Protecting Last Four Rounds of CLEFIA is Not Enough Against Differential Fault Analysis

Sk Subidh Ali and Debdeep Mukhopadhyay

Dept. of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India.
{subidh,debdeep}@cse.iitkgp.ernet.in

Abstract. In this paper we propose a new differential fault analysis (DFA) on CLEFIA of 128-bit key. The proposed attack requires to induce byte faults at the fourteenth round of CLEFIA encryption. The attack uses only two pairs of fault-free and faulty ciphertexts and uniquely determines the 128-bit secret key. The attacker does not need to know the plaintext. The most efficient reported fault attack on CLEFIA, needs fault induction at the fifteenth round of encryption and can be performed with two pairs of fault-free and faulty ciphertexts and brute-force search of around 20 bits. Therefore, the proposed attack can evade the countermeasures against the existing DFAs which only protect the last four rounds of encryption. Extensive simulation results have been presented to validate the proposed attack. The simulation results show that the attack can retrieve the 128-bit secret key in around one minute of execution time. To the best of authors' knowledge the proposed attack is the most efficient attack in terms of both the input requirements as well as the complexity.

Keywords: Differential Fault Analysis, DFA, Fault Attack, CLEFIA, Generalized Feistel Structure.

1 Introduction

Modern day's cryptographic primitives are secured against classical cryptanalysis techniques. However, when these primitives are implemented in hardware could leak the secret informations in the form of side-channels. The attack which exploits the implementation based weakness of ciphers are known as side-channel attack [1]. There are different side-channel attacks depending on the properties of the hardware implementations they used. Differential fault analysis is one of the most lethal form of side-channel cryptanalysis technique [2]. In this technique the attacker induces faults into the hardware running a cryptographic algorithm by some external stimulus like electromagnetic radiation, thermal variation, glitch in the input lines etc. Then analyzing the fault-free and faulty output she retrieves the secret key.

Fault based attack was originally proposed by Boneh *et* al. [3]. They have shown that faults in the hardware implementation of a RSA crypto-system can leak the entire secret key. The differential fault analysis which uses both the concept of differential cryptanalysis and fault analysis, was introduced by Biham *e*t al. [4]. The first DFA was mounted against Data Encryption Standard (DES) implementation. The attack required to analyze around 50 to 1500 faulty ciphertexts to retrieve the entire secret key. Later on many DFA attacks were implemented on different ciphers like Triple-DES [5], RSA [6–8], ECC [9–11], IDEA [12]. Among these ciphers, the most extensive research was done on Advanced Encryption Standard (AES) [13–20]. The recent attacks on AES show that only a single byte fault induction can reveal upto 120 out of 128-bit secret key.

There were some contributions [21–24], where the authors have shown simple techniques to practically induce faults into the crypto-systems using less expensive devices. These results show that fault based attacks poses a potent threat to the modern day cipher implementations. Therefore, the crypto-systems need to be protected against this kind of attacks. However, the amount of protection required by the implementation is dependent on the potential of the threat posed by the attack. From, the designer's perspective, protecting the crypto-system against attacks cause functional and area overhead. Therefore, she would want a countermeasure with less over-head but with full protection against the existing DFAs. For example, the recent attacks on AES required to induce fault in between seventh and eighth round of the cipher. Therefore, the designer only needs to protect the last four rounds of the encryption. On the other hand the attacker would like to develop an attack which can evade the existing countermeasures.

In 2007, the Sony Corporation introduced a new 128-bit block cipher named CLEFIA [25]. It is a generalized Feistel structure consist of four 32-bit data lines. The cipher is suitable for small and high speed implementations. The first DFA against CLEFIA was proposed by Chen *et al.* [26]. The attack repeatedly induce byte-faults from seventeenth round to fifteenth round of CLEFIA. Using 18 pairs of fault-free and faulty ciphertexts the attack retrieves the secret key. The attack does not require any brute-force search. Fukunaga *et al.* proposed an improved attack on CLEFIA which required only two pairs of fault-free and faulty ciphertexts [27]. They induced two byte-faults at the two F-functions of fifteenth round of encryption to get two faulty ciphertexts. It was observed that a single fault corrupts both input and output of three F-functions. Therefore, a single fault induction not only gives the final round key but also gives the informations of previous round key. However, the attack required to do brute-force search on around 20 bits. This implies that the attacker need to known the plaintext. The existing two attacks shows that to secure CLEFIA against DFA, the designer needs to protect the last four rounds of encryption.

In this paper we propose a new DFA on CLEFIA by inducing byte-faults at the F-functions of fourteenth round of encryption. The proposed attack required two pairs of fault-free and faulty ciphertexts and uniquely determines the 128-bit key. Our attack can also be applied to the CLEFIA implementation where the last four round of the encryption is protected against DFA. Extensive simulation results have been provided which show that the attack takes on an average around one minute of execution time on a *Intel Core*^{TM} 2 *Duo* desktop machine of 3 GHz speed to retrieve the entire secret key.

2 The CLEFIA Block Cipher

In this section we briefly describe CLEFIA. Its is a 128-bit block cipher comes in three different security level with three different key lengths 128, 192 and 256 bits. It follows a generalized Feistel structure with four data line each of width 32 bits. In this section we briefly describe CLEFIA with 128 bits key. For more details one can refer to the CLEFIA specification [25].

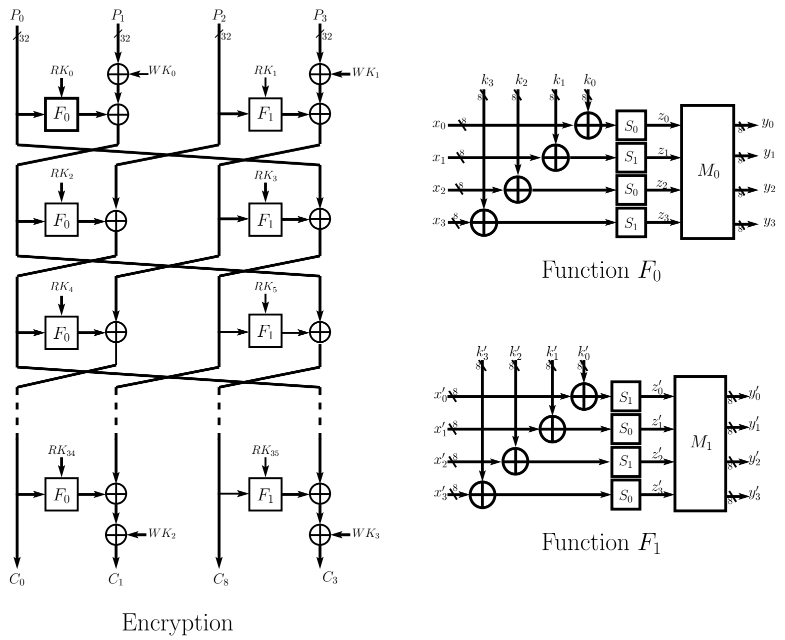


Fig. 1. CLEFIA block diagram

The block diagram of CLEFIA-128 is shown in Figure 1. It follows four way generalized Feistel structure $GFN_{4,18}$, for encryption and decryption. The 16 byte plaintext and Ciphertext are divided into four quartets P_0, P_1, P_2, P_3 and C_0, C_1, C_2, C_3 respectively. The encryption takes four whitening keys WK_0, WK_1, WK_2, WK_3 and 36 round keys RK_{0-35}. The encryption using $GFN_{4,18}$ is shown in Figure 1. Each round of the encryption consists of two F-functions, F_0 and F_1. Both the F-functions uses two non-linear eight bit S-boxes, S_0 and S_1 but in different order. The non-linear operations are followed by a diffusion layer. The diffusion layer is provided by one of the two diffusion matrices M_0 and M_1.

$$M_0 = \begin{pmatrix} 1 & 2 & 4 & 6 \\ 2 & 1 & 6 & 4 \\ 4 & 6 & 1 & 2 \\ 6 & 4 & 2 & 1 \end{pmatrix} \qquad M_1 = \begin{pmatrix} 1 & 8 & 2 & A \\ 8 & 1 & A & 2 \\ 2 & A & 1 & 8 \\ A & 2 & 8 & 1 \end{pmatrix}$$

So, the F-functions take the 32-bit input and ex-ored it with the round key and which then followed by confusion (S-box) and diffusion operations. The out put of F_0 and F_1 are ex-ored with the previous round input of F_1 and F_0 respectively. Then the result is passed to subsequent rounds.

The four whitening keys WK_{0-3} is the copy of 128-bit initial key K. The round keys are generated in two steps, first the intermediate key L is generated from K and then using L the 36 round keys are generated. L is generated by applying the 12 rounds of the four way Feistel structure $GFN_{4,12}$ using K as the input and 24 constant values of 32 bits each as the round keys. The key schedule operation is as follows:

Step 1: $WK_0|WK_1|WK_2|WK_3 \leftarrow K$
Step 2: For $i \leftarrow 0$ to 8
$T \leftarrow L \oplus (CON_{24+4i}|CON_{24+4i+1}|CON_{24+4i+2}|CON_{24+4i+3})$
$L \leftarrow \Sigma(L)$
if i is odd: $T \leftarrow T \oplus K$
$RK4i|RK4i+1|RK4i+2|RK4i+3 \leftarrow T$

where the Σ is known as DoubleSwap function which is expressed as

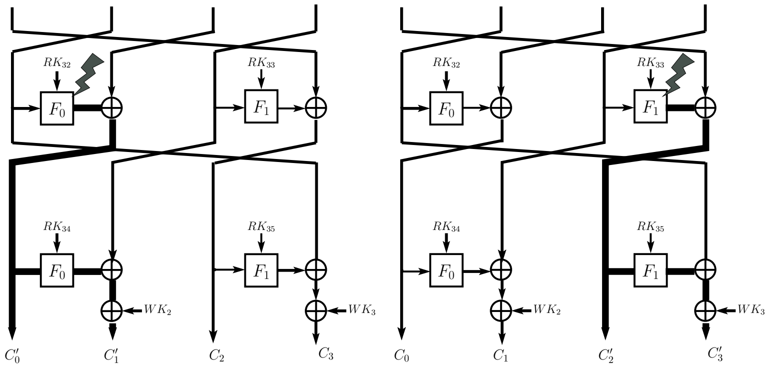
$$\Sigma(L) \leftarrow L_{(7\cdots 63)}|L_{(121\cdots 127)}|L_{(0\cdots 6)}|L_{(64\cdots 120)} \tag{1}$$

3 Related Works

In this section we briefly explain the existing two DFAs on CLEFIA. The attack proposed by Chen *et al.* [26], required repeated induction of byte faults in $(r-1)^{th}$ round so that the fault infect 32-bit input of one of the F-functions of r^{th} round. Then using the fault-free and faulty input and out of the corresponding F-function, differential equations are generated. Solving those equations, corresponding round keys are recovered.

Initially the attacker induce a byte fault in F_0 of the penultimate round (Figure 2(a)). Therefore, the attacker can get the fault-free and faulty inputs (C_0, C_0') of F_0 of final round. She can also get the output difference of F_0 as $\delta_1 = C_1 \oplus C_1'$. Using the value of δ_1, the attacker retrieves the M_0^{-1}. Say $\Delta y_1 = M^{-1}(\delta_1)$. So, now she can deduce following four differential equations:

$$\begin{aligned} S_0(C_{0(0)} \oplus K_{34(0)}) \oplus S_0(C'_{0(0)} \oplus K_{34(0)}) &= \Delta y_{1(0)} \\ S_1(C_{0(1)} \oplus K_{34(1)}) \oplus S_1(C'_{0(1)} \oplus K_{34(1)}) &= \Delta y_{1(0)} \\ S_0(C_{0(2)} \oplus K_{34(2)}) \oplus S_0(C'_{0(2)} \oplus K_{34(2)}) &= \Delta y_{1(0)} \\ S_1(C_{0(3)} \oplus K_{34(3)}) \oplus S_1(C'_{0(3)} \oplus K_{34(3)}) &= \Delta y_{1(0)} \end{aligned} \tag{2}$$



(a) Fault in F_0 (b) Fault in F_1

Fig. 2. Fault induction at seventeenth round

As the input and output differences in the above equations are known therefore the attacker can retrieve the key using S-box difference table. However, due to the differential properties of S-boxes the attacker need on an average three faulty ciphertexts to uniquely determine the four key bytes of RK_{34}.

The attacker follows the same technique to get the value of RK_{35} by inducing fault at F_1 of seventeenth round (Figure 2(b). In the same way the attacker induce faults in the fifteenth and sixteenth rounds and get the values of $RK_{32} \oplus WK_3$, $RK_{33} \oplus WK_2$, RK_{30} and RK_{31}. Then using inverse of Σ function the attack retrieves RK_{32} and RK_{33}. Subsequently, she performs inverse key scheduling operation on the last four round-keys RK_{32-35} and get the master key. So, the attack can retrieve the 128-bit secret key using 18 faulty ciphertexts.

The improved attack on CLEFIA uses only two faulty ciphertexts [27]. In this attack the byte-faults are induced only at the F-functions of fifteenth round. The attacker first retrieves the possible choices of RK_{34} and RK_{35} using the two faulty ciphertexts and then using these values she retrieves the values of $RK_{32} \oplus WK_3$ and $RK_{33} \oplus WK_2$. Then again the values of RK_{30} and RK_{31}. Then she follows the existing technique to get the possible choices of the master key. The authors have theoretically shown that the expected size of possible master key is $2^{19.02}$. So, the attack needs to do brute-force search on the possible keys using the known plaintexts.

Both the two existing attacks exposed the potent threat to CLEFIA implementation when there is a fault. In order to defend CLEFIA implementation against these DFAs, the designer need to protect the last four rounds of encryptions.

In the next section we propose a DFA on CLEFIA which can retrieve the secret key even if the last four rounds are protected. We also show that the key is uniquely determined using two faults, thus requiring no knowledge of plaintexts.

4 Proposed DFA on CLEFIA

In this section we proposed a new DFA on CLEFIA where we assume that the last four rounds of encryption is protected against DFA. Therefore, the attacker can not induce fault in the last four rounds. So she induces faults at the F-functions of the fourteenth round of encryption. The proposed attack is based on the usual single byte fault model and requires two byte-fault induction in the two F-functions. Figure 3 shows the location where the faults are induced. As the faults are induced before the diffusion operation of the F-functions therefore after the diffusion function the fault spread to all the four bytes.

However, the spread of fault follows a pattern. Say the byte faults are induced at x_0 of F_0 and F_1, and the corresponding fault values are p and p'. Due to diffusion matrices, the out put fault pattern becomes $\{p, 2p, 4p, 6p\}$ and $\{p', 8p', 2p', ap'\}$, where 2, 4, 6, 8, a are the 4-bit hexadecimal values and p,p' are the non-zero bytes.

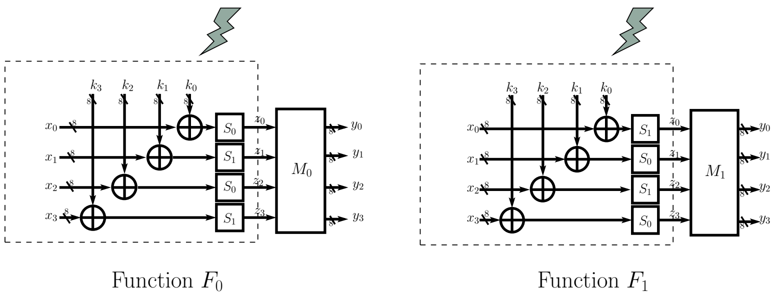
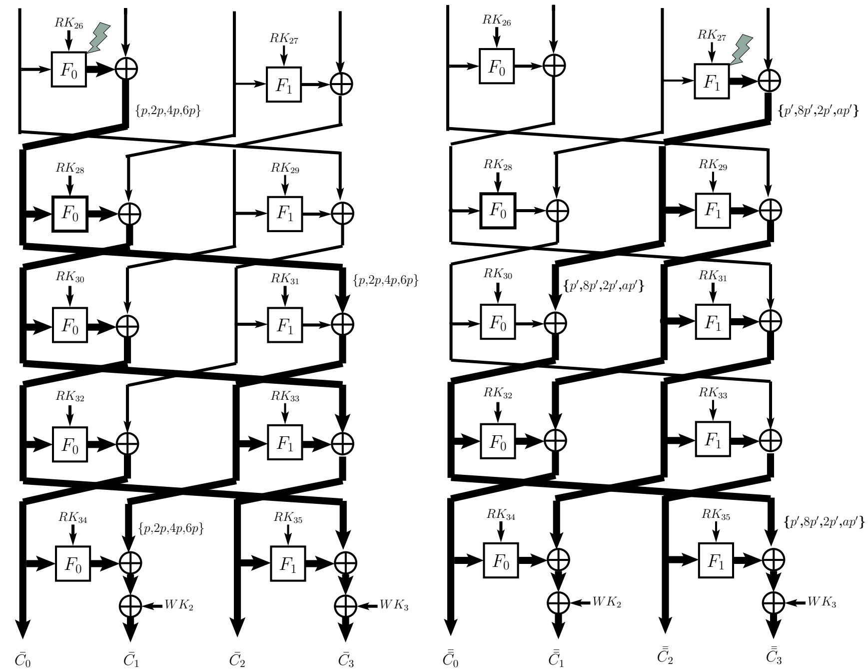


Fig. 3. Fault induction in F-function

The flow of fault in the last five rounds is shown in Figure 4(a) and Figure 4(b). The attacker does not know the value of p and p'. Therefore, she guesses the possible values of (p, p') and for each value she will try to get the round keys in step by step fashion. The attack is divided into two phases. In the first phase, the attacker retrieves the round-keys corresponding to last three rounds and deduce the possible 128-bit master keys. In the second phase the master key is uniquely determined.

4.1 First Phase of the Attack

In this section we first determine the values of RK_{32} RK_{33} RK_{34} RK_{35} corresponding to one choice of (p, p'). Then using these values we determine the possible choices of the master key. We first start with the technique to determine RK_{34} and RK_{35}.



(a) Fault in F_0 of fourteenth round (b) Fault in F_1 of fourteenth round

Fig. 4. Flow of fault in last five rounds

Determining RK_{34} and RK_{35} The value of fault-free ciphertext C and the two faulty ciphertexts $\bar{C}, \bar{\bar{C}}$ are known. Following Figure 4(a) and Figure 4(b), we can directly get the inputs to the last round F-functions. However, getting the output difference is not so obvious. For F_0, the 32-bit input difference is $\Delta\bar{I}_0^{18} = C_0 \oplus \bar{C}_0$ (Figure 4). The corresponding 32-bit output difference $\Delta\bar{Y}_0^{18}$ is given as:

$$
\begin{aligned}
\Delta\bar{Y}_{0(0)}^{18} &= C_{1(0)} \oplus \bar{C}_{1(0)} \oplus p \\
\Delta\bar{Y}_{0(1)}^{18} &= C_{1(1)} \oplus \bar{C}_{1(1)} \oplus 2p \\
\Delta\bar{Y}_{0(2)}^{18} &= C_{1(2)} \oplus \bar{C}_{1(2)} \oplus 4p \\
\Delta\bar{Y}_{0(3)}^{18} &= C_{1(3)} \oplus \bar{C}_{1(3)} \oplus 6p
\end{aligned}
\qquad (3)
$$

For one choice of (p, p') we get one choice $\Delta\bar{Y}_0^{18}$. Using the value $\Delta\bar{Y}_0^{18}$ we can get the corresponding value of inverse of M_0 as $\Delta\bar{Y}^{18} = M_0^{-1}(\Delta\bar{Y}_0^{18})$. So, now we have the input and output difference of the four S-boxes of F_0 using which we can deduce following four differential equations.

$$
\begin{aligned}
S_0(C_{0(0)} \oplus K_{34(0)}) \oplus S_0(C_{0(0)} \oplus K_{34(0)} \oplus \Delta\bar{I}^{18}_{0(0)}) &= \Delta\bar{Y}^{18}_{0(0)} \\
S_1(C_{0(1)} \oplus K_{34(1)}) \oplus S_1(C_{0(1)} \oplus K_{34(1)} \oplus \Delta\bar{I}^{18}_{0(1)}) &= \Delta\bar{Y}^{18}_{0(1)} \\
S_0(C_{0(2)} \oplus K_{34(2)}) \oplus S_0(C_{0(2)} \oplus K_{34(2)} \oplus \Delta\bar{I}^{18}_{0(2)}) &= \Delta\bar{Y}^{18}_{0(2)} \\
S_1(C_{0(3)} \oplus K_{34(3)}) \oplus S_1(C_{0(3)} \oplus K_{34(3)} \oplus \Delta\bar{I}^{18}_{0(3)}) &= \Delta\bar{Y}^{18}_{0(3)}
\end{aligned} \tag{4}
$$

In order to solve the above four equations we use the difference table of S_0 and S_1. The above equations can be generalized to $\Delta Y = S(K \oplus \Delta I) \oplus S(K)$. In order to get the S-box difference table, we store all the values K corresponding to one choice of the input-output difference pair $(\Delta Y, \Delta I)$. In the above four equations $\Delta\bar{I}$ corresponds to ΔI and $\Delta\bar{Y}_0^{18}$ corresponds to ΔY. Therefore, using the input-output deference pair we can get value $C_0 \oplus K_{34}$ from the difference table. To get the actual key we need to ex-or the result with C_0.

However, the number of solutions of K given the input-output difference pair $(\Delta Y, \Delta I)$, is depend on the S-box table we choose. For S_0, we can get 0, 2, 4, 6, 8, or 10 solutions of K. The expected number of nonzero solutions is 2.257 [27]. For S_1, the number of solutions could be 0, 2, or 4, and the expected number of non-zero solutions is 2.024.

Therefore, from the above four equations, for one choice of p we get an expected $2.257^2 \times 2.024^2 = 2^{4.76}$ choices of the 32-bit key K_{34}. We follow the same technique for getting the round key RK_{35} using the faulty ciphertext $\bar{\bar{C}}$ where the input-output differences are given as follows:

$$
\begin{aligned}
\bar{\bar{I}}^{18}_1 &= C_2 \oplus \bar{\bar{C}}_2 \\
\bar{\bar{Y}}^{18}_1 &= M_0^{-1}\big(C_{3(0)} \oplus \bar{\bar{C}}_{3(0)} \oplus p' \;| \\
&\qquad C_{3(1)} \oplus \bar{\bar{C}}_{3(1)} \oplus 8p'| \\
&\qquad C_{3(2)} \oplus \bar{\bar{C}}_{3(2)} \oplus 2p'| \\
&\qquad C_{3(3)} \oplus \bar{\bar{C}}_{3(3)} \oplus ap'\big)
\end{aligned} \tag{5}
$$

Determining $RK_{32} \oplus WK_3$ and $RK_{33} \oplus WK_2$ Once we have the value of RK_{34} and RK_{35}, we can get the outputs of both the F-functions in the last round. We use the first faulty ciphertext $\bar{C}$ to get the values of $RK_{32} \oplus WK_3$. The input output differences of F_0 of seventeenth round is given as follows:

$$
\begin{aligned}
\Delta\bar{I}^{17}_0 &= F_1(C_2, RK_{35}) \oplus F_1(\bar{C}_1, RK_{35}) \oplus C_3 \oplus \bar{C}_3 \\
\Delta\bar{Y}^{17}_0 &= M_0^{-1}(C_0 \oplus \bar{C}_0)
\end{aligned} \tag{6}
$$

It may be noted that the above input and output differences correspond to $RK_{32} \oplus WK_3$, not the actual round key RK_{32}. Therefore, using the S-box difference table we will get the value of $RK_{32} \oplus WK_3$.

Similarly, using the faulty ciphertext $\bar{\bar{C}}$ we can get the following input and output differences of seventeenth round F_1.

$$\begin{aligned} \Delta\bar{\bar{I}}_1^{17} &= F_0(C_0, RK_{34}) \oplus F_0(\bar{\bar{C}}_1, RK_{34}) \oplus C_1 \oplus \bar{\bar{C}}_1 \\ \Delta\bar{\bar{Y}}_1^{17} &= M_1^{-1}(C_2 \oplus \bar{\bar{C}}_2) \end{aligned} \tag{7}$$

Using the above differences we retrieve $RK_{33} \oplus WK_2$.

Determining $\boldsymbol{RK_{30}}$ and $\boldsymbol{RK_{31}}$ In order to get RK_{30} we again use first faulty ciphertext $\bar{C}$ and the already determined round keys. The input and output differences to the sixteenth round F_0 is determined as follows:

$$\begin{aligned} \Delta\bar{I}_0^{16} =& F_1(F_0(C_0, RK_{34}) \oplus C_1, RK_{33} \oplus WK_2) \oplus \\ & F_1(F_0(\bar{C}_0, RK_{34}) \oplus \bar{C}_1, RK_{33} \oplus WK_2) \oplus C_2 \oplus \bar{C}_2 \\ \Delta\bar{Y}_0^{16} =& M_0^{-1}(F_1(C_2, RK_{35}) \oplus F_1(\bar{C}_2, RK_{35}) \oplus C_3 \oplus \bar{C}_3) \end{aligned} \tag{8}$$

It may be observed that the value WK_2 cancels in the above differences. Therefore, from S-box difference table we only get the value of RK_{31}.

In case of F_1 of sixteenth round, the input and output differences are determined from the second faulty ciphertext $\bar{\bar{C}}$. The differences are as follows:

$$\begin{aligned} \Delta\bar{\bar{I}}_1^{16} =& F_0\big(F_1(C_2, RK_{35}) \oplus C_3, RK_{32} \oplus WK_3\big) \oplus \\ & F_0\big(F_1(\bar{\bar{C}}_2, RK_{35}) \oplus \bar{\bar{C}}_3, RK_{32} \oplus WK_3\big) \oplus C_3 \oplus \bar{\bar{C}}_3 \\ \Delta\bar{\bar{Y}}_1^{16} =& M_1^{-1}\big(F_0(C_0, RK_{34}) \oplus F_0(\bar{\bar{C}}_0, RK_{34}) \oplus C_1 \oplus \bar{\bar{C}}_1\big) \end{aligned} \tag{9}$$

So, using above differences we get RK_{31}.

Determining the Possible final Round Keys At this point we have the values of $RK_{34}, RK_{35}, RK_{33} \oplus WK_2$, $RK_{32} \oplus WK_3$, RK_{30} and RK_{31}. In order to extract the values RK_{34} and RK_{35}, from $RK_{33} \oplus WK_2$, and $RK_{32} \oplus WK_3$ we use the key scheduling algorithm of CLEFIA. From RK_{34} and RK_{35}, we can get the right half of the final round intermediate key L ($\Sigma^8(L)$) from the key expansion part of the CLEFIA key schedule. When $i = 8$ we have,

$$(L_2|L_3) = (RK_{34} \oplus CON_{58}^{128} | RK_{35} \oplus CON_{59}^{128}) \tag{10}$$

We do inverse DoubleSwap on L to get the first 57 bits of $(WK_2|WK_3)$. This implies we get the value of WK_2 and 25 bits of WK_3. Using the value of WK_2 we get the value of RK_{33} from $RK_{33} \oplus WK_2$. Again using the value of RK_{33} we get the last seven bits of WK_3. So, finally we get the value of entire last two round keys. We follow the inverse key scheduling algorithm and get the initial key K.

4.2 Second Phase of the Attack

In order to determine the actual key from the possible choices first master keys, we deduce the values of RK_{28} and RK_{29} from the values of $RK_{34}, RK_{35}, RK_{33} \oplus$

WK_2, and $RK_{32}{\oplus}WK_3$. Using the key expansion phase of CLEFIA we can again deduce the value of RK_{28} and RK_{29} from the master keys generated in the first phase of the attack. The intersection of these two lists will uniquely determine the master key.

In Figure 4(a), we can see that the input difference of fifteenth round F_0 is

$$\Delta\bar{I}_0^{15} = (p, 2p, 4p, 6p) \tag{11}$$

The output difference $\Delta\bar{Y}_0^{15}$ can be written as:

$$\begin{aligned}\Delta\bar{Y}_0^{15} = M_0^{-1}(&F_1(F_0(C_0, RK_{34}) \oplus C_1, RK_{33} \oplus WK_2)\oplus \\ &F_1(F_0(\bar{C}_0, RK_{34})\bar{C}_1, RK_{33} \oplus WK_2) \oplus C_2 \oplus \bar{C}_2)\end{aligned} \tag{12}$$

Therefore, using these input and output differences we can retrieve the value of RK_{28}.

In Figure 4(a), the input differences of fifteenth round F_1 is

$$\Delta\bar{I}_1^{15} = (p', 8p', 2p', ap') \tag{13}$$

The output difference can be given as,

$$\begin{aligned}\Delta\bar{Y}_1^{15} = M_1^{-1}\big(&F_0(F_1(C_2, RK_{35}) \oplus C_3, RK_{32} \oplus WK_3)\oplus \\ &F_0(F_1(\bar{\bar{C}}_0, RK_{34}) \oplus \bar{\bar{C}}_3, RK_{33} \oplus WK_2) \oplus C_0 \oplus \bar{\bar{C}}_0\big)\end{aligned} \tag{14}$$

This input output difference will retrieve the value of RK_{29}.

From the first phase, we already know the value of L when $i = 7$ (i.e. $\Sigma^7(L)$), and the value of WK_0 and WK_1, which corresponds to master key. Therefore, from the CLEFIA key expansion phase we can get the values of RK_{28} and RK_{29} using the value of L and WK_0 and WK_1.

$$(RK_{28}|RK_{29}) = (L_0 \oplus WK_0 \oplus CON_{52}^{128}|L_1 \oplus WK_1 \oplus CON_{53}^{128}) \tag{15}$$

So, now we have the values of $(RK_{28}|RK_{29})$ from the first phase as well as deduced in this phase. We make the inter-section of these two lists of values. Only one value will come out of the inter-section. The key corresponding the that value of $(RK_{28}|RK_{29})$ will be the actual master key. The summary of the two phase attack is given in Algorithm 1

4.3 Analysis of the Attack

It is obvious from the existing analysis in [27] that for one choice of (p, p') the expected number of choices of final key from the first phase is $2^{19.02}$. However, for all possible choices 2^8 of p only $2^8 \times 0.037 = 9.472$ will produce RK_{34} ($\langle RK_{34}\rangle \neq \emptyset$) [28, §5.3]. Similarly, for p' only 9.472 choices will produce the value of RK_{35}. Therefore, the expected number of master keys from the first phase is $9.472 \times 9.472 \times 2^{19.02} = 2^{25.507}$.

Algorithm 1: DFA on CLEFIA

Input: $C, \bar{C}, \bar{\bar{C}}$
Output: K

for *Each candidates of* $\{p, q\}$ **do**
  Get $\{RK_{34}, RK_{35}\}$.
  for *Each candidates of* $\{RK_{34}, RK_{35}\}$ **do**
    Get $RK_{32} \oplus WK_3$ and $RK_{33} \oplus WK_2$
    for *Each candidates of* $\{RK_{32} \oplus WK_3,\ RK_{33} \oplus WK_2\}$ **do**
      Get RK_{30}, and RK_{31}.
      Get RK_{28}, and RK_{29}.
      Get RK_{32}, and RK_{33} from $RK_{34}, RK_{35}, RK_{32} \oplus WK_3$, and $RK_{33} \oplus WK_2$.
      for *Each candidates of* $\{RK_{32}, RK_{33}\}$ **do**
        Get L and K from $\{RK_{32}, RK_{33}, RK_{34}, RK_{35}\}$.
        Do $L \leftarrow \Sigma^{-1}(L)$
        if $((RK_{28}|RK_{29}) == (L_0 \oplus WK_0 \oplus CON_{52}^{128}|L_1 \oplus WK_1 \oplus CON_{53}^{128}))$ **then**
          Save K.

In the second phase, we deduce the values of RK_{28} and RK_{29} from $\{RK_{34}, RK_{33} \oplus WK_2\}$ and $\{RK_{35}, RK_{32} \oplus WK_3\}$ respectively. The expected value of $|\langle RK_{34}, RK_{33} \oplus WK_2 \rangle|$ or $|\langle RK_{32}, RK_{32} \oplus WK_3 \rangle|$ is $2^{4.76} \times 9.472 = 2^8$. Therefore, the expected value of $|\langle RK_{28} \rangle|$ or $|\langle RK_{29} \rangle|$ is $2^8 \times 0.037 \times 2^{4.76} = 2^8$ [28, §6.4]. This implies that there are total $2^8 \times 2^8 = 2^{16}$ choices of $\{RK_{28}, RK_{29}\}$. We also deduce the values of $\{RK_{28}, RK_{29}\}$ from the master keys generated in the first phase of the attack which has the expected size of $2^{25.507}$. The intersection of these two list uniquely determines the value of $\{RK_{28}, RK_{29}\}$, which corresponds to the actual master key.

5 Experimental Results

We have performed extensive simulation of the proposed attack. The attack simulation code was written in *C* programming language and compiled using gcc-4.4.3 with *O*3 flag on. The code was executed in Ubuntu-10.4 operating system running on a *Intel Core*^{TM}2 *Duo* desktop machine of 3 GHz speed. In each experiment we used an arbitrary 128-bit key and induced two random single byte faults in the two F-functions of fourteenth round to get the two faulty ciphertexts. Each experiment on a random key was repeated for 256 times. The simulation was performed over a 100 random key-plaintext pairs.

Table 1 shows the results of five such experiments on five random keys. Each row represents the average results of 256 simulations corresponding to a random key. The first column represents the 128-bit random key that has been attacked.

Table 1. Experimental Results

Random 128-bit CLEFIA key	Number of Keys in First Phase	Number of Keys in Second Phase	Running Time (Seconds)
$71b344b86320d3716f566c915bfaa5c2$	$16162164.46 = 2^{23.946}$	1	82.630
$7a052bfbf63a246c838f09766d53aee8$	$12483201.97 = 2^{23.573}$	1	63.688
$b0e8e24e38b682e46abf4767368bcd6b$	$13403507.87 = 2^{23.676}$	1	69.489
$3702237433bd2f12542f4bec01734e64$	$9547074.13 = 2^{23.186}$	1	52.711
$d5659a20b8a945a9566dd7f9f0f886ae$	$11671681.65 = 2^{23.476}$	1	58.819

Second column shows the average number of possible keys generated in the first phase of the attack. The third column shows the number final key deduced from the second phase. The third column shows the average time to perform a successful attack. It is obvious from the above table that the simulated attack on a random key takes around one minute of time to uniquely determine the 128-bit secret key.

6 Comparison

In this section we compare our attack with the existing attacks on CLEFIA. We compare with the help of table 2. The DFA proposed by Chen *et al.* [26] required eighteen pairs of fault-free and faulty ciphertexts to uniquely determine the key. However, the attack need to induce faults at three of the last four rounds. This implies that if only two of the last four rounds of the encryption is protected then the attack will not work. The proposed improved attack [27] required only two pairs of fault-free and faulty ciphertexts. However, the improved attack does not uniquely determine the master key. It needs to do brute-force search on a around $2^{19.02}$ possible keys. Therefore, the attack also needs to know the input plaintexts. The improved DFA present more challenge to the designer as she need to protect last four rounds of encryption against the DFA.

Compared to the existing two attacks our attack requires to induce fault at the fourteenth round of encryption. Therefore, even if the last four round is protected, our attack still can retrieve the secret key. The proposed attack requires only two pairs of fault-free and faulty ciphertexts and uniquely determine the key, thus not require any knowledge of the plaintext.

Table 2. Comparison with existing attacks on CLEFIA

Reference	Fault Model	Fault Location	Number of Faults	Exhaustive Search
[26]	Single byte fault	Fifteenth Round	18	1
[27]	Single byte fault	Fifteenth Round	2	$2^{19.02}$
Our Attack	Single byte fault	Fourteenth Round	2	1

7 Conclusions

In this paper we proposed a new differential fault attack using two pairs of fault-free and faulty ciphertexts where the byte-faults are induced at the fourteenth round of CLEFIA encryption. We retrieved the entire 128-bit secret key without doing any brute-force search. The proposed attack can evade the existing countermeasures which protects the last four round of encryption. The attack will also work in the situation when the attacker only has the access to the ciphertexts and does not have so to the plaintext. The simulation results show that the average time for a successful attack is around one minute on a standard desktop machine, which is indeed practical. To the best of our knowledge this is the most efficient DFA on CLEFIA reported in the literature.

References

1. P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *CRYPTO* (M. J. Wiener, ed.), vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, 1999.
2. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The Sorcerers Apprentice Guide to Fault Attacks." Cryptology ePrint Archive, Report 2004/100, 2004. `http://eprint.iacr.org/`.
3. D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract)," in *EUROCRYPT*, pp. 37–51, 1997.
4. E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," in *CRYPTO* (B. S. K. Jr., ed.), vol. 1294 of *Lecture Notes in Computer Science*, pp. 513–525, Springer, 1997.
5. L. Hemme, "A Differential Fault Attack Against Early Rounds of (Triple-)DES," in *CHES* (M. Joye and J.-J. Quisquater, eds.), vol. 3156 of *Lecture Notes in Computer Science*, pp. 254–267, Springer, 2004.
6. E. Trichina and R. Korkikyan, "Multi Fault Laser Attacks on Protected CRT-RSA," in Breveglieri *et al.* [29], pp. 75–86.
7. A. Pellegrini, V. Bertacco, and T. M. Austin, "Fault-based attack of RSA authentication," in *DATE*, pp. 855–860, IEEE, 2010.

8. J.-S. Coron, C. Giraud, N. Morin, G. Piret, and D. Vigilant, "Fault Attacks and Countermeasures on Vigilant's RSA-CRT Algorithm," in Breveglieri *et al.* [29], pp. 89–96.
9. I. Biehl, B. Meyer, and V. Müller, "Differential Fault Attacks on Elliptic Curve Cryptosystems," in *CRYPTO* (M. Bellare, ed.), vol. 1880 of *Lecture Notes in Computer Science*, pp. 131–146, Springer, 2000.
10. J. Blömer, M. Otto, and J.-P. Seifert, "Sign Change Fault Attacks on Elliptic Curve Cryptosystems," in *FDTC* (L. Breveglieri, I. Koren, D. Naccache, and J.-P. Seifert, eds.), vol. 4236 of *Lecture Notes in Computer Science*, pp. 36–52, Springer, 2006.
11. M. Ciet and M. Joye, "Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Faults," *Des. Codes Cryptography*, vol. 36, no. 1, pp. 33–43, 2005.
12. C. Clavier, B. Gierlichs, and I. Verbauwhede, "Fault analysis study of idea," in *CT-RSA* (T. Malkin, ed.), vol. 4964 of *Lecture Notes in Computer Science*, pp. 274–287, Springer, 2008.
13. Christophe Giraud, "DFA on AES," in *AES Conference* (H. Dobbertin, V. Rijmen, and A. Sowa, eds.), vol. 3373 of *Lecture Notes in Computer Science*, pp. 27–41, Springer, 2004.
14. G. Piret and J.-J. Quisquater, "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD," in *CHES* (C. D. Walter, Çetin Kaya Koç, and C. Paar, eds.), vol. 2779 of *Lecture Notes in Computer Science*, pp. 77–88, Springer, 2003.
15. A. Moradi, M. T. M. Shalmani, and M. Salmasizadeh, "A Generalized Method of Differential Fault Attack Against AES Cryptosystem," in *CHES* (L. Goubin and M. Matsui, eds.), vol. 4249 of *Lecture Notes in Computer Science*, pp. 91–100, Springer, 2006.
16. D. Mukhopadhyay, "An Improved Fault Based Attack of the Advanced Encryption Standard," in *AFRICACRYPT* (B. Preneel, ed.), vol. 5580 of *Lecture Notes in Computer Science*, pp. 421–434, Springer, 2009.
17. M. Tunstall, D. Mukhopadhyay, and S. S. Ali, "Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault," in *WISTP* (C. A. Ardagna and J. Zhou, eds.), vol. 6633 of *Lecture Notes in Computer Science*, pp. 224–233, Springer, 2011.
18. C. H. Kim, "Differential Fault Analysis against AES-192 and AES-256 with Minimal Faults," in Breveglieri *et al.* [29], pp. 3–9.
19. S. Ali and D. Mukhopadhyay, "An Improved Differential Fault Analysis on AES-256," in *AFRICACRYPT* (A. Nitaj and D. Pointcheval, eds.), vol. 6737 of *Lecture Notes in Computer Science*, pp. 332–347, Springer, 2011.
20. S. Ali and D. Mukhopadhyay, "A Differential Fault Analysis on AES Key Schedule Using Single Fault," in *FDTC* (L. Breveglieri, S. Guilley, I. Koren, D. Naccache, and J. Takahashi, eds.), pp. 35–42, IEEE, 2011.
21. S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in *CHES* (B. S. K. Jr., Çetin Kaya Koç, and C. Paar, eds.), vol. 2523 of *Lecture Notes in Computer Science*, pp. 2–12, Springer, 2002.
22. T. Fukunaga and J. Takahashi, "Practical Fault Attack on a Cryptographic LSI with ISO/IEC 18033-3 Block Ciphers," in *FDTC* (L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J.-P. Seifert, eds.), pp. 84–92, IEEE Computer Society, 2009.
23. N. Selmane, S. Guilley, and J.-L. Danger, "Practical Setup Time Violation Attacks on AES," in *EDCC*, pp. 91–96, IEEE Computer Society, 2008.

24. G. Canivet, P. Maistri, R. Leveugle, J. Clédière, F. Valette, and M. Renaudin, "Glitch and Laser Fault Attacks onto a Secure AES Implementation on a SRAM-Based FPGA," *J. Cryptology*, vol. 24, no. 2, pp. 247–268, 2011.
25. S. Corporation, "The 128-bit Blockcipher CLEFIA Algorithm Specification (Revision 1.0,Junw 1, 2007)." `http://www.sony.net/Products/clefia/`.
26. H. Chen, W. Wu, and D. Feng, "Differential Fault Analysis on CLEFIA," in *ICICS* (S. Qing, H. Imai, and G. Wang, eds.), vol. 4861 of *Lecture Notes in Computer Science*, pp. 284–295, Springer, 2007.
27. J. Takahashi and T. Fukunaga, "Improved Differential Fault Analysis on CLEFIA," in *FDTC* (L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J.-P. Seifert, eds.), pp. 25–34, IEEE Computer Society, 2008.
28. M. Tunstall and D. Mukhopadhyay, "Differential Fault Analysis of the Advanced Encryption Standard using a Single Fault." Cryptology ePrint Archive, Report 2009/575, 2009. `http://eprint.iacr.org/`.
29. L. Breveglieri, M. Joye, I. Koren, D. Naccache, and I. Verbauwhede, eds., *2010 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2010, Santa Barbara, California, USA, 21 August 2010*, IEEE Computer Society, 2010.