Multiple Bytes Differential Fault Analysis on CLEFIA

Xin-jie ZHAO, Tao WANG, Jing-zhe GAO

Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China, zhaoxinjieem@163.com

Abstract. This paper examines the strength of CLEFIA against multiple bytes differential fault attack. Firstly, it presents the principle of CLEFIA algorithm and differential fault analysis; then, according to injecting faults into the r^{th},r-1^{th},r-2^{th} CLEFIA round three conditions, proposes three fault models and corresponding analysis methods; finally, all of the fault model and analysis methods above have been verified through software simulation. Experiment results demonstrate that: CLEFIA is vulnerable to differential fault attack due to its Feistel structure and S-box feature, 5-6,6-8,2 faults are needed to recover CLEFIA-128 based on the three fault models in this paper respectively, multiple byte faults model can greatly improve the attack practicality and even the attack efficiency, and the fault analysis methods in this paper can provide some fault analysis ideas on other block ciphers using S-box

1 Introduction

The idea of fault attack was first suggested in 1997 by Boneh, DeMillo and Lipton[1], which makes use of the faults during the execution of a cryptographic algorithm. Under the idea, the attack was successfully exploited to break an RSA-CRT with both a correct and a faulty signature of the same message. Shortly after, Biham and Shamir proposed an attack on secret key cryptosystems called Differential Fault Analysis (DFA)[2], which combined the ideas of fault attack and differential attack. Since that, many research papers have been published using this cryptanalysis technique to successfully attack various cryptosystems, including ECC[1],3DES[4],AES[5-9],Camellia[10-12],ARIA[13],CLEFIA[14-15],SMS4[16-20],RC4[21-22],Trivium[23-24] and so on. It's clear to see that no matter symmetric ciphers or public ciphers are all facing serious threats of fault attacks.

CLEFIA is proposed by Sony Corporation in 2007 FSE conference, it is a 128-bit Feistel structure block cipher with the key length of 128, 192 and 256 bits, which are compatible with AES. Sony claimed that the CLEFIA is designed to concentrate state-of-the-art cryptanalysis techniques, and achieve sufficient immunity against known cryptanalytic attacks. Sony will seek to establish an environment in which CLEFIA can be used across various applications and products such as AV devices. Since the presentation of CLEFIA in FSE2007, cryptographists of the world have made many security analyses on it.

In the previous DFA attacks on CLEFIA, Chen et al.[14] proposed the first DFA attack on CLEFIA, they injected one byte fault in the r-1^{th} round left register to recover 4 bytes of the r^{th} round key, after attacking the last 3 rounds and 9 rounds, the CLEFIA-128,192/256 key was recovered using 18 and 54 ciphertexts on average respectively. In 2008, Junko et al.[15] proposed an improved DFA methods on CLEFIA, they injected 4 byte faults in the r-2^{th} round left registers to deduce the last 3 rounds key, finally they obtained CLEFIA-128,192/256 key by 2 and 10.78 ciphertexts on average respectively.

This work takes different approach to DFA attacks against CLEFIA, we examine the strength of CLEFIA against multiple bytes differential fault attack, propose 3 DFA methods on injecting m multiple byte faults in the $r^{\text{th}}(1\leq m\leq 8)$, r-$1^{\text{th}}(1\leq m\leq 4)$, r-$2^{\text{th}}(1\leq m\leq 4)$ CLEFIA round respectively, and verify all the analysis methods above through simulation experiments. Generally speaking, comparing with all the previous attacks, our attack conditions are much more loosely, and the attack are much more practical. Specifically speaking, attack in [14] is the special case of our r-1^{th} round multiple bytes DFA attack when m=1, the research in this paper also demonstrate that [14] didn't make full analysis on the r-1^{th} round fault(missed the analyzing of the faults in the r-1^{th} round); attack in [15] is the special case of our r-2^{th} round multiple bytes DFA attack when m=4, their faulty model has the defects of difficult to identify the ideal proper faults, but the third fault model in the r-2^{th} round can identify the ideal faults correctly at most cases, and the analysis efficiency is also quite high, attack in [15] takes that one time 4 byte faults can reduce RK_{34} key search space from 2^{32} to $27.1=2^{4.76}$, but the analysis and the concrete experiment demonstrates that the RK_{34} key search space should be reduced to $37.8=2^{5.24}$.

This work is organized as follows. Section 2 presents the preliminaries of DFA on CLEFIA. Section 3,4,5 presents several DFA methods on injecting one byte or multiple byte faults in the r^{th}, r-1^{th}, r-2^{th} CLEFIA round respectively. Section 6 displays the complexity analysis and experimental results of the attacks. Section 7 is the conclusion.

2 Preliminaries

2.1 Notations

P, X_r, C: 128-bit plaintext, the r^{th} round intermediate state, the ciphertext

P^i, X_r^i, C^i: the i^{th} 32-bit word of the plaintext, the r^{th} round intermediate state, the ciphertext

$P^{i,j}$, $X_r^{i,j}$, $C^{i,j}$: the j^{th} byte in the i^{th} 32-bit word of the plaintext, the r^{th} round intermediate state, the ciphertext

$a|b$: the concatenation of a and b

2.2 CLEFIA

CLEFIA[25] employs a generalized Feistel structure with four data lines, and the width of each data line is 32-bit. Additionally, there are key whitening parts at the beginning and the end of the cipher. Fig. 1(a) shows the encryption process of *r*-round CLEFIA.

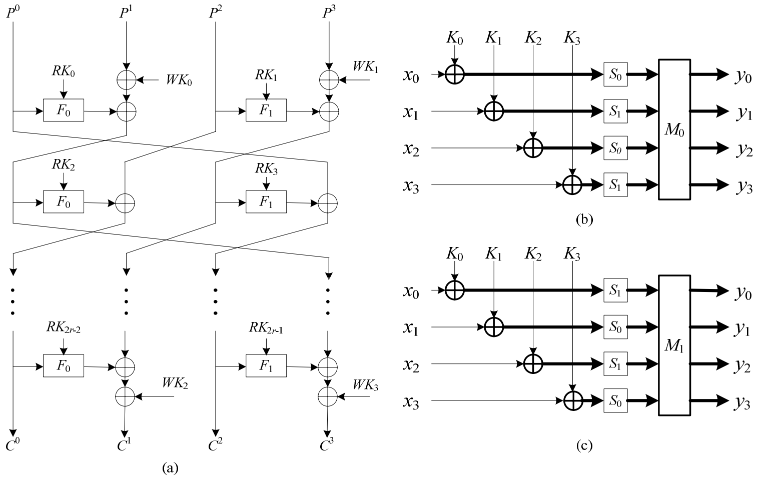

Fig. 1. Encryption Process of *r*-round CLEFIA

The number of rounds *r* can be 18, 22 and 26 for CLEFIA-128, CLEFIA-192 and CLEFIA-256 respectively. 2r 32-bit round subkeys ($RK_0,RK_1,\ldots,RK_{2r-1}$), and 4 32-bit whitening keys (WK_0,WK_1,WK_2,WK_3) are used in the encryption. Let (X_i^0,X_i^1,X_i^2,X_i^3) denote the four 32-bit input of the i+1 round. The encryption process is shown as following 3 steps:

1. Pre-whitening:

The 128-bit plaintext is divided into 4 32-bit words (P^0,P^1,P^2,P^3), the 2rd and 4th words are XORed with the first two 32-bit whitening keys.

$$(X_0^0, X_0^1, X_0^2, X_0^3) = (P^0, P^1 \oplus WK_0, P^2, P^3 \oplus WK_1)$$

2. The first *r*-1th round encryption

$$X_i^0=X_{i-1}^1 \oplus F_0(X_{i-1}^0,RK_{2i-2}), \qquad X_i^1=X_{i-1}^2,$$
$$X_i^2=X_{i-1}^3 \oplus F_1\,(X_{i-1}^2,RK_{2i-1}), \qquad X_i^3=X_{i-1}^0.$$

In order to keep the same structure of the decryption process as the encryption process, the *r*th round has a little different with previous rounds.

$$X_i^0=X_{i-1}^1, \qquad X_i^1=X_{i-1}^1 \oplus F_0(X_{i-1}^0,RK_{2i-2}),$$
$$X_i^2=X_{i-1}^2, \qquad X_i^3=X_{i-1}^3 \oplus F_1(X_{i-1}^2,RK_{2i-1}).$$

3. Post-whitening:

The X_r^1 and X_r^3 are XORed with the last two 32-bit whitening keys.

$$(C^0, C^1, C^2, C^3) = (X_r^0, X_r^1 \oplus WK_2, X_r^2, X_r^3 \oplus WK_3)$$

The description of F_0 and F_1 function is shown in Fig. 1(b) and (c). Two nonlinear 8-bit S-boxes S_0 and $S_1$are used in the substitution function, but the sequence is different. M_0 and M_1 are 4 × 4 Hadamard-type matrixes used in the permutation function, they are defined as follows:

$$M_0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix} \quad M_1 = \begin{pmatrix} 01 & 08 & 02 & 0a \\ 08 & 01 & 0a & 02 \\ 02 & 0a & 01 & 08 \\ 0a & 02 & 08 & 01 \end{pmatrix}$$

The key expansion algorithm of CLEFIA is divided into two part: the first is to generate L by K and the second is to generate $WK_i(0 \leqslant i < 4)$ and $RK_j(0 \leqslant j < 2r)$ by K and L. For the 128-bit key scheduling, the 128-bit intermediate key L is generated by applying $GFN_{4,12}$ which takes 24 32-bit constant values $CON128_i$, $0 \le i < 24$ as round keys and $K=K_0|K_1|K_2|K_3$ as an input. Then K and L are used to generate $WK_i(0 \leqslant i < 4)$ and $RK_j(0 \leqslant j < 36)$ in the following steps.

```
1. $L \leftarrow GFN_{4,12}(CON_0^{128}, \cdots, CON_{23}^{128}, K_0, \cdots, K_3)$;
2. $WK_0 \mid WK_1 \mid WK_2 \mid WK_3 \leftarrow K$;
3. For i=0 to i=8
        {$T=L \oplus (CON_{24+4i}^{128}|CON_{24+4i+1}^{128}|CON_{24+4i+2}^{128}|CON_{24+4i+3}^{128})$;
        $L \leftarrow \sum(L)$;
        If i is an odd, $T \leftarrow T \oplus K$;
        $RK_{4i}|RK_{4i+1}|RK_{4i+2}|RK_{4i+3} \leftarrow T$;}
```

The *DoubleSwap* function $\sum: X\{0, 1\}^{128} \rightarrow Y\{0, 1\}^{128}$ is defined as follows:

$$Y = X[7-63]X[121-127]X[0-6]X[64-120]$$

2.3 CLEFIA DFA principle

In order to enhance the resistance of linear and differential analysis techniques, modern block ciphers usually apply the S-box to improve the cipher non-linearity, avalanche and the implementation efficiency, but it is just because of non-full coverage feature in the differential S-box which plunge the block ciphers into serious threat of fault attacks.

Suppose during the S-box lookups, a random fault is injected into the unknown S-box input value a, usually, the attacker can obtain f', which is the XORed differential value of the correct and faulty S-box output. a, f, f' can satisfy:

$$S[a] \oplus S[a \oplus f] = f' . \tag{1}$$

According to equation (1), as to Feistel block ciphers, it usually satisfies such deduction.

Deduction1: If the input and out differential of the S-box are known, then very limited candidates for the input and output of the S-box lookup can be obtained.

Generally speaking, the length of a (typical is 8-bit) is limited. If we iterate all the candidates of a into equation (1), due to the avalanche of the S-box, as to given f and f', very limited candidates of a can be obtained..

As is shown in Fig.2, due to the generalized double Feistel structure, if a single byte fault f is injected into the r^{th} round left register, after the XORed with the round key RK_{2r-2}, the fault is unchanged, after the r^{th} round S-box lookup, f is transferred into f', after the r^{th} round M permutation function, f' is transferred into $M(f')$, and finally the result is XORed with the two post-whitening keys to generate the ciphertext. Obviously, due to the Feistel structure, the r^{th} round S-box input differential *f'* and *M* function output differential *M*(*f'*) is known by the ciphertext differential. As the r^{th} round S-box output differential *f'* can be computed by the M^{-1} function. According to Deduction 1, the r^{th} round S-box input value ($X_{r\text{-}1}{}^{0,i} \oplus RK_{2r\text{-}2}{}^{i}$) can be recovered, as $X_{r\text{-}1}{}^{0,i} = C^{0,i}$, $RK_{2r\text{-}2}{}^{i}$ can be recovered.

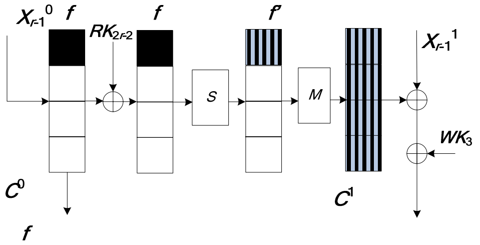

Fig. 2. The basic CLEFIA DFA model

Inducing more faults into $X_{r\text{-}1}{}^{0}$, the round key $RK_{2r\text{-}2}$ can be obtained, inducing more faults into $X_{r\text{-}1}{}^{2}$, the round key $RK_{2r\text{-}1}$ can be obtained. Then the attackers could decrypt the right ciphertext to obtain the input of the last round, which is the output of the penultimate round. They repeat the above procedure to recover more subkeys until the secret key is obtained by the key schedule. On the basis of the above procedure, the attackers must recover the pure subkeys before deducing the secret key in the previous DFA research.

The following Sections will present the specific DFA analysis on the r^{th}, r-1th, r-2th round of CLEFIA respectively.

3 DFA on Fault in the r^{th} CLEFIA round

None of the previous DFA attacks on CLEFIA has considered injecting faults into the r^{th} round left register to analyze the round key. The fault model of this Section is to induce multiple byte faults in the r^{th} round CLEFIA left register. We hold that, due to the broad fault width (at most 64-bit) and high round key recovery efficiency, it can be seen as the most powerful DFA attack to CLEFIA.

Fault model: Inject *m*, *n* ($1 \leq m,n \leq 4$) random faults into $X_{r\text{-}1}{}^{0}$, $X_{r\text{-}1}{}^{2}$ respectively, the faulty bytes number, location and differential value is unknown.

1. Deduce RK_{34},RK_{35}

Let's take CLEFIA-128 as an example. If we induce *m*(*m*=3), *n*(*n*=4) byte faults into $X_{17}{}^{0}$, $X_{17}{}^{2}$, the fault propagation process is depicted in Fig.3.

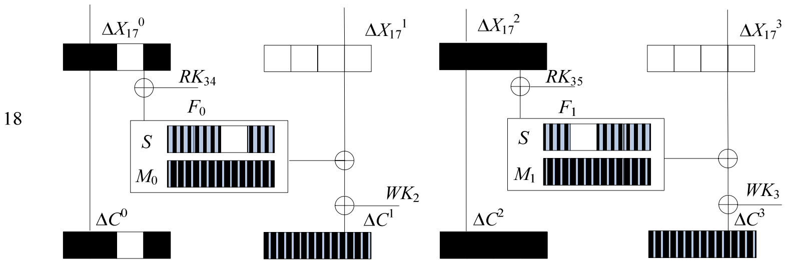

Fig. 3. Fault propagation process of injecting faults into the r^{th} CLEFIA round

The key recovery process is as follows:

Step1：Deduce RK_{34}

Step1.1：Deduce the 18th S-box input differential $\Delta X_{17}{}^0$

From Fig.3, it's easy to known that $\Delta X_{17}{}^0$ is equal to ΔC^0.

Step1.2：Deduce the 18th S-box output differential $\Delta S_{17}{}^0$.

From Fig.3, it's easy to known that$\Delta S_{17}{}^0$ is equal to $M_0{}^{-1}(\Delta C^1)$.

Step1.3：Deduce RK_{34}

From Fig.3, combined the Deduction 1 of Section 2.3,we can recover limited candidates of $RK_{34}{}^0$， $RK_{34}{}^1$， $RK_{34}{}^3$. Repeat to inject random multiple fault into $X_{17}{}^0$ and repeat Step1 to recover RK_{34}.

Step2：Deduce RK_{35}

Step2.1：Deduce the 18th S-box input differential $\Delta X_{17}{}^2$

$\Delta X_{17}{}^2$ is equal to ΔC^2.

Step2.2：Deduce the 18th S-box output differential $\Delta S_{17}{}^1$

$\Delta S_{17}{}^1$ is equal to $M_1{}^{-1}(\Delta C^3)$.

Step2.3：Deduce RK_{35}

Combing the Deduction 1 of Section 2.3, limited candidates of RK_{35} can be obtained, then injecting and analyzing more faults to obtain RK_{35}.

2. Deduce $RK_{32} \oplus WK_3$、$RK_{33} \oplus WK_2$

Inject multiple byte faults into the 17th round left register $X_{16}{}^0$,$X_{16}{}^2$, deduce the 17th round output differential by RK_{34},RK_{35} deduced before, then deduce the 17th round S-box input and output differential $\Delta X_{16}{}^0$,$\Delta S_{16}{}^0$, combined the Deduction 1 to obtain $RK_{32} \oplus WK_3$,$RK_{33} \oplus WK_2$.

3. Deduce RK_{30}、RK_{31}

Inject multiple byte faults into the 16th round left register $X_{15}{}^0$,$X_{15}{}^2$, deduce the 16th round S-box input and output differential $\Delta X_{15}{}^0$,$\Delta S_{15}{}^0$, combined the Deduction 1 to obtain RK_{30}, RK_{31}.

4. Deduce and verify K

Combing the key reverse techniques, the input L value of the 9th iteration in CLEFIA schedule can be obtained, and by applying the inverse of the Doubleswap and the $GFN^1{}_{4,12}$ function, K can be obtained. The last 4 round keys generating procedure, the DFA results above, and the specific key reverse procedure are depicted in Fig.4 (a),(b),(c) respectively.

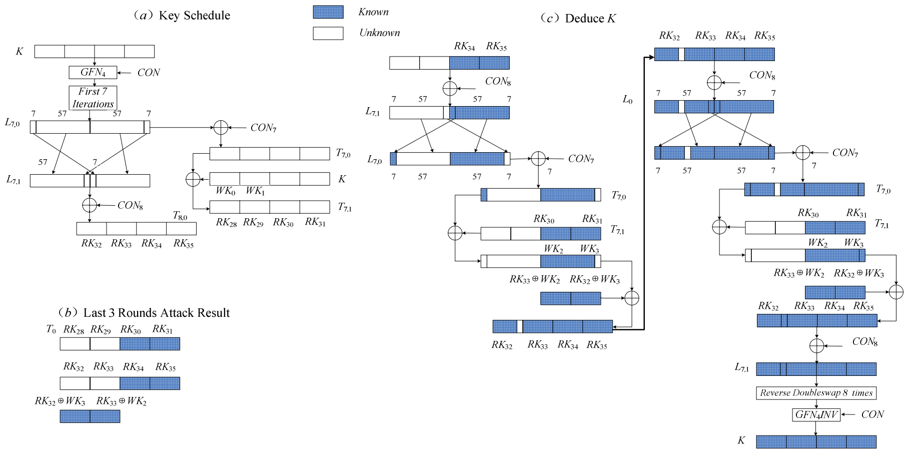

Fig. 4. CLEFIA-128 key deduce procedure

4 DFA on Fault in the *r*-1th CLEFIA round

4.1 Related Work

In the previous DFA attacks on CLEFIA, only Chen et al.[14] proposed the DFA attack on CLEFIA by injecting one byte fault in the *r*-1th round left register to recover 4 bytes of the *r*th round key.

Their fault model is: Inject single byte random faults into the *r*-1th round left register X_{r-2}^{0}, X_{r-2}^{2} respectively, the faulty byte location and value is unknown.

Fig. 5(a) depicts the fault propagation of injecting one byte fault into X_{16}^{0} in [14]. After analyzing the ciphertext differential, the18th round input and output differential can be obtained, using the Deduction 1 in Section 2.3 to obtain RK_{34}, and then analyze faults injected in X_{16}^{2} to deduce RK_{35}. Repeat the same methods by inject faults into the 16th and 15th round to deduce $RK_{32} \oplus WK_3$,$RK_{33} \oplus WK_2$,RK_{34},RK_{35}, then combined the key schedule to recover the CLEFIA-128 initial key.

4.2 Improved Multiple Bytes DFA atgtack

The fault model of [14] was based on single byte fault, it made full analysis on the faults propagated in the *r*th round, but missed the analysis of the single byte fault to deduce one byte of the *r*-1th round key in the *r*-1th round.

In this Section, we make an improved attack based on [14] as follows:

1. Broaden the fault width by injecting multiple byte faults into X_{r-2}^{i} in the *r*-1th round
2. Make full use of the *r*th and *r*-1th round fault to analyze the last two round keys.

The specific fault model and analysis method is shown as follows:

Fault model: Inject m, n ($1\leq m,n\leq 4$) random faults into the r-1th round left register X_{r-2}^0, X_{r-2}^2 respectively, the faulty bytes number, location and differential value is unknown.

Let's also take CLEFIA-128 as an example. If we induce m (m=1,4) byte faults into X_{16}^0, the fault propagation process is depicted in Fig.5. The key analysis procedure is as follows:

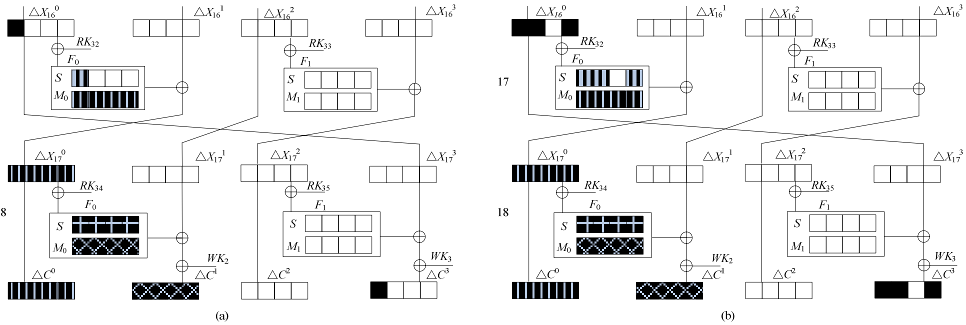

Fig. 5. Fault propagation process of injecting faults into the r-1th CLEFIA round

1. Deduce RK_{34}

The ciphertext differential ΔC^0 is the 18th round left S-box input differential, and $M_0^{-1}(\Delta C^1)$ is the 18th round left S-box output differential, combined the Deduction 1 to deduce RK_{34}.

2. Deduce $RK_{32}\oplus WK_3$

The ciphertext differential ΔC^3 is the 17th round left S-box input differential, and $M_0^{-1}(\Delta C^0)$ is the 17th round left S-box output differential, combined the Deduction 1 to deduce $RK_{32}\oplus WK_3$.

3. Deduce RK_{35}、 $RK_{33}\oplus WK_2$

Inject multiple byte faults into X_{16}^2, ΔC^2, $M_1^{-1}(\Delta C^3)$ are the 18th round right S-box input and output differential, then RK_{35} can be deduced. ΔC^1, $M_1^{-1}(\Delta C^2)$ are the 17th round right S-box input and output differential, and then $RK_{33}\oplus WK_2$ can be deduced.

4. Deduce RK_{30}、 RK_{31}

Inject multiple byte faults into X_{14}^2, using $RK_{30}, RK_{31}, RK_{32}\oplus WK_3, RK_{33}\oplus WK_2$ to computer the 16th round output differential, then combined the Deduction 1 to deduce RK_{30}, RK_{31}.

5. Deduce and verify K

According to CLEFIA key recovery technique of Section 3 to obtain and verify K.

5 DFA on Fault in the *r*-2th CLEFIA round

5.1 Related Work

In the previous DFA attacks on CLEFIA, only Junko et al.[15] proposed the DFA attack on CLEFIA by injecting 4 byte faults in the r-2th round left register to recover last 3 rounds key.

Their fault model is: Inject 4 byte random faults into the r-2th round left register X_{r-3}^0, X_{r-3}^2 respectively, the faulty bytes value is unknown.

Fig. 6(a) depicts the fault propagation of injecting 4 byte faults into X_{15}^0 in [15]. The specific analysis procedure is as follows:

1. Inject 4 byte faults into X_{15}^0, analyze the ciphertext differential to obtain the 18th round left S-box input and output differential, then deduce RK_{34}.
2. Inject 4 byte faults into X_{15}^2, analyze the ciphertext differential to obtain the 18th round right S-box input and output differential, then deduce RK_{35}.
3. Compute the 17th round left S-box input and output differential by RK_{35} analyzed in step 2 to deduce $RK_{32} \oplus WK_3$.
4. Compute the 17th round right S-box input and output differential by RK_{34} analyzed in step 1 to deduce $RK_{33} \oplus WK_2$.
5. Compute the 16th round left S-box input and output differential by the round key analyzed in step 1,2,4 to deduce RK_{30}.
6. Compute the 16th round right S-box input and output differential by the round key analyzed in step 1,2,3 to deduce RK_{31}.
7. Deduce and verify K

5.2 Improved Multiple Bytes DFA attack

The fault model of [15] was based on injecting 4 byte faults into the r-2 round left registers X_{r-3}^0 and X_{r-3}^2, it's a strictly 4 byte faults model, and more importantly, it has the defects of difficult to identify the ideal faulty ciphertexts.

In this Section, we make an improved attack based on [15] as follows:

1. Loose the fault width condition by injecting 1-4 byte faults into X_{r-3}^0 and X_{r-3}^2.
2. Improve the identification of ideal faulty ciphertexts.
3. Attack in [15] is a special case of our attack when the fault byte number is 4.

The specific fault model and analysis method is shown as follows:

Fault model: Inject m, n ($1 \le m,n \le 4$) random faults into the r-2th round left register X_{r-3}^0, X_{r-3}^2 respectively, the faulty bytes number, location and differential value is unknown.

If we induce m (m=4,3) byte faults into X_{15}^0, the fault propagation process is depicted in Fig.6. The key analysis procedure is as follows:

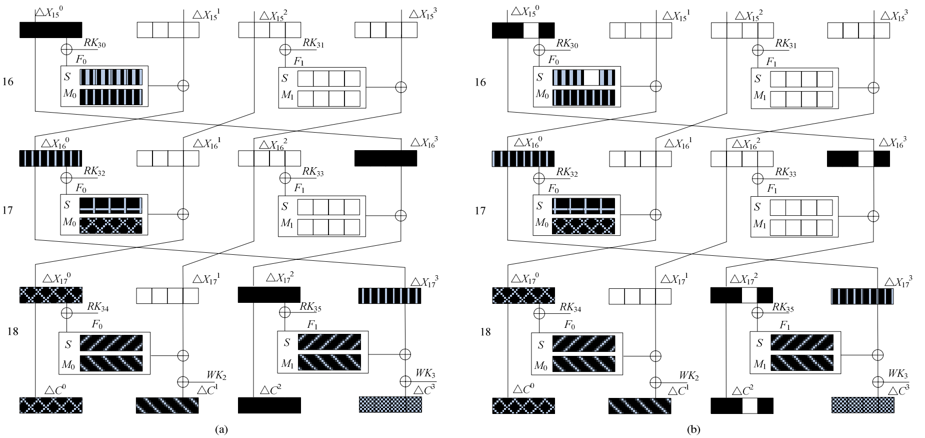

Fig. 6. Fault propagation process of injecting faults into the *r*-2th CLEFIA round

1. Deduce RK_{34}

Inject multiple byte faults into X_{15}^0, RK_{34} can be deduced after analyzing the 18th round left S-box input and out differential ΔC^0 and $M_0^{-1}(\Delta C^1)$.

2. Deduce RK_{35}

Inject multiple byte faults into X_{15}^2, RK_{35} can be deduced after analyzing the 18th round right S-box input and out differential ΔC^2 and $M_0^{-1}(\Delta C^3)$.

3. Deduce $RK_{32} \oplus WK_3$

The 17th round left S-box output differential is $M_0^{-1}(\Delta C^0)$, combined ciphertext and RK_{35} candidates, the 17th round left S-box input differential ΔX_{16}^0 can be deduced by

$$\Delta X_{16}^0 = F_1(C^2, RK_{35}) \oplus F_1(C^{2*}, RK_{35}) \oplus \Delta C^3 . \quad \textbf{(2)}$$

4. Deduce $RK_{33} \oplus WK_2$

The 17th round right S-box output differential is $M_1^{-1}(\Delta C^2)$, combined ciphertext and RK_{34} candidates, the 17th round right S-box input differential ΔX_{16}^2 can be deduced by

$$\Delta X_{16}^2 = F_0(C^0, RK_{34}) \oplus F_0(C^{0*}, RK_{34}) \oplus \Delta C^1 . \quad \textbf{(3)}$$

5. Deduce RK_{30}

The 16th round left S-box input differential is ΔC^2, combined ciphertext and RK_{35} candidates, the 16th round left S-box output differential ΔS_{16}^0 can be deduced by

$$\Delta S_{16}^0 = M_0^{-1}(\Delta X_{16}^0) = M_0^{-1}(F_1(C^2, RK_{35}) \oplus F_1(C^{2*}, RK_{35}) \oplus \Delta C^3) \quad \textbf{(4)}$$

Combing the Deduction 1, the 16th round left S-box input value $X_{15}^0 \oplus RK_{30}$ can be deduced. In order to deduce RK_{30}, we should first compute X_{15}^0, X_{15}^0 can be deduced by

$$X_{15}^0 = X_{16}^3 = C^2 \oplus F_1(F_0(C^0, RK_{34}) \oplus C^1 \oplus WK_2, RK_{33}) \quad \textbf{(5)}$$

As the 17th round right S-box input value $F_0(C^0, RK_{34}) \oplus C^1 \oplus WK_2 \oplus RK_{33}$ is known, according to equation(5), X_{15}^0 can be deduced, and then RK_{30} can be obtained.

6. Deduce RK_{31}

The 16th round right S-box input differential is ΔC^0, combined ciphertext and RK_{34} candidates, the 16th round right S-box output differential ΔS_{16}^2 can be deduced by

$$\Delta S_{16}^2=M_1^{-1}(\Delta X_{16}^2)=M_1^{-1}(F_0(C^0,RK_{34}) \oplus F_0(C^{0*},RK_{34}) \oplus \Delta C^1) \qquad \textbf{(6)}$$

Combing the Deduction 1, the 16th round right S-box input value $X_{15}^2 \oplus RK_{31}$ can be deduced. In order to deduce RK_{31}, we should first compute X_{15}^2, X_{15}^2 can be deduced by

$$X_{15}^2=X_{16}^1=C^0 \oplus F_0(F_1(C^2,RK_{35}) \oplus C^3 \oplus WK_3,RK_{32}) \qquad \textbf{(7)}$$

As the 17th round left S-box input value $F_1(C^2,RK_{35}) \oplus C^3 \oplus WK_3 \oplus RK_{32}$ is known, according to equation(7), X_{15}^2 can be deduced, and then RK_{31} can be obtained.

7. Deduce and verify K

According to CLEFIA key recovery technique of Section 3 to obtain and verify K.

6 Complexity Analysis and Experimental Results

6.1 Complexity Analysis

Suppose single byte fault is injected into the CLEFIA-128 last round, as is shown in Fig.2. Then the fault analysis should satisfy

$$S[C^0 \oplus k] \oplus S[C^0 \oplus k \oplus f]=f' \qquad \textbf{(8)}$$

If we input every possible candidates of $C^0 \oplus k$, $f(f\neq 0x00)$, f' into equation (8), the key DFA analysis efficiency of the two CLEFIA S-box S_0, S_1 are shown in Table 1 and Table 2.

Table 1. CLEFIA key DFA analysis efficiency by S_0

k count	frequency	probability	variance
2	9984512	0.5975	1.1950
4	5157888	0.3086	1.2344
6	1302528	0.0779	0.4674
8	243712	0.0146	0.1168
10	23040	0.0014	0.0140
total	16711680=(2^{24}-256)	1	3.0276=$2^{1.60}$

Table 2. CLEFIA key DFA analysis efficiency by S_1

k count	frequency	probability	variance
2	16450560	0.9844	1.9688
4	261120	0.0156	0.0624
total	16711680=(2^{24}-256)	1	2.0312=$2^{1.02}$

According to the analysis of Table 1 and Table 2, after the fault injecting and analysis of the same byte for two times, the unique key byte recovery probability is over 98.8%, but attack in [14] insists that 3 times same location faulty bytes are needed to obtain the correct key byte. And if full 4 byte of X_{17}^0 or X_{17}^2 has been injected faults, the key search space of RK_{34} or RK_{35} should be reduced from 2^{32} to $37.8=2^{5.24}$, but attack in 15 insists that it should be $27.1=2^{4.76}$, our analysis theory can be verified through simulation experiment in Section 6.2.

In the DFA analysis of Section3, if 8 bytes faults are injected into X_{17}^0 or X_{17}^2 by one time, it can reduce the key search space of RK_{34},RK_{35} from 2^{64} to $2^{10.48}$. Two times 8 byte faults can obtain RK_{34}、 RK_{35}, after injecting 8 byte faults into the 17th and 16th round, through analysis of Section 3, totally 5-6 times faults are enough to obtain CLEFIA-128 key.

In the DFA analysis of Section 4, if we inject 4 byte faults into X_{16}^0 and X_{16}^2 for 2 times, RK_{34},RK_{35},$RK_{32} \oplus WK_3$,$RK_{33} \oplus WK_2$ can be deduced. Then we inject 4 byte fault into X_{14}^0 and X_{14}^2 for one time, RK_{30},RK_{31} key search space can be reduced to $2^{10.48}$, two times are enough to obtain RK_{30},RK_{31} directly, so totally 6-8 times faults are enough to obtain CLEIA-128 key.

In the DFA analysis of Section 5, if we inject 4 byte faults into X_{15}^0 and X_{15}^2, totally 2 faults are enough to reduce the CLEFIA-128 key search space to 2^{19}.

6.2 Experimental Results

We have implemented simulations of all the attacks given in this paper. The simulations are written in Visual C++6.0 on Windows XP. Our simulations run on a personal computer (Athlon 64-bit 3000+ 1.81 GHz CPU and 1GB RAM) and successfully extract the CLEFIA-128/192/256 key.

As we know from Table 1 and Table 2, due to the DFA analysis of RK_{34}(2 times S-box lookup for S_0 and S_1), one time 4 byte fault analysis can reduce the RK_{34} searching space from 2^{32} to 3.0276*3.0276*2.0312*2.0312=37.82. We have done 20 sets of statistics to compute the RK_{34} candidates' number, every set has been done 2000 times of repeat DFA attacks, finally get the average RK_{34} candidates' number, as is depicted in Fig.7. It's clear to see that the experimental result (average number is 37.78) is almost the same as the theoretical value (average number is 37.82), which also verified the correctness of the analysis in Section 6.1 and proved that the analysis of $27.1=2^{4.76}$ in 15 has some deviations.

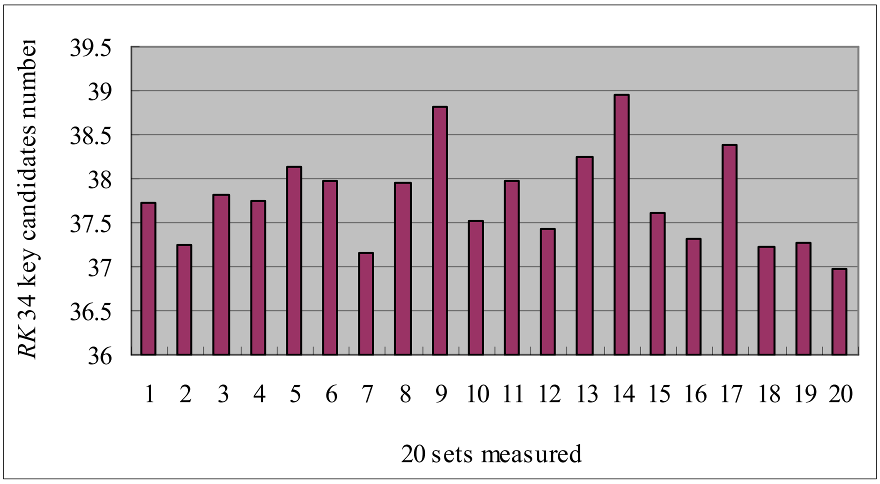

Fig. 7. Statistics of 20 sets for 2000 sample's average 4 byte faults in X_{17} and RK_{34} candidates' number

The DFA attack result of the recent years is shown in Table 3. It's clear to see that CLEFIA is quite vulnerable to multiple byte fault attacks and the key analysis efficiency is also quite high.

Table 3. Summary of CLEFIA-128 DFA attack results

Attack	fault type	fault location	sample
[14]	single byte	X_{17}^0; X_{17}^2; X_{16}^0; X_{16}^2; X_{15}^0; X_{15}^2	18
[15]	4 bytes	X_{15}^0; X_{15}^2	2
Section 3	1-8 bytes	X_{17}^0, X_{17}^2; X_{16}^0, X_{16}^2; X_{15}^0, X_{15}^2	at least 5-6
Section 4	1-4 faults	X_{17}^0; X_{17}^2; X_{15}^0; X_{15}^2	at least 6-8
Section 5	1-4 faults	X_{15}^0; X_{15}^2	at least 2

Compared with previous attacks, the attacks in this work have the following features:

1. The fault model of the Section 3 is the most loosely one, can as widely as 8 byte faults, and can analysis the two related round keys, while attack in r[14] and [15] can not, the least fault number is also quite low, 3 faulty ciphertexts can reduce the CLEFIA-key search space down to 2^{31}.
2. The fault model of Section 4 has extended the fault model of [14], instead injecting single byte fault in the r-1th round left register X_{r-2}^i, we inject multiple byte faults into the r-1th round left register X_{r-2}^i; also point out that [14] didn' t make full use the fault and missed the analysis of the single byte fault in the r-1 round, and we give the specific analysis method to make full use of the faults; Section 6.1 has made the analysis of the single byte fault analysis efficiency, we point out that 2 times of the same location single byte fault can obtain unique related key byte at 98.8% probabilities, while attack in [14] insists that 3 times single byte faults are enough to obtain the unique related key byte.
3. The fault model of Section 5 has extended the fault model of 15, first we loose the fault width, instead of injecting 4 byte faults into X_{r-3}^i , we inject 1-4 byte faults into X_{r-3}^i, the fault model of 15 is the special case when the fault width is 4 bytes,

attack in 15 has the defects of difficult to identify the ideal ciphertexts, while our fault model can improve the fault identification at certain level.

7 Conclusion

In this study we have analyzed the multiple byte faults DFA methods against CLEFIA block cipher, and displayed the complexity analysis and experimental results. Experimental results demonstrate that: due to its Feistel structure and differential S-box features, CLEFIA is quite weak for multiple byte faults attacks, so implementing CLEFIA should be done with great care and private countermeasures should be incorporated. The DFA analysis in this paper can provide some ideas on multiple byte faults DFA analysis on other Feistel structure block ciphers. Following works should be done in the future: the first is to research on the DFA attack on the CLEFIA key schedule; the second is to examine the strength of CLEFIA against hardware fault attacks; the third is to search the countermeasures of DFA attacks against CLEFIA.

Acknowledgements

The authors would like to thank the anonymous reviewers for many helpful comments and suggestions. The research presented in this paper was supported by National Natural Science Foundation of China (Grant No. 60772082) and the Natural Science Foundation of Hebei Province. China (Grant No. 08M010).

References

1. D.Boneh, R.A.DeMillo, R.J.Lipton. On the Importance of Checking Cryptographic Protocols for Faults[J]. Lecture Notes in Computer Science, vol.1233.1997:37-51. (1997)
2. E.biham, A.Shamir. Differential Fault Analysis of Secret Key Cryptosystems[J].Lecture Notes in Computer Science,vol.1294.1997:512-525. (1997)
3. Biehl, I., Meyer, B., Muller, V. Differential fault analysis on elliptic curve cryptosystems. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 131–146. Springer, Heidelberg. (2000)
4. Hemme, L.: A differential fault attack against early rounds of (Triple-) DES. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 254–267. Springer, Heidelberg. (2004)
5. Blomer, J., Seifert, J.P.: Fault based cryptanalysis of the Advanced Encryption Standard (AES). In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 162–181. Springer, Heidelberg.
6. Piret, G., Quisquater, J.J. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 77–88. Springer, Heidelberg. (2003)

7. Debdeep Mukhopadhyay. An Improved Fault Based Attack of the Advanced Encryption Standard. In: B. Preneel (eds.) AFRICACRYPT 2009, LNCS 5580, pp. 421–434. (2009)
8. Michael Tunstall, Debdeep Mukhopadhyay. Differential Fault Analysis of the Advanced Encryption Standard using a single Fault. Cryptology ePrint Archive, http://eprint.iacr.org/2009/575.(2009)
9. Dhiman Saha, Debdeep Mukhopadhyay, Dipanwita RoyChowdhury. A Diagonal Fault Attack on the Advanced Encryption Standard. Cryptology ePrint Archive, http://eprint.iacr.org/2009/581.(2009)
10. Yongbin ZHOU, Wengling WU, Nannan XU, Dengguo FENG. Differential Fault Attack on Camellia. Chinese Journal of Electronics, Vol.18, No.1, pp. 13–19. (2009)
11. Xin-jie ZHAO, Tao WANG. An Improved Differential Fault Attack on Camellia. Cryptology ePrint Archive, http://eprint.iacr.org/2009/585. (2009)
12. Xin-jie ZHAO, Tao WANG. Further Improved Differential Fault Attacks on Camellia by Exploring Fault Width and Depth. Cryptology ePrint Archive, http://eprint.iacr.org/2010/026. (2010)
13. Wei LI, Dawu GU, Juanru LI . Differential fault analysis on the ARIA algorithm. Information Sciences. Elsevier Inc. pp.3727–3737.(2008)
14. Hua CHEN, Wenling WU, and Dengguo FENG. Differential Fault Analysis on CLEFIA. In S. Qing, H. Imai, and G. Wang (Eds.): ICICS 2007, LNCS 4861, pp. 284–295. Springer Heidelberg. (2007)
15. Junko Takahashi and ToshinoriFukunaga. Improved Differential Fault Analysis on CLEFIA. Proceedings of the 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC, IEEE Computer Society, pp 25-34. (2008)
16. Lei ZHANG, Wenling WU, Differential Fault Analysis on SMS4. Chinese Journal of Computers, Vol.29, No.9, pp. 1596–1602. (2006)
17. Wei LI, Dawu GU. An improved method of differential fault analysis on the SMS4 cryptosystem[A]. The First International Symposium on Data, Privacy, and E-Commerce-ISDPE 2007[C]. Chengdu, China, IEEE Computer Society, pp.175-180. (2007)
18. Wei LI, Dawu GU. Differential fault analysis on the SMS4 cipher by inducing faults to the key schedule. Journal on Communications. Vol.29, No.10, pp. 135–142. (2008)
19. Wei LI, Da-wu GU, Yi Wang. Differential fault analysis on the contracting UFN structure, with application to SMS4 and MacGuffin. The Journal of Systems and Software. 2009, pp. 346–354. (2008)
20. Ruilin Li, Bing Sun, Chao Li and JianXiong You. Differential Fault Analysis on SMS4 Using a Single Fault, Cryptology ePrint Archive, http://eprint.iacr.org/2010/063. (2010)
21. Hoch, J.J., Shamir, A.: Fault analysis of stream ciphers. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 240–253. Springer, Heidelberg. (2004)
22. Biham, E., Granboulan, L., Nguyn, P.Q.: Impossible fault analysis of RC4 and differential fault analysis of RC4. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 359–367. Springer, Heidelberg. (2005)

23. M. Hojsik and B. Rudolf. “Floating fault analysis of Trivium,” In: D.R. Chowdhury, V. Rijmen, and A. Das (eds.) INDOCRYPT 2008. LNCS, Heidelberg,Springer,2008,vol. 5365, pp. 239–250. (2008)
24. Yupu HU, Juntao GAO and Qing Liu. Hard Fault Analysis of Trivium. Cryptology ePrint Archive, http://eprint.iacr.org/2009/333. (2009)
25. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit Block cipher CLEFIA. In: Biryukov, A. (ed.) FSE 2007, vol. 4593, pp. 181–195. Springer, Heidelberg.(2007)
26. C.Giraud, H.Thiebeauld, “A survey on fault attacks”. Proceeding of 6th International Conference on Smart Card Research and Advanced Applications(CARDIS’O4), Toulouse,France, pp. 22—27. (2004)