Impossible Differential-Linear Cryptanalysis of Reduced-Round CLEFIA-128 ⋆

Zheng Yuan[1,2,⋆⋆], Xian Li[1,2]

[1] Beijing Electronic Science & Technology Institute, Beijing 100070, P.R. China
zyuan@tsinghua.edu.cn, sharonlee0915@163.com
[2] School of Telecommunications Engineering, Xidian University, Xi'an. Shaanxi 710071, P.R. China

Abstract. CLEFIA is a 128-bit block cipher proposed by Sony Corporation in 2007. Our paper introduces a new chosen text attack, the impossible differential-linear attack, on iterated cryptosystems. The attack is efficient for 16-round CLEFIA with whitening keys. In the paper, we construct a 13-round impossible differential-linear distinguisher. Based on the distinguisher, we present an effective attack on 16-round CLEFIA-128 with data complexity of $2^{122.73}$, recovering 96-bit subkeys in total. Our attack can also be applied to CLEFIA-192 and CLEFIA-256.

Keywords:CLEFIA, impossible differential-linear cryptanalysis, impossible differential cryptanalysis, linear approximation.

1 Introduction

CLEFIA [6] is a 128-bit block cipher supporting key lengths of 128, 192 and 256 bits. It achieves enough immunity against known attacks and is flexible enough for efficient implementation in both hardware and software. As a block cipher proposed by Sony Corporation in 2007, CLEFIA has received significant amount of cryptanalytic attention. However, except for the evaluation report [7] from the designer's, there are only a few significant cryptanalytic results about its security against various cryptanalytic techniques.

At present, the most powerful attack on CLEFIA is a series of impossible differential attacks on reduced rounds of it. The first one is proposed by its designers in the evaluation report of CLEFIA [7]. In FSE 2008, Tsunoo et al. introduced

⋆ This work is supported by the National Natural Science Foundation of China (No.61070250, 61170175), Beijing Natural Science Foundation (N0.4132066), the 12^{th} Five-year Cryptography Development Foundation of China (No.MMJJ201101026), the Fundamental Research Funds for the Central Universities, Scientific Research and Postgraduate Training Cooperation Project-Scientific Research Base-New Theory of Block Cipher and Obfuscation and their Application Research, and Foundation of State Key Laboratory of Information Security(Nos.01-01, 01-02-6).

⋆⋆ To whom correspondence should be addressed.

new 9-round impossible differentials for CLEFIA, and presented a 12-round attack on CLEFIA-128 with $2^{118.9}$ chosen plaintexts and 2^{119} encryptions[8]. Later, by the same impossible differential distinguisher, Zhang et al. presented an attack on 14-round CLEFIA, in which the design team pointed out a flaw and showed that it is not successful[9]. In IndoCrypt 2010, Tezcan proposed improbable differential cryptanalysis and applied it to 13/14/15-round CLEFIA-128/196/256 by taking advantage of relations among the round keys [4].

Our Contribution. In this paper, we will propose a new method, the impossible differential-linear attack, to analyze the CLEFIA block cipher. By constructing a 13-round distinguisher, using the new method, and combining it with key relations we found, we propose an attack on 16-round CLEFIA-128 with data complexity of $2^{122.73}$ and time complexity of $2^{122.73}$. Furthermore, Appendix A presents another distinguisher construction. The attacks to another 16-round is also given in Appendix B, we also show an attack on 15-round CLEFIA-128 in Appendix C. Our attacks are more efficient comparison to the present results. In Appendix D we also provide some key relations.

Outline. This paper is organized as follows: Section 2 provides a brief description of CLEFIA, and Section 3 introduces our new method of impossible differential-linear attack. In section 4 we present details of the 13-round impossible differential-linear distinguisher. The 16-round impossible differential-linear attack on CLEFIA-128 is discussed in detail in section 5. We summarize our results in section 6.

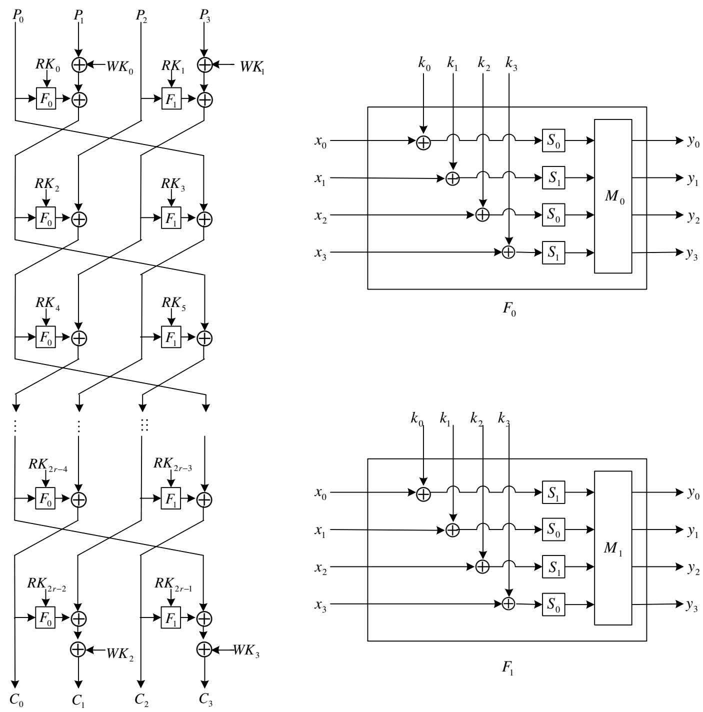
2 Description of CLEFIA

2.1 Notation

$a\,\vert b$	:	The concatenation of a and b;
$a_{(b)}$	:	b is the bit length of a;
a^T	:	The transposition of a vector a;
$P = (P_0, P_1, P_2, P_3)$	:	A 128-bit plaintext, $P_i \in \{0,1\}^{32} (0 \leq i \leq 3)$;
$C = (C_0, C_1, C_2, C_3)$	:	A 128-bit ciphertext, $C_i \in \{0,1\}^{32} (0 \leq i \leq 3)$;
$(X_i^0, X_i^1, X_i^2, X_i^3)$	:	The i^{th} round input data, $X_i^j \in \{0,1\}^{32}$
ΔX	:	The XOR value of X and X^*;

2.2 CLEFIA

CLEFIA is a 128-bit block cipher having a generalized Feistel structure with four 32-bit data lines. For the key lengths of 128, 192, and 256 bits, CLEFIA has 18, 22, and 26 rounds respectively. The encryption function uses four 32-bit whitening keys $WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ and $2r$ 32-bit round keys, where r is the number of rounds. $RK_i \in \{0,1\}^{32} (0 \leq i < 2r)$ represents round key, and $WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ are whitening keys. We denote d-branch r-round generalized Feistel network employed in CLEFIA as $GFN_{d,r}$.



(a) Encryption function ENC_r (b) Functions F_0 and F_1

Fig. 1. CLEFIA

The encryption process can be seen in Fig. 1(a). The details of $GFN_{4,r}$ are as follows:

- Step 1. $T_0\,|T_1\,|T_2\,|T_3 \leftarrow P_0\,|(P_1 \oplus WK_0)\,|P_2\,|(P_3 \oplus WK_1)$
- Step 2. For $i = 0$ to $r-1$ do the following:
  $$T_1 \leftarrow T_1 \oplus F_0(T_0, RK_{2i}), \quad T_3 \leftarrow T_3 \oplus F_1(T_2, RK_{2i+1})$$
  $$T_0\ |T_1\ |T_2\ |T_3 \leftarrow T_1\ |T_2\ \ |T_3\ |T_0$$
- Step 3. $C_0\ |C_1\ |C_2\ |C_3 \leftarrow T_3\ \ |\,(T_0 \oplus WK_2\ )\ \,|\,T_1\ |\,(T_2 \oplus WK_3\,)$

Each round contains two parallel F functions, F_0 and F_1, and their structures are shown in Fig. 1(b) where S_0 and S_1 are 8×8-bit S-boxes. The details of F_0 are as follows:

- Step 1. $T_0\,|T_1\,|T_2\,|T_3 \leftarrow RK \oplus x, \quad T_i \in \{0,1\}^8,\ \ x \in \{0,1\}^{32}$
- Step 2. $T_0 \leftarrow S_0(T_0)\ ,\ T_1 \leftarrow S_1(T_1)\ ,\ T_2 \leftarrow S_0(T_2)\ ,\ T_3 \leftarrow S_1(T_3)$
- Step 3. $y = M_0 \cdot (T_0, T_1, T_2, T_3)^T,\ \ y \in \{0,1\}^{32}$

F_1 is defined by replacing the terms in F_0 as follows: S_0 is replaced with S_1, S_1 with S_0, and M_0 with M_1.

The two matrices M_0 and M_1 used in the F-functions are defined as follows.

$$M_0 = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix}, M_1 = \begin{pmatrix} 0x01 & 0x08 & 0x02 & 0x0a \\ 0x08 & 0x01 & 0x0a & 0x02 \\ 0x02 & 0x0a & 0x01 & 0x08 \\ 0x0a & 0x02 & 0x08 & 0x01 \end{pmatrix}$$

2.3 Key Scheduling

For the 128-bit key, the Double Swap function $\Sigma : \{0,1\}^{128} \to \{0,1\}^{128}$ is defined as follows:

$$X_{(128)} \mapsto X[7-63]\,|X[121-127]\,|X[0-6]\,|X[64-120]$$

where $X[a-b]$ denotes a bit string cut from the a-th bit to the b-th bit of X.

Let $K = K_0\,|K_1\,|K_2\,|K_3$ be the key and L be an intermediate key, the key scheduling consists of the following 3 steps:

- Step 1. $L \leftarrow GFN_{4,12}(CON_0, \cdots, CON_{23}, K_0, \cdots, K_3)$
- Step 2.$WK_0\,|WK_1\,|WK_2\,|WK_3 \leftarrow K$
- Step 3. For i=0 to 8 do the following:
  $T \leftarrow L \oplus (CON_{24+4i}\,|CON_{24+4i+1}\,|CON_{24+4i+2}\,|CON_{24+4i+3})$
  $L \leftarrow \Sigma(L)$
  If i is odd: $T \leftarrow T \oplus K$
  $RK_{4i}\,|RK_{4i+1}\,|RK_{4i+2}\,|RK_{4i+3} \leftarrow T$

We need 60 constant values $CON_{(s)} (0 \leq s \leq 59)$ in the 128-bit key scheduling algorithm. Let $R = 0xb7e1 (= (e-2) \cdot 2^{16})$ and $Q = 0x243f (= (\pi - 3) \cdot 2^{16})$, where e is the base of the natural logarithm (2.71828...) and π is the circle ratio (3.14159...). $CON_{(s)}$ are generated by the following way, in which $IV^{128} = 0x428a (= (\sqrt[3]{2} - 1) \cdot 2^{16})$

- Step 1. $T \leftarrow IV^{128}$
- Step 2. For $j = 0$ to 29 do the following:
  $CON_{2j} \leftarrow (T \oplus R)|(\overline{T} <<< 1)$
  $CON_{2j+1} \leftarrow (\overline{T} \oplus Q)|(T <<< 8)$
  $T \leftarrow T \cdot 0x0002^{-1} (\mathrm{mod}\ z^{16} + z^{15} + z^{13} + z^{11} + z^5 + z^4 + 1)$

The key relations we found are illustrated in Appendix D.

3 The Impossible Differential-Linear Attack

Inspired by the differential-linear attack, first introduced by Langford and Hellman in [3], we propose a new cryptanalytic method called impossible differential-linear attack, because it combines the impossible differential cryptanalysis and linear cryptanalysis together. The attack is not completely new, since the impossible differential attack and linear attack were typical and widely used in previous attacks on various cryptosystems. However, no previous work has been done on combining these two together.

The block cipher E is represented as $E = E_1 \circ E_0$, where E_0 and E_1 are two subciphers. We use an impossible differential $\Omega_P \not\to \Omega_T$ with probability 1 for E_0, and a linear approximation $\lambda_P \to \lambda_T$ with probability $1/2 + q$ for E_1, where $\lambda_P \cdot \Omega_T$ is a fixed value $\pi(0 \text{ or } 1)$. Our impossible differential-linear attack procedures are described as follows:

1. Encrypting stage: Let M be the set of chosen plaintext pairs whose difference $P \oplus P^*$ is Ω_P, we encrypt distinct plaintexts in M.
   In E_0, we have $E_0(P) \oplus E_0(P^*) \neq \Omega_T$ with probability 1.
   In E_1, we can get equations
   $$\lambda_P \cdot E_0(P) \oplus \lambda_T \cdot E_1(E_0(P)) \oplus \lambda_K \cdot K = 0 \tag{1}$$
   and $\lambda_P \cdot E_0(P^*) \oplus \lambda_T \cdot E_1(E_0(P^*)) \oplus \lambda_K \cdot K = 0$. Both of their probabilities are $1/2 + q$. Consequently, using the piling up lemma presented in [5], we can get
   $$\lambda_P \cdot (E_0(P) \oplus E_0(P^*)) = \lambda_T \cdot E_1(E_0(P)) \oplus \lambda_T \cdot E_1(E_0(P^*)) \tag{2}$$
   with probability $1/2 + 2q^2$.
2. Decrypting stage: In this stage we will guess part of the subkeys. Then decrypt some rounds of the ciphertext pairs with the guessed subkeys. Decrypting process is also separated into two subciphers E_1^{-1} and E_0^{-1}, i.e., $D = E_0^{-1} \circ E_1^{-1}$. In the first decrypting subcipher E_1^{-1}, take $\lambda_P \cdot \Omega_T = \pi$ into Eq.(2), it can be rewritten as $\lambda_T \cdot E_1(E_0(P)) \oplus \lambda_T \cdot E_1(E_0(P^*)) = \lambda_P \cdot \Omega_T = \pi$.
   In fact, we just partially decrypt all ciphertext pairs (C, C^*)=$(E_1(E_0(P)), E_1(E_0(P^*)))$ with each guessed subkeys in the first decrypting subcipher E_1^{-1}. The subkeys, with the maximal probability not suiting Eq.(3)
   $$\lambda_P \cdot E_1^{-1}(C) \oplus \ \lambda_P \cdot E_1^{-1}(C^*) = \pi, \tag{3}$$
   is regard as the correct subkeys.
   Denote the set of pairs satisfying $E_1^{-1}(C) \oplus \ E_1^{-1}(C^*) = \Omega_T$ as T, and denote the set of pairs satisfying $\lambda_P \cdot E_1^{-1}(C) \oplus \ \lambda_P \cdot E_1^{-1}(C^*) = \pi$ as V. It is certain that $T \subset V$.

Property 1. For $\mathsf{T} \subset \mathsf{V}$, if an impossible map $\mathsf{M} \not\to \mathsf{V}$, another impossible map $\mathsf{M} \not\to \mathsf{T}$ also holds.

Proof. Assume that there is a map $\bar{F} : \mathsf{M} \to \mathsf{T}$. Randomly choose $p \in \mathsf{M}$, and compute $t = \bar{F}(p) \in \mathsf{T}$. Since $\mathsf{T} \subset \mathsf{V}$, we can get $t \in \mathsf{V}$, which indicates that there is $t = \bar{F}(p) \in \mathsf{V}$, i.e., the map $\bar{F}' : \mathsf{M} \to \mathsf{V}$ holds. It contradicts to the known condition.

◇

3. Sieving stage: Guess part of the first rounds subkeys, and eliminate those wrong values by showing that the impossible property holds if these subkeys are used. That is, we eliminate those wrong values in terms of $E_0(P) \oplus E_0(P^*) \neq E_1^{-1}(C) \oplus E_1^{-1}(C^*)$, where $\lambda_P \cdot E_1^{-1}(C) \oplus \lambda_P \cdot E_1^{-1}(C^*) = \pi$. From Property1, after sieving stage, the right values must satisfy that $E_0(P) \oplus E_0(P^*) \neq \Omega_T$.

Property 2. In *Sieving stage*, the success probability of sieving guessed values by $E_0(P) \oplus E_0(P^*) \neq E_1^{-1}(C) \oplus E_1^{-1}(C^*)$, where $\lambda_P \cdot E_1^{-1}(C) \oplus \lambda_P \cdot E_1^{-1}(C^*) = \pi$, is much higher than the filtering probability using $E_0(P) \oplus E_0(P^*) \neq \Omega_T$.

Proof. In *Sieving stage*, we eliminate those wrong guessed key values which satisfy Eq.(3). When the number of eliminated values is less than the total number, the more eliminated values, the higher successful sieving probability. After the first decrypting subcipher E_1^{-1}, The number of the key values satisfying Eq.(3) is more than the number of the wrong key values with $E_1^{-1}(C) \oplus E_1^{-1}(C^*) = \Omega_T$. So *Property* 2 is established.

$\diamond$

We named all the above as an impossible differential-linear distinguisher. The probability of our distinguisher is dominated by the above steps 1-2, which can be estimated separately as follows. The success rates are $1/2+2q^2$ and 1 in Encrypting stage and Decrypting stage, respectively. Because our elimination principle is sieving the values using the condition $E_0(P) \oplus E_0(P^*) \neq E_1^{-1}(E_1(E_0(P))) \oplus E_1^{-1}(E_1(E_0(P^*)))$, where $\lambda_P \cdot E_1^{-1}(E_1(E_0(P))) \oplus \lambda_P \cdot E_1^{-1}(E_1(E_0(P^*))) = \pi$, the total probability of our distinguisher is $1-(1/2+2q^2)$, i.e., $1/2-2q^2$. The key recovery attack requires about $8 \times \mathrm{O}(q^{-4})$ chosen plaintext pairs.

4 The 13-Round Impossible Differential-Linear Distinguisher

In this section, we first present a 13-round impossible differential-linear distinguisher, which consists of a 9-round impossible differential characteristic followed by a 4-round linear approximation.

4.1 9-Round Impossible Differential Characteristic

Paper [8] presented several 9-round impossible differential characteristics. We choose the following one that is the most efficient and suitable to our attack:

$$(0, \varpi, 0, 0) \not\rightarrow (0, \beta, 0, 0), \ \ where \ \ \varpi = (0,0,0,x), \beta = (y,0,0,0)$$

After the encryption of 9 rounds, the input difference of the 10^{th} round cannot have the following form:

$$\Delta X_9 = (\Delta X_9^0, \Delta X_9^1, \Delta X_9^2, \Delta X_9^3) = (\beta, 0, 0, 0) \tag{4}$$

with probability 1 as illustrated in Fig.2.

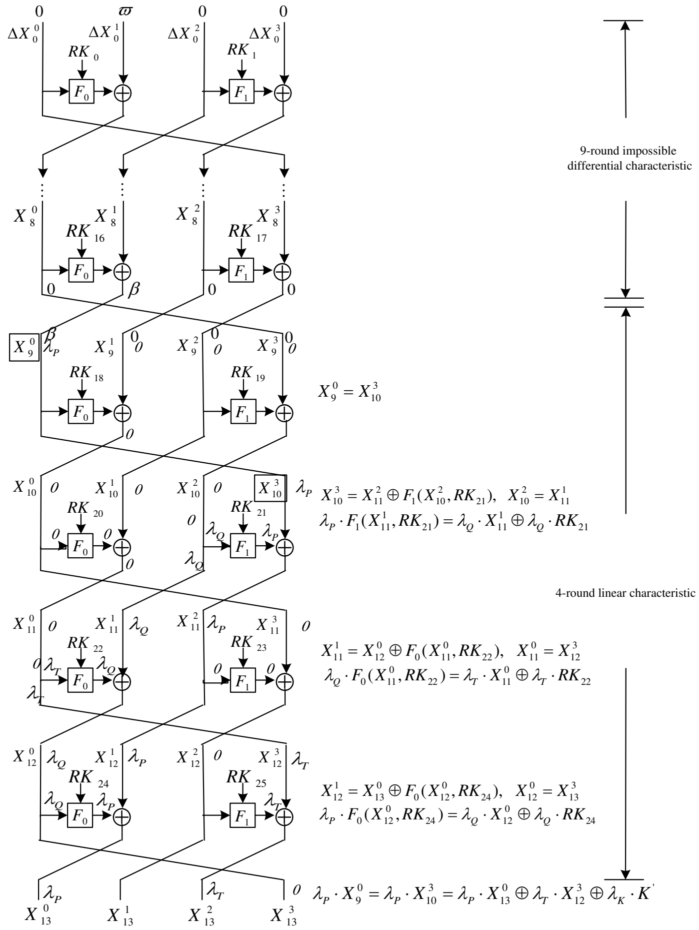


Fig. 2. 13-round impossible differential-linear distinguisher

4.2 4-Round Linear Characteristic

Here we will describe the construction of a 4-round linear characteristic illustrated in Fig.2, which is from round 10 to round 13. Details of the 4-round linear characteristic are described as follows.

In the 10^{th} round, we get $X_9^0 = X_{10}^3$.

In the 11^{th} round, based on the definition of the round function F_1, we can get the following two equations:

$$X_{10}^3 \oplus F_1(X_{10}^2, RK_{21}) = X_{11}^2, X_{10}^2 = X_{11}^1$$

Using linear approximations for the non-linear S-boxes in F_1, we can get the following equation.

$$\lambda_P \cdot F_1(X_{11}^1, RK_{21}) = \lambda_Q \cdot X_{11}^1 \oplus \lambda_Q \cdot RK_{21}$$

As a result, the linear characteristic of the 11^{th} round can be expressed by the following equation:

$$\lambda_P \cdot X_{10}^3 = \lambda_P \cdot X_{11}^2 \oplus \lambda_Q \cdot X_{11}^1 \oplus \lambda_Q \cdot RK_{21}, p_1 = 1/2 + q_1 \quad (5)$$

Similarly, the linear characteristic of the 12^{th} round can be expressed as

$$\lambda_Q \cdot X_{11}^1 = \lambda_Q \cdot X_{12}^0 \oplus \lambda_T \cdot X_{12}^3 \oplus \lambda_T \cdot RK_{22}, p_2 = 1/2 + q_2 \quad (6)$$

In the 13^{th} round, we can first get the following equation.

$$X_{12}^1 \oplus F_0(X_{12}^0, RK_{24}) = X_{13}^0, X_{12}^0 = X_{13}^3$$

Next, we can choose an appropriate pair of values (λ_P, λ_Q) by taking the linear characteristics expressed in Eq.(5) and Eq.(6) into account and get the linear characteristic of the 13^{th} round as follows:

$$\lambda_P \cdot X_{12}^1 = \lambda_P \cdot X_{13}^0 \oplus \lambda_Q \cdot X_{13}^3 \oplus \lambda_Q \cdot RK_{24}, p_3 = 1/2 + q_3 \quad (7)$$

Finally, by concentrating the above linear characteristics of rounds 10-13 together, we can have the following property:

Property 3. If Eq.(5)-(7) hold, we can get the following 4-round linear characteristic of CLEFIA from round 10 to round 13:

$$\lambda_P \cdot X_9^0 = \lambda_P \cdot X_{10}^3 = \lambda_P \cdot X_{13}^0 \oplus \lambda_T \cdot X_{12}^3 \oplus \lambda_K \cdot K', p = 1/2 + 2^2 q_1 q_2 q_3 \quad (8)$$

Proof. If Eq.(5)-(7) are true, this property is obvious from the CLEFIA structure.

◇

Note 1. Similarly, if we arbitrarily choose a 4-round CLEFIA from round $i (i \geq 0)$ to round $i+3$, we can rewrite Eq.(5)-(7) as following Eq.(5′)-(7′):

$$\lambda_P \cdot X_{i+1}^3 = \lambda_P \cdot X_{i+2}^2 \oplus \lambda_Q \cdot X_{i+2}^1 \oplus \lambda_Q \cdot RK_{2(i+1)+1}, p_1' = 1/2 + q_1' \quad (5')$$

$$\lambda_Q \cdot X_{i+2}^1 = \lambda_Q \cdot X_{i+3}^0 \oplus \lambda_T \cdot X_{i+3}^3 \oplus \lambda_T \cdot RK_{2(i+2)}, p_2' = 1/2 + q_2' \quad (6')$$

$$\lambda_P \cdot X_{i+3}^1 = \lambda_P \cdot X_{i+4}^0 \oplus \lambda_Q \cdot X_{i+4}^3 \oplus \lambda_Q \cdot RK_{2(i+3)}, p_3' = 1/2 + q_3' \quad (7')$$

and we can obtain the following 4-round linear characteristic

$$\lambda_P \cdot X_i^0 = \lambda_P \cdot X_{i+1}^3 = \lambda_P \cdot X_{i+4}^0 \oplus \lambda_T \cdot X_{i+3}^3 \oplus \lambda_K \cdot K', p' = 1/2 + 2^2 q_1' q_2' q_3'$$

4-Round Linear Approximations. Furthermore, we can derive the 4-round linear input mask from the above 4-round linear characteristic.

Let $(v,u)_{RK_k}$ be an approximation of a 32-bit invertible function $F_s(X_j^i, RK_k)$, $(0 \le s \le 1, 0 \le k \le 35)$. Eq.(5) suggests that the linear approximation of function $F_1(X_{10}^2, RK_{21})$ is

$$(v,u)_{RK_{21}} = (\lambda_Q, \lambda_P) \tag{9}$$

Eq.(6) indicates that the linear approximation of function $F_0(X_{11}^0, RK_{22})$ is

$$(v,u)_{RK_{22}} = (\lambda_T, \lambda_Q) \tag{10}$$

and Eq.(7) indicates

$$(v,u)_{RK_{24}} = (\lambda_Q, \lambda_P). \tag{11}$$

Denoting the input mask of the j^{th} 32-bit input data in the i^{th} round as $IM_{X_{i-1}^j}(1 \le i \le 16, 0 \le j \le 3)$, which is also the output mask of the $((j+1)mod4)^{th}$ output data in the $(i-1)^{th}$ round, we can get $IM_{X_{10}^3} = IM_{X_{11}^2} = \lambda_P$ from Eq.(9), $IM_{X_{11}^1} = IM_{X_{12}^0} = \lambda_Q$ and $IM_{X_{12}^3} = \lambda_T$ from Eq.(10), and $IM_{X_{12}^1} = IM_{X_{13}^0} = \lambda_P$ from Eq.(11). In addition, we can also derive $IM_{X_{10}^2} = 0$, $IM_{X_{10}^1} = 0$, and $IM_{X_{10}^1} = 0$, and so on.

As a result, we can derive some 128-bit input masks as follows:

Property 4. If Property 3 holds, the 128-bit input masks of 4-round CLEFIA are:

In the 10^{th} round: $IM_{X_9} = (IM_{X_9^0}, IM_{X_9^1}, IM_{X_9^2}, IM_{X_9^3}) = (\lambda_P, 0,0,0)$.

In the 11^{th} round: $IM_{X_{10}} = (IM_{X_{10}^0}, IM_{X_{10}^1}, IM_{X_{10}^2}, IM_{X_{10}^3}) = (0,0,0,\lambda_P)$.

In the 12^{th} round: $IM_{X_{11}} = (IM_{X_{11}^0}, IM_{X_{11}^1}, IM_{X_{11}^2}, IM_{X_{11}^3}) = (0, \lambda_Q, \lambda_P, 0)$.

In the 13^{th} round: $IM_{X_{12}} = (IM_{X_{12}^0}, IM_{X_{12}^1}, IM_{X_{12}^2}, IM_{X_{12}^3}) = (\lambda_Q, \lambda_P, 0, \lambda_T)$.

In the 14^{th} round, $IM_{X_{13}} = (IM_{X_{13}^0}, IM_{X_{13}^1}, IM_{X_{13}^2}, IM_{X_{13}^3}) = (\lambda_P, *, \lambda_T, 0)$, where "$*$" denotes an unknow 32-bit input mask.

Note 2. If a 4-round CLEFIA, from round $i(i \ge 0)$ to round $i+3$, satisfies Eq.(5′)-(7′), their 128-bit input masks are $IM_{X_i} = (\lambda_P, 0,0,0)$, $IM_{X_{i+1}} = (0,0,0,\lambda_P)$, $IM_{X_{i+2}} = (0, \lambda_Q, \lambda_P, 0)$, and $IM_{X_{i+3}} = (\lambda_Q, \lambda_P, 0, \lambda_T)$, respectively. Additionally, $IM_{X_{i+4}} = (\lambda_P, *, \lambda_T, 0)$.

4.3 The 13-Round Impossible Differential-Linear Distinguisher

Here, we first propose a new property, impossible differential-linear property, which is a concatenation of impossible differential characteristic and linear characteristic. To concatenate the above two parts together, the core technology resides in how to link the output differential $\Delta X_9 = (\beta, 0, 0, 0)$ and the input masks $IM_{X_9} = (\lambda_P, 0,0,0)$ of the 10^{th} round together?

From section 4.1, we have $\beta = (y, 0, 0, 0), y \in F_2^8 \backslash \{0\}$. If choosing $\lambda_P = (0, \lambda_1, \lambda_1, \lambda_1)$,$\lambda_1 \in \{01, 02, ...ff\}$, by Eq.(4), we can get the following equation with probability 1:

$$\lambda_P \cdot \Delta X_9^0 = \lambda_P \cdot (X_9^0 \oplus X_9^{0*}) = 0. \tag{12}$$

As a result, we always have $IM_{X_9} \cdot \Delta X_9 = (0, 0, 0, 0)$ in the 10^{th} round, which links the output differential ΔX_9 and the input masks IM_{X_9} together.

Property 5. For a pair of plaintexts (P, P^*) whose difference is $(0, \varpi, 0, 0)$ with $\varpi = (0, 0, 0, x)$, if we choose $\lambda_P = (0, \lambda_1, \lambda_1, \lambda_1)$,$\lambda_1 \in \{01, 02, ...ff\}$, the 4-round linear characteristic can be concatenated to the 9-round impossible differential characteristic based on Eq.(8)and Eq.(12) to form the following 13-round impossible differential-linear distinguisher.

$$\lambda_P \cdot (X_{13}^0 \oplus X_{13}^{0*}) \oplus \lambda_T \cdot (X_{12}^3 \oplus X_{12}^{3*}) = 0 \qquad (13)$$

Details of another 13-round impossible differential distinguisher are discussed in Appendix A.

4.4 Selection of $\boldsymbol{\lambda}$

In this subsection, we show how to select the values for λ_P, λ_Q and λ_T to make the bias of the 4-round linear characteristic as high as possible.

At first, we analyze the linear approximation of F_1 in the 11^{th} round as follows.

$$\lambda_P \cdot F_1(X_{11}^1, RK_{21}) = \lambda_Q \cdot X_{11}^1 \oplus \lambda_Q \cdot RK_{21}$$

The four bytes output of the S-boxes are denoted as (u, v, z, w). Then the round function can be expressed as:

$$F_1(X_{11}^1, RK_{21}) = M_1(S(X_{11}^1 \oplus RK_{21})) = M_1(u, v, z, w)$$

According to the definition in section 2, we can get the following equation:

$$M_1(u, v, z, w)^T = \begin{pmatrix} u \oplus (08 \times v) \oplus (02 \times z) \oplus (0a \times w) \\ (08 \times u) \oplus v \oplus (0a \times z) \oplus (02 \times w) \\ (02 \times u) \oplus (0a \times v) \oplus z \oplus (08 \times w) \\ (0a \times u) \oplus (02 \times v) \oplus (08 \times z) \oplus w \end{pmatrix}$$

Next, based on the discussion in section 4.3 about how to choose value for λ_P, the left part of the linear approximation can be computed as follows:

$$\lambda_P \cdot F_1(X_{11}^1, RK_{21}) = \{00 \;\; \lambda_1 \;\; \lambda_1 \;\; \lambda_1\} \cdot M_1(u, v, z, w)^T$$

$$= \lambda_1 \cdot (v \oplus (08 \times v) \oplus z \oplus (02 \times z) \oplus w \oplus (0a \times w))$$

Note that the primitive polynomial used in the multiplication is $z^8 + z^4 + z^3 + z^2 + 1$, which can be denoted as a binary string 100011101. Hence, we can compute the parity of $\lambda_1 \cdot (02 \times z)$ as follows:

$$\lambda_1 \cdot (02 \times z) = \begin{matrix} \lambda_1 \cdot (z << 1), & z_7 = 0 \\ \lambda_1 \cdot (z << 1 \oplus 00011101), & z_7 = 1 \end{matrix}$$

where z_7 denotes the left-most bit of z. By choosing an appropriate value of λ_1 such that $\lambda_1 \cdot 00011101 = 0$, the above two cases can both be transformed into the following equation:

$$\lambda_1 \cdot (02 \times z) = \lambda_1 \cdot (z << 1) = (\lambda_1 >> 1) \cdot z$$

no matter what the left-most bit of z is.

Similarly, when λ_1 also satisfies $(\lambda_1 >> 1) \cdot 00011101 = 0$, the parity of $\lambda_1 \cdot (08 \cdot v)$ and $\lambda_1 \cdot (0a \cdot w)$ can be computed respectively as follows:

$$\lambda_1 \cdot (08 \times v) = \lambda_1 \cdot (v << 3) = (\lambda_1 >> 3) \cdot v$$

$$\lambda_1 \cdot (0a \times w) = \lambda_1 \cdot ((02 \times w) \oplus (08 \times w)) = ((\lambda_1 >> 1) \oplus (\lambda_1 >> 3)) \cdot w$$

Therefore, the left part of the linear approximation can be transformed into the following equation:

$\lambda_P \cdot F_1(X_{11}^1, RK_{21})$

$$= (\lambda_1 \oplus (\lambda_1 >> 3)) \cdot v \oplus (\lambda_1 \oplus (\lambda_1 >> 1)) \cdot z \oplus (\lambda_1 \oplus (\lambda_1 >> 1) \oplus (\lambda_1 >> 3)) \cdot w$$

$$= \{00, \lambda_1 \oplus (\lambda_1 >> 3), \lambda_1 \oplus (\lambda_1 >> 1), \lambda_1 \oplus (\lambda_1 >> 1) \oplus (\lambda_1 >> 3)\} \cdot (u, v, z, w)$$

By utilizing the linear distribution table of each S-box, we use the following linear approximation for each S-box (ε denotes the bias of the linear approximation).

$$(\lambda_1 \oplus (\lambda_1 >> 3)) \cdot v = \lambda_2 \cdot (X_{11}^1 \oplus RK_{21})_1, \quad p_4 = 1/2 + \varepsilon_1$$

$$(\lambda_1 \oplus (\lambda_1 >> 1)) \cdot z = \lambda_2 \cdot (X_{11}^1 \oplus RK_{21})_2, \quad p_5 = 1/2 + \varepsilon_2$$

$$(\lambda_1 \oplus (\lambda_1 >> 1) \oplus (\lambda_1 >> 3)) \cdot w = \lambda_2 \cdot (X_{11}^1 \oplus RK_{21})_3, \quad p_6 = 1/2 + \varepsilon_3$$

where $(X_{11}^1 \oplus RK_{21})_j$ stands for the j^{th} byte of $(X_{11}^1 \oplus RK_{21})(0 \le j \le 3)$, and (u, v, z, w) denotes the corresponding output of each S-box respectively.

As a result, we get the following linear approximation for the function F_1 in the 11^{th} round.

$$\lambda_P \cdot F_1(X_{11}^1 \oplus RK_{21}) = \{00, \lambda_2, \lambda_2, \lambda_2\} \cdot (X_{11}^1 \oplus RK_{21}), \quad p = 1/2 + 2^2\varepsilon_1\varepsilon_2\varepsilon_3$$

Note that we choose λ_Q as the form of $\lambda_Q = \{00, \lambda_2, \lambda_2, \lambda_2\}$, such that we can make use of the property of the linear transformation as described in section 4.1.

Similar analysis can be applied to the linear approximation used in the 12^{th} and 13^{th} round. By running through all the possible values of λ_P, λ_Q and λ_T that satisfies the above conditions, we can choose the following three linear approximations which achieve the highest biases.

$$\{00, f6, f6, f6\} \cdot F_1(X_{11}^1 \oplus RK_{21}) = \{00, eb, eb, eb\} \cdot (X_{11}^1 \oplus RK_{21})$$

whose probability is $p \approx 1/2 - 2^{-11.61}$.

$$\{00, eb, eb, eb\} \cdot F_0(X_{12}^3 \oplus RK_{22}) = \{00, 49, 49, 49\} \cdot (X_{12}^3 \oplus RK_{22})$$

whose probability is $p \approx 1/2 - 2^{-10.83}$.

$$\{00, f6, f6, f6\} \cdot F_0(X_{13}^3 \oplus RK_{24}) = \{00, eb, eb, eb\} \cdot (X_{13}^3 \oplus RK_{24})$$

whose probability is $p \approx 1/2 - 2^{-10.19}$.

Plugging the corresponding values of λ_P, λ_Q and λ_T into Eq.(5)-(8), we can get the following 3-round linear characteristic of CLEFIA.

$$\{00, f6, f6, f6\} \cdot X_{10}^3 = \{00, f6, f6, f6\} \cdot X_{13}^0 \oplus \{00, 49, 49, 49\} \cdot X_{12}^3 \oplus \lambda_K \cdot K^{'} \quad (14)$$

whose probability is $p \approx 1/2 - 2^{-30.63}$.

Finally, the decrypting stage of the 13-round impossible differential-linear distinguisher can be expressed as:

$$\{00, f6, f6, f6\} \cdot (X_{13}^0 \oplus X_{13}^{0*}) \oplus \{00, 49, 49, 49\} \cdot (X_{12}^3 \oplus X_{12}^{3*}) = 0 \quad (15)$$

The total probability of the 13-round impossible differential-linear distinguisher can be computed as described in section 3, which is about $1/2 - 2^{-60.26}$.

5 The Impossible Differential-Linear Attack on 16-Round CLEFIA-128

In this section, we explain our impossible differential-linear attack on 16-round CLEFIA-128 with whitening keys. In this attack, we set the above 13-round impossible differential-linear distinguisher as rounds 3-15, and extend two rounds backward and one round forward as shown in Fig.3.

The expression of the decrypting stage of the 13-round impossible differential-linear distinguisher should be transformed to the following form:

$$\{00, f6, f6, f6\} \cdot (X_{15}^0 \oplus X_{15}^{0*}) \oplus \{00, 49, 49, 49\} \cdot (X_{14}^3 \oplus X_{14}^{3*}) = 0, \quad (16)$$

and the total probability of the 13-round impossible differential-linear distinguisher is around $1/2 - 2^{-60.26}$, theoretically. Based on the analysis in section 3, we can know that approximately $12 \times (2^{-61.26})^{-2} \approx 2^{126.1}$ correct pairs are needed to mount the key recovery attack.

In the following, we first introduce how to obtain the plaintext pairs, then describe the attack procedure in detail as illustrated in Fig.3. In the end, we estimate the data complexity and time complexity of our attack.

5.1 Chosen Plaintext

We choose a structure composed of 2^{72} plaintexts that is defined as follows:

$$S_P = (X_0^0, X_0^1, X_0^2, X_0^3)_j, 1 \leq j \leq 2^{72},$$

If we choose plaintext pairs (P, P^*) where $P = (X_0^0, X_0^1, X_0^2, X_0^3)$ and $P^* = (X_0^{0*}, X_0^{1*}, X_0^{2*}, X_0^{3*}) = (X_0^0 \oplus \delta, X_0^1 \oplus \gamma, X_0^2, X_0^3 \oplus \varpi)$, whose difference takes the form $\Delta P = (\delta, \gamma, 0, \varpi)$ with $\varpi = (0, 0, 0, x)$, $\delta = (aw, 2w, 8w, w)$, $w = M_0(S(x')) \oplus M_0(S(x' \oplus x))$ $(x' \in F_2^8)$, and $\gamma = (v_0, v_1, v_2, v_3)$. We can get $\Delta X_2 = (0, \varpi, 0, 0)$. For the computations of δ and γ, please refer to Fig.3. Thus, we have 255 possible values of both ϖ and δ, $2^{32} - 1$ possible values of γ, and one structure can produce about 2^{119} distinct plaintext pairs.

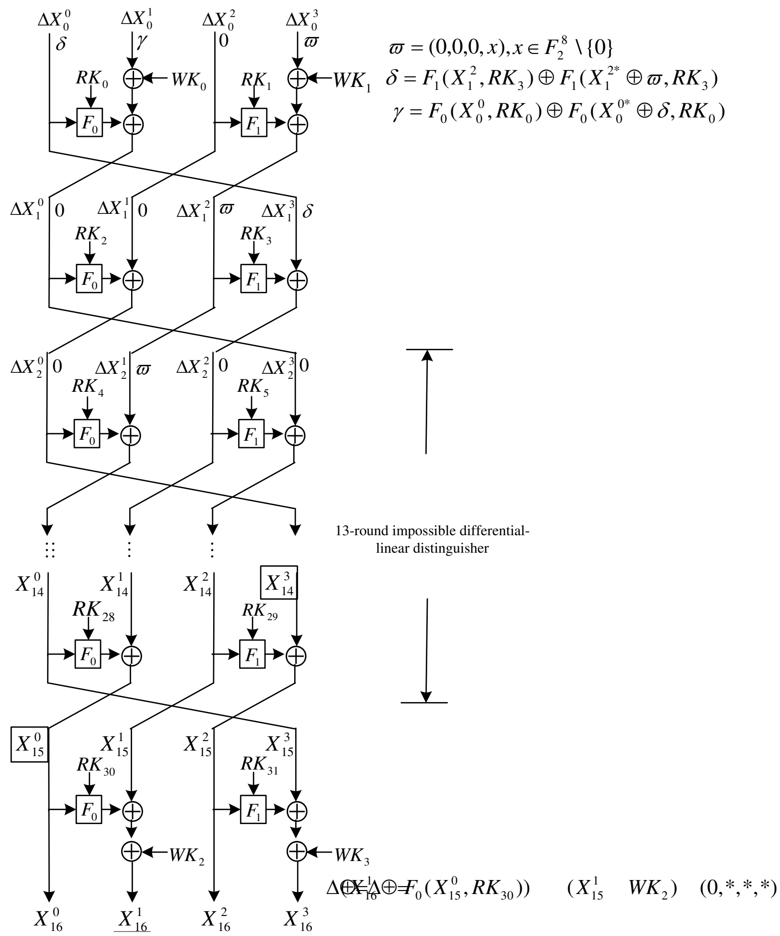


Fig. 3. 16-round impossible differential-linear attack

5.2 The Impossible Differential-Linear Attack on 16-Round CLEFIA-128 with Whitening Keys

In the following, we will discuss our impossible differential-linear attack on 16-round CLEFIA-128 with whitening keys in detail. In Fig.3, plaintext $P = X_0$, ciphertext $C = X_{16}$.

Step 1. Take $2^{50.73}$ structures defined above, i.e. $2^{72} \times 2^{50.73} = 2^{122.73}$ plaintexts, so $2^{119} \times 2^{50.73} = 2^{169.73}$ plaintext pairs. Encrypt $2^{122.73}$ distinct plaintexts for 16 rounds. Insert all ciphertexts into a table T_0 indexed by X_{15}^0, ($X_{15}^0 = X_{16}^0$).

Step 2. Let 32-bit subkey RK_{30} and 24-bit subkey $(RK_{29} \oplus WK_2)$ be indexed by $N_1, \ldots, N_{2^{56}}$ and reset $N_i (1 \leq i \leq 2^{56})$.

Create a table T_1 of $F_0(X_{15}^0, RK_{30})$, indexed by all 2^{32} values of RK_{30} and 2^{32} values of X_{15}^0. For every guess of RK_{30} (32-bit), look up the value of $F_0(X_{15}^0, RK_{30})$ in T_1 for each X_{15}^0, and obtain the value of $X_{15}^1 \oplus WK_2 = X_{16}^1 \oplus F_0(X_{15}^0, RK_{30})$ for each X_{16}^1. Select only the pairs whose difference are equal in the first byte of $(X_{15}^1 \oplus WK_2)$, the expected number of such pairs is $2^{169.73} \times 2^{-8} = 2^{161.73}$.

Then for every guess of the last three bytes of subkey $RK_{29} \oplus WK_2$ (24-bit), we can partially compute the value of $\lambda_Q \cdot X_{14}^3 = \lambda_Q \cdot (F_1(X_{15}^1 \oplus WK_2, RK_{29} \oplus WK_2) \oplus X_{16}^2)$ for each X_{16}^2, and the value of $\lambda_P \cdot X_{15}^0 \oplus \lambda_Q \cdot X_{14}^3$. If the pair satisfies Eq.(16), increment the corresponding N_i by 1.

After running all 2^{56} guesses, we output the minimum value of N_i as the 56-bit correct subkeys. Based on the analysis in section 3 and [1,5], we know that approximately $8 \times (2^{60.26})^2 \approx 2^{123.52}$ plaintext pairs are needed for the 13-round impossible linear distinguisher, we expect to have $2^{38.21}$ pairs left with this condition.

Step 3. We eliminate those wrong 40-bit values for the first two rounds subkey (RK_0, RK_3^3) (The first three zero bytes of ϖ only lead to the last byte of RK_3 that affects ΔF_1, so 32-bit RK_0 and 8-bit RK_3) by showing that the impossible property holds if these subkeys are used. To do so, we use a precomputation stage. At this precomputation stage, we consider all pairs whose difference $(\Delta X_2^0, \Delta X_2^1, \Delta X_2^2, \Delta X_2^3) = (0, \varpi, 0, 0)$ after the first two rounds encryption. To achieve this, we need to perform two step, the first step makes sure that $\Delta X_1^0 = 0$, and the second step enables $\Delta X_2^2 = 0$.

1). If $\Delta X_1^0 = 0$, there are 2^{32} possible values for X_1^0. We perform $A_1 = F_0^{-1}(X_1^0 \oplus WK_0 \oplus X_0^1)$ and create a hash table H_1 containing one of the outputs of A_1 and the XOR of the two outputs $(X_0^0 \oplus RK_0)$. There are 2^{32} possible values for $(X_0^0 \oplus RK_0)$, and on average one value of X_1^0 corresponds to each value of $(X_0^0 \oplus RK_0)$. Now for each of the $2^{38.21}$ remaining pairs we compute $(X_0^0 \oplus RK_0)$, and use the table H_1 to fetch one possibility of X_1^0 that corresponds to the computed $(X_0^0 \oplus RK_0)$. The process identifies roughly one wrong value for the subkey RK_0 by XORing the plaintext and A_1. The probability of a wrong 32-bit value for RK_0 is $(1 - 2^{-32})$. After analyzing all $2^{38.21}$ pairs, we expect only $2^{32} \times (1 - 2^{-32})^{2^{38.21} \times 1} \approx 2^{-75}$ wrong values of RK_0 remaining.

2). In round 2, if $\Delta X_2^2 = 0$, there are 2^{32} possible values for X_2^2. We perform $A_2 = F_1^{-1}(X_2^2 \oplus X_1^3) = F_1^{-1}(X_2^2 \oplus X_0^0)$ and create a hash table H_2 containing one of the outputs of A_2 and XOR of the two outputs $(X_1^2 \oplus RK_3)$. There are 2^{32} possible values for $(X_1^2 \oplus RK_3)$ and 2^8 possible values for RK_3^3. Now for each of the $2^{38.21}$ remaining pairs we compute $(X_1^2 \oplus RK_3)$, and use the table H_2 to fetch one possibility of X_2^2 that corresponds to the computed $(X_1^2 \oplus RK_3)$. The process identifies roughly one wrong value for the subkey RK_3 by XORing the plaintext and A_2. The probability of a wrong 8-bit value for RK_3^3 is $(1 - 2^{-32})$. After analyzing all $2^{38.21}$ pairs, we expect only $2^8 \times (1 - 2^{-32})^{2^{38.21} \times 1} \approx 2^{-99}$ wrong values of RK_3^3.

Therefore, wrong values of the 40-bit of (RK_0, RK_3^3) can be established unless the initial guess of the 32-bit value of RK_{30} or 24-bit value of $(RK_{29} \oplus WK_2)$ is correct. It is expected that we can eliminate the whole 40-bit value of RK_0 and RK_3 in this step, since the wrong values of $(RK_0, RK_3^3, (RK_{29} \oplus WK_2), RK_{30})$ remains with a small probability of $\max\{2^{56} \times 2^{-75} = 2^{-19}, 2^{56} \times 2^{-99} = 2^{-43}\}$. Hence if there remains a value of (RK_0, RK_3^3), we can assume that the guessed 56-bit values for $(RK_{29} \oplus WK_2)$ and RK_{30} are correct. Our attack can recover 96-bit subkeys.

Complexity Analysis. According to the above analysis , a structure has 2^{72} plaintexts, we need about $2^{50.73}$ structures, so the data complexity of our attack is about $2^{72} \times 2^{50.73} = 2^{122.73}$.

Step 1 need $2^{122.73}$ encrypting operations, Step 2 requires $(2 \times 2^{32} \times 2^{32} \times 2^{32}) \approx 2^{97}$ F operations, which is equal to 2^{96} one round operation. The required time for memory access in step 3 is less than $(2^{32} \times 2^{32}) \times 2^{38.21} + (2^8 \times 2^{32}) \times 2^{38.21} \approx 2^{102.21}$ F operations, i.e. $2^{101.21}$ operations one round. Therefore, the total time complexity of our attack can be estimate as about $2^{122.73} + (2^{96} + 2^{101.21})/16 \approx 2^{122.73}$.

$2^{122.73}/2^3 = 2^{119.73}$ bytes of memory are needed to store the table T_0, $2^{96}/2^3 = 2^{93}$ bytes of memory are needed to store the list of deleted key values $(RK_{30}, RK_{29}, RK_0, RK_3^3)$, $2 \times 2^{31} = 2^{32}$ bytes of memory are needed to store the hash table (H_1, H_2), and $2^{64}/2^3 = 2^{61}$ bytes of memory are needed to store table T_1. Our attack can recover 96-bit subkeys $(RK_0, RK_3^3, RK_{29} \oplus WK_2, RK_{30})$.

Note 3. For another 13-round impossible differential-linear distinguisher and another 16-round attack to CLEFIA-128, please refer to Appendix A and Appendix B, respectively. For attacks to 15-round CLEFIA-128, please refer to Appendix C. Our attack is also effective to CLEFIA-192 and CLEFIA-256.

6 Conclusion

In this paper, we present a new attack, impossible differential-linear attack, and achieve a result of 16-round CLEFIA-128 with $2^{122.73}$ CP, and time complexity is also $2^{122.73}$. The comparison of cryptanalytic results with CLEFIA is illustrated in Table 1, which shows that our attack is more efficient than the present results. The attack is also effective to 15-round CLEFIA-128, given in Appendix C.

References

1. E. Biham, O. Dunkelman, N. Keller. Enhancing Differential-Linear Cryptanalysis, Advances in Cryptology, Proceedings of ASIACRYPT 2002, Lecture Notes in Computer Science 2501, pp. 254-266, Springer, (2002)
2. A. Bogdanov, V. Rijmen. Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers. In: http://eprint.iacr.org/2011/123.
3. S. K. Langford, M. E. Hellman. Differential-Linear Cryptanalysis, Advances in Cryptology. In:*Proceedings of CRYPTO94*. LNCS, vol. 839, pp. 17-25. Springer, Heidelberg (1994)

Table 1. Comparison of Cryptanalysis Results of CLEFIA-128

Reference	Rounds	Recover Key	Data Complexity	Time Complexity
[6,7]	10	32-bit	$2^{101.7}$	2^{102}
[8]	12	72-bit	$2^{118.9}$	2^{119}
[9]	12	80-bit	$2^{118.9}$	2^{82}
this paper	16	96-bit	$2^{122.73}$	$2^{122.73}$
this paper	16	104-bit	$2^{124.52}$	2^{131}
this paper	15	64-bit	$2^{124.52}$	$2^{93.1}$
this paper	15	64-bit	$2^{124.52}$	$2^{99.1}$

4. C. Langford. Improbable Differential Attack-Cryptanalysis of Reduced Round CLEFIA, Advances in Cryptology. In:*Proceedings of INDOCRYPT2010*. LNCS, vol. 6498, pp. 197-209. Springer, Heidelberg (2010)
5. M. Matsui. Linear Cryptanalysis Method for DES Cipher, Advances in Cryptology. In:*Proceedings of EUROCRYPT93*. LNCS, vol. 765, pp. 386-397. Spinger, Heidelberg (1994)
6. T. Shirai, K. Shibutani, T. Akishita, S. Moriai, T. Iwata. The 128-bit Blockcipher CLEFIA. In:*Proceedings of Fast Software Encryption 2007*, LNCS, vol. 4593, pp. 181-195.(2007)
7. Sony Corporation. The 128-bit blockcipher CLEFIA: Security and performance evaluations. Revision 1.0, On-Line document, 2007.June 1 (2007), http://www.sony.co.jp/Products/clefia/technical/data/clefia-eval-1.0.pdf
8. Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki and H. Kubo. Impossible Differential Cryptanalysis of CLEFIA. In:*Fast Software Encryption-FSE 2008*, LNCS, vol. 5086, pp. 398-411. Springer, Verlag (2008).
9. W. Zhang , J. Han. Impossible Differential Analysis of Reduced Round CLEFIA. In:*Beijing, China. Proc of Inscrypt'08* . pp. 181-191. (2008)

Appendix A. Another 13-round impossible differential-linear distinguisher

Another 13-round impossible differential-linear distinguisher concatenates an impossible differential

$$(0, 0, 0, \varpi) \nrightarrow (0, 0, 0, \beta)^{[3]}$$

with a 4-round linear characteristic. For details please refer to Fig.4.

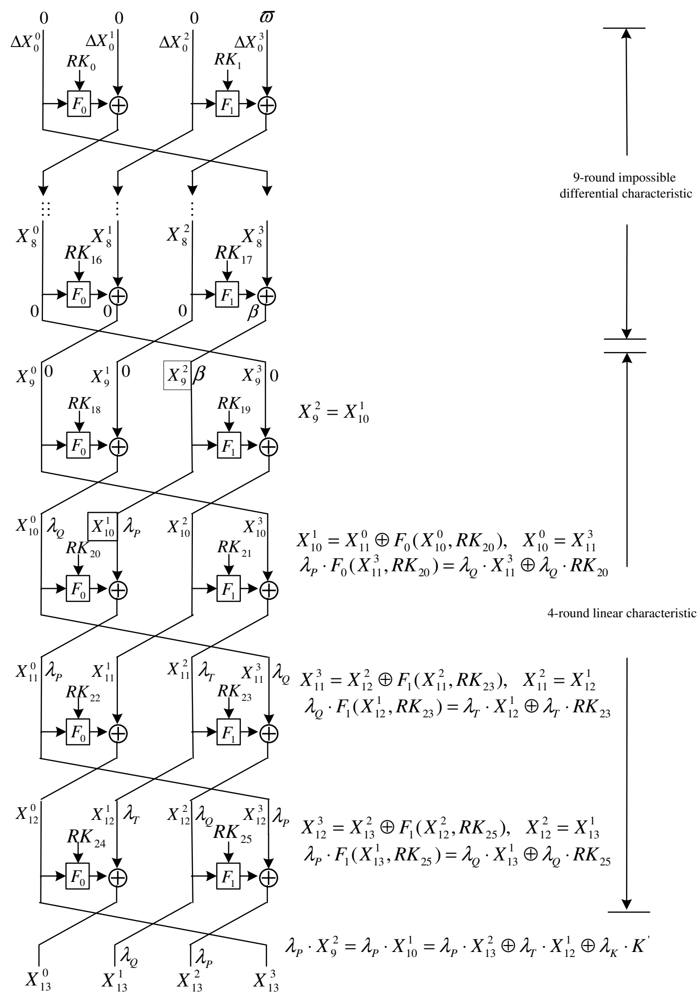


Fig. 4. 13-round impossible differential-linear distinguisher

Appendix B. Another Attack on 16-round CLEFIA-128

Another 16-round attack on CLEFIA-128 is illustrated in Fig.5 with the 13-round impossible differential-linear distinguisher in section 4, and three rounds extension on plaintext side. Its main ideas is: Choose a structure composed of 2^{104} plaintexts, whose corresponding plaintext pairs are of the form $\Delta P = (\varpi, \xi, \gamma, \theta)$. Encrypt all 2^{183} plaintext pairs, select only the pairs whose ciphertexts are equal in the first byte of X_{16}^2. According to section 5.2, we can recover 104-bit subkeys.

The data complexity is about

$$2^{104} \times \left[8 \times \left(2^{60.26}\right)^2 \Big/ \left(2^{183} \times 2^{-8} \times 2^{-32} \times 2^{-32} \times 2^{-8}\right)\right] \approx 2^{124.52}.$$

The time complexity is

$$[(2^{32} \times 2^{32})/2 + ((2^{32} \times 2^{32}) \times 2^{72} + (2^{32} \times 2^{32}) \times 2^{72} + (2^{8} \times 2^{32}) \times 2^{72})/2]/16 \approx 2^{131}$$

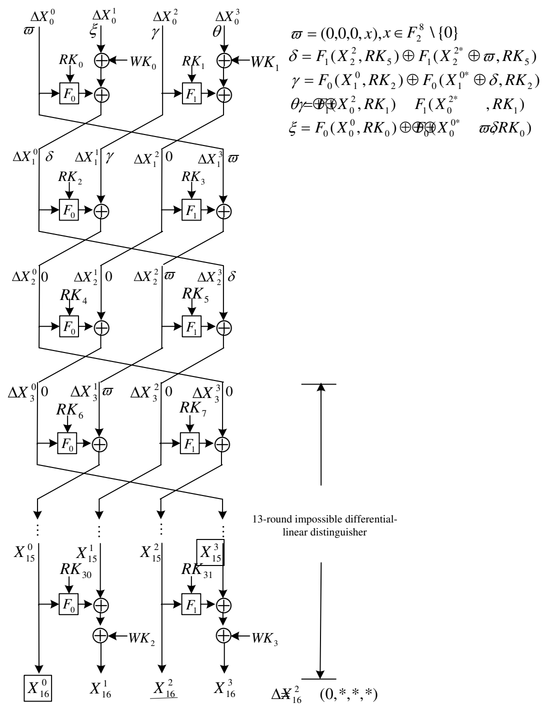


Fig. 5. 16-round impossible differential-linear attack

Appendix C. Attacks on 15-round CLEFIA-128

The attacks to 15-round CLEFIA-128 below are all with whitening keys. The details of the attack can be divided into two cases.

The first extension is one round on plaintext side, and one round on ciphertext side as illustrated in Fig.6. We can choose a structure composed of 2^{40} plaintexts, whose plaintext differences is of the form $\Delta P = (0, 0, \varpi, \delta)$. Obviously, one structure can produce about 2^{55} different plaintext pairs. Similar to the section 5.2, we can recover 64-bit subkey composed of RK_1^0(8bit), RK_{27}(24bit), and RK_{28}(32bit), with impossible differential-linear attack.

The data complexity is

$$2^{40} \times [8 \times (2^{60.26})^2 \Big/ (2^{55} \cdot 2^{-8} \cdot 2^{-8})] \approx 2^{124.52}$$

The time complexity is

$$[(2 \times 2^{32} \times 2^{32} \times 2^{32})/2 + ((2^8 \times 2^{32}) \times 2^8)/2]/15 \approx 2^{93.1}$$

The second extension is two rounds on plaintext side, illustrated in Fig. 7. We can choose a structure composed of 2^{72} plaintexts, whose plaintext differences is of the form $\Delta P = (\delta, \gamma, 0, \varpi)$. Obviously, one structure can produce about 2^{119} distinct plaintext pairs. Similar to section 5.2, we can recover 64-bit subkey, that is RK_0(32bit), RK_3^0(8bit), and RK_{29}(24bit), with impossible differential-linear attack.

The data complexity is

$$2^{72} \times [8 \times (2^{60.26})^2 \Big/ (2^{119} \cdot 2^{-8} \cdot 2^{-32} \cdot 2^{-8})] \approx 2^{124.52}$$

The time complexity is

$$[(2^{32} \times 2^{32})/2 + ((2^{32} \times 2^{32}) \times 2^{40} + (2^8 \times 2^{32}) \times 2^{40})/2]/15 \approx 2^{99.1}$$

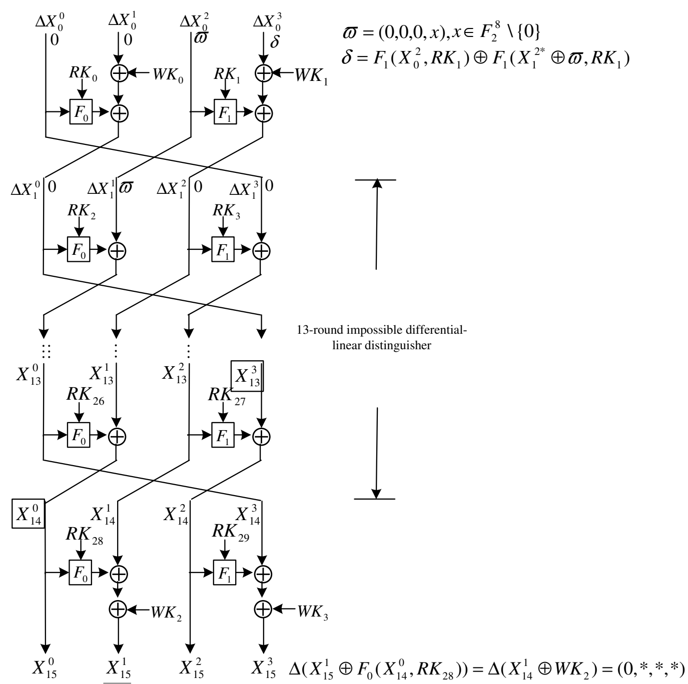


Fig. 6. 15-round impossible differential-linear attack

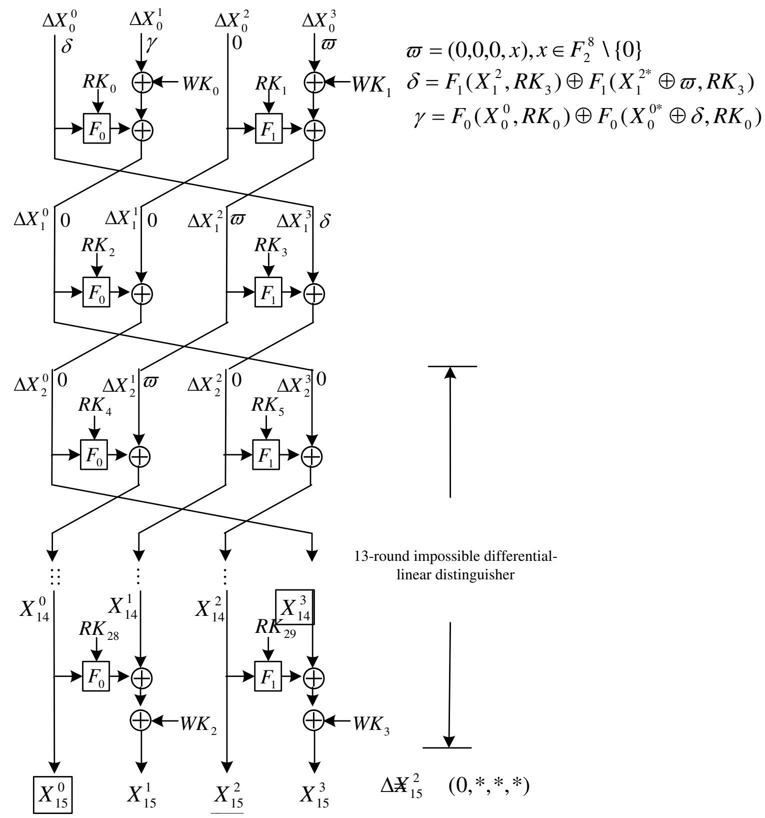


Fig. 7. 15-round impossible differential-linear attack

Appendix D. Round Key Relation

According to the description in section 2, we can get the relationship between generated round keys and related data as follows:

$RK_0\,|RK_1\,|RK_2\,|RK_3 \leftarrow L \oplus CON_{24}\,|CON_{25}\,|CON_{26}\,|CON_{27}$
$RK_4\,|RK_5\,|RK_6\,|RK_7 \leftarrow \Sigma(L) \oplus K \oplus CON_{28}\,|CON_{29}\,|CON_{30}\,|CON_{31}$
$RK_8\,|RK_9\,|RK_{10}\,|RK_{11} \leftarrow \Sigma^2(L) \oplus CON_{32}\,|CON_{33}\,|CON_{34}\,|CON_{35}$
$RK_{12}\,|RK_{13}\,|RK_{14}\,|RK_{15} \leftarrow \Sigma^3(L) \oplus K \oplus CON_{36}\,|CON_{37}\,|CON_{38}\,|CON_{39}$
$RK_{16}\,|RK_{17}\,|RK_{18}\,|RK_{19} \leftarrow \Sigma^4(L) \oplus CON_{40}\,|CON_{41}\,|CON_{42}\,|CON_{43}$
$RK_{20}\,|RK_{21}\,|RK_{22}\,|RK_{23} \leftarrow \Sigma^5(L) \oplus K \oplus CON_{44}\,|CON_{45}\,|CON_{46}\,|CON_{47}$
$RK_{32}\,|RK_{33}\,|RK_{34}\,|RK_{35} \leftarrow \Sigma^8(L) \oplus CON_{56}\,|CON_{57}\,|CON_{58}\,|CON_{59}$

Based on the properties proved in [5], we get the following key relations:

$RK_{32} \oplus C_1 = RK_1[56-63]\,|RK_3[100-102]\,|RK_3[107-127]$
$RK_{33} \oplus C_2 = RK_2[72-95]\,|RK_3[96-99]\,|RK_3[103-106]$
$RK_{34} \oplus C_3 = RK_0[21-24]\,|RK_0[28-31]\,|RK_1[32-55]$
$RK_{35} \oplus C_4 = RK_0[0-20]\,|RK_0[25-27]\,|RK_2[64-71]$

where

$C_1 = CON_{56} \oplus (CON_{25}[56-63]\,|CON_{27}[100-102]\,|CON_{27}[107-127]\,)$
$C_2 = CON_{57} \oplus (CON_{26}[72-95]\,|CON_{27}[96-99]\,|CON_{27}[103-106]\,)$
$C_3 = CON_{58} \oplus (CON_{24}[21-24]\,|CON_{24}[28-31]\,|CON_{25}[32-55]\,)$
$C_4 = CON_{59} \oplus (CON_{24}[0-20]\,|CON_{24}[25-27]\,|CON_{26}[64-71]\,)$

Thus we get the following properties from the above derivations:

Property 6. . If 32 bits RK_{33} are known, we can get 24 bits $RK_2[72-95]$, and 8 bits $RK_3[96-99]\,|RK_3[103-106]$.

Property 7. . If 32 bits RK_{34} are known, then we can get 8 bits $RK_0[21-24]\,|RK_0[28-31]$, and 24 bits $RK_1[32-55]$.