ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.9, September 2009, pp.2587−2596
doi: 10.3724/SP.J.1001.2009.00576

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

CLEFIA-128/192/256 的不可能差分分析*

王　薇[1]，王小云[1,2+]

[1](山东大学 密码技术与信息安全教育部重点实验室,山东 济南　250100)
[2](清华大学 高等研究中心,北京　100084)

Impossible Differential Cryptanalysis of CLEFIA-128/192/256

WANG Wei[1],　WANG Xiao-Yun[1,2+]

[1](Key Laboratory of Cryptologic Technology and Information Security (Ministry of Education), Shandong University, Ji'nan 250100, China)
[2](Institute for Advanced Study, Tsinghua University, Beijing 100084, China)
+ Corresponding author: E-mail: xiaoyunwang@mail.tsinghua.edu.cn

Wang W, Wang XY. Impossible differential cryptanalysis of CLEFIA-128/192/256. *Journal of Software*, 2009,20(9):2587−2596. http://www.jos.org.cn/1000-9825/576.htm

Abstract: An improved impossible differential attack on the block cipher CLEFIA is presented. CLEFIA was proposed by Sony Corporation at FSE 2007. Combining some observations with new tricks, the wrong keys are filtered out more efficiently, and the original impossible differential attack on 11-round CLEFIA-192/256 published by the designers, is extended to CLEFIA-128/192/256, with about $2^{103.1}$ encryptions and $2^{103.1}$ chosen plaintexts. By putting more constraint conditions on plaintext pairs, we present an attack on 12-round CLEFIA for all three key lengths with $2^{119.1}$ encryptions and $2^{119.1}$ chosen plaintexts. Moreover, a birthday sieve method is introduced to decrease the complexity of the precomputation. And an error about the time complexity evaluation in Tsunoo et al.'s attack on 12-round CLEFIA is pointed out and corrected.
Key words: block cipher; cryptanalysis; impossible differential cryptanalysis; birthday sieve; CLEFIA

摘　要: 对分组密码算法 CLEFIA 进行不可能差分分析.CLEFIA 算法是索尼公司在 2007 年快速软件加密大会(FSE)上提出来的.结合新发现和新技巧,可有效过滤错误密钥,从而将算法设计者在评估报告中给出的对 11 圈 CLEFIA-192/256 的攻击扩展到 11 圈 CLEFIA-128/192/256,复杂度为 $2^{103.1}$ 次加密和 $2^{103.1}$ 个明文.通过对明文附加更多限制条件,给出对 12 圈 CLEFIA-128/192/256 的攻击,复杂度为 $2^{119.1}$ 次加密和 $2^{119.1}$ 个明文.而且,引入一种新的生日筛法以降低预计算的时间复杂度.此外,指出并改正了 Tsunoo 等人对 12 圈 CLEFIA 的攻击中复杂度计算方面的错误.
关键词: 分组密码;密码分析;不可能差分分析;生日筛法;CLEFIA
中图法分类号: TP309　　**文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant No.60525201 (国家自然科学基金); the National Basic Research Program of China under Grant No.2007CB807902 (国家重点基础研究发展计划(973))

Received 2008-04-26; Accepted 2009-01-14

1 Introduction

Impossible differential cryptanalysis[1] is a sieving attack which considers a differential with probability 0. If a pair of plaintexts (or ciphertexts) 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. Impossible differentials depend on the basic structure of the block ciphers which are often used, and this method is a particular threat to the generalized Feistel structure.

CLEFIA[2,3] is a new 128-bit block cipher, 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. The fundamental structure of CLEFIA is a generalized Feistel structure consisting of 4 data lines. Sony claimed that the CLEFIA is designed to concentrate state-of-the-art cryptanalysis techniques, and achieves sufficient immunity against known cryptanalytic attacks.

Since CLEFIA was unveiled at Fast Software Encryption (FSE) 2007[2], there have been several papers on its security analysis. The security and performance evaluations[4] published by Sony Corporation examines its security against some well-known attacks, such as differential cryptanalysis, linear cryptanalysis, impossible differential cryptanalysis, etc. Differential fault analysis was presented in Ref.[5], which shows that only about 18 faulty ciphertexts are needed to recover the entire 128-bit secret key and about 54 faulty ciphertexts are enough for 192/256-bit key. References [6,7] propose impossible differential cryptanalysis on 12-round CLEFIA-128/192/256, 13-round CLEFIA-192/256, and 14-round CLEFIA-256, independently.

Using a structure-dependent 9-round impossible differential, the impossible differential attacks presented in Ref.[4] analyze the 10-round CLEFIA-128/192/256, 11-round CLEFIA-192/256, and 12-round CLEFIA-256 without key whitenings. Observing the inner structure of the F-functions, we conclude that the time complexity of these attacks can be decreased by some table lookups and sieving less subkey space. And a birthday sieve method is introduced to reduce the time complexity of the precomputation. By these observations, our attack on 11-round CLEFIA only takes $2^{103.1}$ encryptions and $2^{103.1}$ chosen plaintexts, instead of the original 2^{188} encryptions and $2^{103.5}$ chosen plaintexts. Moreover, combining with a special way to choose plaintext pairs, we show that attack on 12-round CLEFIA-128/192/256 takes $2^{119.1}$ time complexity and $2^{119.1}$ data complexity. Reference [6] explores the relations with the branch number of the matrices, and publishes some new 9-round impossible differences, of which the complexity of the attack on 11-round CLEFIA is $2^{118.8}$ chosen plaintexts and $2^{118.8}$ encryptions, and $2^{118.9}$ chosen plaintexts and $2^{118.9}$ encryptions for 12-round version. However, we found out that Ref.[6] neglects the time complexity of the precomputation (which will be explained later), so that the time complexity of the attack on 12-round CLEFIA is $2^{125.8}$ encryptions actually. Using a similar kind of birthday sieve method to choose plaintext pairs and doing the key recovery process as described in our attacks, we correct this mistake.

This paper is organized as follows: in Section 2, we give a brief description of CLEFIA. Section 3 summarizes some important observations on CLEFIA. We present the attacks applicable to 11-12 round CLEFIA with all three key variants, and correct the error of Ref.[6] in Section 4. Finally, Section 5 concludes this paper.

2 Description of CLEFIA

2.1 Notations

We first describe the notations used throughout this paper.

P or *P*′: A 128-bit plaintext;

C or *C*′: A 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, i=0,1,2,3;

ΔA: The XOR value of A and A', i.e., the value of $A \oplus A'$;

F_i^r : The function F_i involved in the r-th round, i=0,1;

InS_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-bit positions;

$A<<<x$: The rotation of A to the left by x-bit positions;

$a|b$: The concatenation of a and b;

a^T: The transposition of a vector a.

2.2 Data processing part of CLEFIA

CLEFIA[2,3] is a 128-bit block cipher with key length of 128, 192 and 256 bits. It employs a generalized Feistel structure with four data lines, where 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.

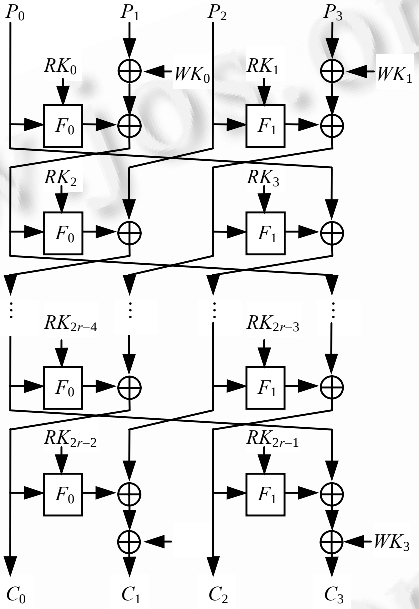

Fig.1 Encryption process of r-round CLEFIA

Let WK_0, WK_1, WK_2, $WK_3 \in \{0,1\}^{32}$ be whitening keys, and $RK_i \in \{0,1\}^{32}$ ($0 \le i < 2r$) 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 round number 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 following.

Denote the 32-bit output of F-function as T_i, where $T_i = T_{i,0} | T_{i,1} | T_{i,2} | T_{i,3}$, $T_{i,j} \in \{0,1\}^8$ (j=0,1,2,3). Then $F_0(C_0^{i-1}, RK_{2i-2})$ (1≤i≤r) is computed as follows (see Fig.2):

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

2) $T_{i,0} = S_0(T_{i,0}),\ T_{i,1} = S_1(T_{i,1}),\ T_{i,2} = S_0(T_{i,2}),\ T_{i,3} = S_1(T_{i,3})$.

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. The computation of $F_1(C_2^{i-1}, RK_{2i-1})$ (1≤i≤r) is similar to that of F_0, where S_0, S_1 and M_0 are replaced with S_1, S_0 and M_1, respectively (See Fig.2).

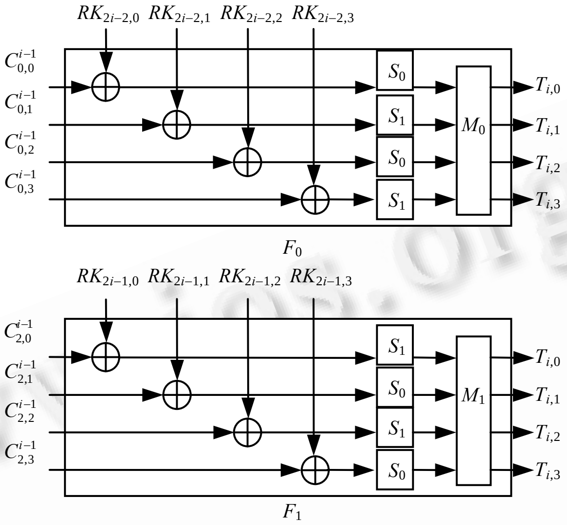

Fig.2 *F*-Functions

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 on CLEFIA which are the basis of our efficient attacks on reduced CLEFIA. Proposition 1 recalls the two 9-round impossible differentials presented in Ref.[4]. Our attacks utilize the same impossible differentials. However, we explore more technique details, such as Proposition 2 and 3, to achieve a prominent improvement. Independently, similar observations are used in Ref.[6].

Proposition 1 (**impossible differentials of 9-round CLEFIA**[4]). For 9-round CLEFIA, given a plaintext pair with difference $(0,\alpha,0,0)$ (or $(0,0,0,\alpha)$), where $\alpha \in \{0,1\}^{32}$ is any non-zero value, the output difference can't equal $(0,\alpha,0,0)$ (or $(0,0,0,\alpha)$). Denote the two 9-round impossible differentials as

$$(0,\alpha,0,0) \not\rightarrow (0,\alpha,0,0) \quad \text{and} \quad (0,0,0,\alpha) \not\rightarrow (0,0,0,\alpha).$$

The correctness of Proposition 1 can be verified easily.

By observing the inner structure of F-functions, we find that the time complexity of attacks in Ref.[4] can be decreased by fast searching the 32-bit subkeys involved in F-functions with the help of XOR distribution tables of S-boxes[8].

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

Proof: Because the diffusion matrix M is linear and invertible and ΔOut is known, we can easily compute the input difference of M, i.e., the output differences of four S-boxes. Therefore, for each S-box in F, we get the input XOR and the corresponding output XOR. Then it is easy to obtain the input of each S-box by searching the XOR distribution tables of the S-box. From the description of F-function, we can see that the concatenation of the four inputs to the four S-boxes is the XOR value of In and RK.

Thus, the 32-bit subkey RK can be derived from In. The time complexity is about one F-computation. □

Usually, the efficiency of the impossible differential attack depends on the subkey space related to the impossible differential. For 11-round CLEFIA-192/256, impossible differential attack[4] sieves 128-bit subkeys involved in rounds 10 and 11. The following proposition is an important phenomenon that can be used to sieve only 96-bit subkey instead of 128-bit.

Proposition 3. For r-round CLEFIA, let RK_{2r-3} and RK_{2r-4} be subkeys in the (r−1)-th round, RK_{2r-1} and RK_{2r-2} be subkeys in the r-th round, WK_2 and WK_3 be the whitening keys in the final round, and $C^r=(C_0^r\,|\,C_1^r\,|\,C_2^r\,|\,C_3^r)$ be the ciphertext, the following two equations reveal the correlations among subkeys WK_2, WK_3, RK_{2r-3} and RK_{2r-4}

$$WK_3\oplus RK_{2r-4}=InS_0^{r-1}\oplus F_1^r(C_2^r,RK_{2r-1})\oplus C_3^r \tag{1}$$

$$WK_2\oplus RK_{2r-3}=InS_1^{r-1}\oplus F_0^r(C_0^r,RK_{2r-2})\oplus C_1^r \tag{2}$$

Here, InS_0^{r-1} and InS_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 $C_2^r=C_2^{r-1}$ and $C_3^r=C_3^{r-1}\oplus F_1^r(C_2^{r-1},RK_{2r-1})\oplus WK_3$. 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 $C_3^{r-1}=C_0^{r-2}$ and $InS_0^{r-1}=C_0^{r-2}\oplus RK_{2r-4}$, we know that

$$C_3^r=C_0^{r-2}\oplus F_1^r(C_2^r,RK_{2r-1})\oplus WK_3=InS_0^{r-1}\oplus RK_{2r-4}\oplus F_1^r(C_2^r,RK_{2r-1})\oplus WK_3$$

i.e., $WK_3\oplus RK_{2r-4}=InS_0^{r-1}\oplus F_1^r(C_2^r,RK_{2r-1})\oplus C_3^r$.

Similarly, we can prove that Eq.(2) holds. □

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

$$WK_0\oplus RK_2=InS_0^2\oplus F_0^1(P_0,RK_0)\oplus P_1 \text{ and } WK_1\oplus RK_3=InS_1^2\oplus F_1^1(P_2,RK_1)\oplus P_3.$$

4 Impossible Differential Attacks on CLEFIA-128/192/256

This section presents impossible differential attacks on 11-12 rounds CLEFIA based on the 9-round impossible differentials in Ref.[4], and corrects the mistake of the attack on 12-round variant presented in Ref.[6]. The main attack process is: Firstly, select many structures of specific 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 Attack on 11-round CLEFIA

This section describes the key recovery attack on 11-round CLEFIA with two additional rounds at the end of the 9-round impossible differential as the preparation for attacks in the following sections. For simplicity of explanation, we regard the first-round output as plaintext and present the attack procedure for the 11-round from the second to the 12-th round. We use the same 9-round impossible differential $(0,\alpha,0,0)\not\rightarrow(0,\alpha,0,0)$ in Sections 4.1 and 4.2. Different from Ref.[4], the attack recovers the 96-bit subkey (RK_{22},RK_{23},$RK_{20}\oplus WK_3$) by Proposition 3 instead of recovering the 128-bit subkey (RK_{20},RK_{22},RK_{23},WK_3). Combining with Proposition 2, the total time complexity can be improved from 2^{188} encryptions to $2^{103.1}$ encryptions with $2^{103.1}$ chosen plaintexts. And the result is better than Ref.[6], which needs about $2^{118.8}$ encryptions and $2^{118.8}$ chosen plaintexts. See Fig.3 for the following attack.

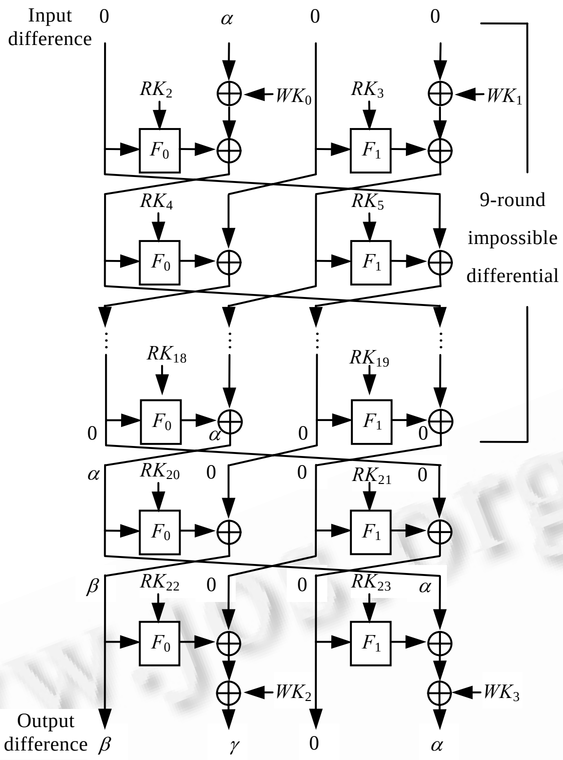

Fig.3 Impossible differential attack on 11-round CLEFIA

Sieving pairs.

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

$S=\{P_0,P_1\oplus\alpha,P_2,P_3|P_0,P_1,P_2,P_3$ are fixed, non-zero $\alpha\in\{0,1\}^{32}\}$.

By the encryption process of CLEFIA, only the plaintext pair with ciphertext difference $\Delta C=(\beta,\gamma,0,\alpha)$ may result from $\Delta C^{10}=(\alpha,0,0,0)$, 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 $\hat{C}=(P>>>64)\oplus C$, where C is the corresponding ciphertext.

2) Store the 2^{32} values of $\hat{C}$ in a table.

3) Search (P,P') with the corresponding $\Delta\hat{C}=(\beta,\gamma,0,0)$ by the birthday attack.

4) Output (P,P').

It is clear that $\Delta C=(\beta,\gamma,0,\alpha)$ if and only if $\Delta\hat{C}=(\beta,\gamma,0,0)$. So, the above algorithm outputs one plaintext pair corresponding to $\Delta C=(\beta,\gamma,0,\alpha)$ with probability 1/2. From the birthday attack[9], 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_{22},RK_{23},$RK_{20}\oplus WK_3$).

We discard the subkeys which cause the partial decryption of the selected pair to match $\Delta C^{10}=(\alpha,0,0,0)$.

For each pair with ciphertext difference $\Delta C=(\beta,\gamma,0,\alpha)$, it is obvious that $\Delta C_0^{10}=\Delta C_3^{11}=\Delta C_3^{12}=\alpha$.

From $C_3^{10}=C_2^{11}\oplus F_1^{11}(C_2^{10},RK_{21})$, $C_2^{11}=C_2^{12}$ and $\Delta C_2^{12}=0$, it is clear that $\Delta C_3^{10}=0$ if and only if $\Delta C_2^{10}=0$.

Thus, we only need to discard the subkeys which lead to $\Delta C_1^{10}=0$ and $\Delta C_2^{10}=0$.

For each ciphertext pair (C,C') with $\Delta C=(\beta,\gamma,0,\alpha)$, we can detect 2^{32} wrong subkeys (RK_{22},RK_{23},$RK_{20}\oplus WK_3$) which suggest the impossible differential as follows:

1) For $\Delta C_2^{10}=0$, since $C_1^{11}=C_2^{10}$, it is equivalent to $\Delta C_1^{11}=0$.

We obtain $\Delta F_0^{12}=\Delta C_1^{12}$ by $C_1^{12}=C_1^{11}\oplus F_0^{12}(C_0^{11},RK_{22})\oplus WK_2$.

From $C_0^{11}=C_0^{12}$ and $C_0'^{11}=C_0'^{12}$, the subkey RK_{22} can be calculated with one F-computation according to Proposition 2.

2) For $\Delta C_1^{10}=0$, we have $\Delta F_0^{11}=\Delta C_0^{12}$ by $C_0^{11}=C_1^{10}\oplus F_0^{11}(C_0^{10},RK_{20})$ and $C_0^{11}=C_0^{12}$.

Because the corresponding input XOR $\Delta C_0^{10}=\alpha$, InS_0^{11} is calculated by Proposition 2.

For each $RK_{23}\in\{0,1\}^{32}$, by Proposition 3, we can deduce that

$$RK_{20}\oplus WK_3=InS_0^{11}\oplus F_1^{12}(C_2^{12},RK_{23})\oplus C_3^{12}.$$

So, we totally obtain 2^{32} possible values of $RK_{20}\oplus WK_3$ with about 2^{32} F-computations.

Summing up 1) and 2), for each pair, we can filter out 2^{32} wrong subkeys (RK_{22},RK_{23},$RK_{20}\oplus WK_3$) which support the impossible differential in about 2^{32} F-computations. A wrong (RK_{22},RK_{23},$RK_{20}\oplus WK_3$) 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_{22},RK_{23},$RK_{20}\oplus WK_3$) is left. This completes our attack.

Complexity evaluation.

The data complexity of the attack is about $2^{70.1+32+1}=2^{103.1}$ chosen plaintexts. The time complexity for obtaining the ciphertexts is $2^{103.1}$ encryptions and the time complexity of sieving the right key 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 Attack on 12-round CLEFIA

We extend the attack on 11-round variant described above to 12-round, by one additional round on the plaintext side. However, the direct extension needs to recover the 32-bit subkey RK_1 in addition to RK_{22}, RK_{23} and $RK_{20}\oplus WK_3$, which will a little exceed the complexity of the exhaustive attack. Thus we put more constraint conditions on the plaintext difference to enforce the first two bytes of ΔC_2^1 and ΔC_1^{10} to be zero. In this way, instead of the original 128-bit subkey (RK_1,RK_{22},RK_{23},$RK_{20}\oplus WK_3$), there exists only 96-bit subkey ($RK_{1,2}$,$RK_{1,3}$, $RK'_{20,2}$, $RK'_{20,3}$, RK_{22}, RK_{23}), which is related to the impossible differential, where $RK'_{20,2}$ and $RK'_{20,3}$ denote the last two bytes of $RK_{20}\oplus WK_3$, respectively.

Sieving pairs.

For all the 2^{16} possible α, of which the first two bytes are zero, we compute a table H_1 to store the 2^{16} values of $M_1(\alpha)$ and a table H_0 to store the 2^{16} values of $M_0(\alpha)$. Because M_1 is linear, for $\delta_1,\delta_2\in H_1$, it is obvious that $\delta_1\oplus\delta_2\in H_1$. So is M_0.

Choose a structure of 2^{32} plaintexts as follows (See Fig.4):

$S=\{P_0,P_1,P_2\oplus\alpha,P_3\oplus\delta|P_0,P_1,P_2,P_3$ are fixed, the first two bytes of α are zero and the other two take 2^{16} possibilities,$\delta\in H_1\}$.

Fulfilling Algorithm 1 in Section 4.1, in which $\hat{C}$ is selected as (P>>>32)⊕C, we can easily search a pair such that ΔC=(β,γ,0,α), where $\beta\in\{0,1\}^{32}$ and $\gamma\in\{0,1\}^{32}$ are non-zero. Then choose the ciphertext pairs satisfying $\beta\in H_0$, of which the probability is 2^{-16}. Thus, $2^{-17}n$ such pairs can be found by searching n structures, where n is determined later.

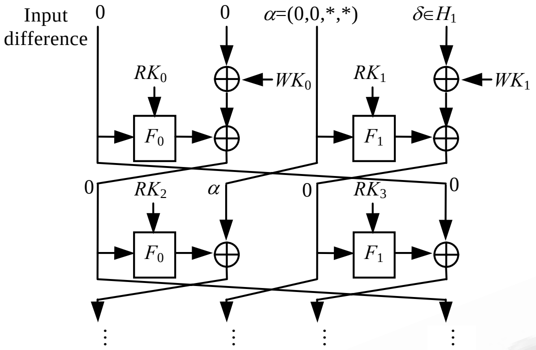

Fig.4 Choice of plaintext in 12-round attack

Recovering the subkey ($RK_{1,2}$,$RK_{1,3}$,RK_{22},RK_{23}, $RK'_{20,2}$, $RK'_{20,3}$).

For each selected pair, because the first two bytes of $M_1^{-1}(\delta)$ and 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. Therefore, only the last 16-bit ($RK_{1,2}$,$RK_{1,3}$) of RK_1 affects ΔC_2^1. Similarly, since the input XOR and output XOR of the first two S-boxes involved in F_0^{11} are zero, only the last 16-bit ($RK'_{20,2}$, $RK'_{20,3}$) affects ΔC_1^{10}. Thus, we only need to discard 96-bit wrong subkeys ($RK_{1,2}$, $RK_{1,3}$,RK_{22},RK_{23}, $RK'_{20,2}$, $RK'_{20,3}$) involved in the impossible differential in the following way.

1) To take advantage of the attack presented in Section 4.1, we need to guarantee that the output difference of the first round ΔC^1=(0,α,0,0). By $C_0^1=P_1\oplus F_0^1(P_0,RK_0)\oplus WK_0$ and ΔP=(0,0,α,δ), the selected pairs already satisfy $\Delta C_0^1=\Delta P_1=0$, $\Delta C_1^1=\Delta P_2=\alpha$ and $\Delta C_3^1=\Delta P_0=0$. So we only compute the 16-bit subkey ($RK_{1,2}$,$RK_{1,3}$) which cause the partial encryption of the pair to match $\Delta C_2^1=0$.

   From $C_2^1=F_1^1(P_2,RK_1)\oplus P_3\oplus WK_1$, 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 $\Delta F_1^1=\Delta P_3$. As the two inputs of F_1^1 are P_2 and P'_2 , one 16-bit subkey ($RK_{1,2}$,$RK_{1,3}$) can be computed with one F-computation on average by Proposition 2.

2) Then we can deduce 2^{32} wrong subkeys (RK_{22},RK_{23}, $RK'_{20,2}$, $RK'_{20,3}$) by the same method as that in Section 4.1. This step takes about 2^{32} F-computations.

To sum up, for each collected pair, we can filter out 2^{32} wrong 96-bit subkeys ($RK_{1,2}$,$RK_{1,3}$,RK_{22},RK_{23}, $RK'_{20,2}$, $RK'_{20,3}$) in about 2^{32} F-computations. The expected n is about $2^{17}\cdot2^{64}\cdot96\cdot\ln2\approx2^{87.1}$ by $2^{96}\cdot\left(1-\frac{2^{32}}{2^{96}}\right)^{2^{-17}n}<1$. Therefore, after analyzing $2^{87.1}\cdot2^{-17}=2^{70.1}$ pairs, only the right ($RK_{1,2}$,$RK_{1,3}$,RK_{22},RK_{23}, $RK'_{20,2}$, $RK'_{20,3}$) is left.

Complexity evaluation.

The data complexity is about $2^{32}\cdot n=2^{119.1}$ chosen plaintexts.

The time complexity of obtaining the ciphertexts is $2^{119.1}$ encryptions.

The time complexity of choosing the useful pairs is $2^{119.1}\div2^{32}\cdot2^{32}+2^{119.1}\div2^{32}\cdot2^{-1}\approx2^{119.1}$ XOR computations. Here, $2^{119.1}\div2^{32}$ is the number of structures. For each structure, we have to do 2^{32} XOR operations to apply the birthday sieve. Then for the $2^{119.1}\div2^{32}\cdot2^{-1}$ sieved pairs, we compute the XOR of them to choose the one we want.

The time complexity of sieving the right key is $2^{70.1}\cdot 2^{32}=2^{102.1}$ F-computations, which equals $2^{98.1}$ encryptions.

4.3 Analysis of Tsunoo et al's attack on 12-round CLEFIA

In Section 3.3 of Tsunoo, *et al*.'s attack[6], the authors present an impossible differential attack on 12-round CLEFIA, with complexity of $2^{118.9}$ chosen plaintexts and $2^{118.9}$ encryptions. However, they neglect the time complexity of the precomputation, i.e., the time complexity of choosing plaintext pairs, which is about $2^{125.9}$ encryptions. First, we explain this error, and then correct it using the birthday sieve method and key recovery process similar to Section 4.2.

4.3.1 An error in Tsunoo *et al*'s attack on 12-round CLEFIA

The authors use a new 9-round impossible differential $(0,0,0,(0,0,0,X)) \not\Rightarrow (0,0,0,(Y,0,0,0))$, where $(0,0,0,X)$ and $(Y,0,0,0)$ are 32-bit word, X and Y are non-zero bytes. Replacing the original impossible differential (in Fig.3) with the new one, we can easily see that the difference of ciphertexts pair must be $(0,(Y,0,0,0),\beta,\gamma)$ and the difference of the second round input must be $(0,0,0,(0,0,0,X))$. Here, β represents the 255 values that can be obtained as the output difference when the input difference for M_1 is $(Y,0,0,0)$, and γ is a 32-bit non-zero word.

The left is to choose the plaintext pairs of which the difference of the first-round output is $(0,0,0,(0,0,0,X))$. Ref.[6] does as follows:

1) Choose a set of 2^{40} plaintexts, where $\Delta P=((0,0,0,X)\delta,0,0)$, and δ is a non-zero 32-bit word.

2) For a chosen plaintexts set, guess $RK_{0,3}\oplus WK_{0,3}=RK'_{0,3}\in\{0,1\}^8$ to split the 2^{40} plaintexts set into 2^{32} structures. Each structure contains 2^8 plaintexts, of which any two satisfy $\Delta C^1=(0,0,0,(0,0,0,X))$.

It is obvious that, for each set and each guessed $RK'_{0,3}$, we have to do $2^{32}\cdot(2^8\cdot 2^7)=2^{47}$ XOR operations to select the proper pairs with $\Delta C=(0,(Y,0,0,0),\beta,\gamma)$. Thus, for all the 2^8 possible values of $RK'_{0,3}$ and the $2^{118.9}$ chosen plaintexts which is equivalent to $2^{78.9}$ sets, the time complexity of the plaintext choice method is about $2^8\cdot 2^{78.9}\cdot 2^{47}=2^{133.9}$ XOR operations. Using rough equivalence of 2^8 XOR operations to one encryption, the $2^{133.9}$ XOR operations equals to about $2^{125.9}$ encryptions, which is larger than $2^{118.9}$ encryptions.

4.3.2 Correction of the error

We can correct this error by doing the key recovery attack as that in Section 4.2.

Sieving pairs.

For all the 2^8 possible $\alpha=(0,0,0,X)$, compute a table H'_1 to store the 2^8 values of $M_1(\alpha)$ and a table H'_0 to store the 2^8 values of $M_0(\alpha)$. Choose a structure of 2^{16} plaintexts as follows:

$$S=\{P_0\oplus(0,0,0,X),P_1\oplus\delta,P_2,P_3|P_0,P_1,P_2,P_3 \text{ are fixed},\delta\in H'_0\}.$$

Due to the birthday attack, for each structure, we can collect one pair, of which the first word difference of the ciphertext is zero, in 2^{16} XOR operations with probability 2^{-1}. For $2^{118.9}$ chosen plaintexts, we can get $2^{118.9}\div 2^{16}\cdot 2^{-1}=2^{101.9}$ such pairs. Then by simply compute the XOR values of any two ciphertexts, we sieve the ones with $\Delta C=(0,(Y,0,0,0),\beta,\gamma)$, where $\beta\in H'_1$. In this way, we can get $2^{101.9}\cdot 2^{-48}=2^{53.9}$ pairs. This precomputation takes about $2^{118.9}\div 2^{16}\cdot 2^{16}+2^{101.9}\approx 2^{118.9}$ XOR operations $<2^{118.9}$ encryptions.

Recovering the subkey $(RK_{0,3},RK_{22},RK_{23},RK'_{21,0})$.

This part is similar to Ref.[6], except that we compute the value of $RK'_{0,3}$ instead of guessing it, which is just as what done in Section 4.2. Therefore, for each collected pair, we can filter out 2^{32} wrong 80-bit subkeys $(RK'_{0,3}, RK_{22},RK_{23},RK'_{21,0})$ in about 2^{32} F-computations. After analyzing $2^{53.9}$ pairs, only the right $(RK'_{0,3},RK_{22},RK_{23},RK'_{21,0})$ is left.

By the above method, the data and time complexity of the 12-round attack in Ref.[6] is $2^{118.9}$ chosen plaintexts and $2^{118.9}$ encryptions, which is the same as they announced.

5 Conclusions

In this paper, we present a chosen-plaintext attack on reduced CLEFIA-128/192/256. Table 1 shows the comparison among the attacks in Refs.[2,4,6] and ours. Note that there is an error in 12-round attack of Ref.[6], so we mark the time complexity of them with “*”. We explore some observations and tricks to extend the impossible differential attack to 11-12 rounds CLEFIA-128/192/256 based on the impossible differentials presented in Refs.[2,4], which only cryptanalyze 10-round variant. Moreover, a birthday sieve method is introduced to greatly reduce the time complexity of the precomputation. Finally, we point out the error in Ref.[6], correct it with birthday sieve to choose plaintext pairs, and present a different key recovery process.

Table 1 Summary of impossible differential attacks on reduced CLEFIA-128/192/256

	Refs.[2,4]		Ref.[6]		This paper	
Number of rounds	10	11	12	11	12	12^1
Data complexity	$2^{101.7}$	$2^{118.8}$	$2^{118.9}$	$2^{103.1}$	$2^{119.1}$	$2^{118.9}$
Time complexity	$2^{101.7}$	$2^{118.8}$	$2^{118.9}$*	$2^{103.1}$	$2^{119.1}$	$2^{118.9}$

1: We correct Tsunoo, *et al*.’s attack on 12-round CLEFIA[6]

Acknowledgement We thank the anonymous referees for their useful comments.

References:

[1] Biham E, Biryukov A, Shamir A. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Stern J, ed. Advances in Cryptology—EUROCRYPT’99. LNCS 1592, Berlin: Springer-Verlag, 1999. 12−23.

[2] Shirai T, Shibutani K, Akishita T, Moriai S, Iwata T. The 128-bit blockcipher CLEFIA (extended abstract). In: Biryukov A, ed. Proc. of the Fast Software Encryption (FSE 2007). LNCS 4593, Berlin: Springer-Verlag, 2007. 181−195.

[3] Sony Corporation. The 128-bit blockcipher CLEFIA: Algorithm specification. Revision 1.0, On-Line document, 2007. http://www.sony.net/Products/clefia/technical/data/clefia-spec-1.0.pdf

[4] Sony Corporation. The 128-bit blockcipher CLEFIA: Security and performance evaluations. Revision 1.0, On-Line document, 2007. http://www.sony.co.jp/Products/clefia/technical/data/clefia-eval-1.0.pdf

[5] Chen H, Wu WL, Feng DG. Differential fault analysis on CLEFIA. In: Qing S, Imai H, Wang G, eds. Proc. of the Int’l Conf. on Information and Communications Security (ICICS 2007). LNCS 4861, Berlin: Springer-Verlag, 2007. 284−295.

[6] Tsunoo Y, Tsujihara E, Shigeri M, Saito T, Suzaki T, Kubo H. Impossible differential cryptanalysis of CLEFIA. In: Nyberg K, ed. Proc. of the Fast Software Encryption (FSE 2008). LNCS 5086, Berlin: Springer-Verlag, 2008. 398−411.

[7] Wang W, Wang XY. Improved impossible differential cryptanalysis of CLEFIA. Report 2007/466. Cryptology ePrint Archive, 2007. http://eprint.iacr.org/2007/466

[8] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 1991,4(1):3−72.

[9] Menezes AJ, Oorschot PC, Vanstone SA. Handbook of Applied Cryptography. 5th ed., CRC Press, 2001. 369−370.

WANG Wei was born in 1983. She is a Ph.D. candidate at the Shandong University. Her current research areas are cryptanalysis of block cipher and hash function.

WANG Xiao-Yun was born in 1966. She is a professor and doctoral supervisor at the Tsinghua University and Shandong University. Her research areas are cryptography theory and technique.