Differential Fault Analysis on SMS4 Using a Single Fault

Ruilin Li[1], Bing Sun[1], Chao Li[1,2], and JianXiong You[1]

[1]Department of Mathematics and System Science, Science College, National
University of Defense Technology, Changsha, 410073, China
securitylrl@gmail.com
[2]State Key Laboratory of Information Security, Institute of Software,
Chinese Academy of Sciences, Beijing, 100190, China

Abstract. Differential Fault Analysis (DFA) attack is a powerful cryptanalytic technique that could be used to retrieve the secret key by exploiting computational errors in the encryption (decryption) procedure. In the present paper, we propose a new DFA attack on SMS4 using a single fault. We show that if a random byte fault is induced into either the second, third, or forth word register at the input of the 28-th round, the 128-bit master key could be recovered with an exhaustive search of 22.11 bits on average. The proposed attack makes use of the characteristic of the cipher's structure, the speciality of the diffusion layer, and the differential property of the S-box. Furthermore, it can be tailored to any block cipher employing a similar structure and an SPN-style round function as that of SMS4.

Keywords: fault attacks, differential fault analysis, block cipher, SMS4

1 Introduction

Fault attacks are where an adversary tries to derive the secret key by accidental or intentional injecting faults in a cryptographic device during its computation of an algorithm. The idea of fault attack was introduced by Boneh, Demillo, and Lipton [3] from Bellcore in 1996. They exploited errors injected during the encryption process and showed that a single faulty encryption could break a CRT-RSA based signature cryptosystem. Later on, such kind of attack was extended by Biham and Shamir to DES-like secret key cryptosystems together with the technique of differential cryptanalysis [5] and referred as *Differential Fault Analysis* (DFA) [4]. Since then, DFA attack has applied to many other block ciphers, such as AES [7, 10, 13, 21, 23], ARIA [19], IDEA [8], Camellia[32], CLEFIA [9, 28], etc.

When applying fault attacks, it is usually assumed that the adversary has physical access to the tamper-proof device under attack and that he could induce faults by some special equipments. There are lots of methods for fault injection [1, 2, 6, 25], such as changing the power supply voltage or the frequency of the external clock, varying the environmental temperature, and exposing the circuits of the device to intense lights or lasers. Most of these methods could induce faults at byte level, due to the 8-bit size of a register for most current cryptographic security modules (e.g. smart cards).

Generally speaking, most DFA attacks against block ciphers target the last few rounds, i.e. they exploit computational errors in the last few rounds to extract the secret key. However, in 2003, Hemme showed the possibility of breaking DES with injected faults in the early rounds [14]. And recently, Rivain demonstrates the feasibility of recovering DES key even when faults are injected in the middle rounds [24]. These significant results again confirm that fault attack is really a terrible threat for many real life cryptosystems and it may be not sufficient to protect only the last few rounds of a cipher against fault attacks

SMS4 is the underlying block cipher used in the WAPI standard, which is the Chinese national standard for securing Wireless LANs. The detail of SMS4 was made public in 2006 by the Chinese government [26] and its English version was translated by Diffie and Ledin [11] at the end of 2008. After its publication, there are many traditional cryptanalytic works evaluating its security including differential attack [30, 31], linear attack [12], integral attack [16], algebraic attack [15], rectangle attack [20, 27, 30] and impossible differential attack [20, 27]. Besides traditional cryptanalysis, several authors mounted DFA attacks on SMS4 (see e.g. [17, 18, 29]).

In the present paper, we propose a new DFA attack on SMS4 using a single fault. We generalize the attack described by Takahashi et al. in [28] and consider a more realistic fault model. The main idea is based on the observation of the special characteristic of the cipher's structure and its round function. We show that if a random byte fault is induced into either the second, third or forth word register at the input of the 28-th round, the 128-bit master key could be derived with an exhaustive search of 22.11 bits on average. Moreover, by using the concept of *differential distribution table* of the S-box, the efficiency of the proposed attack could be greatly improved, which has been verified by our computer simulations.

This paper is organized as follows: a brief description of SMS4 is described in Section 2, some useful properties of the components of SMS4 related to our fault attack is proofed in Section 3. Fault model and attack procedure are proposed in Section 4. Section 5 includes some simulation results of our fault attack on SMS4. Finally, Section 6 concludes this paper.

2 Description of SMS4 Algorithm

2.1 Notation

The following notations are used throughout this paper.

- $\mathbb{F}_2$ denotes the finite field with elements 0 and 1.
- $\mathbb{F}_2^8$ denotes the set of 8-bit bytes.
- $\mathbb{F}_2^{32}$ denotes the set of 32-bit words.
- Given a word $U \in \mathbb{F}_2^{32}$, $U \lll n$ denotes left rotation of U by n bits.

- Any word $U \in \mathbb{F}_2^{32}$ can be divided into four bytes (u_0, u_1, u_2, u_3), where $u_i \in \mathbb{F}_2^8$, $i = 0, 1, 2, 3$.

2.2 Encryption and Decryption

SMS4 is a 128-bit block cipher with 128-bit key length. It iterates a simple round function 32 times. The encryption and decryption of SMS4 share the same procedure except that the round sub-keys for decryption are used in the reverse order. The overall structure of SMS4 is depicted in Fig.1 and the encryption procedure is described below.

1. The 128-bit plaintext is divided into four 32-bit words (X_0, X_1, X_2, X_3).
2. For $i = 0$ to 31, the words are updated according to the following rule:

$$\begin{aligned}(X_i, X_{i+1}, X_{i+2}, X_{i+3}) &\mapsto (X_{i+1}, X_{i+2}, X_{i+3}, X_{i+4})\\ X_{i+4} &= X_i \oplus F(X_{i+1} \oplus X_{i+2} \oplus X_{i+3}, RK_i)\end{aligned}$$

   where $F : \mathbb{F}_2^{32} \times \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is the round function and RK_i is the round-key.
3. The ciphertext is obtained through the following switch transform R,

$$(Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32}) \ .$$

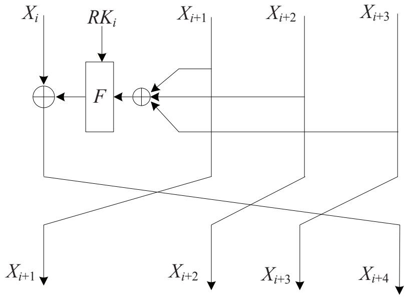

Fig. 1. The overall structure of SMS4

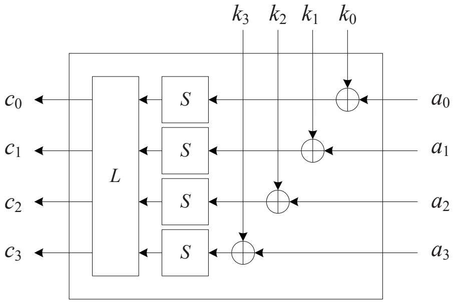

Fig. 2. The round function of SMS4

The round function of SMS4, as depicted in Fig. 2, is composed of three parts: the round-key addition layer σ, the substitution layer τ and the diffusion layer L, which are described as follows:

- The round-key addition $\sigma : \mathbb{F}_2^{32} \times \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is simply XORed the input A with a round key K, i.e.

$$\sigma(A, K) = \sigma_K(A) = A \oplus K = (a_0 \oplus k_0, a_1 \oplus k_1, a_2 \oplus k_2, a_3 \oplus k_3) \ .$$

- The nonlinear transformation $\tau : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ applies four S-boxes in parallel. Let B be the output of τ, and $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ be an 8×8 S-box, then
$$B = \tau(A \oplus K) \Leftrightarrow (b_0, b_1, b_2, b_3) = (S(a_0 \oplus k_0), S(a_1 \oplus k_1), S(a_2 \oplus k_2), S(a_3 \oplus k_3)) \ .$$
- The linear transformation $L : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ is defined as follow
$$C = L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24) \ ,$$
where C is the output of L, B is the input of L as well as the output of τ.

2.3 Key Schedule

SMS4 only supports 128-bit key and its key schedule is similar to the encryption function. A 128-bit master key is passed to the key schedule to generate 32 words in total for round keys.

Given the system parameter $FK = (FK_0, FK_1, FK_2, FK_3)$, and the fixed parameters $CK = (CK_0, CK_1, \ldots, CK_{31})$, both FK_i, $i = 0, 1, 2, 3$, and CK_j, $j = 0, 1, \ldots 31$, are some constant words which can be found in [26].

Let the master key be $MK = (MK_0, MK_1, MK_2, MK_3)$, then the generation of the round keys $(RK_0, RK_1, \ldots, RK_{31})$ can be described as follows:

1. $(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$.
2. For $i = 0, 1, \ldots, 31$,
$$RK_i = K_{i+4} = K_i \oplus L' \circ \tau \ (K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i) \ ,$$

where the non-linear transformation function $\tau(\cdot)$ is the same as that of the encryption function and the linear transformation of $L'(\cdot)$ is defined by $L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23)$.

The procedure of the round key generation indicates that the master key can be easily retrieved from any four consecutive round keys.

3 Some Properties of the Components of SMS4

In this section, several properties of the components of SMS4 are studied, which are related to our fault attack. Their proofs can be found in Appendix A.

Definition 1. *(Differential distribution table) Let $S : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ be an 8×8 S-box and let $\#\Omega$ represent the cardinality of the set Ω. Given $\alpha, \beta \in \mathbb{F}_2^8$, let*
$$\begin{aligned} \mathrm{IN}_S(\alpha, \beta) &= \ \ \{x \in \mathbb{F}_2^8 : S(x) \oplus S(x \oplus \alpha) = \beta\}, \\ \mathrm{N}_S(\alpha, \beta) &= \#\{x \in \mathbb{F}_2^8 : S(x) \oplus S(x \oplus \alpha) = \beta\}, \end{aligned}$$
then the differential distribution table of $S(\cdot)$ is defined by the table that is composed of all possible $(\alpha, \beta, \mathrm{N}_S(\alpha, \beta))$. The row (column) of the table corresponds to α (β), and its entry is $\mathrm{N}_S(\alpha, \beta)$.

Proposition 1. *For the S-box of* SMS4*, given any input difference* $\alpha \neq 0$*, there exist* 127 *possible output differences, of which* 1 *output difference satisfies* $\mathrm{N}_S(\alpha, \beta) = 4$*, and each of the other* 126 *output differences satisfies* $\mathrm{N}_S(\alpha, \beta) = 2$*.*

From Definition 1 and Proposition 1, we can apply differential attack to the S-box of SMS4 in the following model.

Differential Attack Model of the S-box. Given an 8×8 S-box $S(\cdot)$, let the encryption function be $y = S(x \oplus k)$, where x is the input, k is the encryption key, and y is the output. Assume an adversary could get an input pair as (x, x^*), however, he only knows the output difference $\beta = y \oplus y^* = S(x \oplus k) \oplus S(x^* \oplus k)$. How can he derive the encryption key k or the key candidates from the triplet (x, x^*, β)?

One can refer Appendix B for the detail of the differential attack on an S-box. The key point is using the concept of *differential distribution table*, by which one triplet (x, x^*, β) could greatly decreases the key candidates from 2^8 to at most 4 (the case for the S-box of SMS4).

In fact, the triplet (x, x^*, β) corresponds to the following equation

$$S(x \oplus k) \oplus S(x^* \oplus k) = \beta, \text{ with } k \text{ the indeterminate },$$

and

$$x \oplus \mathrm{IN}_S(x \oplus x^*, \beta) = \{x \oplus z : z \in \mathrm{IN}_S(x \oplus x^*, \beta)\}$$

is just the solution of the above equation, thus also the candidate set for the right key.

Remark 1. To obtain the key candidates in the differential attack model of the S-box, it is natural that one can try each possible value $gk \in \mathbb{F}_2^8$, then verifies whether or not $S(x \oplus gk) \oplus S(x^* \oplus gk) = \beta$. This brute-force attack would lead to 2^9 table-lookups. However, if the set $\mathrm{IN}_S(\alpha, \beta)$, with all possible (α, β), is stored in a table in advance, a more efficient attack could be applied by using only one table-lookup as described in Appendix B.

Remark 2. Sometimes, when an adversary faces the above differential attack model of the S-box, the two inputs (x, x^*) as well as their output difference β are not necessary the exact values, since the triplet (x, x^*, β), or part of it, may be obtained through a key guess on some known (exact) values, thus such triplet should be treated as a random one. In other words, if (x, x^*, β) is obtained through the right key guess, then it always leads to the set $x \oplus \mathrm{IN}_S(x \oplus x^*, \beta)$ containing the right key. However, if (x, x^*, β) is obtained through a wrong key guess, it would lead to some other candidate key set, which does not necessarily contain the right key. Even in some special cases, the random triplet (x, x^*, β) results in an empty candidate key set which indicates a wrong key guess.

As discussed above, the following situation should be considered: given a random triplet (x, x^*, β), what's the property of the solution for the equation $S(x \oplus k) \oplus S(x^* \oplus k) = \beta$? The following proposition answers such question and it describes the average cardinality of the candidate key set if the equation has any solution.

Proposition 2. *Let $S(\cdot)$ be the S-box of* SMS4*, (x, x^*, β) be a random triplet in $\mathbb{F}_2^8$, then the following results hold:*

(1) $\mathrm{N}_S(x \oplus x^*, \beta) > 0$ *is satisfied with probability* 0.4942*, or in other words, the equation $S(x \oplus k) \oplus S(x^* \oplus k) = \beta$ has solutions with probability* 0.4942*.*
(2) *If $\mathrm{N}_S(x \oplus x^*, \beta) > 0$, then the expectation of $\mathrm{N}_S(x \oplus x^*, \beta)$ is* 2.0236*. That is to say, if $S(x \oplus k) \oplus S(x^* \oplus k) = \beta$ has any solution, the expectation of the number of solutions is* 2.0236*.*

Next, we present some properties with the linear transformation in the diffusion layer. We mainly discuss the differential brunch number and the inversion expression of the linear transformation L.

Definition 2. *(Differential branch number) Let $L : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$ be a linear transformation, $W(\cdot)$ denotes the byte weight function, that is the number of non-zero bytes, then the differential branch number of L is defined by*

$$\mathcal{B}(L) = \min_{a \in \mathbb{F}_2^{32},\, a \neq 0} (W(a) + W(L(a))) \ .$$

Differential branch number is a good concept for measuring the diffusion effect of a transformation. By computer program, we know that the differential branch number of L in SMS4 is 5, which ensures that input difference with one non-zero byte will lead to output difference with four non-zero bytes. Moreover, if $\mathcal{B}(L) = 5$, by Def. 2, one can easily proof that $\mathcal{B}(L^{-1}) = 5$, where L^{-1} denotes the inversion of L, whose expression is deduced by the following proposition.

Proposition 3. *The inversion of the linear transformation $L(\cdot)$ of* SMS4 *has the following expression:*

$$\begin{aligned} L^{-1}(C) = C &\oplus (C \lll 2) \oplus (C \lll 4) \oplus (C \lll 8) \oplus (C \lll 12) \oplus (C \lll 14) \\ &\oplus (C \lll 16) \oplus (C \lll 18) \oplus (C \lll 22) \oplus (C \lll 24) \oplus (C \lll 30) \ . \end{aligned}$$

By the expression of L^{-1}, the differential attack on the S-box can be easily extended to the round function $F(A, K) = L \circ \tau \circ \sigma_K(A)$, since from the output difference of F, one can easily deduce the output difference of τ, thus he can apply differential attack to each S-box independently.

4 Proposed DFA Attack on SMS4

In this section, we firstly summarize previous fault attacks on SMS4, then propose our fault attack, including the fault model, main idea, attack procedure and complexity analysis.

4.1 Previous DFA Attacks on SMS4

There are three DFA attacks on SMS4 reported in the literature and we summarized them as follows:

The first fault attack on SMS4 was proposed in [29]. By using the byte-oriented model, the 128-bit key could be recovered with 32 faults ideally. The deficiency of such attack is that it can only recover the same round key when injecting faults in some round. Moreover, at least two faults are needed to deduce one byte of the round key in their attack model, thus decreasing the efficiency of fault injections.

An improved fault attack on SMS4 was presented in [17]. By injecting one random byte fault into some word at the input of the 29-th and 27-th round, respectively, the authors claimed that the 128-bit key could be derived efficiently. This improved attack is mainly based on the maximum diffusion property of the linear transform.

Another kind of fault attack [18] on SMS4 is based on injecting faults into the key schedule of SMS4. After carefully studying the property of the round key generation, the authors proofed that 8 or 32 faults are needed to retrieve the master key according to different fault injection points.

4.2 Fault Model and Main Idea

Our proposed fault attack adopts the byte-oriented model, more precisely, it uses the following realistic assumptions:

- The adversary can obtain a pair of correct and faulty ciphertexts both corresponding to the same plaintext and the unknown key.
- The adversary knows the area of the fault injection, e.g. he could inject a random byte fault into the first, second, third or forth word at the input of the 28-th round.
- The adversary does not know either the location of the byte in the word or the value of the fault.

All previous fault attacks on SMS4 are based on the differential attack on the S-box as described in Appendix B, thus by injecting sufficient faults, the last four round keys could be uniquely retrieved. The main idea of our proposed attack, however, is only to deduce the candidates for the last four round keys, then a brute-force attack is needed to find the right one. The attack procedure is briefly described as follow:

- Randomly choose a plaintext, obtain the correct ciphertext.
- For the same plaintext, inject a random byte fault into either the second, third or forth word at the input of the 28-th round, and obtain the faulty ciphertext.
- According to the cipher's structure, apply the *basic attack of the round function*, as will be described later, to the 32-nd, 31-st, 30-th, and 29-th round in sequence, obtain the last four round-key candidates.
- Apply brute-force attack on these candidates to retrieve the master key.

4.3 Attack Procedure

In this subsection, we describe the detailed procedure of the proposed fault attack on SMS4. Without loss of generality, assume a random byte fault occurs at the forth word of the 28-th round (Faults occur at the second or third word are similar to analyze). As shown in Fig.3, this new attack applies differential attack to the last 4 rounds of SMS4, and can reduce the key space from 2^{128} to $2^{22.11}$ on average, thus an exhaustive search is feasible.

We firstly introduce the following notations:

- $A_i = (a_{i,0}, a_{i,1}, a_{i,2}, a_{i,3})$ denotes the input of the round function F in the i-th round, $B_i = (b_{i,0}, b_{i,1}, b_{i,2}, b_{i,3})$ and $C_i = (c_{i,0}, c_{i,1}, c_{i,2}, c_{i,3})$ denote the output of the non-linear function $\tau(\cdot)$ and linear function $L(\cdot)$ in the i-th round, respectively, where each a_i, b_i, $c_i \in \mathbb{F}_2^8$, $i = 1, 2, \ldots, 32$.
- For any word of the correct intermediate state, say W, W^* denotes the counterpart of the faulty intermediate state, and ΔW denotes their difference, i.e. $\mathit{\Delta} W = W \oplus W^*$.
- For any word $U = (u_0, u_1, u_2, u_3) \in \mathbb{F}_2^{32}$, $(U)_i$ denotes the i-th byte of U, i.e. $(U)_i = u_i$.
- $\langle RK_i, RK_{i-1}, \ldots, RK_j \rangle$ denotes the candidate set for round-keys RK_i, RK_{i-1}, $\ldots$, RK_j, where $0 \le j \le i \le 31$.

Basic Attack of the Round Function[1]**.** Given the round function of SMS4 as $F(A, K) = L \circ \tau \circ \sigma_K(A)$, assume the adversary obtains a 32-bit triplet $(A, A^*, \Delta C)$, where (A, A^*) is the input pair for F and ΔC is the the output difference. Both (A, A^*) and ΔC can be either the known values (exact) or guessed values (obtained from other round key candidate). The following basic attack of the round function could be applied to retrieve the round-key candidate set $\langle K \rangle$ and meanwhile reduce the size of possible values of the other round key candidates, by which this triplet is obtained.

1. Compute $\Delta B = L^{-1}(\Delta C)$.
2. For $i = 0, 1, 2, 3$, calculate
   (a) $a_i = (A)_i$, $a_i^* = (A^*)_i$, $\Delta b_i = (\Delta B)_i$;
   (b) $\langle k_i \rangle = a_i \oplus \mathrm{IN}_S(a_i \oplus a_i^*, \Delta b_i)$;
3. If for each $i \in \{0, 1, 2, 3\}$, $\langle k_i \rangle \neq \emptyset$, then the round key candidate set must be $\langle K \rangle = \langle k_0 \rangle \| \langle k_1 \rangle \| \langle k_2 \rangle \| \langle k_3 \rangle \triangleq \{\, gk_0 \| gk_1 \| gk_2 \| gk_3 : gk_i \in \langle k_i \rangle \,\}$, with $\|$ the concatenation.
4. If there exists some $i \in \{0, 1, 2, 3\}$, such that $\langle k_i \rangle = \emptyset$, then this triplet $(A, A^*, \Delta C)$ indicates no key candidates, i.e. the round-key candidate set $\langle K \rangle = \emptyset$. Meanwhile, this also implies that the guessed key (other round key candidate), by which this triplet $(A, A^*, \Delta C)$ is obtained, is incorrect.

According to Proposition 2, given a 32-bit random triplet $(A, A^*, \Delta C)$ for the round function F, the following results hold:

[1] The idea of this basic attack of the round function is the same as described in [28]. Note that the concept of differential distribution table of the S-box is used to significantly reduce the time complexity of the attack.

- The above attack will output a non-empty round-key candidate set $\langle K\rangle$ with probability $0.4942^4 \approx \left(2^{-1.107}\right)^4 = 2^{-4.068}$. This implies that the size of the guessed keys (other round key candidates), by which this triplet is obtained, could be reduced about $2^{-4.068}$.
- If there exists any round-key candidate, the expectation value of $\langle K\rangle$ is $2.0236^4 \approx \left(2^{1.017}\right)^4 = 2^{4.068}$.

Now the detailed attack procedure is described in the following three steps:

Step 1. Obtain the correct and faulty ciphertext. Randomly choose a plaintext $X = (X_0, X_1, X_2, X_3)$, and obtain the correct ciphertext $Y = (Y_1$, Y_2, Y_3, $Y_4)$ under the unknown encryption key MK. Assume the round keys generated by MK is RK_i, where $i = 0, 1, \ldots, 31$. For the same plaintext and the unknown key, inject a random byte fault into the forth word at the input of the 28-th round, obtain the fault ciphertext $Y^* = (Y_1^*, Y_2^*, Y_3^*, Y_4^*)$.

Step 2. Deduce $\langle \boldsymbol{RK_{31}}, \boldsymbol{RK_{30}}, \boldsymbol{RK_{29}}, \boldsymbol{RK_{28}}\rangle$. Due to the switch transformation, $(X_{32}, X_{33}, X_{34}, X_{35}) = (Y_3, Y_2, Y_1, Y_0)$ and $(X_{32}^*$, X_{33}^*, X_{34}^*, $X_{35}^*)$=$(Y_3^*$, Y_2^*, Y_1^*, $Y_0^*)$, thus both the correct and faulty output of the 32-nd round are known. For this output pair, by using the technique of differential attack, obtain the round-key candidates for the 32-nd, 31-st, 30-th and 29-th round.

Let $\Psi = \{(\delta, 0, 0, 0),\ (0, \delta, 0, 0),\ (0, 0, \delta, 0),\ (0, 0, 0, \delta) : 0 \neq \delta \in \mathbb{F}_2^8\}$ be the set that contains all possible values of random byte faults that occurs in ΔX_{30} at the input of the 28-th round, thus $\#\Psi = 255 \times 4 = 1020$.

Since a random byte fault is induced into X_{30} at the input of the 28-th round, we have

$$\Delta X_{27} = \Delta X_{28} = \Delta X_{29} = 0, \text{and } \Delta X_{30} \in \Psi.$$

Thus, the input difference of the 28-th round function is

$$\Delta A_{28} = \Delta X_{28} \oplus \Delta X_{29} \oplus \Delta X_{30} = \Delta X_{30} \in \Psi,$$

after passing through the substitution layer,

$$\Delta B_{28} \in \Psi.$$

By the cipher's structure,

$$\Delta X_{31} = \Delta X_{27} \oplus \Delta C_{28} = \Delta C_{28} = L(\Delta B_{28})\ .$$

According the above analysis, we have the following results:

- $\Delta X_{30} \in \Psi$, thus there are at most 1020 possible values for ΔX_{30}.
- $\Delta X_{31} = L(\Delta B_{31})$, $\Delta B_{31} \in \Psi$, thus there are at most 1020 possible values for ΔX_{31}.

Now we can describe Step 2 in the following four consecutive sub-steps.

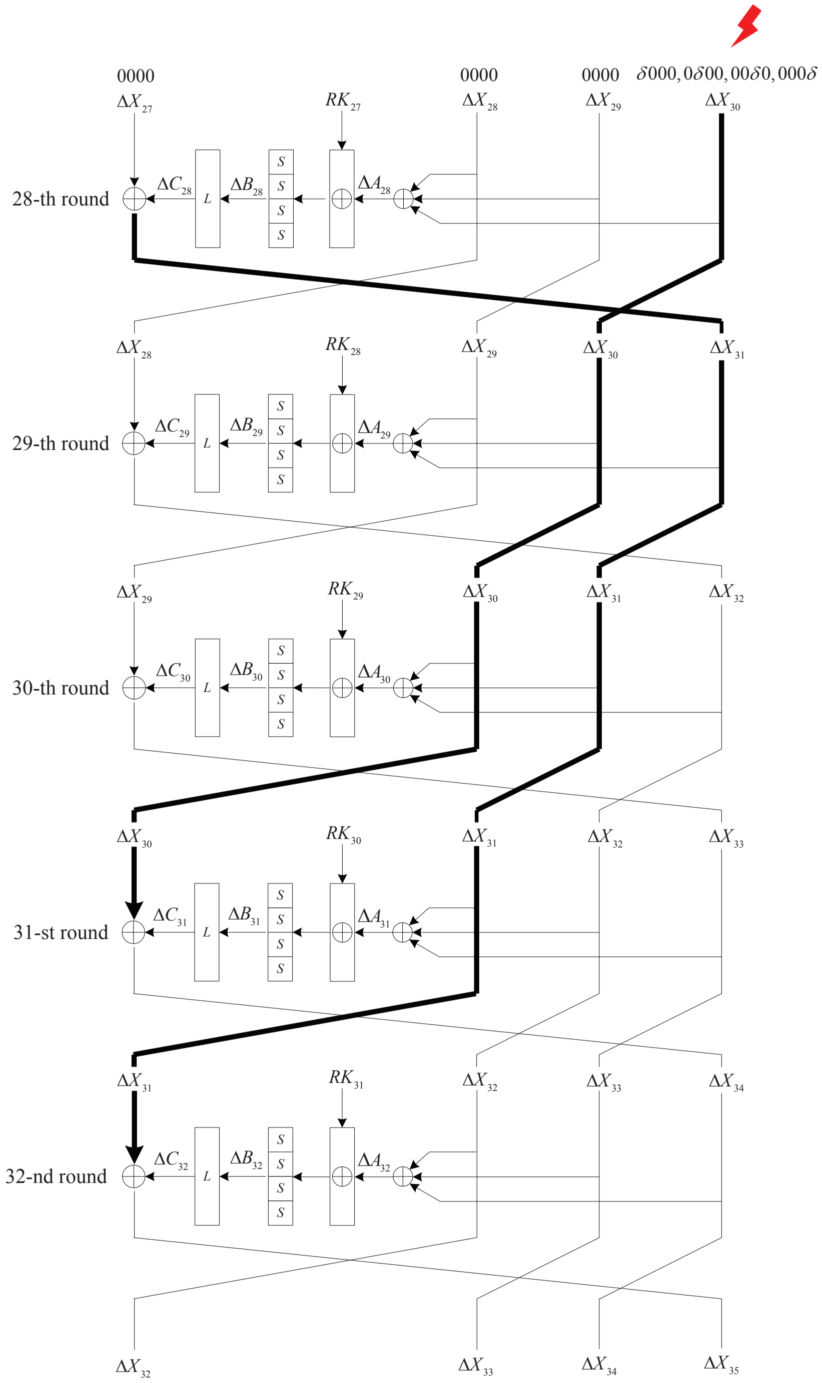

Fig. 3. DFA attack on SMS4 within last four rounds

Step 2.1 Deduce $\langle \boldsymbol{RK_{31}} \rangle$. Consider the 32-nd round, the correct as well as the faulty input of the round function F can be calculated as:

$$\begin{aligned} A_{32} &= X_{32} \oplus X_{33} \oplus X_{34}, \\ \text{and } A^*_{32} &= X^*_{32} \oplus X^*_{33} \oplus X^*_{34}. \end{aligned}$$

Meanwhile, the output difference of F is

$$\Delta C_{32} = \Delta X_{31} \oplus \Delta X_{35}.$$

Use all possible triplets $(A_{32}, A^*_{32}, \Delta C_{32})$ to apply the basic attack on the 32-nd round function, obtain the candidate set $\langle RK_{31} \rangle$.

Step 2.2 Deduce $\langle \boldsymbol{RK_{31}},\ \boldsymbol{RK_{30}} \rangle$. For each $gk_{31} \in \langle RK_{31} \rangle$, "decrypt" the ciphertext pair by one round and obtain:

$$X_{31} = X_{35} \oplus F(X_{32} \oplus X_{33} \oplus X_{34}, gk_{31}), X^*_{31} = X^*_{35} \oplus F(X^*_{32} \oplus X^*_{33} \oplus X^*_{34}, gk_{31}),$$

Consider the 31-st round, calculate [2] :

$$\begin{aligned} A_{31} &= X_{31} \oplus X_{32} \oplus X_{33}, \\ A^*_{31} &= X^*_{31} \oplus X^*_{32} \oplus X^*_{33}, \\ \text{and } \Delta C_{31} &= \Delta X_{30} \oplus \Delta X_{34}. \end{aligned}$$

Use all possible triplets $(A_{31}, A^*_{31}, \Delta C_{31})$ to apply the basic attack on the 31-st round function, obtain the candidate set $\langle RK_{31}, RK_{30} \rangle$.

Step 2.3 Deduce $\langle \boldsymbol{RK_{31}},\boldsymbol{RK_{30}},\boldsymbol{RK_{29}} \rangle$. For each $(gk_{31}, gk_{30}) \in \langle RK_{31}, RK_{30} \rangle$, "decrypt" the ciphertext pair by two rounds and obtain:

$$\begin{aligned} X_{31} = X_{35} \oplus F(X_{32} \oplus X_{33} \oplus X_{34}, gk_{31}),\ X^*_{31} = X^*_{35} \oplus F(X^*_{32} \oplus X^*_{33} \oplus X^*_{34}, gk_{31}), \\ X_{30} = X_{34} \oplus F(X_{31} \oplus X_{32} \oplus X_{33}, gk_{30}),\ X^*_{30} = X^*_{34} \oplus F(X^*_{31} \oplus X^*_{32} \oplus X^*_{33}, gk_{30}), \end{aligned}$$

Consider the 30-th round, calculate :

$$\begin{aligned} A_{30} &= X_{30} \oplus X_{31} \oplus X_{32}, \\ A^*_{30} &= X^*_{30} \oplus X^*_{31} \oplus X^*_{32}, \\ \text{and } \Delta C_{30} &= \Delta X_{29} \oplus \Delta X_{33} = \Delta X_{33}. \end{aligned}$$

Apply the basic attack on the 30-th round function via $(A_{30}, A^*_{30}, \Delta C_{30})$, obtain the candidate set $\langle RK_{31}, RK_{30}, RK_{29} \rangle$.

[2] Both X_{31} and X^*_{31} (thus A_{31} and A^*_{31}) are guessed values and they are not necessary the correct and faulty words unless the guessed round-key gk_{31} is RK_{31}, the same case also exists for some other intermediate states in the following sub-steps.

Step 2.4 Deduce $\langle \boldsymbol{RK_{31}}, \boldsymbol{RK_{30}}, \boldsymbol{RK_{29}}, \boldsymbol{RK_{28}} \rangle$. For each $(gk_{31}, gk_{30}, gk_{29}) \in \langle RK_{31}, RK_{30}, RK_{29} \rangle$, "decrypt" the ciphertext pair by three rounds and obtain:

$$\begin{aligned} X_{31} &= X_{35} \oplus F(X_{32} \oplus X_{33} \oplus X_{34}, gk_{31}), & X_{31}^* &= X_{35}^* \oplus F(X_{32}^* \oplus X_{33}^* \oplus X_{34}^*, gk_{31}), \\ X_{30} &= X_{34} \oplus F(X_{31} \oplus X_{32} \oplus X_{33}, gk_{30}), & X_{30}^* &= X_{34}^* \oplus F(X_{31}^* \oplus X_{32}^* \oplus X_{33}^*, gk_{30}), \\ X_{29} &= X_{33} \oplus F(X_{30} \oplus X_{31} \oplus X_{32}, gk_{29}), & X_{29}^* &= X_{33}^* \oplus F(X_{30}^* \oplus X_{31}^* \oplus X_{32}^*, gk_{29}), \end{aligned}$$

Consider the 29-th round, calculate :

$$\begin{aligned} A_{29} &= X_{29} \oplus X_{30} \oplus X_{31}, \\ A_{29}^* &= X_{29}^* \oplus X_{30}^* \oplus X_{31}^*, \\ \text{and } \Delta C_{29} &= \Delta X_{28} \oplus \Delta X_{32}. \end{aligned}$$

Apply the basic attack on the 29-th round function via $(A_{29}, A_{29}^*, \Delta C_{29})$, obtain the candidate set $\langle RK_{31}, RK_{30}, RK_{29}, RK_{28} \rangle$.

Step 3. Retrieve the master key *MK*. According to the key schedule, we use each possible 4-word round key candidate after Step 2 to decrypt the right ciphertext Y, then check whether the plaintext is X. Through a brute-force attack, there will be only one 4-word round-key candidate surviving the filtration, in which case, the master key MK can be easily deduced via key schedule (If not the case, try another plaintext/ciphertext pair to verify).

4.4 Complexity Analysis

As described in Section 4.3, to recover the encryption key MK, a brute-force attack is needed, thus we have to evaluate the expected value of the size of the round key candidate set $\langle RK_{31}, RK_{30}, RK_{29}, RK_{28} \rangle$ derived from Step 2.

Expected value of $\#\langle \boldsymbol{RK_{31}} \rangle$ after step 2.1. Consider the triplet (A_{32}, A_{32}^*, ΔC_{32}) for the 32-nd round function, since $\Delta B_{28} \in \Psi$, $\Delta X_{31} = L(\Delta B_{28})$, and $\Delta C_{32} = \Delta X_{31} \oplus \Delta X_{35}$, we have

$$\Delta B_{32} = L^{-1}\left(\Delta C_{32}\right) = L^{-1}(\Delta X_{31}) \oplus L^{-1}(\Delta X_{35}) = \Delta B_{28} \oplus L^{-1}(\Delta X_{35}).$$

Calculate $L^{-1}(\Delta X_{35}) = L^{-1}(X_{35} \oplus X_{35}^*) \triangleq (d_0, d_1, d_2, d_3)$, where $d_0, d_1, d_2, d_3 \in \mathbb{F}_2^8$, and all of them are known. Let $0 \neq \gamma \in \mathbb{F}_2^8$, thus ΔB_{32} must be one of the following 4 kinds of differences (in total there are 1020 possible values):

$$\begin{array}{llllr} (\gamma \oplus d_0, & d_1, & d_2, & d_3) & (1) \\ (\quad d_0, & \gamma \oplus d_1, & d_2, & d_3) & (2) \\ (\quad d_0, & d_1, & \gamma \oplus d_2, & d_3) & (3) \\ (\quad d_0, & d_1, & d_2, & \gamma \oplus d_3) & (4) \end{array}$$

Let $(\Delta a_{32,0}, \Delta a_{32,1}, \Delta a_{32,2}, \Delta a_{32,3}) = \left((\Delta A_{32})_0, (\Delta A_{32})_1, (\Delta A_{32})_2, (\Delta A_{32})_3\right)$, then for each $0 \leq i \leq 3$, one check whether or not $\mathrm{N}_S(\Delta a_{32,i}, d_i) > 0$. According to the above analysis, at lest three of (d_0, d_1, d_2, d_3) satisfy $\mathrm{N}_S(\Delta a_{32,i}, d_i) > 0$. Without loss of generality, assume

$$\mathrm{N}_S(\Delta_{32,0}, d_0) > 0, \mathrm{N}_S(\Delta_{32,1}, d_1) > 0, \mathrm{N}_S(\Delta_{32,2}, d_2) > 0,$$

so next, the adversary checks whether or not $\mathrm{N}_S(\Delta_{32,3}, d_3) > 0$:

- If $\mathrm{N}_S(\Delta a_{32,3}, d_3) = 0$, then differences corresponding to (1)(2)(3) should be discarded, which implies that the number of possible values of ΔB_{32} is 255 and that the exact position of the fault is the forth byte of X_{30}. In this situation, the basic attack of the 32-nd round will return $(2.0236)^3 \times 2^8 = 2^{11.051}$ round-key candidates on overage.
- If $\mathrm{N}_S(\Delta_{32,3}, d_3) > 0$, then the number of possible values of ΔB_{32} is 1020. Thus the basic attack of the 32-nd round will return $(2.0236)^3 \times (2^8 - 2.0236) \times 4 = 2^{13.039}$ round-key candidates on overage.

Now by Proposition 2, we can conclude that the expected value of $\#\langle RK_{\mathbf{31}}\rangle$ after Step 2.1 is $0.4942 \times 2^{13.039} + (1 - 0.4942) \times 2^{11.051} \approx 2^{12.353}$.

Expected value of $\#\langle \boldsymbol{RK_{31}}, \boldsymbol{RK_{30}}\rangle$ after step 2.2. For each candidate key $gk_{31} \in \langle RK_{31}\rangle$, we obtain the guessed triplet $(\Delta A_{31}, \Delta A^*_{31}, \Delta C_{31})$ to apply the basic attack of the 31-st round function. This would decrease the size of the possible round-key candidates for the 32-nd round, the detailed analysis is as follow:

By $\Delta C_{31} = \Delta X_{30} \oplus \Delta X_{34}$, we have

$$\Delta B_{31} = L^{-1}(\Delta C_{31}) = L^{-1}(\Delta X_{30}) \oplus L^{-1}(\Delta X_{34}).$$

Since ΔX_{34} is known, so is $L^{-1}(\Delta X_{34})$. Moreover, $\Delta X_{30} \in \Psi$ indicates that $L^{-1}(\Delta X_{30})$ has four non-zero bytes. Now according to the value of $\#\langle RK_{31}\rangle$, we can analyze this situation in the following two cases:

- If $\#\langle RK_{31}\rangle = 2^{11.051}$, then the number of possible values of ΔX_{30} is 255, thus the expected value of $\#\langle RK_{31}, RK_{30}\rangle$ after this step is $2^{11.051} \times 2^{-4.068} \times 2^{4.068} \times 255 = 2^{19.045}$.
- If $\#\langle RK_{31}\rangle = 2^{13.039}$, then the number of possible values of ΔX_{30} is 1020, thus the expected value of $\#\langle RK_{31}, RK_{30}\rangle$ after this step is $2^{13.039} \times 2^{-4.068} \times 2^{4.068} \times 1020 = 2^{23.033}$.

In total, we conclude that the expected value of $\#\langle RK_{31}, RK_{30}\rangle$ after this step is $0.4942 \times 2^{23.033} + (1 - 0.4942) \times 2^{19.045} \approx 2^{22.11}$.

Expected value of $\#\langle \boldsymbol{RK_{31}}, \boldsymbol{RK_{30}}, \boldsymbol{RK_{29}}\rangle$ after step 2.3. As discussed in Step 2.3, the expected value of $\#\langle RK_{31}, RK_{30}, RK_{31}\rangle$ after this step is $0.4942 \times 2^{23.033} \times 2^{-4.068} \times 2^{4.068} + (1 - 0.4942) \times 2^{19.045} \times 2^{-4.068} \times 2^{4.068} \approx 2^{22.11}$.

Expected value of $\#\langle \boldsymbol{RK_{31}}, \boldsymbol{RK_{30}}, \boldsymbol{RK_{29}}, \boldsymbol{RK_{28}}\rangle$ after step 2.4. As discussed in Step 2.4, the expected value of $\#\langle RK_{31}, RK_{30}, RK_{29}, RK_{28}\rangle$ after this step is $0.4942 \times 2^{23.033} \times 2^{-4.068} \times 2^{4.068} \times 2^{-4.068} \times 2^{4.068} + (1 - 0.4942) \times 2^{19.045} \times 2^{-4.068} \times 2^{4.068} \times 2^{-4.068} \times 2^{4.068} \approx 2^{22.11}$.

5 Simulation Results

We implement our proposed DFA attack on SMS4 in C++ code and execute it on a PC with Intel Pentium 1.80 GHz processor. Our simulation experiment is based on 1000 samples and the plaintext as well as the master key in each attack are randomly generated. The distributions of exhaustive search bits after each sub-steps in step 2 are depicted in Fig.4.

Our experimental result indicates that the average bit space for brute-force search after each sub-steps in step 2 is well agreed with the previous theoretical predications.

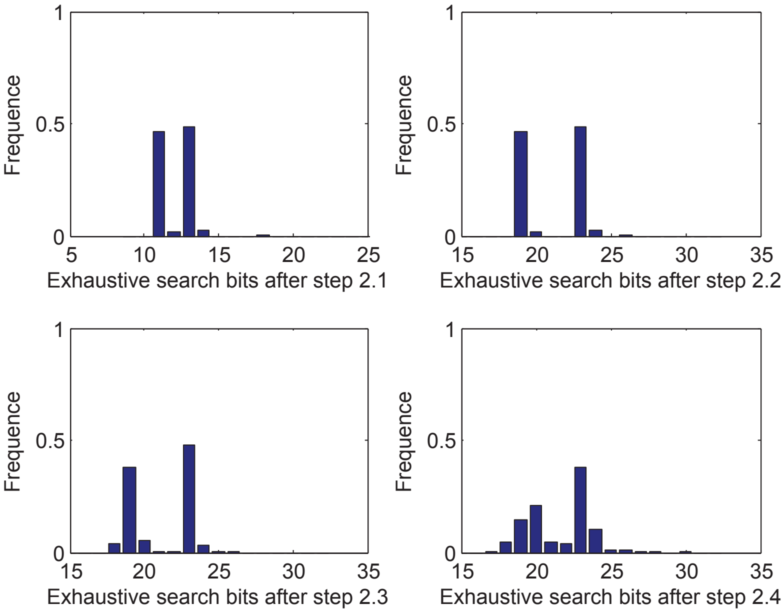

Fig. 4. Exhaustive search bits after each sub-steps in step 2

6 Conclusion

In this paper, we present a new DFA attack on SMS4 using a single fault. We show that if a random byte fault is injected into either the second, third or forth word at the input of the 28-th round, the 128-bit master key could be retrieved by applying an exhaustive search of 22.11 bits on average. Table 1 lists our work compared with previous fault attacks on SMS4 and Table 2 is the comparison of detailed fault injection points with different attack scenarios. These results indicate that SMS4 can be broken easily using fault based method, thus cryptographic devices supporting SMS4 should be carefully protected.

Table 1. Comparison with existing fault attacks

Fault Model	Fault Injection Region	No. of Fault Injection Points	No. of Faulty Encryptions	Brute-force Attack	Ref.
Disturb 1 byte	Data process	4	32	–	[29]
Disturb 1 byte	Data process	2	2	–	[17]
Disturb 1 byte	Key schedule	4	32	–	[18]
Disturb 1 byte	Key schedule	4	8	–	[18]
Disturb 1 byte	Data process	1	1	$2^{22.11}$	Sect. 4

Table 2. Comparison with existing fault attacks by fault locations

32-nd	31-st	30-th	29-th	28-th	27-th	Ref.
X_{32}, X_{33}, X_{34}	X_{31}, X_{32}, X_{33}	X_{30}, X_{31}, X_{32}	X_{29}, X_{30}, X_{31}	–	–	[29]
–	–	–	X_{28}, X_{29}	–	X_{26}, X_{27}	[17]
K_{32}, K_{33}, K_{34}	K_{31}, K_{32}, K_{33}	K_{30}, K_{31}, K_{32}	K_{29}, K_{30}, K_{31}	–	–	[18]
K_{31}	K_{30}	K_{29}	K_{28}	–	–	[18]
–	–	–	–	X_{28}, X_{29}, X_{30}	–	Sect. 4

It should be pointed out that our proposed fault attack can be extended to a more generalized case. Any block cipher that employs a similar structure and an SPN-style round function as that of SMS4 could be suffered from our attack. Assume such block cipher contains n sub-blocks with $n \geq 2$ ($n = 2$ corresponds to Feistel structure), by injecting a random byte fault into either the second, third, ..., or $(n-1)$-th word at the input of the last $(n+1)$-th round, the expected number of round-key candidates for the last n rounds could be significantly reduced. Even if the linear transformation of the round function is not optimal (i.e. the differential branch number of the linear transformation does not achieve the maximum), these round keys could be uniquely determined via a very small quantity of extra fault injections.

Acknowledgments

The work in this paper is supported by the Natural Science Foundation of China (No: 60803156) and the open research fund of State Key Laboratory of Information Security(No: 01-07).

References

1. R. Anderson and M. Kuhn. Tamper resistance – a cautionary note. Second USENIX workshop on eletronic commerce, 1996, pp. 1–11.
2. R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. Security Protocols 1997, LNCS 1361, pp. 125–136, Springer-Verlag, 1997.
3. Dan Boneh, Richard A. DeMillo, Richard J. Lipton. On the importance of checking cryptographic protocols for faults. EUROCRYPT'97, LNCS 1233, pp. 37–51, Springer-Verlag, 1997.
4. Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. CRYPTO 97, LNCS 1294, pp. 513–525, Springer-Verlag, 1997.
5. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. Journal of Cryptology, Vol 4, pp.3–72, Springer-Verlag, 1991.
6. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks. Proceedings IEEE Vol 94(2): 370–386, 2006.
7. Johannes Blömer and Jean-Pierre Seifert. Fault based cryptanalysis of the Advanced Encryption Standard (AES). FC 2003, LNCS 2742, pp. 162–181, Springer-Verlag, 2003.
8. Christophe Clavier, Benedikt Gierlichs, and Ingrid Verbauwhede. Fault analysis study of IDEA. CT-RSA 2008, LNCS 4964, pp. 274–287, Springer-Verlag, 2008.
9. Hua Chen, Wenling Wu, and Dengguo Feng. Differential fault analysis on CLEFIA. ICICS 2007, LNCS 4861, pp. 284–295, Springer-Verlag, 2007.
10. Pierre Dusart, Gilles Letourneux and Olivier Vivolo. Differential fault analysis on A.E.S. ACNS 2003, LNCS 2846, pp. 293–306, Springer-Verlag, 2003.
11. Whitfield Diffie and George Ledin(translators). SMS4 encryption algorithm for wireless networks. (English version of Ref.[26]) http://eprint.iacr.org/2008/329.
12. Jonathan Etrog and Matt J.B. Robshaw. The cryptanalysis of reduced-round SMS4. SAC 2008, LNCS 5381, pp. 51–65, Springer-Verlag, 2009.
13. Christophe Giraud. DFA on AES. AES 2004, LNCS 3373, pp. 27–41, Springer-Verlag, 2005.
14. Ludger Hemme. A differential fault attack against early rounds of (Triple-)DES. CHES 2004, LNCS 3156, pp. 254–267, Springer-Verlag, 2004.
15. Wen Ji, Lei Hu. New description of SMS4 by an embedding over GF(2^8). Indocrypt 2007, LNCS 4859, pp. 238–251, Springer-Verlag, 2007.
16. Fen Liu, Wen Ji, Lei Hu, Jintai Ding, Shuwang Lv, Andrei Pyshkin, and Ralf-Philipp Weinmann. Analysis of the SMS4 block cipher. ACISP 2007, LNCS 4586, pp. 158–170, Springer-Verlag, 2007.
17. Wei Li, Dawu Gu. An improved method of differential fault analysis on the SMS4 cryptosystem. ISDPE 2007, pp. 175–180, IEEE Computer Society, 2007.
18. Wei Li, Dawu Gu. Differential fault analysis on the SMS4 cipher by inducing faults to the key schedule. Chinese Journal on Communications, 2008, 29(10): 135–142.
19. Wei Li, Dawu Gu, Juanru Li. Differential fault analysis on the ARIA algorithm. Information Sciences, 2008, 178(19): 3727–3737.
20. Jiqiang Lu. Attacking reduced-round versions of the SMS4 block cipher in the Chinese WAPI standard. ICICS 2007, LNCS 4861, pp. 306–318, Springer-Verlag, 2007.
21. Debdeep Mukhopadhyay. An improved fault based attack of the Advanced Encryption Standard. Africacrypt 2009, LNCS 5580, pp. 421–434, 2009.
22. Kaisa Nyberg. Differentially uniform mappings for cryptography. EUROCRYPT 1993, LNCS 765, pp. 55–64, Springer-Verlag, 1994.

23. Gilles Piret and Jean-Jacques Quisquater. A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. CHES 2003, LNCS 2779, pp. 77–88, Springer-Verlag, 2003.
24. Matthieu Rivain. Differential fault analysis on DES middle rounds. In CHES 2009, LNCS 5747, pp. 457–469, 2009.
25. S.P. Skorobogatov, R.J. Anderson. Optical fault induction attacks. CHES 2002, LNCS 2523, pp. 2–12, Springer-Verlag, 2003.
26. Specification of SMS4, Block cipher for WLAN products–SMS4 (in Chinese), http://www.oscca.gov.cn/UpFile/200621016423197990.pdf
27. Deniz Toz and Orr Dunkelman. Analysis of two attacks on reduced-round versions of the SMS4. ICICS 2008, LNCS 5308, pp. 141–156, Springer-Verlag, 2008.
28. Junko Takahashi and Toshinori Fukunaga. Improved differential fault analysis on CLEFIA. FDTC 2008, pp. 25–34, IEEE Computer Society, 2008.
29. Lei Zhang, Wenling Wu. Differential fault analysis on SMS4, Chinese Journal of Computers, 2006, 29(9): 1596–1602.
30. Lei Zhang, Wentao Zhang and Wenling Wu. Cryptanalysis of reduced-round SMS4 block cipher. ACISP 2008, LNCS 5107, pp. 216–229, Springer-Verlag, 2008.
31. Wentao Zhang, Wenling Wu, Dengguo Feng, and Bozhan Su. Some new observations on the SMS4 block cipher in the Chinese WAPI standard. ISPEC 2009, LNCS 5451, pp. 324–335, 2009.
32. Yongbin Zhou, Wengling Wu, Nannan Xu, Dengguo Feng. Differential fault attack on Camellia. Chinese Journal of Electronics, 2009, 18(1): 13–19.

A Proofs of the Propositions in Sect. 3

A.1 Proof of Proposition 1

According to [16], the S-box of SMS4 is affine equivalent to the patched multiplicative inverse over $GF(2^8)$, say $I(\cdot)$, thus the differential property of $S(\cdot)$ is the same as that of $I(\cdot)$.

By Proposition 6 of [22], for any given $0 \neq \alpha \in GF(2^8)$,

$$\mathrm{N}_I(\alpha,\beta) = \begin{cases} 0 \text{ or } 2 & \text{if } \beta \neq \alpha^{-1} \\ 4 & \text{if } \beta = \alpha^{-1} \end{cases}$$

Moreover, if $\beta \neq \alpha^{-1}$, then $\mathrm{N}_S(\alpha,\beta) = 2$ iff $\mathrm{Tr}\left((\alpha\beta)^{-1}\right) = 0$. Here $\mathrm{Tr}(\cdot)$ denotes the trace map of $\mathbb{F}_{2^8}$ over $\mathbb{F}_2$. Since the trace map is a balanced function, the number of β such that $\mathrm{Tr}\left((\alpha\beta)^{-1}\right) = 0$ is 128. Excluding $\beta = 0$ and $\beta = \alpha^{-1}$, we conclude that there are 126 possible β satisfying $\mathrm{N}_S(\alpha,\beta) = 2$. □

A.2 Proof of Proposition 2

(1) If $x = x^*$, then the equation has 256 solutions; if $x \neq x^*$, let $\alpha = x \oplus x^*$, by Proposition 1, the possible values of Δ such that $\mathrm{N}_S(\alpha,\beta) > 0$ is 127 and in this case the equation will have solutions. Thus the equation has solutions with probability

$$\frac{256 + 256 \times 255 \times 127}{2^8 \times 2^8 \times 2^8} \approx 0.4942.$$

(2) From the result of (1) and the differential distribution table of S, when the equation $S(x \oplus k) \oplus S(x^* \oplus k) = \beta$ has solutions, the expectation of the number of solutions can be calculated as follow

$$\frac{256 \times 256 + 256 \times 255 \times (126 \times 2 + 1 \times 4)}{256 + 256 \times 255 \times 127} \approx 2.0236.$$

□

A.3 Proof of Proposition 3

Let $\mathbb{F}_2[x]$ denote the polynomial ring over $\mathbb{F}_2$. Consider $\mathbb{F}_2[x]/(x^{32} \oplus 1)$ as the residue class of the ring $\mathbb{F}_2[x]$ modulo the ideal $(x^{32} \oplus 1)$. For each

$$B = (B_{31}, B_{30}, \ldots, B_1, B_0) \in \mathbb{F}_2^{32},$$

there exists a corresponding element

$$B(x) = B_{31}x^{31} \oplus B_{30}x^{30} \oplus \ldots \oplus B_1 x \oplus B_0 \in \mathbb{F}_2[x]/(x^{32} \oplus 1),$$

i.e. a polynomial with degree no more than 32 and vice versa.

Since $B \lll i$ is equivalent to $B(x) \cdot x^i \pmod{x^{32} \oplus 1}$, by the definition of $L(\cdot)$, let $l(x) = 1 \oplus x^2 \oplus x^{10} \oplus x^{18} \oplus x^{24}$, then we can build the following relationship between $L(\cdot)$ and $\mathcal{L}(\cdot)$:

$$\begin{tikzcd} L : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32} \arrow[d, <->] \arrow[r, phantom, "\Leftrightarrow"] & \mathcal{L} : \mathbb{F}_2[x]/(x^{32} \oplus 1) \arrow[d, <->] \\ B \mapsto L(B) \arrow[r, phantom, "\Leftrightarrow"] & B(x) \mapsto \mathcal{L}(B(x)) = B(x) \cdot l(x) \pmod{x^{32} \oplus 1} \end{tikzcd}$$

Notice that $x = 1$ is not the solution of the equation $l(x) = 0$, which implies that $x \oplus 1 \nmid l(x)$. Since $x^{32} \oplus 1 = (x \oplus 1)^{32}$, we have $\gcd(l(x), x^{32} \oplus 1) = 1$. Thus there exists $l^{-1}(x) \in \mathbb{F}_2[x]/(x^{32} \oplus 1)$, such that

$$l(x) \cdot l^{-1}(x) \equiv 1 \pmod{x^{32} \oplus 1}.$$

By the extended Euclid algorithm,

$$l^{-1}(x) = 1 \oplus x^2 \oplus x^4 \oplus x^8 \oplus x^{12} \oplus x^{14} \oplus x^{16} \oplus x^{18} \oplus x^{22} \oplus x^{24} \oplus x^{30}.$$

From the relationship between the $L(\cdot)$ and $\mathcal{L}(\cdot)$, the concrete expression of $L^{-1}(\cdot)$ can be easily deduced as follow

$$\begin{aligned} L^{-1}(C) = C &\oplus (C \lll 2) \oplus (C \lll 4) \oplus (C \lll 8) \oplus (C \lll 12) \oplus (C \lll 14) \\ &\oplus (C \lll 16) \oplus (C \lll 18) \oplus (C \lll 22) \oplus (C \lll 24) \oplus (C \lll 30), \end{aligned}$$

which ends the proof. □

B Differential Attack on an S-box

In this appendix, we describe how to apply differential attack on an 8×8 S-box $S(\cdot)$ from a triplet (x, x^*, β), where $\beta = S(x \oplus k) \oplus S(x^* \oplus k)$ and k is the encryption key. One can also refer [5] for the detail of differential attack on the S-box of DES.

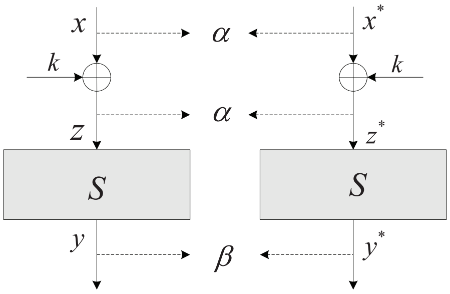

Fig. 5. Differential attack on the S-box

Let $\alpha = x \oplus x^*$, $\beta = y \oplus y^*$, since $z \oplus z^* = (x \oplus k) \oplus (x^* \oplus k) = x \oplus x^* = \alpha$, then $S(x \oplus k) \oplus S(x^* \oplus k) = \beta \Leftrightarrow S(z) \oplus S(z \oplus \alpha) = \beta$, with $z = x \oplus k$, thus if the adversary firstly store the set $\mathrm{IN}_S(\alpha, \beta)$, with all possible (α, β), in advance, he could do the following attack procedure:

1. Set $\Omega = \mathbb{F}_2^8$.
2. According to the differential distribution table of the S-box,
$$x \oplus k = z \in \mathrm{IN}_S(\alpha, \beta) \ .$$
Thus, the right key k must be in
$$x \oplus \mathrm{IN}_S(\alpha, \beta) = \{x \oplus z : z \in \mathrm{IN}_S(\alpha, \beta)\} \ .$$
Set $\Omega = \Omega \cap (x \oplus \mathrm{IN}_S(\alpha, \beta))$, go to step 3.
3. If $\#\Omega = 1$, then the right key k is uniquely deduced. Otherwise, obtain another triplet (x, x^*, β), and go to step 2.

Notice that, the number of triplets (x, x^*, β) that are needed to uniquely determine the encryption key k is significantly related to the differential distribution table of the S-box. If only one triplet (x, x^*, β) can be obtained, the adversary only gets the key candidate set $x \oplus \mathrm{IN}_S(x \oplus x^*, \beta)$. If another triplet could be obtained, however, its input difference is the same as the fore triplet, then these two triplets could not retrieve the unique key all the same. In other words, in this case, at least two key candidates will be left.