Improved Differential Fault Analysis of CLEFIA

Sk. Subidh Ali
CRISSP-AD
New York University Abu Dhabi
Abu Dhabi,UAE
Email: sa112-at-nyu-dot-edu

Debdeep Mukhopadhyay
Dept. of Computer Sc. and Engg.
Indian Institute of Technology, India
Email: debdeep-at-cse-dot-iitkgp-dot-ernet-in

***Abstract*—CLEFIA is already shown to be vulnerable to differential fault analysis (DFA). The existing state-of-the-art DFA shows that two faults are enough to break CLEFIA-128, whereas for CLEFIA-192 and CLEFIA-256 ten faults are needed. Side-by-side it emphasizes the need for protecting last four rounds of the cipher in order to make it secure against the attack. In this paper we propose an improved DFA on CLEFIA. The analysis shows that an attack is possible even if the last four rounds of CLEFIA are protected against DFA. Further, the proposed attacks on CLEFIA-192 and CLEFIA-256 show that 8 faults are sufficient to successfully retrieve the 192 and 256-bit key respectively. The work shows improvement over the previous work. 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 whereas the attack on 192 and 256-bit key requires around one second to retrieve the secret key.**

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

I. Introduction

Modern day cryptographic primitives are secured against classical cryptanalysis techniques. However, when these primitives are implemented in hardware could leak the secret information's in the form of side-channels. This kind of physical attack which exploit the implementation based weakness of ciphers are known as side-channel attacks [1]. There is another kind of physical attack which induces faults into the hardware running the encryption and then analysing the fault-free and faulty ciphertexts retrieves the secret key. Fault based attack was originally proposed by Boneh *et al.* [2]. 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 *et al.* [3]. 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 RSA [4], ECC [5], and IDEA [6]. Among these ciphers, the most extensive research was done on Advanced Encryption Standard (AES) [7]–[12], [12]–[14].

The practical aspect of fault analysis is also extensively researched in [15]–[18]. The recent results have shown that fault can be easily induced into the hardware circuit using glitch in the clock input line, voltage variation or laser beam, thus making the fault attack practical.

In 2007, Sony Corporation introduced a new 128-bit block cipher named CLEFIA [19], which later internationally recognized as a lightweight block cipher [20]. CLEFIA has three different versions namely: CLEFIA-128, CLEFIA-192 and CLEFIA-256 based on three different key lengths 128, 192 and 256 bits. It is a 4-branch generalized Feistel structure consists of four 32-bit data lines. The cipher is suitable for small and high speed implementations like Radio Frequency Identification (RFID) tags; wireless sensor networks (WSNs), hand-held smart devices, low-energy wearable/implantable medical devices, and smart cards. The first DFA against CLEFIA was proposed by Chen *et al.* [21]. The attack required to repeatedly induce byte-faults from the seventeenth round to the fifteenth round of CLEFIA-128 and using 18 pairs of fault-free and faulty ciphertexts retrieves the secret key. In case CLEFIA-192 and CLEFIA-256 the proposed DFA required 54 pairs of fault-free and faulty ciphertexts each.

Fukunaga *et al.* proposed an improved attack on CLEFIA. The attack on CLEFIA-128 required only two pairs of fault-free and faulty ciphertexts [22]. The proposed attack is also applicable to CLEFIA-192 and CLEFIA-256, each required 10 faults. In case of CLEFIA-128, 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 information of previous round keys. The existing two attacks show that to secure CLEFIA against DFA, the designer needs to protect the last four rounds of encryption against fault injection.

In this paper we propose improved DFAs on CLEFIA. We propose attacks on all the three versions of CLEFIA with 128, 192 and 256 bits key length. In each of the three attacks the fault is induced in the $(r-4)^{th}$ round, where r is the number of rounds. This implies, our DFA attacks require lesser number of faults, thus making the attack more

978-0-7695-5059-6/13 $26.00 © 2013 IEEE
DOI 10.1109/FDTC.2013.11

practical.

II. The CLEFIA Block Cipher

In this section we briefly describe CLEFIA. It is a 128-bit block cipher comes in three different versions CLEFIA-128, CLEFIA-192, CLEFIA-256, with three different key lengths 128, 192 and 256 bits.

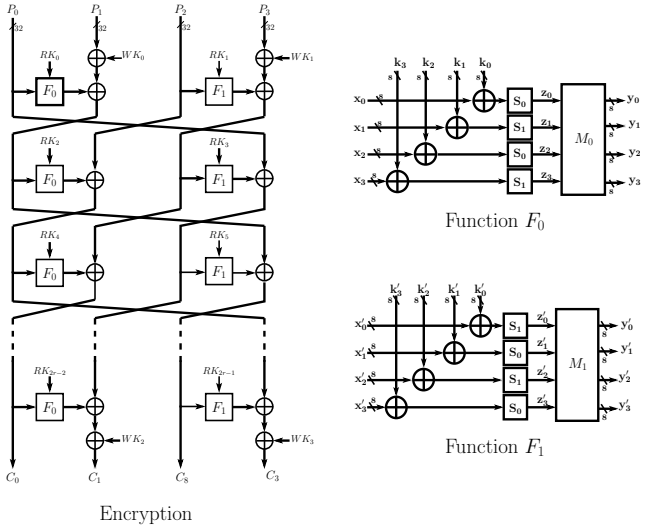


Figure 1. CLEFIA block diagram

The block diagram of CLEFIA is shown in Figure 1. It follows four way generalized Feistel structure $GFN_{4,r}$, for encryption and decryption, where r is the number of rounds. The 16 byte plaintext and ciphertext are divided into four quartets $\langle P_0, P_1, P_2, P_3\rangle$ and $\langle C_0, C_1, C_2, C_3\rangle$ respectively. The encryption takes four whitening keys WK_0, WK_1, WK_2, WK_3 and $2r$ round keys $RK_{0-(2r-1)}$. Each round of the encryption consists of two F-functions, F_0 and F_1. Both the F-functions use 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 output 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 $WK_0 \ldots WK_3$ are generated from the initial key K. The round keys are generated in two steps, first the intermediate key is generated from K and then using the intermediate key and the initial key, the round keys are generated. For CLEFIA-128, the intermediate key 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 of CLEFIA-128 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 Σ is known as DoubleSwap function which is expressed as

$$\Sigma(L) \leftarrow L_{(7\cdots63)}|L_{(121\cdots127)}|L_{(0\cdots6)}|L_{(64\cdots120)} \qquad (1)$$

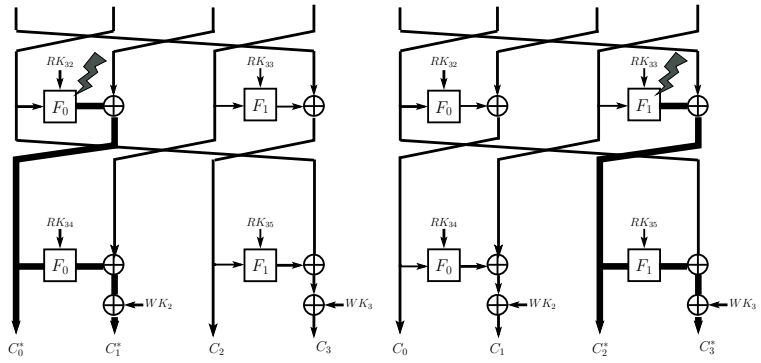
In case of CLEFIA-192 and CLEFIA-256 the two 128 bits values K_L and K_R are generated from the initial key K. Then the intermediate keys L_L and L_R are generated using $GFN_{8,10}$ with $CON_i^k (0 \leq i < 40$ and $r = \{192, 256\})$ as round keys and $K_L|K_R$ as a 256-bit input. The key schedule of CLEFIA-192 and CLEFIA-256 is shown below,

Step 1: Set $k = 192$ or 256
Step 2: If $K = 192 : K_L \leftarrow K_0|K_1|K_2|K_3,\ K_R \leftarrow K_4|K_5|\overline{K_0}|\overline{K_1}$
else if $K = 256 : K_L \leftarrow K_0|K_1|K_2|K_3, K_R \leftarrow K_4|K_5|K_6|K_7$
Step 3: Let $K_L \leftarrow K_{L0}|K_{L1}|K_{L2}|K_{L3}, K_R \leftarrow K_{R0}|K_{R1}|K_{R2}|K_{R3}$
$L_L|L_R \leftarrow GFN_{8,10}(CON_0^{(k)}, \ldots, CON_{39}^{(k)}, K_{L0}, \ldots, K_{L3},$
$K_{R0}, \ldots, K_{R3})$ (Expanding K_L, K_R and L_R, L_L for a k-bit key)
Step 4: $WK_0|WK_1|WK_2|WK_3 \leftarrow K_L \oplus K_R$
Step 5: For $i \leftarrow 0$ to 10(if k==192),or 12(if k=256)
If(i mod 4)=0 or 1:
$T \leftarrow L \oplus (CON_{40+4i}^k|CON_{40+4i+1}^k|CON_{40+4i+2}^k|$
$CON_{40+4i+3}^k)$
$L_L \leftarrow \Sigma(L_L)$
if i is odd: $T \leftarrow T \oplus K_R$
$RK4i|RK4i+1|RK4i+2|RK4i+3 \leftarrow T$
else:
$T \leftarrow L_R \oplus (CON_{40+4i}^k|CON_{40+4i+1}^k|CON_{40+4i+2}^k|$
$CON_{40+4i+3}^k)$
$L_R \leftarrow \Sigma(L_R)$
if i is odd: $T \leftarrow T \oplus K_L$
$RK4i|RK4i+1|RK4i+2|RK4i+3 \leftarrow T$

III. Related Works

In this section we briefly explain the existing two DFAs on CLEFIA. The attack proposed by Chen *et al.* [21], 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 output of the corresponding F-functions, differential equations are generated. Solving those equations, the corresponding round key is recovered.

Initially the attacker induces 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



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

Figure 2. Fault induction at seventeenth round

the M_0^{-1}. Say $\Delta y_1 = M^{-1}(\delta_1)$. So, now she can deduce following four differential equations:

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

As the input and output differences in the above equations are known, the attacker can retrieve the key using S-box difference distribution table. However, due to the differential properties of S-boxes the attack needs on an average three faulty ciphertexts to eniquely 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 induces faults in the fifteenth and the sixteenth rounds and gets 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 gets 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 [22]. In this attack the byte-faults are induced only at the F-functions of the 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.

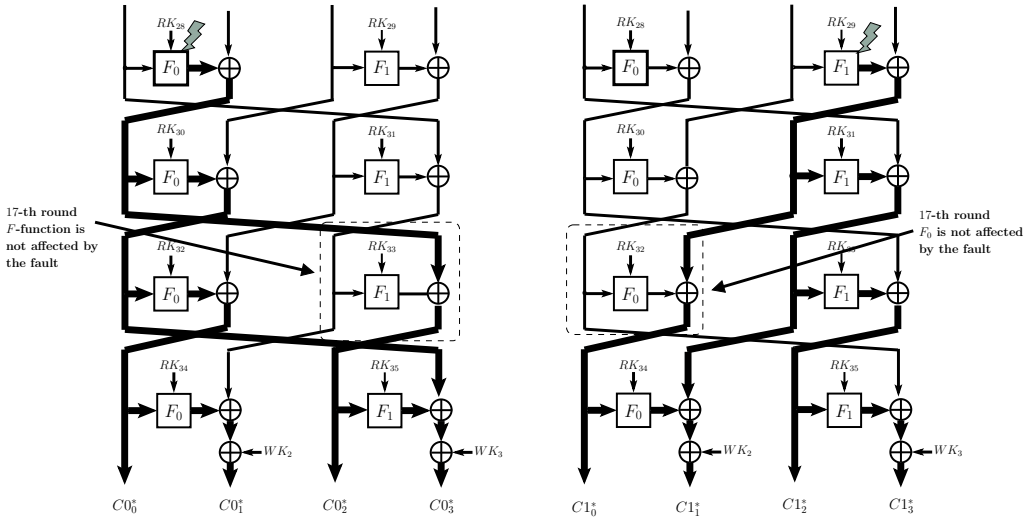
Both the two existing attacks exposed the potent threat to CLEFIA implementation when there is a fault. In order to defend CLEFIA implementation against these DFAs, the designer need to protect the last four rounds of encryptions. In the next section we propose DFA on CLEFIA which can retrieve the secret key even if the last four rounds are protected.

IV. Do the Existing Attacks Fully Utilized the Faults?

The number of required faulty ciphertexts itself shows that the first DFA on CLEFIA [21], does not fully utilised the faults. The practical feasibility of a DFA lies in the number faulty ciphertexts required by the attack. More faulty ciphertexts mean less practical attack. In this regard the first DFA requires a large number of faulty ciphertexts making it impractical. In the second DFA on CLEFIA [23], the utilization of faults is more than the first attack. However, the attack does not use the maximum utilization of faults. In order to retrieve the maximum information from a fault, an attacker has to carefully choose the fault injection point. In case of AES we have noticed that this point is in between 6-th and 7-th round `MixColumns` operations [24]. In this regard the way the fault is induced in the state-of-the-art DFA on CLEFIA, does not spread to all the last four F-functions. Figure 3 shows the fault injection points and the corresponding fault coverage as per the attack. It shows that for the first fault, the 17-th round F_1 does not have any input difference. Therefore, the fault will not give any information of the corresponding round key. Similarly, the second fault will not give any information of the round key corresponding to 17-th round F_0. Therefore, the challenge is to develop an attack by inducing faults one round earlier so that it covers the last four F-functions. Side by side the attack should take minimum number of faulty ciphertexts.

V. Effect of Faults in One Round Earlier

The existing best known DFA on CLEFIA, the attacker retrieves the secret key by inducing two byte faults in the two F-functions: F_0 and F_1 of fifteenth round. For iterative implementation, the fault injection round is not a matter; protecting one round is sufficient to protect the entire algorithm. However, for pipeline implementation it does matter in which round the fault is being injected. In order to protect such implementation of CLEFIA from DFA, the designer not only need to protect last four rounds but also



(a) Fault in F_0 of fifteenth round

(b) Fault in F_1 of fifteenth round

Figure 3. Flow of fault in the last four rounds of CLEFIA

need to protect the first four rounds, as the attacker may have access to decryption process. However, if the fault is induced at the fourteenth round, then on one hand the designer need to protect ten rounds out of eighteen rounds: five rounds at the beginning and five rounds at the end. On the other hand each induced faults will give more information about the key than the existing one. Therefore, from attackers side the advantage is of two folds: first he can attack CLEFIA even if the first and the last four rounds are protected against DFA. Second, he will get information about six round keys from each fault inductions, as the fault passes through six F-functions. For CLEFIA-128, the second advantage might not help but for other two versions of CLEFIA: CLEFIA-192 and CLEFIA-256 a DFA is possible with lesser number of faults.

In the next section we present a DFA on CLEFIA-128 based on fault in fourteenth round F-functions.

VI. Proposed DFA on CLEFIA-128

In this section we propose a DFA attack on CLEFIA-128 where we assume that the last four rounds of encryption are protected against DFA. Therefore, the attacker cannot induce fault in the last four rounds. So she induces faults at the F-functions of the 14^{th} round of encryption. The attack requires two byte-fault inductions in the two F-functions. Figure 4 shows the location where the faults are induced. As the faults are induced before the diffusion operation of the F-functions, the fault spreads to all the four bytes at the output of the F-functions.

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

The flow of faults in the last five rounds are shown in Figure 5(a) and Figure 5(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.

A. First Phase of the Attack

In this section we first determine the values of $RK_{32} \oplus WK_3$, $RK_{33} \oplus WK_2$, RK_{34}, RK_{35} corresponding to one choice of (p, p'). We start with the technique to determine RK_{34} and RK_{35}.

1) Determining RK_{34} and RK_{35}: The value of fault-free ciphertext C and the two faulty ciphertexts $(C0^*, C1^*)$ are known. Following Figure 5(a) and Figure 5(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 $\Delta I_0^{18} = C_0 \oplus C0_0^*$ (Figure 5). The corresponding 32-bit output difference ΔY_0^{18} is given as:

$$\begin{aligned} \Delta Y_{0(0)}^{18} &= C_{1(0)} \oplus C0_{1(0)}^* \oplus p \\ \Delta Y_{0(1)}^{18} &= C_{1(1)} \oplus C0_{1(1)}^* \oplus 2p \\ \Delta Y_{0(2)}^{18} &= C_{1(2)} \oplus C0_{1(2)}^* \oplus 4p \\ \Delta Y_{0(3)}^{18} &= C_{1(3)} \oplus C0_{1(3)}^* \oplus 6p \end{aligned} \tag{3}$$

For one choice of (p, p') we get one choice ΔY_0^{18}. Using the value ΔY_0^{18} we can get the corresponding value of inverse of M_0 as $\Delta Y_0^{18} = M_0^{-1}(\Delta Y_0^{18})$. So, now we have the input

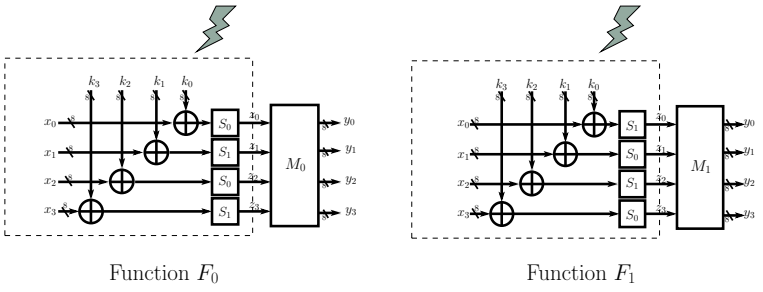
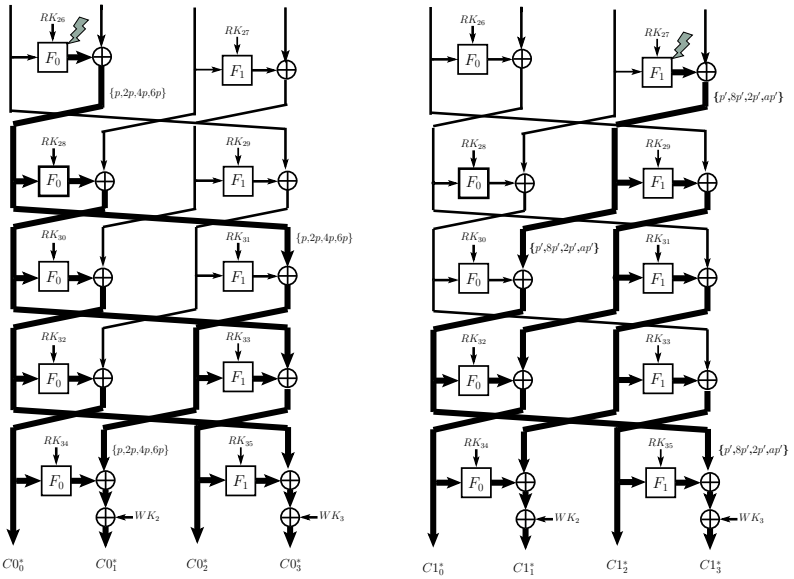


Figure 4. Fault induction in F-function



(a) Fault in F_0 of fourteenth round

(b) Fault in F_1 of fourteenth round

Figure 5. Flow of fault in last five rounds

and output difference of the four S-boxes of F_0 using which we can deduce following four differential equations.

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

In order to solve the above four equations we use the difference distribution table of S_0 and S_1. The above equations can be generalized to $\Delta A = S(K \oplus \Delta B) \oplus S(K)$. In order to get the S-box difference distribution table, we store all the values K corresponding to one choice of the input-output difference pair $(\Delta A, \Delta B)$. In the above four equations ${\Delta I_0}^{18}$ corresponds to ΔB and ΔY_0^{18} corresponds to ΔA. Therefore, using the input-output deference pair we can get value $C_0 \oplus K_{34}$ from the difference distribution table. To get the actual key we need to XOR the result with C_0.

However, the number of solutions of K given the input-output difference pair $(\Delta A, \Delta B)$, is dependent 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 [22]. 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 $C1^*$ where the input-output differences are given as follows:

$$\begin{aligned}
I_1^{18} &= C_2 \oplus C1_2^* \\
Y_1^{18} &= M_0^{-1}(C_{3(0)} \oplus {C1^*}_{3(0)} \oplus p' \ | \\
&\quad C_{3(1)} \oplus {C1^*}_{3(1)} \oplus 8p' | \\
&\quad C_{3(2)} \oplus {C1^*}_{3(2)} \oplus 2p' | \\
&\quad C_{3(3)} \oplus {C1^*}_{3(3)} \oplus ap')
\end{aligned} \tag{5}$$

2) 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 $C0^*$ to get the values of $RK_{32} \oplus WK_3$. The input output differences of F_0 of seventeenth round is given as follows:

$$\begin{aligned} \Delta I_0^{17} &= F_1(C_2, RK_{35}) \oplus F_1(C0_2^*, RK_{35}) \oplus C_3 \oplus C0_3^* \\ \Delta Y_0^{17} &= M_0^{-1}(C_0 \oplus C0_0^*) \end{aligned} \tag{6}$$

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 distribution table we will get the value of $RK_{32} \oplus WK_3$.

Similarly, using the faulty ciphertext $C1^*$ we can get the following input and output differences of seventeenth round F_1.

$$\begin{aligned} \Delta I_1^{17} &= F_0(C_0, RK_{34}) \oplus F_0(C1_0^*, RK_{34}) \oplus C_1 \oplus C1_1^* \\ \Delta Y_1^{17} &= M_1^{-1}(C_2 \oplus C1_2^*) \end{aligned} \tag{7}$$

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

3) Determining RK_{30} and RK_{31}: In order to get RK_{30} we again use first faulty ciphertext $C0^*$ and the already determined round keys. The input and output differences to the sixteenth round F_0 is determined as follows:

$$\begin{aligned} \Delta I_0^{16} =& F_1(F_0(C_0, RK_{34}) \oplus C_1, RK_{33} \oplus WK_2) \oplus \\ & F_1(F_0(C_0, RK_{34}) \oplus C0_1^*, RK_{33} \oplus WK_2) \oplus C_2 \oplus C0_2^* \\ \Delta Y_0^{16} =& M_0^{-1}(F_1(C_2, RK_{35}) \oplus F_1(C0_2^*, RK_{35}) \oplus C_3 \oplus C0_3^*) \end{aligned} \tag{8}$$

It may be observed that the value WK_2 cancels in the above differences. Therefore, from S-box difference distribution 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 $C1^*$. The differences are as follows:

$$\begin{aligned} \Delta I_1^{16} =& F_0\big(F_1(C_2, RK_{35}) \oplus C_3, RK_{32} \oplus WK_3\big) \oplus \\ & F_0\big(F_1(C1_2^*, RK_{35}) \oplus C1_3^*, RK_{32} \oplus WK_3\big) \oplus C_3 \oplus C1_3^* \\ \Delta Y_1^{16} =& M_1^{-1}\big(F_0(C_0, RK_{34}) \oplus F_0(C1_0^*, RK_{34}) \oplus C_1 \oplus C1_1^*\big) \end{aligned} \tag{9}$$

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

4) Determining $RK_{28} \oplus WK_2$ and $RK_{29} \oplus WK_3$: One added advantage of inducing fault in the fourteenth round is that we can determine one more round key than the existing attack. Though for our attack only the last four round keys are sufficient to recover the master key. However, for the sake of generality of all the three proposed attack we determine the fifteenth round keys. In Figure 5(a), we can see that the input difference of fifteenth round F_0 is

$$\Delta I_0^{15} = (p, 2p, 4p, 6p) \tag{10}$$

The output difference ΔY_0^{15} can be written as:

$$\begin{aligned} \Delta Y_0^{15} = M_0^{-1}(& F_1(F_0(C_0, RK_{34}) \oplus C_1, RK_{33} \oplus WK_2) \oplus \\ & F_1(F_0(C0_0^*, RK_{34}) \oplus C0_1^*, RK_{33} \oplus WK_2) \oplus C_2 \oplus C0_2^*) \end{aligned} \tag{11}$$

Therefore, using these input and output differences we can retrieve the value of $RK_{28} \oplus WK_2$.

In Figure 5(a), the input difference of fifteenth round F_1 is

$$\Delta I_1^{15} = (p', 8p', 2p', ap') \tag{12}$$

The output difference can be given as,

$$\begin{aligned} \Delta Y_1^{15} = M_1^{-1}\big(& F_0(F_1(C_2, RK_{35}) \oplus C_3, RK_{32} \oplus WK_3) \oplus \\ & F_0(F_1(C1_0^*, RK_{34}) \oplus C1_3^*, RK_{33} \oplus WK_2) \oplus C_0 \oplus C1_0^*\big) \end{aligned} \tag{13}$$

This input output differences will retrieve the value of $RK_{29} \oplus WK_3$.

B. Second Phase of the Attack

From the first phase of the attack we already determine the possible values of $RK_{34}, RK_{35}, RK_{33} \oplus WK_2$, $RK_{32} \oplus WK_3$, RK_{30} and RK_{31}. In order to get the master key we use the key scheduling algorithm of CLEFIA. From RK_{34} and RK_{35}, we can get the right half of $\Sigma^8(L)$ by following the key expansion part of CLEFIA-128 key schedule. As per the key schedule, when $i = 8$ we have

$$(RK_{34}|RK_{35}) = \Sigma^8(L_2|L_3) \oplus (CON_{58}^{128}|CON_{59}^{128}) \tag{14}$$

As the value of RK_{34} and RK_{35} are known therefore we can get the value of $\Sigma(L_2|L_3)$. We do inverse DoubleSwap on $\Sigma^8(L)$ to get the first 57 bits of $\Sigma^7(L_2|L_3)$. Again from the key schedule we have following relation when $i = 7$,

$$(RK_{30}|RK_{31}) = \Sigma^7(L_2|L_3) \oplus (K_2|K_3) \oplus (CON_{54}^{128}|CON_{55}^{128}) \tag{15}$$

So, now we can get the most significant 57 bits of $(K_2|K_3)$ which corresponds to $(WK_2|WK_3)$. 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. Finally we get the value of entire last two round keys and the value of L. We do the inverse of $GFN_{4,12}$ on L and get the value of K. In order to uniquely determine the master key we check whether the last two words of K is same as the value of $(WK_2|WK_3)$. Due to the properties of $GFN_{4,12}$, only one choice of master key will satisfy the condition.

The summary of the two phase attack is given in Algorithm 1

C. Analysis of the Attack

It is obvious from the existing analysis in [22] that for one choice of (p, p') the expected number of choices of final key from the first phase is $2^{19.02}$. However, all the 2^8 possible choices of p will not produce RK_{34}. It may be noted that in equation (4), all the four elements of ΔY_0^{18}

Algorithm 1: DFA on CLEFIA

Input: $C, C0^*, C1^*$
Output: K

```
for Each candidates of {p, p'} do
    Get {RK34, RK35}.
    for Each candidates of {RK34, RK35} do
        Get RK32 ⊕ WK3 and RK33 ⊕ WK2
        for Each candidates of {RK32 ⊕ WK3,
        RK33 ⊕ WK2} do
            Get RK30, and RK31.
            Get right half of Σ^8(L) from RK34, and RK35.
            Get WK2 from Σ^7(L) ⊕ K.
            Get RK33, WK3 and RK32.
            Get L from RK32, RK33, RK34, RK35.
            for Each candidates of L do
                Get K
                if ((WK2|WK3) == (K2|K3)) then
                    Save K.
```

are not dependent on p. Only the first element $\Delta Y^{18}_{0(0)}$ is dependent on p, rest of the three elements are independent. For example in the second element $\Delta Y^{18}_{0(1)}$, the value of p can be written as $2 \cdot p \oplus 1 \cdot 2p \oplus 4 \cdot 6p \oplus 6.4p$. Therefore, p will cancel out in the expression. This is also true for equation (5). So, now the number of possible choices p, which produce RK_{34} can be written as, $2^8 \times 0.611 = 156.416 = 2^{7.289}$ ($\langle RK_{34} \rangle \neq \emptyset$) [22, §5.3]. Similarly, for p', only $2^8 \times 0.506 = 129.536 = 2^{7.017}$ will produce RK_{35}. Following steps show the possible candidates of the round keys.

Step 1. Expected value of $\langle p, p' \rangle$ is $2^{7.289} \times 2^{7.017} = 2^{14.306}$.

Step 2. Expected value of $\langle p, p', RK_{34}, RK_{35} \rangle$ producing $RK_{32} \oplus WK_3$ is $2^{14.306} \times 2^{4.76} \times 2^{4.76} \times 0.037 = 2^{19.066}$.

Step 3. Expected value of $\langle p, p', RK_{34}, RK_{35}, RK_{32} \oplus WK_3 \rangle$ producing $RK_{33} \oplus WK_2$ is $2^{19.066} \times 2^{4.76} \times 0.037 = 2^{19.066}$.

Step 4. Expected value of $\langle p, p', RK_{34}, RK_{35}, RK_{32} \oplus WK_3, RK_{33} \oplus WK_2 \rangle$ producing RK_{30} is $2^{19.066} \times 2^{4.76} \times 0.037 = 2^{19.066}$.

Step 5. Expected value of $\langle p, p', RK_{34}, RK_{35}, RK_{32} \oplus WK_3, RK_{33} \oplus WK_2, RK_{30} \rangle$ producing RK_{31} is $2^{19.066} \times 2^{4.76} \times 0.037 = 2^{19.066}$.

Step 6. Expected value of $\langle p, p', RK_{34}, RK_{35}, RK_{32} \oplus WK_3, RK_{33} \oplus WK_2, RK_{30}, RK_{31} \rangle$ is $2^{19.066} \times 2^{4.76} = 2^{23.826}$

So, the expected number of master key is $2^{23.826}$. Using inverse key schedule operation the master key is uniquely determined. The time complexity of the attack is $2^{23.826}$.

VII. Proposed Attack on CLEFIA-192 and CLEFIA-256

In this section we propose DFA on CLEFIA-192 and CLEFIA-256 with minimum number of faults. The attack strategy is same for CLEFIA-192 and CLEFIA-256. In each case the attack takes eight pairs of fault-free and faulty ciphertexts. Two faults are induced each in the $(r-4)^{th}$ round F_0 and F_1, where r is the number of rounds ($r = 22$ in CLEFIA-192 and $r = 26$ in CLEFIA-256). Therefore, the flow of fault in the last four rounds are same as in Figure 5. Using total four faults we recover the last four round keys. Afterward we induce two more faults each in the $(r-8)^{th}$ round F_0 and F_1 to recover four more round keys.

The existing attack strategy on CLEFIA-192 and CLEFIA-256 requires to recover the last nine round keys by inducing ten faults in three different rounds. For example, in case of CLEFIA-192 the attacker needs to recover RK_{43},RK_{42},$RK_{41} \oplus WK_2$,$RK_{40} \oplus WK_3$, RK_{39},RK_{38},$RK_{37} \oplus WK_3$,$RK_{36} \oplus WK_2$, RK_{35},RK_{34},$RK_{33} \oplus WK_2$,$RK_{32} \oplus WK_3$, RK_{31},RK_{30},$RK_{29} \oplus WK_3$,$RK_{28} \oplus WK_2$, RK_{27},RK_{26}. In our proposed technique we don't need to recover $RK_{27}2$, and RK_{26}. Only the last eight rounds keys are sufficient to determine the master key. Moreover if the last four rounds are protected against DFA, the existing attacks will fail whereas our attacks will still work.

We propose a two phase attack strategy on CLEFIA-192 and CLEFIA-256. The attack on CLEFIA-256 is same as the attack of CLEFIA-192. Therefore, we only describe the attack on CLEFIA-192.

A. First Phase of the Attack

In the first phase of the attack we determine the last eight round keys in step-by-step fashion. At first we induce four faults in the 18^{th} round F-functions: two faults in F_0 and two faults in F_1, and determine the last four round keys. Once we have the last four round keys, we again induce four faults in the 14^{th} round F-functions: two faults in F_0 and two faults in F_1. Based on these four faults we determine four more round keys, round sixteen to round eighteen.

Let $C0^*$, $C1^*$ (fault in F_0) and $C2^*$, $C3^*$ (fault in F_1) be the four faulty ciphertexts corresponding to the four faults at 18^{th} round F-functions. Using these four faulty ciphertexts and the fault-free ciphertext we determine the last four round keys in step-by-step fashion.

1) Determining RK_{42} and RK_{43}: Unlike the DFA on CLEFIA-128, here we have two faulty ciphertexts to determine each of the round keys RK_{42} and RK_{43}. Let p_0, and p_1 be the values of p corresponding to the two faulty ciphertexts $C0^*$ and $C1^*$ (Figure 5(a)). Similarly, consider p'_0, and p'_1 be the values of p' corresponding to the two faulty ciphertexts $C2^*$ and $C3^*$ (Figure 5(b)). Therefore, according to the Section VI-A1, we get two sets of values of RK_{42} corresponding to p_0 and p_1. However, we consider

only those values of RK_{42} which are common to both the sets. The inter-section is based on only the first byte of the RK_{42}. As only the first byte is dependent on the value of p. Therefore, for all possible values of (p_0, p_1) we only consider those values of RK_{42}, which are same in first byte and discard those which are different, for p_0 and p_1. The expected number (p_0, p_1) producing RK_{42} is $2^8 \times 0.506 \times 2^8 \times 0.611 = 2^{14.306}$. Among these values, only $\frac{2^{14.306}}{2^8} = 2^{6.306}$ values are same in the first byte of RK_{42} for p_0 and p_1. We save the values of RK_{42} and corresponding values of (p_0, p_1).

We follow the same technique to determine RK_{43}.

2) Determining $RK_{40} \oplus WK_3$ and $RK_{41} \oplus WK_2$: Once we have the value of RK_{43} we can determine the value of $RK_{40} \oplus WK_3$ by following the technique proposed in Section VI-A2 to determine $RK_{32} \oplus WK_3$ and $RK_{33} \oplus WK_2$. However, in this case we have two input-output differences of 21^{st} round F_0 function corresponding to two faulty ciphertexts $C2^*$ and $C3^*$. Therefore, we get two sets of values of $RK_{40} \oplus WK_3$ corresponding to two faulty ciphertexts. The expected number of values in each set is given by $2^{6.306} \times 0.037 \times 2^{4.76} = 2^{6.306}$. The intersection of these two sets will uniquely determine the value of $RK_{40} \oplus WK_3$. This can be explain as follows. For, one choice of RK_{43}, we get two sets of values of $RK_{40} \oplus WK_3$ corresponding to two faulty ciphertexts. In case of actual key the value of $RK_{40} \oplus WK_3$ must be same for both the faulty ciphertexts. Therefore, we consider only those values of RK_{43} which gives same values of $RK_{40} \oplus WK_3$ corresponding to two different faulty ciphertexts. The probability that two words are same in all four bytes is $\frac{1}{2^{32}}$. Therefore, the expected number of RK_{43} giving same value of $RK_{40} \oplus WK_3$ for two different faulty ciphertexts is $\frac{2^{6.306}}{2^{32}}$. This implies only the actual value of RK_{43} and $RK_{40} \oplus WK_3$ will satisfy the test. Same way we determine $RK_{41} \oplus WK_2$ from RK_{42}.

3) Determining RK_{38} and RK_{39}: In this case we have uniquely determined the value of RK_{42}, RK_{43}, $RK_{40} \oplus WK_3$ and $RK_{41} \oplus WK_2$. Now, we apply the technique proposed in Section VI-A3 to determine RK_{30} and RK_{31}. Here also we get two sets of values of RK_{38} corresponding to two faulty ciphertexts. Intersection of these two sets will uniquely determine the value of RK_{38}. Similarly, we can uniquely determine the value of RK_{39}.

4) Determining $RK_{36} \oplus WK_2$ and $RK_{37} \oplus WK_3$: The advantage of inducing fault at $r-4$ round is that we can determine one extra round keys compare to existing attacks. For our case we can determine $RK_{36} \oplus WK_2$ and $RK_{37} \oplus WK_3$. The technique is already shown in Section VI-A4, where we determine the value of $RK_{28} \oplus WK_2$ and $RK_{29} \oplus WK_3$. In our case, using four pairs of fault-free and faulty ciphertexts we can uniquely determine the value $RK_{36} \oplus WK_2$ and $RK_{37} \oplus WK_3$.

5) Determining Rest of the Round keys: Once we have the last four round keys (or the combination of the round keys), we induce four more faults in the 14^{th} round F-functions: two faults in F_0 and two faults in F_1. We follow the above technique and uniquely determine the values of RK_{35}, RK_{34}, $RK_{33} \oplus WK_2$, $RK_{32} \oplus WK_3$, RK_{31}, RK_{30}, $RK_{29} \oplus WK_3$, $RK_{28} \oplus WK_2$. Therefore, using eight pairs of fault free and faulty ciphertexts we uniquely determine the last eight round keys (or the combination of round keys).

In the second phase of the attack we determine the master key from the last eight round keys.

B. Second Phase of the Attack

We have already uniquely determined the values of RK_{43}, RK_{42}, RK_{39}, RK_{38}, RK_{35}, RK_{34}, RK_{31}, and RK_{30}. From the CLEFIA-192 key schedule we have following relation (when $i = 10$),

$$\begin{aligned}(RK_{40}|RK_{41}|RK_{42}|RK_{43}) =& \Sigma^4(L_R) \oplus (CON_{80}^{192}|CON_{81}^{192}| \\ & CON_{82}^{192}|CON_{83}^{192}) \\ \Rightarrow (RK_{42}|RK_{43}) =& \Sigma^4(L_{R(2)}|L_{R(3)}) \oplus (CON_{82}^{192}|CON_{83}^{192})\end{aligned}$$

We know the value of RK_{43} and RK_{42}, therefore, we can determine the last two words of $\Sigma^4(L_R)$ i.e. $\Sigma^4(L_{R(2)}|L_{R(3)})$. From, these value we can determine the most significant 57 bits of $\Sigma^3(L_{R(2)}|L_{R(3)})$ by performing the inverse of DoubleSwap operation. Therefore, we get the value of $\Sigma^3(L_{R(2)})$ and the most significant 25 bits of $\Sigma^3(L_{R(3)})$. Again when $i = 7$ we get the following relation,

$$\begin{aligned}(RK_{30}|RK_{31}) =& \Sigma^3(L_{R(2)}|L_{R(3)}) \oplus (K_{L(2)}|K_{L(3)}) \oplus \\ & (CON_{70}^{192}|CON_{71}^{192})\end{aligned}$$

So, we can get the most significant 57 bits of the second half of K_L. Similarly, when $i = 8$, we can recover the second half of L_L from the following relation,

$$(RK_{34}|RK_{35}) = \Sigma^4(L_{L(2)}|L_{L(3)}) \oplus (CON_{74}^{192}|CON_{75}^{192})$$

We perform the DoubleSwap operation on $\Sigma^4(L_L)$ and get the least significant 57 bits of $\Sigma^5(L_L)$. We put the value in the following relation (when $i = 9$),

$$(RK_{38}|RK_{39}) = \Sigma^5(L_{L(2)}|L_{L(3)}) \oplus (K_{R(2)}|K_{R(3)} \oplus (CON_{78}^{192}|CON_{79}^{192})$$

Therefore, we get the least significant 57 bits of K_R.

So, now we guess 7 most significant bits of $K_{R(2)}$ and 7 least significant bits of $K_{L(3)}$ and get the values of $(WK_2|WK_3)$. There are 2^{14} such values. For each value of $(WK_2|WK_3)$, we get the corresponding values of RK_{41}, RK_{40}, RK_{37}, RK_{36}, RK_{33}, RK_{32}, RK_{29}, and RK_{28}, from the values $RK_{41} \oplus WK_2$, $RK_{40} \oplus WK_3$, $RK_{37} \oplus WK_3$, $RK_{36} \oplus WK_2$, $RK_{33} \oplus WK_2$, $RK_{32} \oplus WK_3$, $RK_{29} \oplus WK_3$, and $RK_{28} \oplus WK_2$.

Therefore, now we can get the values $\Sigma^4(L_R)$ and $\Sigma^4(L_L)$, subsequently the values of L_R and L_L by performing the inverse DoubleSwap operation on them. For, one choice of $(WK_2|WK_3)$ we get one choice of (L_R, L_L). For each possible choice of (L_R, L_L), we do the $GFN_{(8,10)}^{-1}$, and get the value of (K_R, K_L). For each value (K_R, K_L), we

check whether the value of $(K_{R(2)}|K_{R(3)}) \oplus (K_{L(2)}|K_{L(3)}$ matches with the corresponding value of $(WK_2|WK_3)$. Only the actual value of (K_R, K_L) which corresponds to the master key will satisfy the condition. Therefore, the time complexity of the second phase of the attack is 2^{14}. The time complexity of the first phase of the attack is 2^{16} as we have to do brute-force on all the values of (p_0, p_1) or (p'_0, p'_1). Hence, the time complexity of the entire attack is 2^{16}. Thus using eight pairs of fault-free and faulty ciphertexts we uniquely determine the 192-bit master key with attack time complexity 2^{16}.

The same technique is also applicable to CLEFIA-256.

VIII. Experimental Results

We have performed extensive simulation of the proposed attacks. The attack simulation codes were written in C programming language and compiled using gcc-4.4.3 with $O3$ flag on. The code were executed in Ubuntu-10.4 operating system running on an $Intel\ Core^{TM}2\ Duo$ desktop machine of 3 GHz speed. In each experiment we used an arbitrary key and random byte-faults are induced to get the faulty ciphertexts. The experiment on a random key was repeated for 256 times. Each simulation was performed over a 100 random key-plaintext pairs.

Table I
Experimental Results on CLEFIA-128

Random 128-bit CLEFIA key	Number of Keys in First Phase	Number of Keys in Second Phase	Running Time (Seconds)
$71b344b86320d371$ $6f566c915bfaa5c2$	16162164.46 $= 2^{23.946}$	1	82.630
$7a052bfbf63a246c$ $838f09766d53aee8$	12483201.97 $= 2^{23.573}$	1	63.688
$b0e8e24e38b682e4$ $6abf4767368bcd6b$	13403507.87 $= 2^{23.676}$	1	69.489
$3702237433bd2f12$ $542f4bec01734e64$	9547074.13 $= 2^{23.186}$	1	52.711
$d5659a20b8a945a9$ $566dd7f9f0f886ae$	11671681.65 $2^{23.476}$	1	58.819

Table I shows the simulation results of five such experiments on five random keys of CLEFIA-128. 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. 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. The results show that the simulated attack on CLEFIA-128 takes around one minute of execution time to uniquely determine the secret key. Similar experiments were also performed for CLEFIA-192 and CLEFIA-256. In both the cases the simulated attack revealed the secret key within one second of execution time.

IX. Comparison

In this section we compare our attack with the existing attacks on CLEFIA. We compare with the help of Table II and Table III. In Table III, r refers to the number of rounds. The DFA proposed by Chen *et al.* [21] required 18 and 54 pairs of fault-free and faulty ciphertexts to uniquely determine the CLEFIA-128 and CLEFIA-192/CLEFIA-256 key respectively. However, the attack needs to induce the first fault at the $(r-1)^{th}$ round. This implies that if only the last two rounds of the encryption are protected then the attack will not work. The proposed improved attack in [22], [23] required only two pairs of fault-free and faulty ciphertexts for CLEFIA-128 and around ten pairs of fault-free and faulty ciphertexts for CLEFIA-192/CLEFIA-256 to recover the secret key. In each case the first fault needs to be induced in the $(r-3)^{th}$ round of encryption. Therefore, if the last four rounds are protected then the attack will fail. Further, the attacks on CLEFIA-192/CLEFIA-256 required to induced faults in three different rounds.

Compared to the existing two attacks the proposed attacks require to induce faults at the $(r-4)^{th}$ round of encryption. Therefore, even if the last four rounds are protected, our attacks can retrieve the secret key. The proposed attack on CLEFIA-128 requires only two pairs of fault-free and faulty ciphertexts and uniquely determines the key. The attack on CLEFIA-192 and CLEFIA-256 requires eight faulty ciphertexts each and the faults are induced in only two different rounds of encryption. Therefore, compared to the existing attacks on CLEFIA-192 and CLEFIA-256 the proposed attacks are more efficient in terms of both the number of required faulty ciphertexts and the controllability of the faults. Overall the proposed attacks pose a significant threat to CLEFIA.

Table II
Comparison with existing attacks on CLEFIA-128

Reference	Fault Location	Number of Faults	Exhaustive Search
[21]	15^{th},16^{th},17^{th} Round	18	1
[22], [23]	15^{th} Round	2	$2^{19.02}$
Our Attack	14^{th} Round	2	$2^{23.826}$

X. Conclusions

In this paper we proposed new differential fault attacks on CLEFIA. The results show that the proposed attack on CLEFIA-128 retrieves the secret key using two pairs of fault-free and faulty ciphertexts where the byte-faults are induced at the fourteenth round of CLEFIA encryption.

Table III
COMPARISON WITH EXISTING ATTACKS ON CLEFIA-192 AND CLEFIA-256

Reference	Fault Location (in rounds)	Number of Faults	Exhaustive Search
[21]	$(r-1)^{th}$ to $(r-8)^{th}$	54	1
[22], [23]	$(r-3)^{th}$, $(r-6)^{th}$, $(r-9)^{th}$	10	$2^{19.02}$
Our Attack	$(r-4)^{th}$ and $(r-8)^{th}$	8	2^{16}

The proposed attack on CLEFIA-192 and CLEFIA-256 can retrieve the secret key using only eight pairs of fault-free and faulty ciphertexts. The proposed attacks can evade the existing countermeasures for pipeline implementation of CLEFIA, which protect the last four rounds of encryption. The simulation results show that the attacks are indeed practical. To the best of our knowledge the proposed DFA attacks are the most efficient attacks on CLEFIA reported in the literature.

ACKNOWLEDGMENT

We thank our colleague Chester Rebeiro for sharing his insight knowledge of CLEFIA cipher.

REFERENCES

[1] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *CRYPTO*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 388–397.

[2] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract)," in *EUROCRYPT*, 1997, pp. 37–51.

[3] E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," in *CRYPTO*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Ed., vol. 1294. Springer, 1997, pp. 513–525.

[4] A. Pellegrini, V. Bertacco, and T. M. Austin, "Fault-based attack of RSA authentication," in *DATE*. IEEE, 2010, pp. 855–860.

[5] I. Biehl, B. Meyer, and V. Müller, "Differential Fault Attacks on Elliptic Curve Cryptosystems," in *CRYPTO*, ser. Lecture Notes in Computer Science, M. Bellare, Ed., vol. 1880. Springer, 2000, pp. 131–146.

[6] C. Clavier, B. Gierlichs, and I. Verbauwhede, "Fault Analysis Study of IDEA," in *CT-RSA*, ser. Lecture Notes in Computer Science, T. Malkin, Ed., vol. 4964. Springer, 2008, pp. 274–287.

[7] Christophe Giraud, "DFA on AES," in *AES Conference*, ser. Lecture Notes in Computer Science, H. Dobbertin, V. Rijmen, and A. Sowa, Eds., vol. 3373. Springer, 2004, pp. 27–41.

[8] G. Piret and J.-J. Quisquater, "A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD," in *CHES*, ser. Lecture Notes in Computer Science, C. D. Walter, Çetin Kaya Koç, and C. Paar, Eds., vol. 2779. Springer, 2003, pp. 77–88.

[9] D. Mukhopadhyay, "An Improved Fault Based Attack of the Advanced Encryption Standard," in *AFRICACRYPT*, ser. Lecture Notes in Computer Science, B. Preneel, Ed., vol. 5580. Springer, 2009, pp. 421–434.

[10] M. Tunstall, D. Mukhopadhyay, and S. S. Ali, "Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault," in *WISTP*, ser. Lecture Notes in Computer Science, C. A. Ardagna and J. Zhou, Eds., vol. 6633. Springer, 2011, pp. 224–233.

[11] C. H. Kim, "Differential Fault Analysis against AES-192 and AES-256 with Minimal Faults," in *FDTC*, L. Breveglieri, M. Joye, I. Koren, D. Naccache, and I. Verbauwhede, Eds. IEEE Computer Society, 2010, pp. 3–9.

[12] 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. IEEE, 2011, pp. 35–42.

[13] S. Ali and D. Mukhopadhyay, "Differential Fault Analysis of AES-128 Key Schedule Using a Single Multi-byte Fault," in *CARDIS*, ser. Lecture Notes in Computer Science, E. Prouff, Ed., vol. 7079. Springer, 2011, pp. 50–64.

[14] S. Ali and D. Mukhopadhyay, "An Improved Differential Fault Analysis on AES-256," in *AFRICACRYPT*, ser. Lecture Notes in Computer Science, A. Nitaj and D. Pointcheval, Eds., vol. 6737. Springer, 2011, pp. 332–347.

[15] S. P. Skorobogatov and R. J. Anderson, "Optical Fault Induction Attacks," in *CHES*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Çetin Kaya Koç, and C. Paar, Eds., vol. 2523. Springer, 2002, pp. 2–12.

[16] 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. IEEE Computer Society, 2009, pp. 84–92.

[17] N. Selmane, S. Guilley, and J.-L. Danger, "Practical Setup Time Violation Attacks on AES," in *EDCC*. IEEE Computer Society, 2008, pp. 91–96.

[18] 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.

[19] S. Corporation, "The 128-bit Blockcipher CLEFIA Algorithm Specification (Revision 1.0,Junw 1, 2007)," http://www.sony.net/Products/clefia/.

[20] CLEFIA Standardization in ISO/IEC 29192-2 [Online]. Available:, "The 128-bit Blockcipher CLEFIA Algorithm Specification," 2007, http://www.sony.net/Products/cryptography/clefia/standard/iso.html.

[21] H. Chen, W. Wu, and D. Feng, "Differential Fault Analysis on CLEFIA," in *ICICS*, ser. Lecture Notes in Computer Science, S. Qing, H. Imai, and G. Wang, Eds., vol. 4861. Springer, 2007, pp. 284–295.

[22] 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. IEEE Computer Society, 2008, pp. 25–34.

[23] J. Takahashi and T. Fukunaga, "Differential Fault Analysis on CLEFIA with 128, 192, and 256-Bit Keys," *IEICE Transactions*, vol. 93-A, no. 1, pp. 136–143, 2010.

[24] P. Derbez, P.-A. Fouque, and D. Leresteux, "Meet-in-the-Middle and Impossible Differential Fault Analysis on AES," in *CHES*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer, 2011, pp. 274–291.