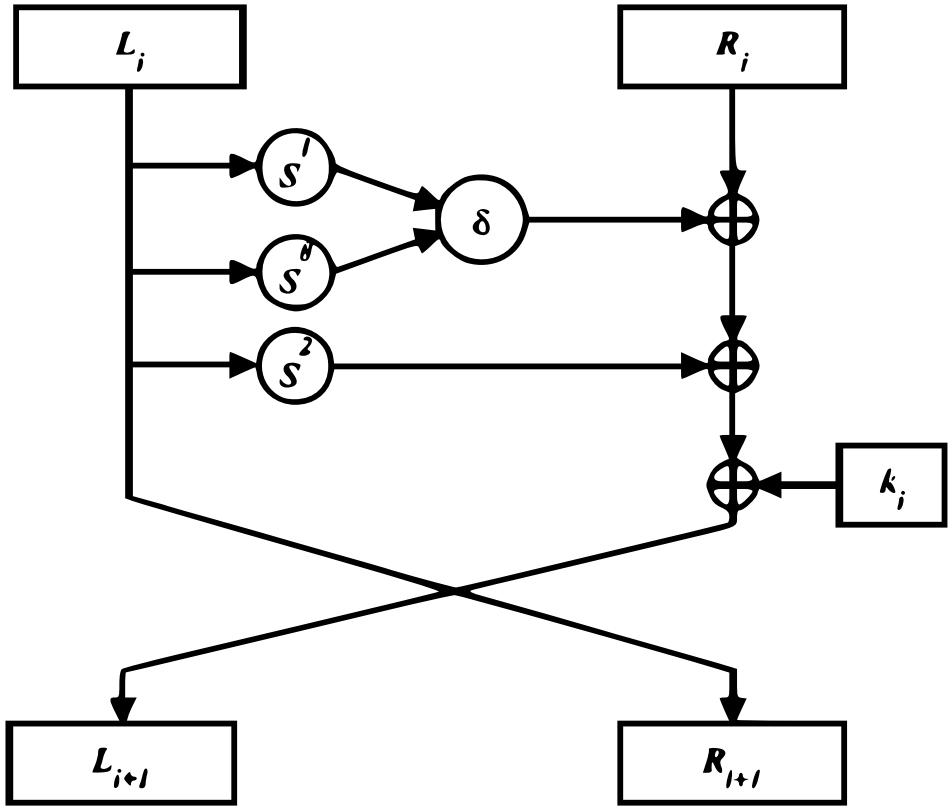


Figure: SIMON round function

Existing DFAs on SIMON

Tupsamudre et al (FDTC 2014)

Proposed one-bit flip and random one-byte flip model. Fault is injected in left input of $(T-2)^{th}$ round and last n-bit round key is recovered.

Takahashi et al (ICISC 2014)

Reduced the number of faults required using a random fault attack

Vasquez et al (FDTC 2015)

Improved the attack by injecting faults in the $(T-3)^{rd}$ round and recovering last two round keys

Chen et al (FDTC 2016)

Injected fault only once in the $(T-m-1)^{th}$ round and recovered all last m round keys

Proposed Differential Fault Attack

- A transient single bit flip model of attack [Biham et al. (1996)]
- Inspired from Differential Fault Attack on Stream Ciphers [Maitra et al. (IEEE TC 2017)]
  - The fault is injected in a particular state of cipher and an ℓ-length keystream is generated by clocking the cipher ℓ times.
  - The cipher is reset to same state and a new fault is injected and the faulty keystream is generated.
  - This process is repeated ρ many times (ρ is the number of faults required)
- In this work we successfully adopt this model to SIMON$2n/4n$

Proposed Differential Fault Attack

- For block ciphers, we need to assume that the plaintext remains fixed
- The secret key is deduced using the differences in faulty and fault-free ciphertexts. For SIMON$2n/4n$ $\ell = 2n$
- The fault is injected in some unknown register location of L or R of SIMON$2n/4n$ at the beginning of some round, say $r = (T - 5)$.

Basic Assumptions of the Fault Attack Model.

Our attack model assumes the following:

1. The adversary has the required technology to inject faults, with precise timing.
2. Fault injection causes a single bit-flip, and the effect propagates to other locations with each clocking.
3. The adversary can reset the cipher using the original secret key.
4. The adversary need not know the exact location of the fault.

Identifying Fault Locations using Signatures

- Consider we encrypt a plaintext $P = \{p_0, p_1, \cdots, p_{(2n-1)}\}$ using key K and obtain ciphertext $C = \{c_0, c_1, \cdots, c_{(2n-1)}\}$.
- Repeat the experiment, where P is encrypted with K, but a 1-bit fault is injected in the r^{th} round, and a faulty ciphertext $C^{(\gamma)} = \{c_0^{(\gamma)}, c_1^{(\gamma)}, \cdots, c_{(2n-1)}^{(\gamma)}\}$, is obtained.
- Determine the location of the injected fault, i.e., γ.

The process of determining γ is same for all the three variants, and consists of two phases, the *offline phase* and the *online phase*.

Offline Phase

Signature Vector

The *signature vector* $S^{(j)}$ is defined as:

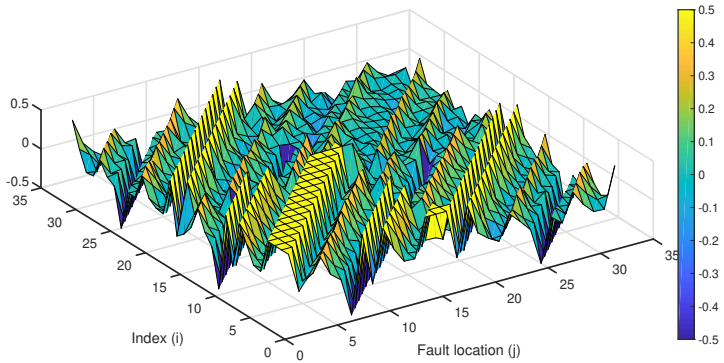
$$S^{(j)} = (s_0^{(j)}, s_1^{(j)}, \cdots, s_{(2n-1)}^{(j)}), \tag{1}$$

where

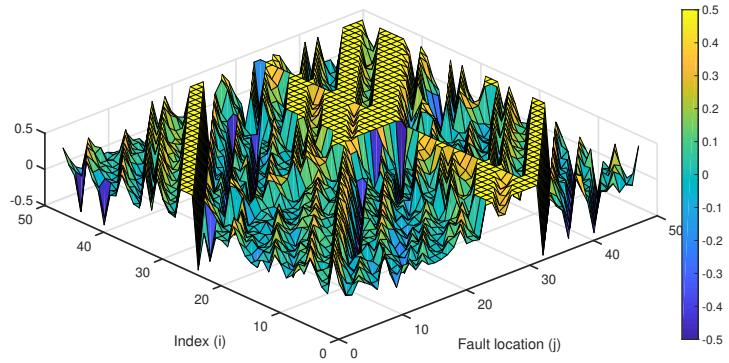
$$s_i^{(j)} = \frac{1}{2} - Pr(c_i \neq c_i^{(j)}), \tag{2}$$

for $j = 0, 1, \cdots, (2n-1)$ and $i = 0, 1, \cdots, (2n-1)$.

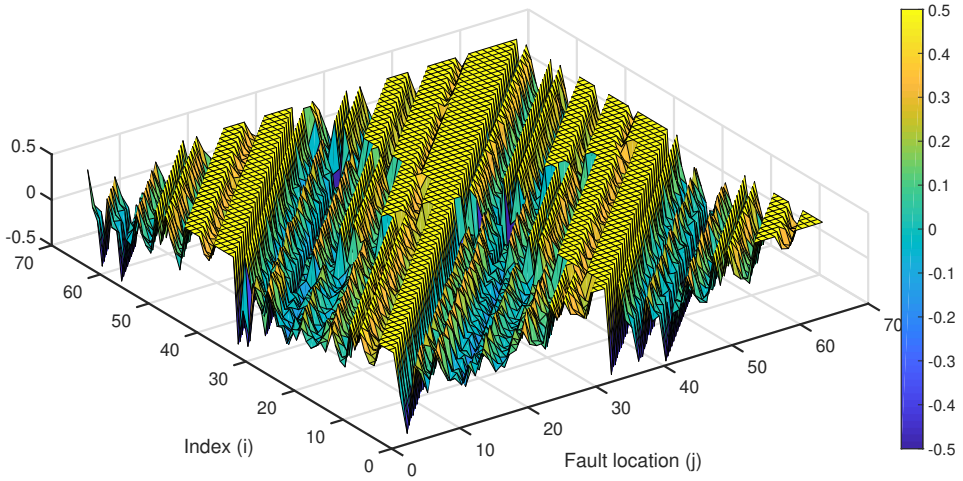
The signatures $S^{(0)}$, $S^{(1)}$, $\cdots$, $S^{(2n-1)}$ are stored for all possible $2n$ fault locations.



(a) SIMON32/64



(b) SIMON48/96



(c) SIMON64/128

Figure: The plot of s_i^j on index (i) and fault location (j) for faults injected in $(T-5)^{th}$ round

Online Phase

- A fault free ciphertext C is obtained for P using key K
- Obtain faulty ciphertext $C^{(\gamma)}$ from P and K by injecting a fault at location γ, $0 \leq \gamma \leq 2n-1$, in the internal state $\mathcal{S}_r$.
- After having λ faulty ciphertexts, calculate *trail* for each $C^{(\gamma)}$

Trail of $C^{(\gamma)}$

$$\tau^{(\gamma)} = (\psi_0^{(\gamma)}, \psi_1^{(\gamma)}, \ldots, \psi_{(2n-1)}^{(\gamma)}), \tag{3}$$

where $\psi_i^{(\gamma)}$ is:

$$\psi_i^{(\gamma)} = \frac{1}{2} - (c_i \oplus c_i^{(\gamma)}). \tag{4}$$

Identifying γ

For each faulty ciphertext $C^{(\gamma)}$, γ is identified by matching $\tau^{(\gamma)}$ and $S^{(j)}$.

Mismatch between a Signature and a Trail

$(s_i^j = \frac{1}{2}, \psi_i^\gamma = -\frac{1}{2})$ or $(s_i^j = -\frac{1}{2}, \psi_i^\gamma = \frac{1}{2}) \Rightarrow$ *Mismatch* for atleast one $j,\ 0 \leq j \leq (2n-1)$.

- For each $\tau^{(\gamma)}$, the adversary calculates the correlation $\mu(S^{(j)}, \tau^{(\gamma)})$ and $\alpha(S^{(\gamma)}) = |\{j : (\mu(S^{(j)}, \tau^{(\gamma)})) > \mu(S^{(\gamma)}, \tau^{(\gamma)})\}|$.
- For every γ, a table $T_{(\gamma)}$ is prepared, in which each fault location j is arranged in the decreasing order of the correlation coefficient $\mu(S^{(j)}, \tau^{(\gamma)})$.

Obtaining the set of Fault locations

- Consider all possible set $S^{(\gamma)}$ of fault locations, where $S^{(\gamma)} = \{j : (\mu(S^{(j)}, \tau^{\gamma})) \geq \mu(S^{\gamma}, \tau^{\gamma})\}$ and $\alpha(S^{(\gamma)}) = |S^{(\gamma)}|$.
- For λ faults injected, $(\alpha(S^{\gamma}))^{\lambda}$ many possible combinations of fault locations are needed.

Table: Expected number of times the SAT solver needs to be run to arrive at a correct set of fault locations.

SIMON2n/4n Variant	Round injected	Number of Faults (λ)	$\alpha(S^{\gamma})$	Number of times SAT solver is run ($=(\alpha(S^{\gamma}))^{\lambda}$)
SIMON32/64	27	4	$9.13 \approx 2^{3.191}$	$2^{12.764}$
SIMON48/96	31	6	$10.07 \approx 2^{3.345}$	$2^{20.070}$
SIMON64/128	39	9	$39.49 \approx 2^{5.311}$	$2^{47.799}$

Recovering the secret key

- Inject λ many faults in the r^{th} round of the internal state register $\mathcal{S}_r$; resetting the cipher to its original state post every fault injection
- Denote fault-free ciphertext by C_0 and the λ faulty ciphertexts by $C_1, C_2, \ldots, C_\lambda$.
- We consider $r = T - 5$ for each variant of SIMON$2n/4n$

Recovering the secret key

Every fault injected in $\mathcal{S}_r$ propagates to the next round $\mathcal{S}_{r+1}$ as per the construction of SIMON

- A fault injected in register L_r propagates as follows:

$$L_{r+1} = F(L_r^*) \oplus R_i \oplus k_r \tag{5}$$
$$R_{r+1} = L_r^* \tag{6}$$

- A fault injected in register R_r, we have

$$L_{r+1} = F(L_r) \oplus R_i^* \oplus k_r \tag{7}$$
$$R_{r+1} = L_r \tag{8}$$

Recovering the secret key

- Initialize $2n$ variables $L_{r,0} \ldots, L_{r,(n-1)}$ and $R_{r,0} \ldots R_{r,(n-1)}$ for the state of SIMON at round r, where $L_{r,j}$ $(R_{r,j})$ is the j^{th} bit of the left (right) block of the internal state $\mathcal{S}_r$.
- After every state update, the variables will be initialized as:

$$L_{r+1,j} = (L_{r,(j-1)\ mod(n)} \ \& \ L_{r,(j-8)\ mod(n)}) \oplus L_{r,(j-2)\ mod(n)} \tag{9}$$

$$\oplus R_{r,j} \oplus k_{r,j} \tag{10}$$

$$R_{r+1,j} = L_{r,j} \tag{11}$$

for $j = 0, 1, \ldots, (n-1)$.

Recovering the secret key

- Introduce new state variables $L_{r'+1,0} \ldots, L_{r'+1,(n-1)}, R_{r'+1,0}, \ldots R_{r'+1,(n-1)}$ for each round r' and formulate equations as:

$$0 \equiv L_{r'+1,j} \oplus (L_{r',(j-1) \ mod(n)} \ \& \ L_{r',(j-8) \ mod(n)}) \oplus L_{r',(j-2) \ mod(n)} \oplus R_{r',j} \oplus k_{r',j} \quad (12)$$

$$0 \equiv R_{r'+1,j} \oplus L_{r',j} \quad (13)$$

- We obtain $5 \cdot 2n = 10n$ variables and $5 \cdot 2 \cdot 2n = 20n$ equations
- We have $(\lambda + 1)$ cipher-texts, hence we have $10n \cdot (\lambda + 1)$ variables and $20n \cdot (\lambda + 1)$ equations, forming a system of Boolean equations.
- We use SAT solver to obtain a solution set satisfying these equations

Recovering the secret key

- The complexity of the equations increases drastically with the increase in the number of rounds
- We guess the entire register R and these guessed bits are substituted directly into the equations.
- The sharp rise in non-linearity of the equations prevents us from formulating equations for faults injected before $T - 7$ rounds
- The computation time of the same increases significantly.

Experimental Results

Table: Fault requirements for DFA on SIMON.

	Round fault is injected in	Number of Faults (λ)	Number of bits guessed in R	Time taken by SAT solver
SIMON32/64	27	4	16	191.230 *sec*
SIMON48/96	31	6	24	290.997 *sec*
SIMON64/128	39	9	32	403.035 *sec*

These experiments were conducted on a consumer grade laptop HP-15D103TX with CPU specifications Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz running SageMath version 8.1 along with Cryptominisat package on Ubuntu Bionic Beaver (development branch).

Experimental Results

Table: Comparison of the experimental number of the fault injections

SIMON2n/mn	Random n-bit model	Random byte model	Random bit model		
	Takahashi et al.	Tupsamudre et al.	Tupsamudre et al.	Vasquez et al.	This work
SIMON32/64	12.20	24	100	50.85	4
SIMON48/96	13.22	36	172	87.19	6
SIMON64/128	13.93	52	248	126.29	9

Table: Comparison of the rounds in which faults are injected in case of each attack model

SIMON2n/mn	Random n-bit model	Random byte model	Random bit model		
	Takahashi et al.	Tupsamudre et al.	Tupsamudre et al.	Vasquez et al.	This work
SIMON32/64	$L_{27}, L_{28}, L_{29}, L_{30}$	$L_{27}, L_{28}, L_{29}, L_{30}$	$L_{27}, L_{28}, L_{29}, L_{30}$	L_{27}, L_{29}	L_{27}, R_{27}
SIMON48/96	$L_{31}, L_{32}, L_{33}, L_{34}$	$L_{31}, L_{32}, L_{33}, L_{34}$	$L_{31}, L_{32}, L_{33}, L_{34}$	L_{31}, L_{33}	L_{31}, R_{31}
SIMON64/128	$L_{39}, L_{40}, L_{41}, L_{42}$	$L_{39}, L_{40}, L_{41}, L_{42}$	$L_{39}, L_{40}, L_{41}, L_{42}$	L_{39}, L_{41}	L_{39}, R_{39}

Time Complexity

Our attack procedure consists of the following two steps:

1. locating the faults using correlation between faulty and fault-free cipher texts,
2. deriving the secret key by formulating equations from cipher texts.

Consider

- the number of times the SAT solver needs to be run to arrive at a correct set of fault locations $= 2^x$
- number of bits guessed by SAT solver to derive the key $= w$

Then the time complexity of the attack $= 2^x * 2^w * c$,
where c is the time complexity of each execution of the SAT solver.

Time Complexity

Table: Time complexities of DFA on variants of SIMON.

	Fault Requirements	w	x	Time Complexity
SIMON32/64	4	16	12.76	$2^{28.76} \cdot c$
SIMON48/96	6	24	20.07	$2^{44.07} \cdot c$
SIMON64/128	9	32	47.80	$2^{79.80} \cdot c$

Future Work

- Extension of this work to remaining variants of SIMON
- Can this model of attack be mounted on SPECK ?
- How can we adapt this framework of attack to other block ciphers ?
- Is there a better technique to identify fault loations ?

Conclusion

We presented a Differential Fault Attack on SIMON.

- We showed how one can identify the location of injected faults using signatures
- We recovered the key by injecting as few as $4, 6$ and 9 faults in the $(T - m - 1)^{th}$ round of SIMON32/64, SIMON48/96 and SIMON64/128 respectively
- Our work does not compromise its security in normal mode, the attack is achievable under certain constrained environment.

Thank You