Improved Impossible Differential Cryptanalysis of CLEFIA ⋆

Wei Wang[1] and Xiaoyun Wang[2⋆⋆]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China
wwang@math.sdu.edu.cn
[2] Center for Advanced Study, Tsinghua University, Beijing, China
xiaoyunwang@tsinghua.edu.cn

Abstract. This paper presents an improved impossible differential attack on the new block cipher CLEFIA which is proposed by Sony Corporation at FSE 2007. Combining some observations with new tricks, we can filter out the wrong keys more efficiently, and improve the impossible differential attack on 11-round CLEFIA-192/256, which also firstly works for CLEFIA-128. The complexity is about $2^{98.1}$ encryptions and $2^{103.1}$ chosen plaintexts. By putting more constraint conditions on plaintext pairs, we give the first attack on 12-round CLEFIA for all three key lengths with $2^{114.3}$ encryptions and $2^{119.3}$ chosen plaintexts. For CLEFIA-192/256, our attack is applicable to 13-round variant, of which the time complexity is about 2^{181}, and the data complexity is 2^{120}. We also extend our attack to 14-round CLEFIA-256, with about $2^{245.4}$ encryptions and $2^{120.4}$ chosen plaintexts. Moreover, a birthday sieve method is introduced to decrease the complexity of the core precomputation.

Key words: Block ciphers, cryptanalysis, impossible differential attack, CLEFIA

1 Introduction

CLEFIA [5] is a new 128-bit block cipher algorithm, developed by Sony Corporation. Compatible with AES, CLEFIA supports three different key lengths (128, 192 and 256 bits), which is denoted as CLEFIA-128, CLEFIA-192 and CLEFIA-256 respectively. 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.

⋆ This paper is a preprint of a paper submitted to ETRI Journal, and is subject to ETRI Journal Copyright.

⋆⋆ This research was supported by 973 Project (No.2007CB807902) and the National Natural Science Foundation of China (NSFC Grant No.90604036 and No.60525201).

Since CLEFIA was presented at Fast Software Encryption (FSE) 2007 [4], there has been little analysis on its security except the security and performance evaluations [6] published by Sony Corporation and a differential fault analysis [3]. However, because of its advantage in hardware and software implementations and wide potential applications, it's necessary to give further security evaluation. In this paper, we present the impossible differential attacks on reduced CLEFIA with more rounds.

Impossible differential cryptanalysis [1] is a sieving attack which considers a differential with probability 0. If a pair of plaintexts is encrypted or decrypted to such a difference under some trial key, we filter out this trial key from the key space. Thus, the correct key is found by eliminating all the other keys which lead to a contradiction. Reference [6] presented an impossible differential attack on 10-round CLEFIA-128/192/256, 11-round CLEFIA-192/256, and 12-round CLEFIA-256 without key whitenings using a 9-round impossible differential.

This paper improves the impossible differential attack on reduced CLEFIA. Observing the inner structure of the F-functions, we conclude that the time complexity of recovering subkeys can be decreased by some table lookups and sieving less subkey space. By these observations, our attack on 11-round CLEFIA only takes $2^{98.1}$ encryptions and $2^{103.1}$ chosen plaintexts while the result in [6] is 2^{188} encryptions and $2^{103.5}$ chosen plaintexts. We also propose another attack on 11-round variant with $2^{66.5}$ encryptions and $2^{118.5}$ chosen plaintexts. Moreover, combining the above techniques with a special way to choose plaintexts, we present the first attack on 12-round CLEFIA for all three key lengths with $2^{114.3}$ time complexity and $2^{119.3}$ data complexity. The attack can be extended to 13-round CLEFIA-192/256, and the complexity is about 2^{181} encryptions and 2^{120} chosen plaintexts. Finally, we give an attack on 14-round CLEFIA-256, which needs about $2^{245.4}$ time complexity and $2^{120.4}$ data complexity. In addition, we introduce a birthday sieve method to reduce the complexity of searching chosen plaintext pairs in the precomputation.

This paper is organized as follows: in Section 2, we give a brief description of CLEFIA. Section 3 summarizes some important observations on CLEFIA. In Section 4, we present two attacks applicable to 11-round CLEFIA with all three key variants, and extend to 12-round variant. Section 5 describes the attacks on 13-round CLEFIA-192/256 and 14-round CLEFIA-256. Finally, we conclude this paper in Section 6.

2 Description of CLEFIA

2.1 Notations

We first describe the symbols used throughout this paper.

P or P'	: the 128-bit plaintext
C or C'	: the 128-bit ciphertext
C^r	: the 128-bit output of the r-th round
C_i^r	: the i-th 32-bit word of C^r
ΔP or ΔC	: the plaintext or ciphertext difference
ΔC^r	: the XOR value of C^r and $C^{r\prime}$
F_i^r	: F_i involved in the r-th round, $i = 0,1$
ΔF_i^r	: the output XOR of F_i in the r-th round, $i = 0,1$
$InS_{F_i}^r$	: the 32-bit value after the key addition in F_i^r, i. e., the input to the S-boxes involved in F_i^r
$A >>> x$	: the rotation of A to the right by x bits positions
$A <<< x$	: the rotation of A to the left by x bits positions
$a \mid b$	: the concatenation of a and b
a^T	: the transposition of a vector a

2.2 Data Processing Part of CLEFIA

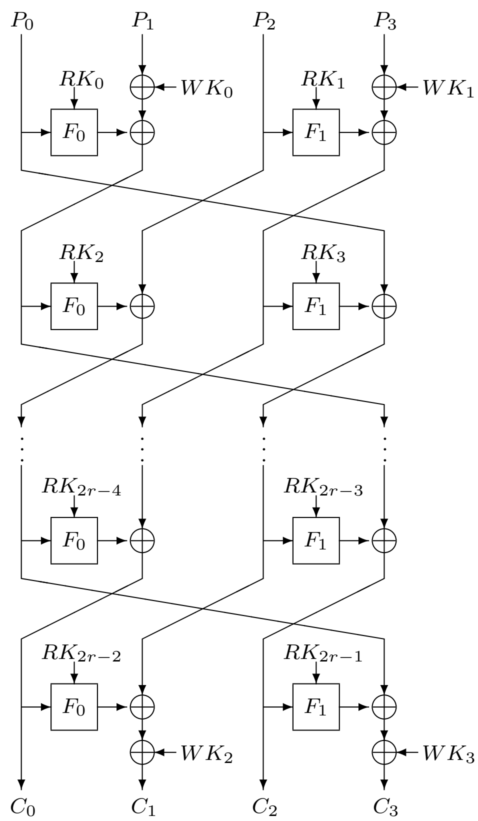

Fig. 1 Encryption Process of r-round CLEFIA

CLEFIA [5] is a 128-bit block cipher with the key length of 128, 192 and 256 bits. It employs a generalized Feistel structure with four data lines, and the width of each data line is 32 bits. Additionally, there are key whitening parts at the beginning and the end of the cipher. Figure 1 shows the encryption process of r-round CLEFIA.

Let $WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ be whitening keys and $RK_i \in \{0,1\}^{32}$ ($0 \leq i < 2r$) be round subkeys produced by the key scheduling part. For a 128-bit plaintext $P = P_0|P_1|P_2|P_3$, we compute the ciphertext $C = C_0|C_1|C_2|C_3$ as follows:

1. $C_0^0 = P_0,\ C_1^0 = P_1 \oplus WK_0,\ C_2^0 = P_2,\ C_3^0 = P_3 \oplus WK_1.$
2. For $i = 1$ to r-1,
   $C_0^i = C_1^{i-1} \oplus F_0(C_0^{i-1}, RK_{2i-2}),\ C_1^i = C_2^{i-1},$
   $C_2^i = C_3^{i-1} \oplus F_1(C_2^{i-1}, RK_{2i-1}),\ C_3^i = C_0^{i-1}.$
3. $C_0^r = C_0^{r-1},\ C_1^r = C_1^{r-1} \oplus F_0(C_0^{r-1}, RK_{2r-2}) \oplus WK_2,$
   $C_2^r = C_2^{r-1},\ C_3^r = C_3^{r-1} \oplus F_1(C_2^{r-1}, RK_{2r-1}) \oplus WK_3.$

The number of rounds r can be 18, 22 and 26 for CLEFIA-128, CLEFIA-192 and CLEFIA-256 respectively, and the two F-functions F_0 and F_1 are described in the next.

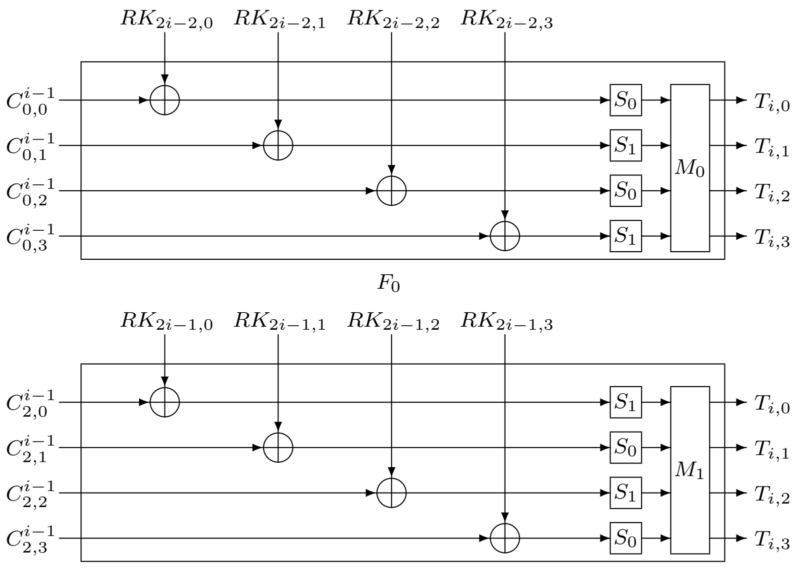

Fig. 2 F-functions

Denote the 32-bit output of F-functions as T_i, $T_i \in \{0,1\}^{32}$. Then $F_0(C_0^{i-1}, RK_{2i-2})$ ($1 \leq i \leq r$) is computed as follows:

1. $T_i = C_0^{i-1} \oplus RK_{2i-2}$.

2. Let $T_i = T_{i,0} \mid T_{i,1} \mid T_{i,2} \mid T_{i,3}$, $T_{i,j} \in \{0,1\}^8 (j = 0,1,2,3)$, then compute $T_{i,0} = S_0(T_{i,0})$, $T_{i,1} = S_1(T_{i,1})$, $T_{i,2} = S_0(T_{i,2})$, $T_{i,3} = S_1(T_{i,3})$.
3. $(T_{i,0}, T_{i,1}, T_{i,2}, T_{i,3})^T = M_0(T_{i,0}, T_{i,1}, T_{i,2}, T_{i,3})^T$.

Here, S_0 and S_1 are two nonlinear 8-bit S-boxes, and M_0 is a 4×4 Hadamard-type matrix. $F_1(C_2^{i-1}, RK_{2i-1})(1 \leq i \leq r)$ is similar to F_0 by replacing S_0 with S_1, S_1 with S_0, and M_0 with another 4×4 Hadamard-type matrix M_1. See Figure 2 for a pictorial depiction of F_0 and F_1.

We suppose that all the round subkeys and whitening keys are independent of each other, and omit the description of the key scheduling part.

3 Some Observations on CLEFIA

This section describes some important observations for analyzing CLEFIA, which lead to more efficient attacks on reduced CLEFIA variants.

Reference [6] presented two 9-round impossible differentials which resulted in the attack on 10-round CLEFIA-128/192/256 and 11-round CLEFIA-192/256. We utilize the same impossible differential, and explore more technique details to achieve a prominent improvement.

Proposition 1. *(Impossible Differentials of 9-round CLEFIA [6]) For 9-round CLEFIA, given a plaintext pair with difference (0, α, 0, 0) (or (0, 0, 0, α)), where $\alpha \in \{0,1\}^{32}$ is any non-zero value, the output difference can't be equal to (0, α, 0, 0) (or (0, 0, 0, α)). We denoted the two 9-round impossible differentials as*

$$(0, \alpha, 0, 0) \nrightarrow (0, \alpha, 0, 0) \text{ and } (0, 0, 0, \alpha) \nrightarrow (0, 0, 0, \alpha).$$

The correctness of Proposition 1 can be verified easily.

By observing the inner structure of F-functions, we find out that the time complexity of attacks in [6] can be can be decreased by fast searching the 32-bit key in F-function with the help of XOR distribution tables of S-boxes [2].

Proposition 2. *For the F-function F (F_0 or F_1), let (In, In') be two 32-bit inputs, and ΔOut be the difference of the corresponding output, the 32-bit round subkey RK involved in F can be deduced with about one F-computation.*

Proof: Because the diffusion matrix M is linear and invertible, we can easily compute the input difference ΔOut^{-1} of M, i. e.,

$$\Delta Out^{-1} = M^{-1}(\Delta Out) = \Delta Out_0^{-1} \mid \Delta Out_1^{-1} \mid \Delta Out_2^{-1} \mid \Delta Out_3^{-1},$$

where ΔOut_0^{-1}, ΔOut_1^{-1}, ΔOut_2^{-1} and ΔOut_3^{-1} are four 8-bit output XOR of the four S-boxes in F respectively.

Therefore, for each S-box in F, we get the input XOR and the corresponding output XOR. It is easy to obtain the four inputs to the four S-boxes by searching the XOR distribution tables of S-boxes. Denote the four inputs as $InS_0, InS_1, InS_2, InS_3$ respectively. Then we get the 32-bit value

$$InS_0 \mid InS_1 \mid InS_2 \mid InS_3 = InS.$$

Thus the 32-bit round subkey RK can be deduced from the equation

$$RK = InS \oplus In.$$

Clearly the time complexity is about one F-computation. $\square$

Usually, the efficiency of the impossible differential attack depends on the subkey space related to the impossible differential. For 11-round CLEFIA-192/256, 9-round impossible differential can be used to sieve 128-bit subkey involved in rounds 10 and 11 [6]. The following proposition is an important phenomenon that can be used to sieve only 96-bit subkey instead of 128-bit subkey.

Proposition 3. *For r-round CLEFIA, let (RK_{2r-3}, RK_{2r-4}) be the subkey in the (r-1)th round, (RK_{2r-1}, RK_{2r-2}) be the subkey key in the r-th round, (WK_2, WK_3) be the whitening key in the final round, and $C^r = (C_0^r, C_1^r, C_2^r, C_3^r)$ be the ciphertext, we have the following two equations which reveal the correlations among subkeys WK_2, WK_3, RK_{2r-3} and RK_{2r-4}:*

$$WK_3 \oplus RK_{2r-4} = InS_{F_0}^{r-1} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus C_3^r, \tag{1}$$

$$WK_2 \oplus RK_{2r-3} = InS_{F_1}^{r-1} \oplus F_0^r(C_0^r, RK_{2r-2}) \oplus C_1^r. \tag{2}$$

Here $InS_{F_0}^{r-1}$ and $InS_{F_1}^{r-1}$ are the inputs to the four S-boxes of F_0^{r-1} and F_1^{r-1} in the (r-1)-th round respectively.

Proof: From the encryption algorithm, we obtain that

$$C_3^r = C_3^{r-1} \oplus F_1^r(C_2^{r-1}, RK_{2r-1}) \oplus WK_3, \text{where } C_2^{r-1} = C_2^r.$$

Then it is clear that

$$C_3^r = C_3^{r-1} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus WK_3.$$

Since

$$C_3^{r-1} = C_0^{r-2} \text{ and } InS_{F_0}^{r-1} = C_0^{r-2} \oplus RK_{2r-4},$$

we know that

$$\begin{aligned} C_3^r &= C_0^{r-2} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus WK_3 \\ &= InS_{F_0}^{r-1} \oplus RK_{2r-4} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus WK_3, \end{aligned}$$

i. e.,

$$WK_3 \oplus RK_{2r-4} = InS_{F_0}^{r-1} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus C_3^r.$$

Similarly, we can prove that Equation (2) holds. $\square$

Furthermore, for the 1st and 2nd rounds, there are two other similar equations about $WK_0 \oplus RK_2$ and $WK_1 \oplus RK_3$:

$$WK_0 \oplus RK_2 = InS_{F_0}^2 \oplus F_0^1(P_0, RK_0) \oplus P_1, \tag{3}$$

$$WK_1 \oplus RK_3 = InS_{F_1}^2 \oplus F_1^1(P_2, RK_1) \oplus P_3. \tag{4}$$

4 Attacks on CLEFIA-128/192/256

In this section, we present two improved impossible differential attacks on 11-round CLEFIA, and extend the attack to 12-round variant. The improved attack works for CLEFIA-128/192/256. Especially for CLEFIA-128, this is the first known attack. The main attack process is as follows. Firstly, select many structures of chosen plaintexts, and sieve the pairs satisfying the required output differences. Secondly, for each sieved pair, discard the wrong subkeys which cause the partial encryption and decryption to match the impossible differential. Finally, analyze enough pairs, and sieve the correct subkey.

4.1 The Improved Attack on 11-round CLEFIA

This section describes the improved key recovery attack on 11-round CLEFIA with two additional rounds at the end of the 9-round impossible differential. We use the same 9-round impossible differential $(0,\alpha,0,0) \nrightarrow (0,\alpha,0,0)$ throughout this paper. The other 9-round impossible differential $(0,0,0,\alpha) \nrightarrow (0,0,0,\alpha)$ can be used in a similar way. Different from [6], the attack recovers the 96-bit subkey $(RK_{20}, RK_{21},\ RK_{18} \oplus WK_3)$ by Proposition 3 instead of recovering the 128-bit subkey $(RK_{18},\ RK_{20}, RK_{21}, WK_3)$. Combining with Proposition 2, the total complexity can be improved from the original 2^{188} encryptions to $2^{98.1}$ encryptions with $2^{103.1}$ chosen plaintexts. See Figure 3 for the following attack.

Sieving pairs

A structure composed of 2^{32} plaintexts is defined as follows:

$$Struc = \{P_0, P_1 \oplus \alpha, P_2, P_3 \mid P_0, P_1, P_2, P_3 \text{ are fixed, } \alpha \in \{0,1\}^{32} \text{ is non-zero}\}.$$

By the encryption process of CLEFIA, only the plaintext pair with ciphertext difference $\Delta C = (\beta,\gamma,0,\alpha)$ may result from $\Delta C^9 = (\alpha,0,0,0)$, where $\beta \in \{0,1\}^{32}$ and $\gamma \in \{0,1\}^{32}$ are non-zero. It is clear that every two structures can produce about one pair with the target ciphertext difference. In our attack, about$2^{70.1}$ such plaintext pairs are necessary to sieve the right key. So, we choose $2^{71.1}$ such structures.

Because there are $2^{134.1}$ plaintext pairs from $2^{71.1}$ structures totally, we need to explore a fast algorithm to obtain the $2^{70.1}$ pairs. We employ a type of ***birthday sieve*** to search these pairs more efficiently.

Birthday Sieve Algorithm 1:
For each structure, we fulfill the following steps:

1. For each plaintext P, compute $\widetilde{C} = (P >>> 64) \oplus C$, where C is the corresponding ciphertext.
2. Store the 2^{32} values of $\widetilde{C}$ in a table.
3. Search (P, P') with the corresponding $\Delta\widetilde{C} = (\beta,\gamma,0,0)$ by the birthday attack.

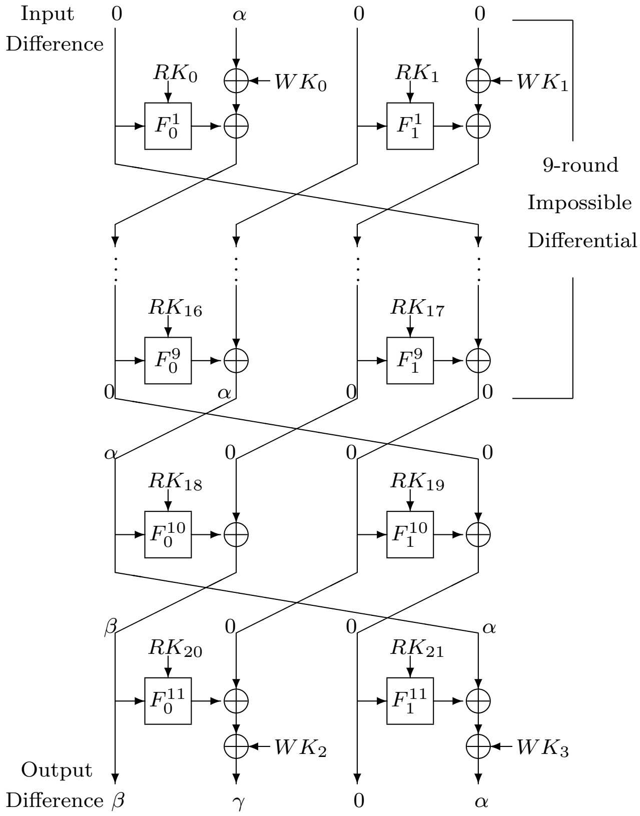

Fig. 3 Impossible Differential Attack on 11-round CLEFIA (1)

4. Output (P, P').

It is clear that $\Delta C = (\beta, \gamma, 0, \alpha)$ if and only if $\Delta\widetilde{C} = (\beta, \gamma, 0, 0)$. So, the above algorithm outputs one plaintext pair corresponding to $\Delta C = (\beta, \gamma, 0, \alpha)$ with probability 1/2. From the birthday attack [7], the time complexity is only 2^{32} XOR computations, and the table memory is about 2^{34} words. Thus, we can obtain $2^{70.1}$ pairs with about $2^{103.1}$ XOR computations by neglecting the table lookups.

Recovering the subkey $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$

We discard the subkeys which cause the partial decryption of the selected pair to match $\Delta C^9 = (\alpha, 0, 0, 0)$.

For each pair with ciphertext difference

$$\Delta C = (\beta, \gamma, 0, \alpha),$$

it is obvious that

$$\Delta C_0^9 = \Delta C_3^{10} = \Delta C_3^{11} = \alpha.$$

From

$$C_3^9 = C_2^{10} \oplus F_1^{10}(C_2^9, RK_{19}), C_2^{10} = C_2^{11} \text{ and } \Delta C_2^{11} = 0,$$

it is clear that

$$\Delta C_3^9 = 0 \text{ if and only if } \Delta C_2^9 = 0.$$

Thus, we only need to discard the subkeys which lead to

$$\Delta C_1^9 = 0 \text{ and } \Delta C_2^9 = 0.$$

For each ciphertext pair (C, C') with $\Delta C = (\beta, \gamma, 0, \alpha)$, we can prove that there are 2^{32} wrong subkeys $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$ which suggest the impossible differential.

1. For $\Delta C_2^9 = 0$, from $C_1^{10} = C_2^9$, it is clear that

   $$\Delta C_2^9 = 0 \text{ if and only if } \Delta C_1^{10} = 0.$$

   Since

   $$C_1^{11} = C_1^{10} \oplus F_0^{11}(C_0^{10}, RK_{20}) \oplus WK_2,$$

   we obtain that

   $$\Delta F_0^{11} = \Delta C_1^{11} \text{ when } \Delta C_1^{10} = 0.$$

   The two corresponding input to F_0^{11} are

   $$C_0^{10} = C_0^{11} \text{ and } C_0^{10\prime} = C_0^{11\prime},$$

   so the subkey RK_{20} can be calculated with about one F-computation by Proposition 2.
2. For $\Delta C_1^9 = 0$, we have

   $$\Delta F_0^{10} = \Delta C_0^{11}$$

   by

   $$C_0^{10} = C_1^9 \oplus F_0^{10}(C_0^9, RK_{18}) \text{ and } C_0^{10} = C_0^{11}.$$

   Because the corresponding input XOR

   $$\Delta C_0^9 = \alpha,$$

   $InS_{F_0}^{10}$ is calculated by Proposition 2.
   For each $RK_{21} \in \{0, 1\}^{32}$, according to Proposition 3, we deduce that

   $$RK_{18} \oplus WK_3 = InS_{F_0}^{10} \oplus F_1^{11}(C_2^{11}, RK_{21}) \oplus C_3^{11}.$$

   So, we totally obtain 2^{32} wrong values of $(RK_{21}, RK_{18} \oplus WK_3)$ with about 2^{32} F-computations.

Summing up 1) and 2), we can filter out 2^{32} wrong subkeys $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$ which support the impossible differential. Thus, for each pair, a wrong subkey $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$ survives with probability $1 - 2^{-64}$. After analyzing $2^{70.1}$ pairs, the number of the remaining subkeys is

$$2^{96} \cdot (1 - 2^{-64})^{2^{70.1}} \approx 0.13 < 1.$$

That is to say, only the right subkey $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$ is left. This completes our attack.

Complexity evaluation

The data complexity for the attack is about $2^{70.1+32+1} = 2^{103.1}$ chosen plaintexts. The time complexity of recovering the 96-bit subkey is about $2^{70.1} \cdot 2^{32} = 2^{102.1}$ F-computations. Using rough equivalence of 2^4 F-computations to one encryption, the $2^{102.1}$ F-computations are equivalent to about $2^{98.1}$ encryptions.

4.2 Another Improved Attack on 11-round CLEFIA

This subsection describes another key recovery attack on 11-round CLEFIA, with one additional round on top of the 9-round impossible differential and one at the end. The complexity of the attack is only about $2^{66.5}$ encryptions and $2^{118.5}$ chosen plaintexts. This attack is about 2^{32} times faster than the first attack because we only need to recover the 64-bit subkey (RK_0, RK_{21}) instead of the 96-bit subkey $(RK_{20}, RK_{21}, RK_{18} \oplus WK_3)$. Although this attack needs more chosen plaintexts, it is necessary to 12-round attack in the next section. See Figure 4 for a pictorial depiction of the following attack.

Sieving pairs

To guarantee the impossible differential hold, we need to select the plaintext pairs with $\Delta P = (0, 0, \alpha, \delta)$ and $\Delta C = (\alpha, \beta, 0, 0)$, where $\alpha \in \{0,1\}^{32}$, $\delta \in \{0,1\}^{32}$ and $\beta \in \{0,1\}^{32}$ are non-zero. Choose a structure of 2^{48} plaintexts, where the first and second 32-bit words are fixed, the third word ranges over all 2^{32} possibilities, and the fourth word takes 2^{16} distinct random values. Similar to Section 4.1, we select pairs with $\Delta C = (\alpha, \beta, 0, 0)$ in the following way.

Birthday Sieve Algorithm 2:

1. For each plaintext P, compute $\widetilde{C} = (P <<< 64) \oplus C$, where C is the corresponding ciphertext.
2. Store the 2^{48} values of $\widetilde{C}$ in a table.
3. Search (P, P') with the corresponding $\Delta\widetilde{C} = (0, \gamma, 0, 0)$ by the birthday attack.
4. Output (P, P').

It is clear that $\Delta C = (\alpha, \beta, 0, 0)$ if and only if $\Delta\widetilde{C} = (0, \gamma, 0, 0)$. The table memory is about 2^{50} words. We collect $2^{69.5}$ such pairs from $2^{70.5}$ structures.

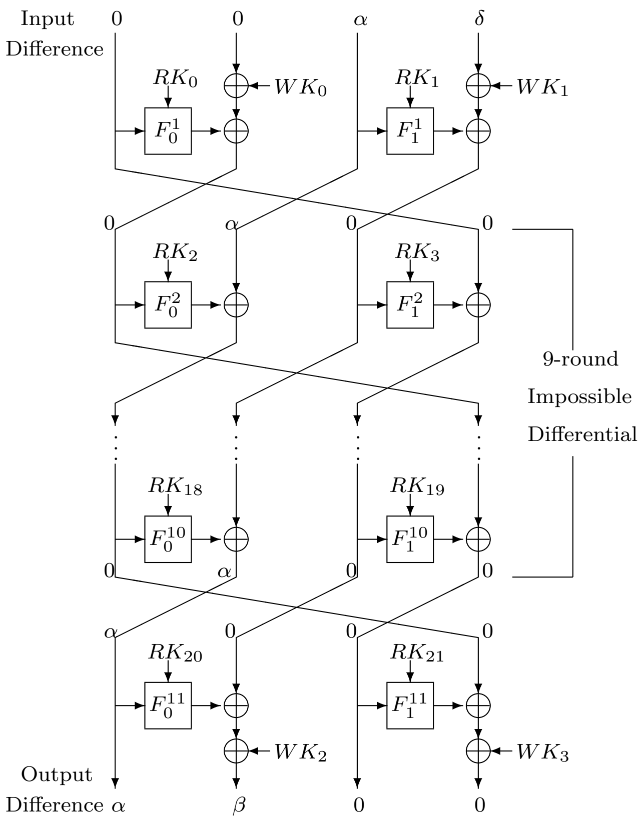

Fig. 4 Impossible Differential Attack on 11-round CLEFIA (2)

Recovering the subkey (RK_1, RK_{20})

For each selected pair (P, P'), the wrong subkeys (RK_1, RK_{20}) resulting in the impossible differential are computed as follows.

1. Compute the subkey RK_1 which produces the partial encryption of the pair to match $\Delta C^1 = (0, \alpha, 0, 0)$.
   From
   $$C_0^1 = P_1 \oplus F_0^1(P_0, RK_0) \oplus WK_0 \text{ and } \Delta P = (0, 0, \alpha, \delta),$$
   the selected pair already satisfies
   $$\Delta C_0^1 = \Delta P_1 = 0, \Delta C_1^1 = \Delta P_2 = \alpha \text{ and } \Delta C_3^1 = \Delta P_0 = 0.$$
   So we only compute subkeys that cause
   $$\Delta C_2^1 = 0.$$
   By
   $$C_2^1 = F_1^1(P_2, RK_1) \oplus P_3 \oplus WK_1,$$

it is clear that

$$F_1^1(P_2, RK_1) = C_2^1 \oplus P_3 \oplus WK_1.$$

Thus, if $\Delta C_2^1 = 0$, then $\Delta F_1^1 = \Delta P_3$ holds.
As the two inputs of F_1^1 are P_2 and P_2', one 32-bit RK_1 can be computed with one F-computation on average by Proposition 2.

2. Compute the subkey RK_{20} which causes the partial decryption of the pair to match $\Delta C^{10} = (\alpha, 0, 0, 0)$.
According to

$$C_3^{10} = C_3^{11} \oplus WK_3 \oplus F_1^{11}(C_2^{11}, RK_{21}) \text{ and } \Delta C = (\alpha, \beta, 0, 0),$$

we derive that

$$\Delta C_0^{10} = \Delta C_0^{11} = \alpha, \Delta C_2^{10} = \Delta C_2^{11} = 0 \text{ and } \Delta C_3^{10} = \Delta C_3^{11} = 0.$$

Hence, it is sufficient to guarantee

$$\Delta C_1^{10} = 0.$$

From

$$C_1^{11} = F_0^{11}(C_0^{10}, RK_{20}) \oplus C_1^{10} \oplus WK_2,$$

it is obvious that

$$\Delta F_0^{11} = \Delta C_1^{11} \text{ when } \Delta C_1^{10} = 0.$$

By

$$C_0^{10} = C_0^{11} \text{ and } C_0^{10\prime} = C_0^{11\prime},$$

RK_{20} can be derived with one F-computation by Proposition 2.

Till now, for each pair, we find one 64-bit wrong subkey (RK_1, RK_{20}) with two F-computations, and discard it from the subkey space. After analyzing $2^{69.5}$ pairs, the number of subkey left in the 2^{64} subkey space is about

$$2^{64} \cdot (1 - 2^{-64})^{2^{69.5}} \approx 2^{64} \cdot e^{-2^{5.5}} \approx 0.41 < 1.$$

So, only the right subkey (RK_1, RK_{20}) is left.

Complexity evaluation

The number of chosen plaintexts is $2^{70.5} \cdot 2^{48} = 2^{118.5}$, and the time complexity is about $2^{70.5}$ F-computations which is equivalent to about $2^{66.5}$ encryptions.

4.3 Attack on 12-round CLEFIA

We try to extend the attack on 11-round variant to 12-round, with one additional round on top of the 9-round impossible differential and two rounds at the end. Basically, the attack is the combination of the two attacks presented above. However, the direct combination will a little exceed the complexity of the exhaustive attack. We put more constraint conditions on the plaintext difference to enforce the first two bytes of ΔC^1 to be zero. So, only 112-bit subkey $(RK_{1,2}, RK_{1,3}, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$ can guarantee the impossible differential, instead of the original 128-bit subkey $(RK_1, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$.

Sieving pairs

For all the 2^{16} possible α, of which the first two bytes are zero, we compute a table TB_1 to store the 2^{16} values of $M_1(\alpha)$. Because M_1 is linear, for $\delta_1, \delta_2 \in TB_1$, it is obvious that $\delta_1 \oplus \delta_2 \in TB_1$.

Choose a structure of 2^{32} plaintexts as follows:
$Struc = \{P_0, P_1, P_2 \oplus \alpha, P_3 \oplus \delta \mid P_0, P_1, P_2, P_3$ are fixed, the first two bytes of α are zero and the other two take 2^{16} possibilities, $\delta \in TB_1\}$.

Fulfilling Algorithm 1 in Section 4.1, in which $\widetilde{C}$ is selected as $(P >>> 32) \oplus C$, we can easily search a pair with $\Delta C = (\beta, \gamma, 0, \alpha)$, where $\beta \in \{0,1\}^{32}$ and $\gamma \in \{0,1\}^{32}$ are non-zero. $2^{-1}n$ such pairs can be found by searching n structures, and n is determined later.

Recovering the subkey $(RK_{1,2}, RK_{1,3}, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$

For each selected pair, because $M_1^{-1}(\delta)$ and the first two bytes of α are zero, we know that the input XOR and output XOR of the first two S-boxes involved in F_1^1 are zero. So only the last 16-bit $(RK_{1,2}, RK_{1,3})$ of RK_1 affects ΔC_2^1. Thus, we only discard 112-bit wrong subkeys $(RK_{1,2}, RK_{1,3}, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$ involved in the impossible differential.

To ensure the impossible differential occur, we need

$$\Delta C^1 = (0, \alpha, 0, 0) \text{ and } \Delta C^{10} = (\alpha, 0, 0, 0).$$

1. For $\Delta C^1 = (0, \alpha, 0, 0)$, the situation is the same as step 1) of Section 4.2. Therefore, we can compute one $(RK_{1,2}, RK_{1,3})$ in one F-computation.
2. For $\Delta C^{10} = (\alpha, 0, 0, 0)$, the output differences of the last two rounds are the same with those shown in Figure 3, and we can deduce 2^{32} wrong subkeys $(RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$ by the same method as Section 4.1. This step takes about 2^{32} F-computations.

For each collected pair, we can filter out 2^{32} wrong 112-bit subkeys $(RK_{1,2}, RK_{1,3}, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$ in about 2^{32} F-computations.

In order to satisfy

$$2^{112} \cdot (1 - \frac{2^{32}}{2^{112}})^{2^{-1}n} < 1,$$

the expected n is about

$$2 \cdot 2^{80} \cdot 112 \cdot \ln 2 \approx 2^{87.3}.$$

Therefore, after analyzing $2^{86.3}$ pairs, the right $(RK_{1,2}, RK_{1,3}, RK_{22}, RK_{23}, RK_{20} \oplus WK_3)$ is left.

Complexity evaluation

The data complexity is about $2^{32} \cdot n = 2^{119.3}$. The time complexity is about $2^{86.3} \cdot 2^{32} = 2^{118.3}$ F-computations, which equals to $2^{114.3}$ encryptions.

5 Attacks on 13-round CLEFIA-192/256 and 14-round CLEFIA-256

5.1 Attack on 13-round CLEFIA-192/256

This section extends our attack to 13-round CLEFIA-192/256, by adding one more round on top of the 12-round attack (See Figure 5). The main purpose is to sieve the related subkey $(RK_1, RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{25}, RK_{22} \oplus WK_3)$ by the similar techniques in Section 4.1 and Section 4.3.

Sieving pairs

Similar to Section 4.3, for all the 2^{16} possible δ, of which the first two bytes are zero, construct a table TB_0 to store the 2^{16} values of $M_0(\delta)$.

A structure is a set of 2^{64} plaintexts defined as follows:
$Struc = \{P_0 \oplus \delta, P_1 \oplus \epsilon, P_2, P_3 \oplus \alpha \mid P_0, P_1, P_2, P_3$ are fixed, the first two bytes of δ are zero and the other two randomly take 2^{16} cases, $\epsilon \in TB_0$, $\alpha \in \{0,1\}^{32}$ is non-zero$\}$.

We select the pairs with $\Delta C = (\beta, \gamma, 0, \alpha)$, where $\beta \in \{0,1\}^{32}$ and $\gamma \in \{0,1\}^{32}$ are non-zero. Select $2^{63}n$ such pairs from n structures.

Recovering the subkey $(RK_1, RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{25}, RK_{22} \oplus WK_3)$

As described in Section 4.3, because $M_0^{-1}(\epsilon)$ and the first two bytes of δ are zero, only $(RK_{0,2}, RK_{0,3})$ is related to the condition $\Delta C_0^1 = 0$.

For each of the $2^{63}n$ remaining pairs, we can filter out 2^{63} wrong 176-bit subkeys $(RK_1, RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{25}, RK_{22} \oplus WK_3)$ by the following steps.

1. Guess $RK_1 \in \{0,1\}^{32}$ and $RK_{25} \in \{0,1\}^{32}$.
2. For the guessed RK_{25}, we compute one wrong $(RK_{24}, RK_{22} \oplus WK_3)$ with the techniques in Section 4.1. This step takes about one F-computation.
3. For the guessed RK_1, we focus on the related subkey $(RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1)$ of the first two rounds.
   It is easy to prove that
   $$\Delta C^2 = (0, \alpha, 0, 0) \text{ if and only if } \Delta C_2^2 = 0 \text{ and } \Delta C_3^2 = 0.$$
   (a) For $\Delta C_3^2 = 0$, according to
   $$C_0^1 = C_3^2 \text{ and } C_0^1 = F_0^1(P_0, RK_0) \oplus P_1 \oplus WK_0,$$
   we can derive the 16-bit key $(RK_{0,2}, RK_{0,3})$ with one F-computation by Proposition 2.
   (b) For $\Delta C_2^2 = 0$, from
   $$C_2^2 = F_1^2(C_2^1, RK_3) \oplus C_3^1,$$

we get

$$\Delta F_1^2 = \Delta C_3^1.$$

By

$$C_2^1 = F_1^1(P_2, RK_1) \oplus P_3 \oplus WK_1 \text{ and } \Delta P_2 = 0,$$

we have

$$\Delta C_2^1 = \Delta P_3.$$

So we search $InS_{F_1}^2$ involved in F_1^2 by Proposition 2.
According to Proposition 3, for the guessed RK_1, we can derive $(RK_3 \oplus WK_1)$ in one F-computation, such that

$$RK_3 \oplus WK_1 = InS_{F_1}^2 \oplus F_1^1(P_2, RK_1) \oplus P_3$$

From a) and b), one wrong $(RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1)$ is computed.

Summing up 1)-3), we filter out one wrong subkey $(RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{22} \oplus WK_3)$ for each guessed (RK_1, RK_{25}). Totally, for each pair, we capture 2^{64} 176-bit wrong subkeys $(RK_1, RK_{25}, RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{22} \oplus WK_3)$ in 2^{66} F-computations, and delete them from the subkey space.

From

$$2^{176} \cdot (1 - 2^{-112})^{2^{63} n} < 1,$$

we know that n is at least 2^{56}.

Complexity evaluation

Clearly, the number of chosen plaintexts is about $2^{56} \cdot 2^{64} = 2^{120}$. The time complexity is $2^{63} \cdot 2^{56} \cdot 2^{66} = 2^{185}$ F-computations, which is about 2^{181} encryptions.

Remark 1. We use a table to keep the list of discarded keys where the entries are initialized to 0, and are set to 1 when the corresponding keys are discarded. As described in [1], for each chosen (RK_1, RK_{25}), we only need to save the 112-bit subkey $(RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1, RK_{24}, RK_{22} \oplus WK_3)$ to sieve the right subkey, so the required memory is 2^{112} bits.

5.2 Attack on 14-round CLEFIA-256

Furthermore, our attack can be applicable to 14-round CLEFIA-256, with three rounds at the end of the 9-round impossible differential and two additional rounds on the top.

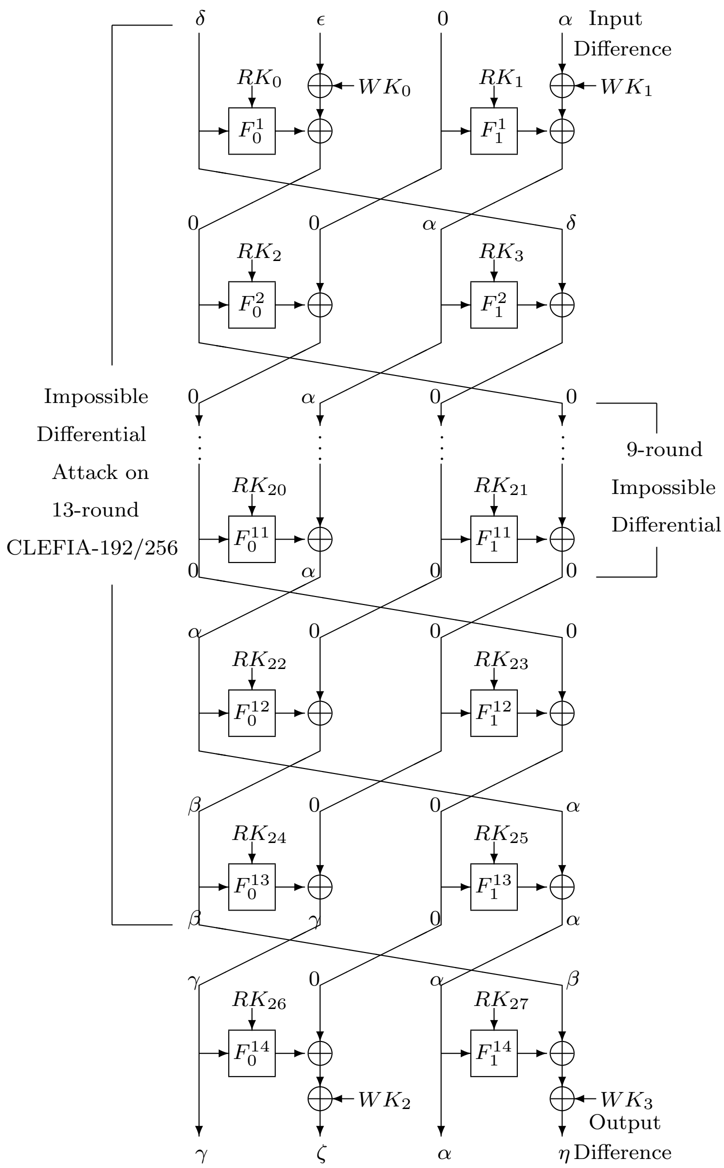

Fig. 5 Impossible Differential Attack on 14-round CLEFIA-256

In our attack, the related subkey $(RK_{0,2}, RK_{0,3}, RK_1, RK_3 \oplus WK_1)$ can be directly found by the same method in Section 5.1. Because there exist more subkeys $(RK_{26}, RK_{22}, RK_{25} \oplus WK_2, RK_{24} \oplus WK_3, RK_{27})$ in the last three rounds which are related to the impossible differential, we will give more computational details about searching the right subkey. The attack on 14-round CLEFIA-256 is illustrated in Figure 5.

Sieving pairs

A structure is composed of 2^{64} plaintexts as described in Section 5.1, $Struc = \{P_0 \oplus \delta, P_1 \oplus \epsilon, P_2, P_3 \oplus \alpha \mid P_0, P_1, P_2, P_3$ are fixed, the first two bytes of δ are zero and the other two randomly take 2^{16} cases, $\epsilon \in TB_0, \alpha \in \{0,1\}^{32}$ is non-zero$\}$

Choose the pairs with $\Delta C = (\gamma, \zeta, \alpha, \eta)$, where $\gamma, \zeta, \eta \in \{0,1\}^{32}$ are non-zero. We find 2^{95} pairs with such ΔC from each structure, and select $2^{95}n$ pairs from n structures.

Recovering the subkey $(RK_{0,2}, RK_{0,3}, RK_1, RK_3 \oplus WK_1, RK_{22}, RK_{24} \oplus WK_3, RK_{25} \oplus WK_2, RK_{26}, RK_{27})$

1. Guess each $RK_1 \in \{0,1\}^{32}$ and $RK_{27} \in \{0,1\}^{32}$.
2. For the guessed RK_1, we deduce one wrong $(RK_{0,2}, RK_{0,3}, RK_3 \oplus WK_1)$ by about two F-computations.
3. For the guessed RK_{27}, we intend to derive the related subkey (RK_{26}, RK_{22}, $RK_{25} \oplus WK_2, RK_{24} \oplus WK_3$) which leads to $\Delta C^{11} = (\alpha, 0, 0, 0)$.
   To match $\Delta C^{11} = (\alpha, 0, 0, 0)$, it is equivalent to satisfy the following three conditions:
   $$\Delta C_1^{11} = 0, \Delta C_1^{12} = 0 \text{ and } \Delta C_1^{13} = 0.$$
   (a) For $\Delta C_1^{13} = 0$, from
   $$C_0^{13} = C_0^{14} \text{ and } C_1^{14} = F_0^{14}(C_0^{13}, RK_{26}) \oplus C_1^{13} \oplus WK_2,$$
   the wrong RK_{26} is derived with one F-computation by Proposition 2.
   (b) For $\Delta C_1^{11} = 0$, we know that
   $$\Delta F_0^{12} = \Delta C_0^{12} = \Delta C_3^{13}$$
   by
   $$C_0^{12} = C_3^{13} \text{ and } C_1^{11} = C_0^{12} \oplus F_0^{12}(C_0^{11}, RK_{22}).$$
   In order to deduce RK_{22}, we have to calculate C_0^{11} and ΔC_3^{13}. It is clear that ΔC_3^{13} can be easily obtained by the final round operation:
   $$C_3^{13} = C_3^{14} \oplus F_1^{14}(C_2^{13}, RK_{27}) \oplus WK_3, \text{ where } C_2^{13} = C_2^{14}.$$
   The following step is to compute C_0^{11}.
   Since
   $$C_0^{11} = C_3^{12}, C_2^{13} = C_3^{12} \oplus F_1^{13}(C_2^{12}, RK_{25}) \text{ and } C_2^{13} = C_2^{14},$$
   we have
   $$C_0^{11} = C_2^{14} \oplus F_1^{13}(C_2^{12}, RK_{25}).$$
   According to the structure of F_1^{13}, it is easy to know
   $$C_2^{12} \oplus RK_{25} = InS_{F_1}^{13} = InS_{F_1,0}^{13} | InS_{F_1,1}^{13} | InS_{F_1,2}^{13} | InS_{F_1,3}^{13}.$$

For each guess of $RK_{25} \oplus WK_2$, we compute

$$\begin{aligned} InS_{F_1}^{13} &= C_2^{12} \oplus RK_{25} \\ &= C_1^{13} \oplus RK_{25} \\ &= C_1^{14} \oplus F_0^{14}(C_0^{14}, RK_{26}) \oplus WK_2 \oplus RK_{25}. \end{aligned}$$

So, $F_1^{13}(C_2^{12}, RK_{25})$ can be computed by

$$M_1[S_1(InS_{F_1,0}^{13}), S_0(InS_{F_1,1}^{13}), S_1(InS_{F_1,2}^{13}), S_0(InS_{F_1,3}^{13})]^T.$$

Thus, we get the value of C_0^{11}.
Combining C_0^{11} with ΔC_3^{13}, RK_{22} can be computed by Proposition 2. Totally, we obtain 2^{32} values of $(RK_{22}, RK_{25} \oplus WK_2)$ in this step, taking about 2^{34} F-computations.

(c) For $\Delta C_1^{12} = 0$, we have

$$\Delta F_0^{13} = \Delta C_0^{13} = \Delta C_0^{14} = \gamma$$

by

$$C_0^{13} = C_1^{12} \oplus F_0^{13}(C_0^{12}, RK_{24}).$$

From

$$\Delta C_0^{12} = \Delta C_3^{13}$$

which is obtained in b), we can deduce $InS_{F_0}^{13}$.
According to Proposition 3, $RK_{24} \oplus WK_3$ can be computed by the following equation

$$RK_{24} \oplus WK_3 = C_3^{14} \oplus F_1^{14}(C_2^{14}, RK_{27}) \oplus InS_{F_0}^{13}.$$

Thus, we calculate 2^{32} values of $(RK_{26}, RK_{22}, RK_{25} \oplus WK_2, RK_{24} \oplus WK_3)$ for each RK_{27} in about 2^{34} F-computations.

So far, we can discard $2^{32} \cdot 2^{64} = 2^{96}$ wrong 240-bit subkeys $(RK_{0,2}, RK_{0,3}, RK_1, RK_3 \oplus WK_1, RK_{26}, RK_{22}, RK_{25} \oplus WK_2, RK_{24} \oplus WK_3, RK_{27})$ with about 2^{98} F-computations. After analyzing $2^{95}n$ pairs, the number of wrong subkeys left is

$$2^{240} \cdot (1 - \frac{2^{96}}{2^{240}})^{2^{95}n} < 1,$$

where n is about $2^{56.4}$.

Complexity evaluation

The number of chosen plaintexts is about $2^{56.4} \cdot 2^{64} = 2^{120.4}$. The time complexity is about $2^{151.4} \cdot 2^{98} = 2^{249.4}$ F-computations, which equals to $2^{245.4}$ encryptions.

As described in Section 5.1, for each RK_1 and RK_{27}, we save only the 176-bit subkeys, so the required memory is about 2^{176} bits.

6 Conclusions

In this paper, we present a chosen-plaintext attack on reduced CLEFIA variants. Table 1 shows the comparison between the attack in [3] and our attack. Reference [3] only cryptanalyzes 10-round CLEFIA-128/192/256, 11-round CLEFIA-192/256 with key whitenings, and 12-round CLEFIA-256 without key whitenings. In our attack, we explore some observations and some tricks to break 11-12 rounds CLEFIA-128/192/256. The attack can be applied to 13-round CLEFIA-192/256 and 14-round CLEFIA-256. It is deserved to notice that all our attacks are applicable to the reduced CLEFIA with key whitenings.

Table 1. Summary of Impossible Differential Attacks on Reduced CLEFIA

Round Num.	Ref. [3]			This paper		
	Key Length	Data	Time	Key Length	Data	Time
10	128/192/256	$2^{101.7}$	2^{102}	-		
11	192/256	$2^{103.5}$	2^{188}	128/192/256	$2^{103.1}$	$2^{98.1}$
12	256^a	$2^{103.8}$	2^{252}	128/192/256	$2^{119.3}$	$2^{114.3}$
13	-			192/256	2^{120}	2^{181}
14	-			256	$2^{120.4}$	$2^{245.4}$

[a] without key whitenings

References

1. E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. *Eurocrypt* 1999, *LNCS* 1592, pp. 12-23, 1999. Springer-Verlag.
2. E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *Journal of Cryptology*, vol. 4, no. 1, pp. 3-72, 1991. Springer-Verlag.
3. H. Chen, W. L. Wu, and D. G. Feng. Differential Fault Analysis on CLEFIA. *ICICS* 2007, *LNCS* 4861, pp. 284-295, 2007. Springer-Verlag.
4. T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata. The 128-bit Blockcipher CLEFIA. *FSE* 2007, *LNCS* 4593, pp.181-195, 2007. Springer-Verlag.
5. Sony Corporation. The 128-bit Blockcipher CLEFIA: Algorithm Specification. Revision 1.0. June 1, 2007.
6. Sony Corporation. The 128-bit Blockcipher CLEFIA: Security and Performance Evaluations. Revision 1.0. June 1, 2007.
7. G. Yuval. How to Swindle Rabin. *Cryptologia*, vol. 3, pp. 187-189, 1979.