

New Impossible Differential Attacks of Reduced-Round Camellia-192 and Camellia-256*

Jiazhe Chen^{1,2}, Keting Jia³, Hongbo Yu⁴, and Xiaoyun Wang^{1,2,3**}

¹ Key Laboratory of Cryptologic Technology and Information Security,
Ministry of Education, Shandong University, Jinan 250100, China

² School of Mathematics, Shandong University, Jinan 250100, China
jiazhechen@mail.sdu.edu.cn

³ Institute for Advanced Study, Tsinghua University, Beijing 100084, China

⁴ Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
{ktjia,yuhongbo,xiaoyunwang}@mail.tsinghua.edu.cn

Abstract. Camellia is a block cipher selected as a standard by ISO/IEC, which has been analyzed by a number of cryptanalysts. In this paper, we propose several 6-round impossible differential paths of Camellia with the FL/FL^{-1} layer in the middle of them. With the impossible differential and a well-organized precomputational table, impossible differential attacks on 10-round Camellia-192 and 11-round Camellia-256 are given, and the time complexity are 2^{175} and $2^{206.8}$ respectively. An impossible differential attack on 15-round Camellia-256 without FL/FL^{-1} layers and whitening is also be given, which needs about $2^{236.1}$ encryptions. To the best of our knowledge, these are the best cryptanalytic results of Camellia-192/-256 with FL/FL^{-1} layers and Camellia-256 without FL/FL^{-1} layers to date.

Key words: Camellia Block Cipher, Cryptanalysis, Impossible Differential Path, Impossible Differential Attack.

1 Introduction

Block cipher Camellia is proposed by NTT and Mitsubishi in 2000 [1]. Its block size is 128 bits and it supports 128-, 192- and 256-bit key sizes with 18, 24 and 24 rounds respectively. Camellia was selected as an e-government recommended cipher by CRYPTREC [5] and recommended in NESSIE [15] block cipher portfolio. Then it was selected as an international standard by ISO/IEC.

The structure of Camellia is Feistel structure with FL/FL^{-1} layers inserted every 6 rounds. The FL and FL^{-1} functions are keyed linear functions which are designed to provide non-regularity across rounds [1]. In the past years, Camellia has attracted the attention of the cryptanalytic community. The square-type attacks are efficient to attack Camellia, which can be used to analysis 9-round Camellia-128 and 10-round Camellia-256 [11]. Furthermore, Hatano et al. used the higher order differential attack to analysis the last 11 rounds Camellia-256 with complexity $2^{255.6}$ [7].

There are a number of results on simple versions of Camellia which exclude the FL/FL^{-1} layers and whitening being given in recent years [6,10,13,14,16,17,18,19]. Among them, the impossible differential attacks [3] are most efficient [13,14,17,18]. Since the existence of FL/FL^{-1} layers will probably destroy the impossibility, non of the impossible differential paths in these attacks includes the FL/FL^{-1} layers. In this paper, we present 6-round impossible differential paths with FL/FL^{-1} layers in the middle, which turn out to be first impossible differential

* Supported by 973 Project (No.2007CB807902), the National Natural Science Foundation of China (Grant No.60931160442) and Graduate Independent Innovation Foundation of Shandong University (No. 11140070613183).

** Corresponding author

paths with FL/FL^{-1} layers. Due to one of these impossible differential paths and a precomputational table that is carefully constructed, we propose impossible differential attacks on 10-round Camellia-192 and 11-round (Round 1-11) Camellia-256 with complexity 2^{175} and $2^{206.8}$ respectively.

For the attacks of Camellia-256 without FL/FL^{-1} layers and whitening, the 14-round attack in [13] was pointed out to be incorrect by [20]. Later Mala *et al.* [14] pointed out a flaw in [20] and showed that the time complexities of the 12-round Camellia-128 and 16-round Camellia-256 attacks were more than exhaustive search. As a result, the best analysis of Camellia-256 without FL/FL^{-1} layers and whitening dated back to [12], which was a 13-round attack with complexity $2^{211.7}$. By carefully using the subkey relations and one of the 8-round impossible differential paths without FL/FL^{-1} layers proposed in [18], we also present an impossible differential attack on 15-round Camellia-256 without FL/FL^{-1} layers and whitening, and the complexity is about $2^{236.1}$ encryptions.

The rest of this paper is organized as follows. We give some notations and a brief description of Camellia in Section 2. Some properties and 6-round impossible differential paths with FL/FL^{-1} layers of Camellia are given in Section 3. Section 4 describes the impossible differential attacks on reduced-round Camellia with FL/FL^{-1} layers and whitening. The impossible differential attack on 15-round Camellia-256 without FL/FL^{-1} layers and whitening is illustrated in Section 5. Finally, we conclude the paper in Section 6.

2 Preliminaries

Some notions used in this paper and a simple description of the Camellia algorithm are given in this section.

2.1 Notations

L^{r-1}, L'^{r-1} : the left half of the 128-bit r -th round input
 R^{r-1}, R'^{r-1} : the right half of the 128-bit r -th round input
 ΔL^{r-1} : the difference of L^{r-1}, L'^{r-1}
 ΔR^{r-1} : the difference of R^{r-1}, R'^{r-1}
 S^r, S'^r : the output value of the S-box of the r -th round
 ΔS^r : the output difference of the S-box of the r -th round
 k^r : the 64-bit r -th round subkey,
 A_i : the i -th byte of a 64-bit value A ($i = 1, \dots, 8$)
 $B \lll j$: left rotation of B by j bits
 $X_{L(64)}$: the left half of a 128-bit word X
 $X_{R(64)}$: the right half of a 128-bit word X
 $Y_{L(32)}$: the left half of a 64-bit word Y
 $Y_{R(32)}$: the right half of a 64-bit word Y
 $||$: the cascade of two words

2.2 The Camellia Algorithm

Camellia [1] is a 128-bit block cipher with Feistel structure. It has 18 rounds for 128-bit key, and 24 rounds for 192-/256-bit key. We give the encryption procedure of Camellia-192/-256 as follows, see Fig. 1.

Encryption Procedure. The input of the encryption procedure is a 128-bit plaintext M , and 64-bit subkeys k^{wi} ($i = 1, \dots, 4$), k^r ($r = 1, \dots, 24$) and kl^j ($j = 1, \dots, 6$). First M is XORed with

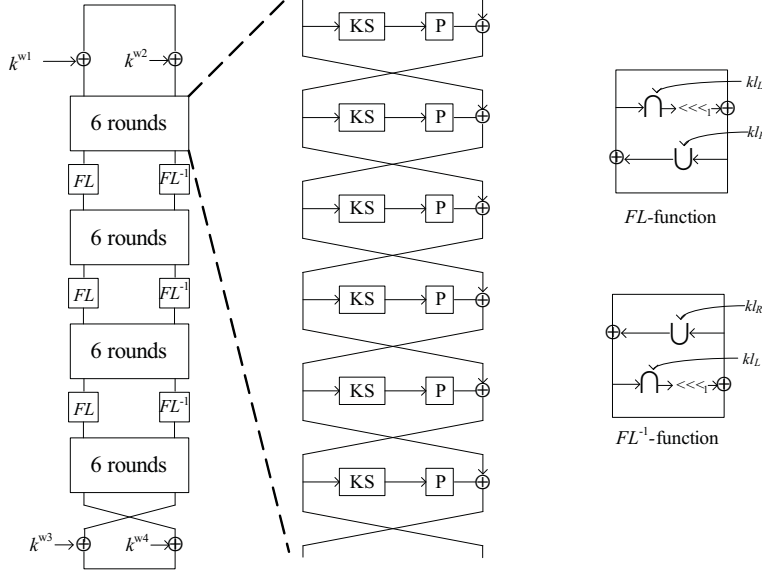


Fig. 1. Camellia-192/-256

k^{w1} and k^{w2} to get two 64-bit intermediate value L^0 and R^0 : $L^0 || R^0 = M \oplus (k^{w1} || k^{w2})$. Then the following operations are carried out for $i = 1$ to 24, except for $r = 6, 12$ and 18 :

$$L^r = R^{r-1} \oplus F(L^{r-1}, k^r), R^r = L^{r-1}.$$

For $r = 6, 12$ and 18 , do the following:

$$\begin{aligned} L^{r'} &= R^{r-1} \oplus F(L^{r-1}, k^r), R^{r'} = L^{r-1}. \\ L^r &= FL(L^{r'}, kl^{2r/6-1}), R^r = FL^{-1}(R^{r'}, kl^{2r/6}). \end{aligned}$$

Finally the 128-bit ciphertext C is computed as: $C = (R^{24} || L^{24}) \oplus (k^{w3} || k^{w4})$.

The FL function is defined as: $(X_{L(32)} || X_{R(32)}, kl_{L(32)} || kl_{R(32)}) \mapsto (Y_{L(32)} || Y_{R(32)})$, where:

$$\begin{aligned} Y_{R(32)} &= ((X_{L(32)} \cap kl_{L(32)}) \lll 1) \oplus X_{R(32)}, \\ Y_{L(32)} &= (Y_{R(32)} \cup kl_{R(32)}) \oplus X_{L(32)}. \end{aligned}$$

The FL^{-1} function is the inverse of FL function, and FL and FL^{-1} are linear as long as the key is fixed [2].

The round function F is composed of the key-addition layer, S-box layer and linear transformation P . In the key-addition layer, the input of the round function is XORed with the subkey. There are $4 \times 8 \times 8$ S-boxes S_1, S_2, S_3, S_4 used in the S-box layer, and each S-box is used twice. Finally, the linear transformation $P : (\{0, 1\}^8)^8 \rightarrow (\{0, 1\}^8)^8$ maps $(z_1, \dots, z_8) \rightarrow (y_1, \dots, y_8)$. P function and its inverse function P^{-1} are:

$$\begin{aligned} y_1 &= z_1 \oplus z_3 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8 & z_1 &= y_2 \oplus y_3 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_8 \\ y_2 &= z_1 \oplus z_2 \oplus z_4 \oplus z_5 \oplus z_7 \oplus z_8 & z_2 &= y_1 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8 \\ y_3 &= z_1 \oplus z_2 \oplus z_3 \oplus z_5 \oplus z_6 \oplus z_8 & z_3 &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_8 \\ y_4 &= z_2 \oplus z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7 & z_4 &= y_1 \oplus y_2 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_7 \\ y_5 &= z_1 \oplus z_2 \oplus z_6 \oplus z_7 \oplus z_8 & z_5 &= y_1 \oplus y_2 \oplus y_5 \oplus y_7 \oplus y_8 \\ y_6 &= z_2 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_8 & z_6 &= y_2 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_8 \\ y_7 &= z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_8 & z_7 &= y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \\ y_8 &= z_1 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7 & z_8 &= y_1 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_8 \end{aligned}$$

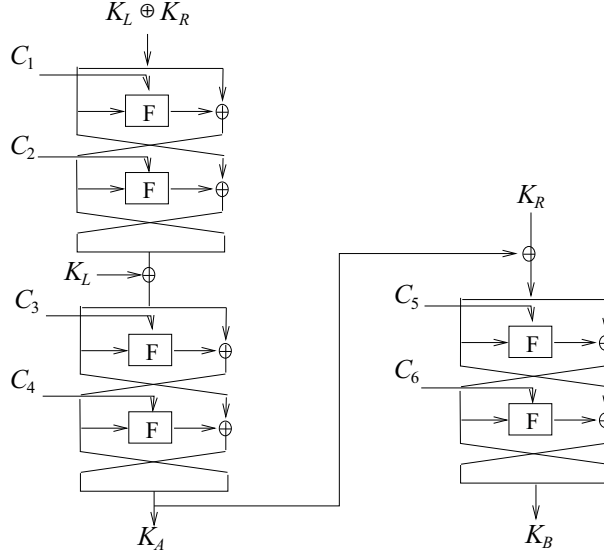


Fig. 2. The Calculation of K_A and K_B

Key Schedule. For Camellia-256, the 256-bit main key $K = K_L || K_R$, where K_L and K_R are 128 bits. And for Camellia-192, the 192-bit main key $K = K_L || K_{RL(64)}$ and $K_{RR(64)} = \overline{K_{RL(64)}}$. Using K_L and K_R , the key schedule algorithm first calculate K_A and K_B , which is described in Fig. 2. Where F is the round function of Camellia and C_i ($1 \leq i \leq 6$) are constants used as the keys. Then the subkeys k^{wi} ($i = 1, \dots, 4$), k^r ($r = 1, \dots, 24$) and kl^j ($j = 1, \dots, 6$) are derived from rotating K_L , K_R , K_A or K_B . For details of Camellia, we refer to [1].

It can be known from Fig. 2 that, if K_B and K_R are known, K_A is known. Therefore, one can get K_L using the relation between K_L and K_A described in [14], Section 3.2. So once K_B and K_R are known, K can be computed.

3 Properties and 6-Round Impossible Differential Paths of Camellia with FL/FL^{-1} Functions

In this section, we first give some useful properties of Camellia and then propose several impossible differential paths.

Property 1 For a 3-round Camellia structure, if the input difference is of the form $\Delta L^i = (0, a, 0, 0, 0, 0, 0, 0)$, $\Delta R^i = (0, 0, 0, 0, 0, 0, 0, 0)$, then:
 $\Delta L^{i+1} = (0, b, b, b, b, 0, 0)$, $\Delta S^{i+2} = (0, b_2, b_3, b_4, b_5, b_6, 0, 0)$,
 $\Delta L^{i+2} = \Delta R^{i+3} = P(a, b_2, b_3 \oplus a, b_4 \oplus a, b_5 \oplus a, b_6 \oplus a, 0, 0)$,
and $\Delta S_l^{i+3} = (P^{-1}(\Delta L^{i+3}))_l$, for $l = 1, 3, 4, \dots, 8$, where $a, b, b_2, b_3, b_4, b_5, b_6$ are non-zero bytes.

Property 2 The necessary conditions of $\Delta L^{i+3} = (0, a, 0, 0, 0, 0, 0, 0)$ and $\Delta R^{i+3} = (0, 0, 0, 0, 0, 0, 0, 0)$ are:
 $\Delta L^{i+1} = (0, b, b, b, b, 0, 0)$, $\Delta S^{i+2} = (0, b_2, b_3, b_4, b_5, b_6, 0, 0)$,
 $\Delta L^i = P(a, b_2, b_3 \oplus a, b_4 \oplus a, b_5 \oplus a, b_6 \oplus a, 0, 0)$,
and $\Delta S_l^{i+1} = (P^{-1}(\Delta R^i))_l$, for $l = 1, 3, 4, \dots, 8$, where $a, b, b_2, b_3, b_4, b_5, b_6$ are non-zero bytes.

To better describe the properties, we also illustrate them in Fig. 3. Actually, the proofs of the properties are similar and the proof Property 1 is given as an example.

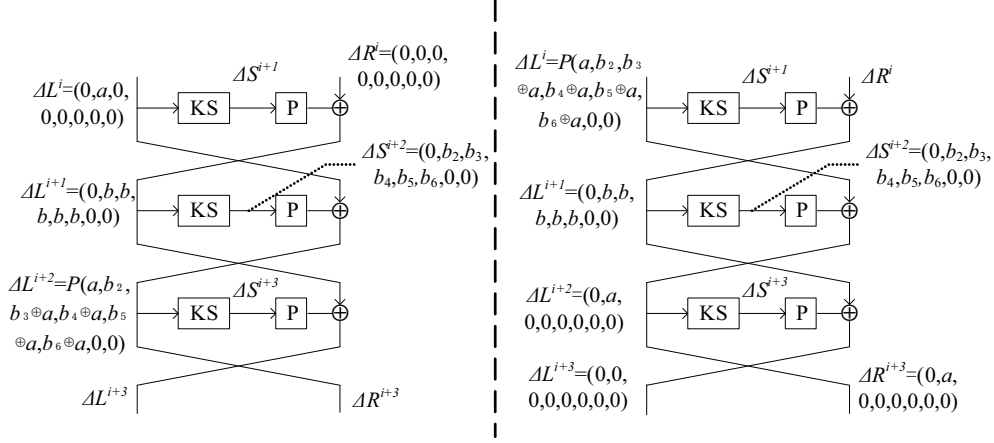


Fig. 3. Properties of 3-round Camellia

Proof. Apparently, ΔS^{i+1} is of the form $(0, b, 0, 0, 0, 0, 0, 0)$, where b is an unknown non-zero byte. And $\Delta L^{i+1} = (0, b, b, b, b, b, 0, 0)$ as P function is linear. After the key-addition layer and S-box layer, it can be obtained that $\Delta S^{i+2} = (0, b_2, b_3, b_4, b_5, b_6, 0, 0)$, where b_2, b_3, b_4, b_5 and b_6 are unknown non-zero bytes.

Since $\Delta L^{i+2} = \Delta S^{i+2} \oplus \Delta L^i$ and $P^{-1}(\Delta L^i) = (a, 0, a, a, a, a, 0, 0)$,

$$\Delta L^{i+2} = P(a, b_2, b_3 \oplus a, b_4 \oplus a, b_5 \oplus a, b_6 \oplus a, 0, 0).$$

Finally, because $\Delta S^{i+3} = P^{-1}(\Delta L^{i+1} \oplus \Delta L^{i+3})$, $P^{-1}(\Delta L^{i+1}) = (0, b, 0, 0, 0, 0, 0, 0)$ and P^{-1} function is linear,

$$\Delta S_l^{i+3} = (P^{-1}(\Delta L^{i+3}))_l, \text{ for } l = 1, 3, 4, \dots, 8. \quad \square$$

Property 3 (from [9]) Let x, x^* be 32-bit values, and $x' = x \oplus x^*$, then the differential properties of AND and OR operations are:

$$\begin{aligned} (x \cap k) \oplus (x^* \cap k) &= (x \oplus x^*) \cap k = x' \cap k \\ (x \cup k) \oplus (x^* \cup k) &= (x \oplus k \oplus (x \cap k)) \oplus (x^* \oplus k \oplus (x^* \cap k)) = x' \oplus (x' \cap k) \end{aligned}$$

Property 4 Let $M = (m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8)$ be the input difference of FL function, and $N = (n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8)$ be the output difference of FL, where n_l, m_l ($l = 1, \dots, 8$) are arbitrary 8-bit values. Then if $n_i = 0$ ($i \in \{5, 6, 7, 8\}$), $n_{i-4} = m_{i-4}$.

Proof. Let us denote the subkey used for AND operation as k_L and the subkey used for OR operation as k_R . By Property 3, the following equations must hold:

$$\begin{aligned} ((M_L \cap k_L) \lll 1) \oplus M_R &= N_R \\ M_L \oplus N_R \oplus (N_R \cap k_R) &= N_L \end{aligned} \quad (1)$$

Then if $n_i = 0$ ($i \in \{5, 6, 7, 8\}$), it can be deduced from Equation (1) that $n_{i-4} = m_{i-4}$. \square

Impossible Differential. Now we demonstrate that the 6-round differential in Fig. 4 is impossible. The input difference is

$$((0, 0, 0, 0, 0, 0, 0, 0); (0, a, 0, 0, 0, 0, 0, 0)),$$

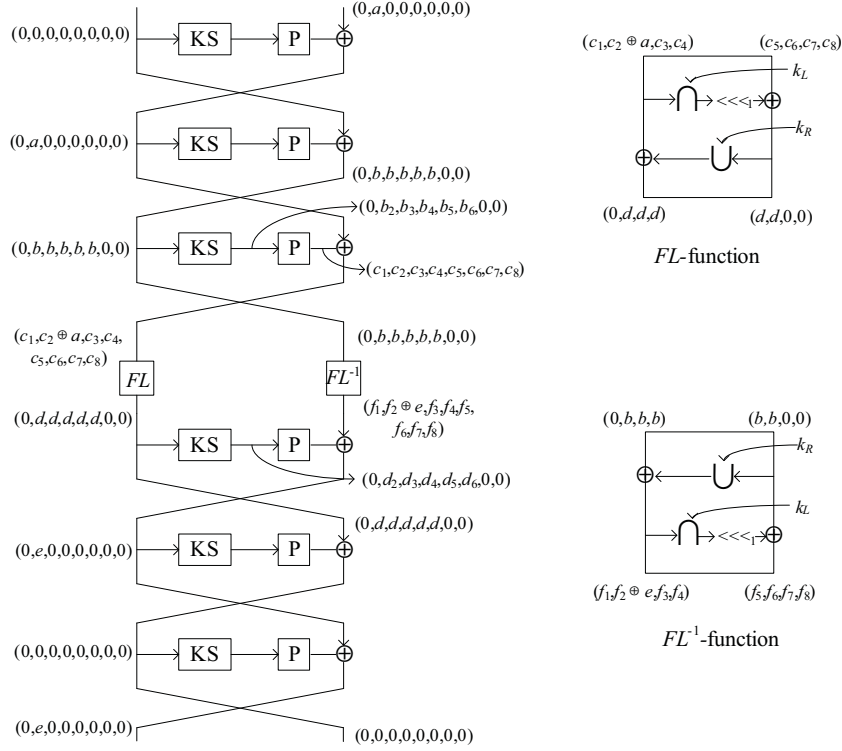


Fig. 4. 6-round impossible differential path with the FL/FL^{-1} layer in the middle

where a is arbitrary non-zero byte. The output difference of the first round is

$$((0, a, 0, 0, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0)).$$

Then by Property 1, the output differences of the second and third round are

$$((0, b, b, b, b, b, 0, 0); (0, a, 0, 0, 0, 0, 0, 0)) \text{ and } ((c_1, c_2 \oplus a, c_3, c_4, c_5, c_6, c_7, c_8); (0, b, b, b, b, b, 0, 0))$$

with probability 1, as long as

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) = P(0, b_2, b_3, b_4, b_5, b_6, 0, 0),$$

where $b, b_2, b_3, b_4, b_5, b_6$ are unknown non-zero bytes, $(0, b_2, b_3, b_4, b_5, b_6, 0, 0)$ is evolved from $(0, b, b, b, b, b, 0, 0)$ after the S-box layer and c_l ($l = 1, \dots, 8$) are unknown bytes.

Similarly, in the backward direction, we know that for arbitrary non-zero byte e , if the output difference of the sixth round is

$$((0, e, 0, 0, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0)),$$

then the input difference of the forth round is

$$((0, d, d, d, d, d, 0, 0); (f_1, f_2 \oplus e, f_3, f_4, f_5, f_6, f_7, f_8)),$$

where d is an unknown non-zero byte and f_l ($l = 1, \dots, 8$) are unknown bytes.

Now the input and output differences of the FL function are determined. It can be deduced from Property 4 that $c_3 = d$ and $c_4 = d$, which means $c_3 = c_4$. But this implies $b_4 = 0$ as

$$\begin{aligned} c_3 &= b_2 \oplus b_3 \oplus b_5 \oplus b_6, \\ c_4 &= b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6, \end{aligned}$$

which contradict $b_4 \neq 0$. (By the input and output difference of FL^{-1} function, we can also deduce another contradiction that $d_4 = 0 \not\Rightarrow d_4 \neq 0$). As a result, the differential

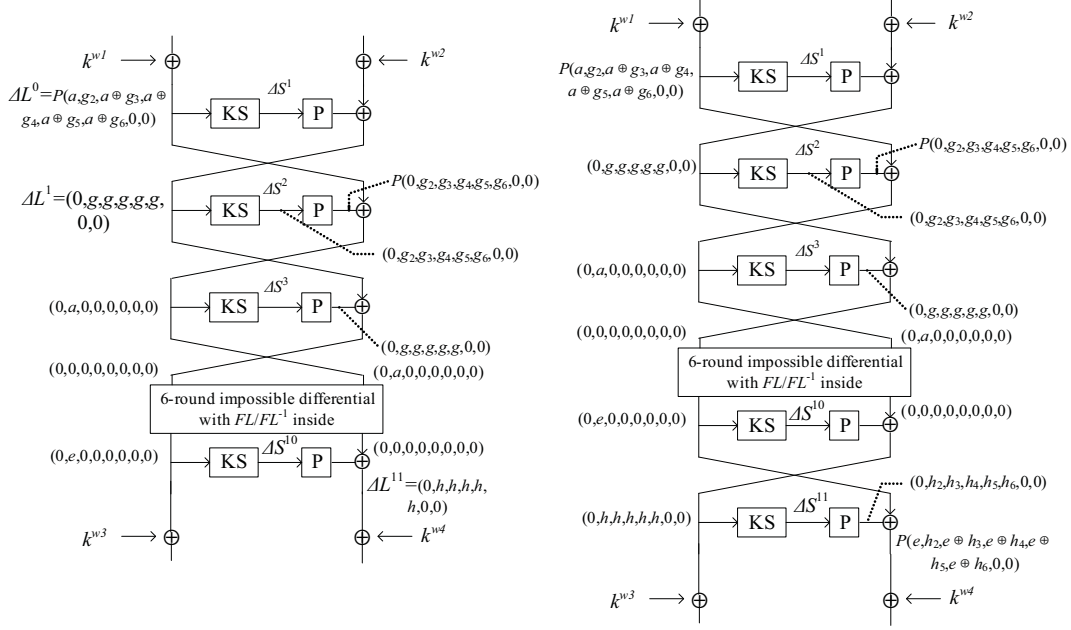


Fig. 5. Impossible Differential Attacks on 10-round Camellia-192 and 11-round Camellia-256 with whitening and FL/FL^{-1}

$$((0, 0, 0, 0, 0, 0, 0, 0); (0, a, 0, 0, 0, 0, 0, 0)) \xrightarrow{6\text{-round}} ((0, e, 0, 0, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0))$$

is impossible.

Actually, there are three more 6-round impossible differential paths with the FL/FL^{-1} layer in the middle, which are:

$$\begin{aligned} &((0, 0, 0, 0, 0, 0, 0, 0); (a, 0, 0, 0, 0, 0, 0, 0)) \xrightarrow{6\text{-round}} ((e, 0, 0, 0, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0)) \\ &((0, 0, 0, 0, 0, 0, 0, 0); (0, 0, a, 0, 0, 0, 0, 0)) \xrightarrow{6\text{-round}} ((0, 0, e, 0, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0)) \\ &((0, 0, 0, 0, 0, 0, 0, 0); (0, 0, 0, a, 0, 0, 0, 0)) \xrightarrow{6\text{-round}} ((0, 0, 0, e, 0, 0, 0, 0); (0, 0, 0, 0, 0, 0, 0, 0)) \end{aligned}$$

4 Impossible Differential Attack on Camellia with FL/FL^{-1} functions and whitening

In this section, we present impossible differential attacks on 11-round Camellia-256 and 10-round Camellia-192 using the impossible differential in Section 3.

4.1 Impossible Differential Attack on 11-Round Camellia-256

We add 3 rounds on the top and 2 rounds on the bottom of the 6-round impossible differential path to analysis 11-round Camellia-256, see Fig. 5 in the right. Denote $k^a = k^{w1} \oplus k^1$, $k^b = k^{w2} \oplus k^2$, $k^c = k^{w1} \oplus k^3$, $k^d = k^{w4} \oplus k^{10}$ and $k^e = k^{w3} \oplus k^{11}$. The attack is started by carrying out a precomputation.

Precomputation. A precomputational table H for Rounds 2-3 is set up here, which contains the all possible pairs that can follow the differential in Rounds 2-3 and their corresponding subkeys k^b , k^c . This table can also be used for Rounds 10-11, as in the backward direction, the differences are the same as that of Rounds 2-3. The table is constructed as follows:

For every $(L^1, g, k^b, L_2^2, a, k_2^c)$, compute $L^1 = L^1 \oplus (0, g, g, g, g, g, 0, 0)$, $T = F(L^1, k^b)$, $T' = F(L^1, k^b)$, $\Delta T = T \oplus T'$ and sieve the ones satisfying $S(L_2^2 \oplus k_2^c) \oplus S(L_2^2 \oplus a \oplus k_2^c) = g$, where g and a are non-zero bytes. There are 2^{160} $(L^1, g, k^b, L_2^2, a, k_2^c)$, and 2^{152} of which remain after the sieve. Then insert (k^b, k_2^c) into the row indexed by $(L^1, g, \Delta T \oplus a, L_2^2 \oplus T_2)$. Because there are only 2^{40} ΔT which lead to 2^{48} $\Delta T \oplus a$, there are 2^{128} rows in H and each row contains 2^{24} 72-bit subkeys (k^b, k_2^c) . Consequently, the memory complexity of the table is about $2^{155.2}$ bytes and the time complexity of the precomputation is less than 2^{161} one round encryptions.

Data Collection. Choose 2^n structures of plaintexts, and each structure contains plaintexts with the following form:

$$(P(y_1, y_2, y_3, y_4, y_5, y_6, \alpha, \beta); (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8))$$

where y_i ($i = 1, \dots, 6$) and x_j ($j = 1, \dots, 8$) take all possible values and α, β are fixed in each structure. As a result, there are 2^{112} plaintexts in each structure and we can get $2^n \times 2^{112 \times 2 - 1} = 2^{n+223}$ plaintext pairs totally. For each of the pairs, $\Delta(P^{-1}(L^0))_7 = 0$, $\Delta(P^{-1}(L^0))_8 = 0$.

Ask for the encryptions of the plaintexts in each structure to get the corresponding ciphertexts, and keep the pairs whose ciphertext differences satisfy the following form by birthday attack:

$$((0, h, h, h, h, h, 0, 0); P(e, h_2, e \oplus h_3, e \oplus h_4, e \oplus h_5, e \oplus h_6, 0, 0)),$$

where e, h, h_2, h_3, h_4, h_5 and h_6 are non-zero bytes. So there are $2^{n+223-72} = 2^{n+151}$ pairs remaining.

Key Recovery. In the key recovery procedure, we use Property 2 and the precomputational table to discard the wrong keys.

1. Individually guess k_l^a ($l = 1, 3, \dots, 8$) and check whether the equation $\Delta S_l^1 = (P^{-1}(\Delta R^0))_l$ holds. About $2^{n+151} \times 2^{-56} = 2^{n+95}$ pairs will be kept. Next guess k_2^a , so (L^1, L^1) can be computed. For each of the remaining pairs, do Step 2.
2. Initialize a table Γ of 2^{144} all possible values (k^b, k_2^c, k^e, k_2^d) , for each of the remaining pairs, access the row $(L^1, \Delta L_2^1, \Delta R^1, R_2^1)$ and the row $(L^{11}, \Delta L_2^{11}, \Delta L^{12}, L_2^{12})$ in table H . Then combining the values in the two rows to get (k^b, k_2^c, k^e, k_2^d) , and remove the corresponding value from Γ .
3. If Γ is not empty, output the 208-bit value $(k^a, k^b, k_2^c, k_2^d, k^e)$, otherwise go to Step 1 and try another guess. The main key can be recovered when $(k^a, k^b, k_2^c, k_2^d, k^e)$ is obtained, which will be described as follows.

The following equations are deduced from Table 3 in [1]:

$$k^a = (K_L \lll 0)_L \oplus (K_B \lll 0)_L, \quad (2)$$

$$k^b = (K_L \lll 0)_R \oplus (K_B \lll 0)_R, \quad (3)$$

$$k^c = (K_L \lll 0)_L \oplus (K_R \lll 15)_L, \quad (4)$$

$$k^e = (K_B \lll 111)_L \oplus (K_A \lll 45)_L, \quad (5)$$

$$k^d = (K_B \lll 111)_R \oplus (K_L \lll 45)_R. \quad (6)$$

We guess every possible value of K_L , for each guess, K_B can be calculated by Equations (2) and (3), then sieve this (K_L, K_B) pair by Equation (6). For the (K_L, K_B) that satisfy Equation (6), further compute 64 bits of K_A by Equation (5). Then guess the other 64-bits of K_A , by the key schedule of Camellia-256, K_R can be fully determined by K_B and K_A . Equation (4) will further reduce the keys by a factor of 2^8 . So we get about $2^{192} \times 2^{-8} \times 2^{-8} = 2^{176}$ (K_L, K_R) and the correct $K = K_L || K_R$ can be obtained by trial encryption.

Complexity. We choose $n = 9$, then the data complexity is 2^{121} chosen plaintexts. Step 2 removes 2^{48} wrong (k^b, k_2^c, k^e, k_2^d) , for each pair remained after Step 1, $\frac{2^{48}}{2^{144}} = 2^{-96}$ of wrong (k^b, k_2^c, k_2^d, k^e) are removed. Consequently, the number of remaining wrong 208-bit value $(k^a, k^b, k_2^c, k_2^d, k^e)$ after analyzing all the pairs is $2^{64} \times 2^{144} \times (1 - 2^{-96})^{2^{104}} \approx 0$.

The complexity of Step 1 is about $2 \times \left(\sum_{i=1}^7 2^{128-8(i-1)} \times 2^{8i} \right) \times \frac{1}{8} + 2 \times 2^{64} \times 2^{104} \approx 2^{169}$ one round encryptions, equivalent to $2^{165.7}$ encryptions. There are 2^{24} values in H , so in Step 2, 2×2^{24} memory access to H and 2^{48} memory access to Γ are needed for each pair, which result in $2^{64} \times 2^{104} \times 2^{48} = 2^{216}$ memory access, equivalent to $2^{216} \times \frac{1}{52} \times \frac{1}{11} \approx 2^{206.8}$ 11-round encryptions. The complexity of Step 3 is about 2^{184} 6-round encryptions, so the time complexity is dominated by Step 2, which about $2^{206.8}$ encryptions. And the memory complexity is about $2^{155.2}$ bytes.

4.2 Impossible Differential Attack on 10-Round Camellia-192

We remove a round from the bottom of the 11-round attack, and give an attack on 10-round Camellia-192, see Fig. 5 in the left. The choice of plaintexts is the same as the 11-round attack, and the ciphertext pairs are sieved by the difference:

$$(0, e, 0, 0, 0, 0, 0, 0; 0, h, h, h, h, h, 0, 0),$$

where e and h are non-zero values. After the sieve, about 2^{120} pairs remain.

Denote the equivalent subkeys $k^a = k^{w1} \oplus k^1$, $k^b = k^{w2} \oplus k^2$, $k^c = k^{w1} \oplus k^3$ and $k^d = k^{w3} \oplus k^{10}$. The key recovery phase is as follows:

1. Guess k_2^d and check whether $\Delta S_2^{10} = \Delta L_2^{11}$, the number of remaining pairs after this step is about 2^{112} .
2. Individually guess k_l^a ($l = 1, 3, \dots, 8$) and check whether the equation $\Delta S_l^1 = (P^{-1}(\Delta R^0))_l$ holds. About $2^{112} \times 2^{-56} = 2^{56}$ pairs will be kept. Next guess k_2^a , so (L^1, L^1) can be computed. For each of the remaining pairs, do Step 3.
3. Initialize a table Γ' of 2^{72} all possible values (k^b, k_2^c) , for each of the remaining pairs, access the row $(L^1, \Delta L_2^1, \Delta R^1, R_2^1)$ in table H . For each value in the row, remove the corresponding value from Γ' .
4. If Γ' is not empty, output the 144-bit value (k^a, k^b, k_2^c, k_2^d) , otherwise try another guess. The main key can be recovered when (k^a, k^b, k_2^c, k_2^d) is obtained by the similar method of the 11-round attack, except that there are only for equations that can be used:

$$k^a = (K_L \lll 0)_L \oplus (K_B \lll 0)_L, \quad (7)$$

$$k^b = (K_L \lll 0)_R \oplus (K_B \lll 0)_R, \quad (8)$$

$$k^c = (K_L \lll 0)_L \oplus (K_R \lll 15)_L, \quad (9)$$

$$k^d = (K_B \lll 111)_L \oplus (K_L \lll 45)_R. \quad (10)$$

Again, we guess every possible value of K_L , for each guess K_B can be calculated by Equations (7) and (8), then sieve this (K_L, K_B) pair by Equation (10). For the (K_L, K_B) that satisfy Equation (10), compute 8 bits of K_R by Equation (9), and further guess the rest unknown 56 bits of K_R . Furthermore, we test whether the (K_L, K_R) can pass the key schedule of Camellia-192. About $2^{184} \times 2^{-8} \times 2^{-128} = 2^{48}$ keys will remain, and the correct $K = K_L || K_R$ can be obtained by trial encryption.

In this attack, the time complexity is dominated by Step 4, which is about 2^{176} 6-round encryptions, equivalent to about 2^{175} 10-round encryptions. The memory complexity is the same as the 11-round attack. If we did not take the pre-/post- whitening key into account, the complexity would be about 2^{144} encryptions.

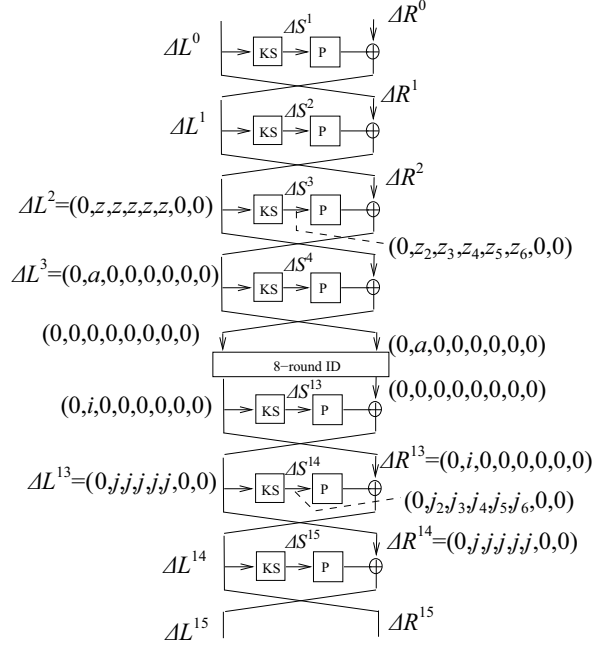


Fig. 6. Impossible Differential Attack on 15-round Camellia-256 without FL/FL^{-1} layers and whitening

5 Impossible Differential Cryptanalysis of 15-round Camellia-256 without FL/FL^{-1} layers and whitening

In this section, we give an improved impossible differential attack on Camellia-256 by using a 8-round impossible differential path without FL/FL^{-1} layer in Fig. 7, which was proposed in [18]. By adding 4 rounds on the top and 3 rounds on the bottom, we can attack 15-round Camellia-256 without FL/FL^{-1} layers and whitening, see Fig. 6.

Precomputation. First, we set up a table Γ_1 for the first two round of the 15-round model. Property 2 implies that $\Delta L^1 = P(a, z_2, z_3 \oplus a, z_4 \oplus a, z_5 \oplus a, z_6 \oplus a, 0, 0)$, so we choose all the 2^{48} ΔL^1 with the required form. Furthermore, all possible ΔL^0 , $L^0 \oplus k^1$ and $(R^0 \oplus k^2)_2$ are chosen. For each of the values, we compute $T = P(S(L^0 \oplus k^1))$, $T' = P(S(L^0 \oplus k^1 \oplus \Delta L^0))$, and get the value $(R^0 \oplus k^2)_l$ ($l = 1, 3, \dots, 8$) by ΔL_l^1 , $(P^{-1}(\Delta L^0))_l$, T , and the corresponding differential table of S-box. Insert $L^0 \oplus k^1$ and $R^0 \oplus k^2$ into the row indexed by $\Delta R^0 = T \oplus T' \oplus \Delta L^1$, ΔL^0 , bits 61 ~ 64 of $L^0 \oplus k^1$, and bits 1 ~ 4, 13 ~ 60 of $R^0 \oplus k^2$. As 2^{184} $(\Delta L^0, \Delta L^1, L^0 \oplus k^1, (R^0 \oplus k^2)_2)$ are totally chosen and there are 2^{184} rows in the table, we get about one $(L^0 \oplus k^1, R^0 \oplus k^2)$ in each row on average.

The complexity of the precomputation is about 2×2^{184} 2-round encryptions, equivalent to $2^{182.1}$ 15-round encryptions. The table requires about $2^{184} \times 16 = 2^{188}$ bytes of memory.

Data Collection. For $2^{122.5}$ known plaintexts, ask for the encryptions and insert the ciphertexts into a hash table indexed by the 7-th and 8-th bytes of $P^{-1}(\Delta R^{15})$. Since by Property 1, the right half of ciphertexts must have the form

$$\Delta R^{15} = \Delta L^{14} = P(i, j_2, j_3 \oplus i, j_4 \oplus i, j_5 \oplus i, j_6 \oplus i, 0, 0).$$

By birthday attack, we can get $2^{244} \times 2^{-16} = 2^{228}$ pairs that the 7-th and 8-th bytes of $P^{-1}(\Delta R^{15})$ are 0.

Key Recovery. We give in Table 2 the corresponding positions of k^1 , k^2 and k^{15} in K_B , and the corresponding positions of k^3 , k^{14} , k_2^4 and k_2^{13} in K_R . It is obvious that there are close relations among the subkeys, i.e., there are common bits in some of the subkeys, which can be used to reduced the complexity of the attack. The key recovery phase is demonstrated as follows.

1. Individually guess k_l^{15} ($l = 1, 3, \dots, 8$) ($K_B : 61 \sim 68, 73 \sim 124$) and remove the pairs do not satisfy $\Delta S_l^{15} = (P^{-1}(\Delta L^{15}))_l$. About $2^{228-7 \times 8} = 2^{172}$ pairs are kept. From Table 2, bits $61 \sim 64$ of k^1 and bits $1 \sim 4, 13 \sim 60$ of k^2 are known.
2. For each of the remaining pairs, compute bits $61 \sim 64$ of $L^0 \oplus k^1$ and bits $1 \sim 4, 13 \sim 60$ of $R^0 \oplus k^2$, then access the value in the corresponding row in Γ_1 . Insert $(\Delta L^0, \Delta R^0, L^0, R^0)$ to a table Γ_2 indexed by bits $1 \sim 60$ of k^1 ($K_B : 1 \sim 60$) and bits $5 \sim 12, 61 \sim 64$ of k^2 ($K_B : 69 \sim 76, 124 \sim 128$). As a result, Γ_2 has 2^{72} rows with about 2^{100} $(\Delta L^0, \Delta R^0, L^0, R^0)$ in each.
3. Guess bits $1 \sim 60$ of k^1 and bits $5 \sim 12, 61 \sim 64$ of k^2 , access the corresponding row in Γ_2 , and compute 2^{100} (L^2, L'^2) and (R^2, R'^2) by two-round encryptions as the whole K_B is known.
4. From Property 1, it is clear that if a pair follows the path in Fig. 6, it has to satisfy $\Delta S_l^{14} = (P^{-1}(\Delta L^{14}))_1 \oplus (P^{-1}(\Delta L^{14}))_l$ ($l = 3, \dots, 6$) and $\Delta S_2^{14} = (P^{-1}(\Delta L^{14}))_2$.
 - (a) Further guess k_2^{14} ($K_R : 5 \sim 12$), partially decrypt round 15 and round 14 to discard the pairs which do not satisfy $\Delta S_2^{14} = (P^{-1}(\Delta L^{14}))_2$. After this procedure, the number of remaining pairs is $2^{100-8} = 2^{92}$.
 - (b) Individually guess k_l^{14} ($l = 3, \dots, 6$) ($k_R : 13 \sim 44$) and keep the pairs which satisfy $\Delta S_l^{14} = (P^{-1}(\Delta L^{14}))_1 \oplus (P^{-1}(\Delta L^{14}))_l$. There are $2^{92-8 \times 4} = 2^{60}$ pairs being kept.
5. (a) Guess bits $45 \sim 47$ of k_R , now k_2^3, k_3^3 , and k_4^3 ($K_R : 24 \sim 47$) are known. Detect if $\Delta S_2^3 = (P^{-1}(\Delta R^2))_2$, $\Delta S_3^3 = (P^{-1}(\Delta R^2))_1 \oplus (P^{-1}(\Delta R^2))_3$, and $\Delta S_4^3 = (P^{-1}(\Delta R^2))_1 \oplus (P^{-1}(\Delta R^2))_4$. The number of remaining pairs is $2^{60-8 \times 3} = 2^{36}$.
 - (b) Individually guess k_l^3 ($l = 5, 6$) ($k_R : 48 \sim 63$) and keep the pairs that satisfy $\Delta S_l^3 = (P^{-1}(\Delta R^2))_1 \oplus (P^{-1}(\Delta R^2))_l$. There are $2^{36-8 \times 2} = 2^{20}$ pairs being kept.
6. Guess k_1^{14} ($K_R : 125 \sim 128, 1 \sim 4$) (now the whole k^{14} is known) and k_2^{13} ($K_R : 69 \sim 76$), keep the pairs that meet $\Delta S_2^{13} = \Delta L_2^{13}$. The number of remaining of pairs will be $2^{20-8} = 2^{12}$.
7. Guess the rest 8 bits of k^3 ($K_R : 64 \sim 68, 77 \sim 79$), now the whole k^3 ($K_R : 16 \sim 79$) is known. We further guess k_2^4 ($K_R : 88 \sim 95$) and check if there is a pair satisfy $\Delta S_2^4 = \Delta L_2^2$. If there is a pair satisfy this, then discard the key guess. Otherwise for every 219-bit key guess, exhaustively search the rest 37 bits of K_R to calculate K_A , use the relation of K_A and K_L to recover K_L , and test the resulting (K_L, K_R) by trial encryption.

Complexity. The data complexity is $2^{122.5}$ known plaintexts. In the data collecting phase, the computation of the 7-th and 8-th bytes of $P^{-1}(\Delta R^{16})$ is less than $2/8$ one round encryption, so the complexity of computing the 7-th and 8-th bytes of $P^{-1}(\Delta R^{16})$ is at most $2^{122.5} \times \frac{1}{4} \times \frac{1}{15} \approx 2^{116.6}$ 15-round encryptions.

Below we elaborate the complexity of each step in the key-recovery phase.

1. The complexity is about $7 \times 2 \times 2^8 \times 2^{228} = 2^{240}$ one round encryptions, which is about $2^{236.1}$ encryptions.
2. This step needs about $2 \times 2^{56} \times 2^{172} = 2^{229}$ memory access and $2^{72} \times 2^{100} \times 16 = 2^{176}$ bytes of memory.
3. The complexity of this step is about $2 \times 2^{128} \times 2^{100} = 2^{229}$ two round encryptions.
4. (a) The complexity of this step is about $2 \times 2^{136} \times 2^{100} = 2^{237}$ one round encryptions, which is about $2^{233.1}$ encryptions.

- (b) The complexity of the each operation in this step is about one round encryption, so the complexity of is about: $2 \times \sum_{i=0}^3 (2^{144+8 \times i} \times 2^{92-8 \times i} \times \frac{1}{15}) \approx 2^{235.1}$.
5. (a) The complexity of this step is about $2 \times \frac{1}{15} \times 2^{171} \times (2^{60} + 2^{52} + 2^{44}) \approx 2^{228.1}$.
 (b) The complexity of this step is about $2 \times \frac{1}{15} \times (2^{179} \times 2^{36} + 2^{187} \times 2^{28}) \approx 2^{213.1}$.
6. This step requires $2 \times 2^{203} \times 2^{20} \times \frac{1}{15} \approx 2^{220.1}$ encryptions.
7. In step 8, we expect $2^{219} \times (1 - 2^{-8})^{2^{12}} \approx 2^{196.6}$ of the key guess remained. So about $2^{196.6+37} = 2^{233.6}$ trail encryptions are request to recover the whole key. The complexity of this step is thus $2 \times 2^{219} \times [1 + (1 - 2^{-8}) + \dots + (1 - 2^{-8})^{2^{12}}] \times \frac{1}{15} + 2^{233.6} \approx 2^{233.6}$. As a result, the time complexity is dominated by Step 1, which is about $2^{236.1}$ 15-round encryptions.

6 Conclusion

In this paper, we present several 6-round impossible differential paths with FL/FL^{-1} layers in the middle, which lead to impossible differential attacks on 10-round Camellia-192 and 11-round Camellia-256. Then an impossible differential cryptanalysis of 15-round Camellia-256 without FL/FL^{-1} layers and whitening is given by carefully using the subkey relation and a precomputational table. A summary of the previous attacks and our analysis of Camellia is given in Table 1.

Table 1. Summary of the attacks on Camellia

Block Size	#Rounds	FL/FL^{-1}	Attack Type	Data	Time	Source
Camellia-128	8	×	Truncated DC	$2^{83.6}$ CP	$2^{55.6}$	[10]
	9	✓	Square Attack	2^{48} CP	2^{122}	[11]
	9	×	Collision Attack	$2^{113.6}$ CP	2^{121}	[19]
	9	×	Square Attack	2^{66} CP	$2^{84.8}$	[6]
	11	×	Impossible DC	2^{118} CP	2^{126} MA	[12]
	12	×	Impossible DC	$2^{116.3}$ CP	$2^{116.6}$	[14]
Camellia-192 /-256	10	✓	Square Attack	2^{48} CP	2^{210}	[11]
	10	✓	Impossible DC	2^{121} CP	2^{175}	this paper
	last 11 rounds	✓	Higher Order DC	2^{93} CC	$2^{255.6}$	[7]
	11	✓	Impossible DC	2^{121} CP	$2^{206.8}$	this paper
	12	×	Impossible DC	2^{120} CP	2^{181}	[18]
	12	×	Linear Attack	2^{119} KP	2^{247}	[16]
	12	×	Square Attack	2^{66} CP	$2^{249.6}$	[6]
	13	×	Impossible DC	2^{120} CP	$2^{211.7}$	[12]
15	×	Impossible DC	$2^{122.5}$ KP	$2^{236.1}$	this paper	

KP: known plaintext; CP: chosen plaintext; CC: chosen ciphertext; DC: differential attack

References

1. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: a 128-bit Block Cipher Suitable for Multiple Platforms-Design and Analysis. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39-56. Springer, Heidelberg (2001)
2. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Specification of Camellia-a 128-bit Block Cipher. version 2.0 (2001), <http://info.isl.ntt.co.jp/crypt/eng/camellia/specifications.html>
3. Biham, E., Shamir, A.: Differential cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
4. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12-23. Springer, Heidelberg (1999)
5. CRYPTREC-Cryptography Research and Evaluation Committees, report, Archive (2002), <http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html>

6. Duo, L., Li, C., Feng, K.: Square like attack on camellia. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 269-283. Springer, Heidelberg (2007)
7. Hatano, Y., Sekine, H., Kaneko, T.: Higher order differential attack of Camellia (II). In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 129-146. Springer, Heidelberg (2003)
8. International Standardization of Organization (ISO), International Standard- ISO/IEC 18033-3, Information technology-Security techniques-Encryption algorithms -Part 3: Block ciphers (2005)
9. Kühn, U.: Improved Cryptanalysis of MISTY1, In: Daemen, J. and Rijmen, V. (Eds.) FSE 2002, LNCS 2365, pp. 61–75. Springer-Verlag, Heidelberg (2002)
10. Lee, S., Hong, S.H., Lee, S.-J., Lim, J.-I., Yoon, S.H.: Truncated differential cryptanalysis of camellia. In: Kim, K.-c. (ed.) ICISC 2001. LNCS, vol. 2288, pp. 32-38. Springer, Heidelberg (2002)
11. Lei, D., Chao, L., Feng, K.: New observation on Camellia. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 51-64. Springer, Heidelberg (2006)
12. Lu, J.: Cryptanalysis of Block Ciphers. PhD Thesis, Department of Mathematics, Royal Holloway, University of London, England (2008)
13. Lu, J., Kim, J.-S., Keller, N., Dunkelman, O.: Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 370-386. Springer, Heidelberg (2008)
14. Mala, H., Shakiba, M., Dakhilalian, M., Bagherikaram G. New Results on Impossible Differential Cryptanalysis of Reduced-Round Camellia-128, SAC 2009, LNCS 5867, pp.281-294, 2009.
15. NESSIE-New European Schemes for Signatures, Integrity, and Encryption, final report of European project IST-1999-12324. Archive (1999), <https://www.cosic.esat.kuleuven.be/nessie/Bookv015.pdf>
16. Shirai, T.: Differential, Linear, Boomerang and Rectangle Cryptanalysis of Reduced-Round Camellia. In: Proceedings of the Third NESSIE Workshop, Munich, Germany, November 6-7 (2002)
17. Sugita, M., Kobara, K., Imai, H.: Security of Reduced Version of the Block Cipher Camellia against Truncated and Impossible Differential Cryptanalysis. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 193-207. Springer, Heidelberg (2001)
18. Wu, W., Zhang, W., and Feng, D. Impossible Differential Cryptanalysis of Reduced-Round ARIA and Camellia. *Journal of Computer Science and Technology*, 22(3), 449-456.
19. Wenling, W., Dengguo, F., Hua, C.: Collision attack and pseudorandomness of reduced-round camellia. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 252-266. Springer, Heidelberg (2004)
20. W. Wu, L. Zhang, and W. Zhang, Improved Impossible Differential Cryptanalysis of Reduced-Round Camellia. In: Avanzi, R., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol.5381, pp. 442-456. Springer, Heidelberg (2009)

A 8-Round Impossible Differential Path without FL/FL^{-1} Layer

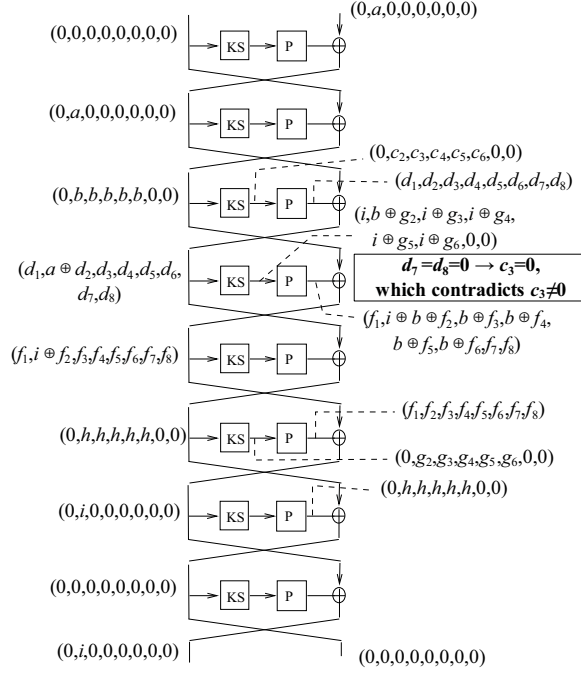


Fig. 7. 8-round Impossible Differential Path without FL/FL^{-1} layer

B The process of key recovery attack on 15-round Camellia-256 without FL/FL^{-1} layers and whitening

Table 2. Corresponding bit positions of the subkeys in K_B and K_R

Subkey bytes	Bit positions in K_B	Subkey bytes	Bit positions in K_B	Subkey bytes	Bit positions in K_R	Subkey bytes	Bit positions in K_R
k_1^1	1 ~ 8	k_5^2	97 ~ 104	k_1^3	16 ~ 23	k_1^{14}	125 ~ 128, 1 ~ 4
k_2^1	9 ~ 16	k_6^2	105 ~ 112	k_2^3	24 ~ 31	k_2^{14}	5 ~ 12
k_3^1	17 ~ 24	k_7^2	113 ~ 120	k_3^3	32 ~ 39	k_3^{14}	13 ~ 20
k_4^1	25 ~ 32	k_8^2	121 ~ 128	k_4^3	40 ~ 47	k_4^{14}	21 ~ 28
k_5^1	33 ~ 40	k_1^{15}	61 ~ 68	k_5^3	48 ~ 55	k_5^{14}	29 ~ 36
k_6^1	41 ~ 48	k_2^{15}	69 ~ 76	k_6^3	56 ~ 63	k_6^{14}	37 ~ 44
k_7^1	49 ~ 56	k_3^{15}	77 ~ 84	k_7^3	64 ~ 71	k_7^{14}	45 ~ 52
k_8^1	57 ~ 64	k_4^{15}	85 ~ 92	k_8^3	72 ~ 79	k_8^{14}	53 ~ 60
k_1^2	65 ~ 72	k_5^{15}	93 ~ 100	k_2^4	88 ~ 95	k_2^{13}	69 ~ 76
k_2^2	73 ~ 80	k_6^{15}	101 ~ 108				
k_3^2	81 ~ 88	k_7^{15}	109 ~ 116				
k_4^2	89 ~ 96	k_8^{15}	117 ~ 124				