Improved Attacks on Reduced-Round Camellia-128/192/256

Xiaoyang Dong[1], Leibo Li[1], Keting Jia[2], and Xiaoyun Wang[1,3*]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, China
{dongxiaoyang,lileibo}@mail.sdu.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University
ktjia@mail.tsinghua.edu.cn
[3] Institute for Advanced Study, Tsinghua University
xiaoyunwang@tsinghua.edu.cn

Abstract. Camellia is a widely used block cipher, which has been selected as an international standard by ISO/IEC. In this paper, we consider a new family of differentials of round-reduced Camellia-128 depending on different key subsets. There are totally 224 key subsets corresponding to 224 types of 8-round differentials, which cover a fraction of $1 - 1/2^{15}$ of the keyspace. And each type of 8-round differential consists of 2^{43} differentials. Combining with the multiple differential attack techniques, we give the key-dependent multiple differential attack on 10-round Camellia-128 with data complexity 2^{91} and time complexity 2^{113}. Furthermore, we propose a 7-round property for Camellia-192 and an 8-round property for Camellia-256, and then mount the meet-in-the-middle attacks on 12-round Camellia-192 and 13-round Camellia-256, with complexity of 2^{180} encryptions and $2^{232.7}$ encryptions, respectively. All these attacks start from the first round in a single key setting.

Keywords: Camellia, Block Cipher, Key-Dependent Attack, Multiple Differential Attack, Meet-in-the-Middle Attack.

1 Introduction

The block cipher Camellia with 128-bit block size has variable key lengths of 128, 192, 256, named as Camellia-128, Camellia-192 and Camellia-256, respectively. It was proposed by NTT and Mitsubishi in 2000 [2]. Now Camellia has become a widely used block cipher as an e-government recommended cipher by CRYPTREC [9]. Besides, Camellia was selected as one of NESSIE block cipher portfolio [26] and international standard by ISO/IEC 18033-3 [14]. Therefore, Camellia has received a great deal of attention from cryptanalysts with various attack methods, including higher order differential attack [13], linear and differential attack [26], truncated differential attacks [15,18,27], collision attack [30], square attacks [19,20], impossible differential attacks [22,25,31,23,21], meet-in-the-middle attacks [24,8] and zero correlation cryptanalysis [5] etc.

An important property of Camellia is FL/FL^{-1} layers inserted every 6 rounds. The FL/FL^{-1} functions are key-dependent functions which provide non-regularity across rounds to resist the differential cryptanalysis. Many previous papers presented attacks on simplified versions of Camellia without the FL/FL^{-1} layers and the whitening layers [20,18,22,25,26,30,31]. For the original Camellia, impossible differential attacks on 10/11/12-round Camellia-128/192/256 were given in [21], and recently improved by Boura *et al.* in [6]. The Meet-in-the-Middle (MITM) attack on Camellia was firstly proposed by Lu *et al.* in [24], which introduced attacks on 10-round Camellia-128, 11-round Camellia-192 and 12-round Camellia-256 utilizing 5-round and 6-round higher-order MITM properties of Camellia. However this attack does not start from the first round and excludes the whitening layers. Chen *et al.* [8] attacked 12-round Camellia from the first round by applying the attack model for AES in [10] to construct a 7-round MITM property of Camellia. Besides, zero-correlation cryptanalysis with FFT method(ZC FFT) was applied to 11-round Camellia-128 and 12-round Camellia-192 in [5], which was slightly better than exhaustive search with almost the full codebook.

[*] Corresponding author

In this paper, we analyze *the original versions of Camellia with FL/FL^{-1} layers and whitening key starting from the first round* by two methods: key-dependent multiple differential attack and meet-in-the-middle attack. Multiple differential attack [4,29] uses multiple differentials to accumulate the advantage of many differentials as a distinguisher. The key-dependent differential attack was proposed by Ben-Aroya and Biham [3] to analyze Lucifer, which covered a fraction of 55% of the keyspace. A similar idea was also used by Knudsen and Rijmen to analyze DFC in [16]. Later, Sun and Lai proposed the key-dependent attack to analyze IDEA [28] by distinguishing the non-random distribution of the intermediate values for different key subsets, which composed the full keyspace.

Our Contributions. In this paper, we first consider the key-dependent multiple differential attack (KDMDA) on Camellia-128, by studying the multiple differentials corresponding to different key subsets. There are 224 types of 8-round differentials corresponding to different key subsets for Camellia, and each includes 2^{43} differentials. Each key subset contains a fraction of 1/4 of the keyspace. All the 224 subsets cover a fraction of $1 - 1/2^{15}$ of the keyspace. Using these differentials, we launch the multiple differential attack on 10-round Camellia-128, which needs 2^{91} chosen plaintexts and $2^{104.5}$ encryptions, and succeeds on a fraction of about 99.99% of the keyspace. It is easy to extend this attack to the full keyspace by exhaustive search on the remaining fraction of $1/2^{15}$ of the keyspace. This is the first differential attack on Camellia with FL/FL^{-1} layers.

The key-dependent multiple differential attack is also possible against Camellia-192/256. In order to get better analysis results, we explore the meet-in-the-middle attack on Camellia-192/256. Combined with the differential enumeration technique and multiset proposed by Dunkelman *et al.* [12], other improved techniques proposed by Derbez *et al.* [11] and the relations of intermediate variables and subkeys, we propose a new 7-round property for Camellia-192 and an 8-round property of Camellia-256 to reduce the number of elements in a multiset. Based on both properties, we attack the 12-round Camellia-192 and 13-round Camellia-256 which costs 2^{113} chosen plaintexts, 2^{180} encryptions and 2^{154} 128-bit memories for Camellia-192, 2^{113} chosen plaintexts, $2^{232.7}$ encryptions and 2^{227} 128-bit memories for Camellia-256, respectively. However, we can not construct a good property for Camellia-128 since the complexity of the precomputation phase are larger than 2^{128} and it should be further explored.

In this paper, we only discuss the attacks on *Camellia with FL/FL^{-1} layers and whitening key starting from the first round.* Table 1 summarizes our results along with the major previous results, where CP and CC refer to the number of chosen plaintexts and chosen ciphertexts, respectively.

Table 1. Summary of the Attacks on Reduced-Round Camellia

Rounds	Percentage of Key Space	Attack Type	Data	Time	Memory	Source
Camellia-128						
10	100%	Impossible Diff	$2^{113.8}$CP	2^{120}Enc	$2^{86.4}$Bytes	[21]
10	99.99%	KDMDA	2^{91}CP	$2^{104.5}$Enc	2^{96}Bytes	Section 4.4
10	100%	KDMDA	2^{91}CP	2^{113}Enc	2^{96}Bytes	Section 4.4
11	100%	ZC FFT	$2^{125.3}$KP	$2^{124.8}$Enc	$2^{112.0}$Bytes	[5]
Camellia-192						
11	100%	Impossible Diff	$2^{113.7}$CP	2^{184}Enc	$2^{143.7}$Bytes	[21]
12	100%	ZC FFT	$2^{125.7}$KP	$2^{188.8}$Enc	2^{112}Bytes	[5]
12	100%	MITM	2^{113}CP	2^{180}Enc	2^{158}Bytes	Section 5.2
Camellia-256						
12	100%	Impossible Diff	$2^{114.8}$CP/CC	2^{240}Enc	$2^{151.8}$Bytes	[21]
12	100%	MITM	2^{19}CP	$2^{231.2}$Enc	2^{229} Bytes	[8]
13	100%	MITM	2^{113}CC	$2^{232.7}$Enc	2^{231}Bytes	Section 5.3

The rest of this paper is organized as follows. Section 2 gives some notations and a brief description of Camellia. Section 3 describes some observations of Camellia used in our cryptanalysis. In Section 4, we give the 8-round multiple differentials of Camellia for different key subsets, and present key-dependent multiple differential attack on 10-round Camellia-128. Section 5 illustrates the meet-in-the-middle attacks on 12/13-round Camellia-192/256. Finally, we conclude the paper in Section 6.

2 Preliminaries

In this section we give the notations used throughout this paper, and then briefly describe the block cipher Camellia.

2.1 Notations

The following notations are used in this paper:

$L_{r-1},\ L'_{r-1}$	the left 64-bit half of the r-th round input
$R_{r-1},\ R'_{r-1}$	the right 64-bit half of the r-th round input
X_r	the state after the key addition layer of the r-th round
Y_r	the state after the substitution transformation layer of the r-th round
Z_r	the state after the diffusion layer of the r-th round
k_r	the subkey used in the r-th round
kw_i	the whitening key used in the beginning and an the end of Camellia, $i = 1, 2, 3, 4$
$X[i]$	the i-th byte of a bit string X $(1 \leq i \leq 8)$, where the left most byte is the first byte
X_L (X_R)	the left (right) half of a bit string X,
$X\{i\}$	the i-th most significant bit of a bit string $X(1 \leq i \leq 128)$, where the left-most bit is the most significant bit
$\varDelta X$	the difference of X and X'
ham(X)	the hamming weight of X, for example, $X = 00100010$, ham(X)=2
zero(X)	the number of X's zero bits, for example, $X = 00100010$, zero(X)=6
$\oplus,\ \wedge,\ \vee$	bitwise exclusive OR (XOR), AND, OR
$\neg x$	bitwise inversion of bit string x, e.g. $\neg 0x22 = 0xdd$
$\bigcup$	the union of sets
$\lvert A \rvert$	the size of the set A
$x \Vert y$	bit string concatenation of x and y
$\lll l$	bit rotation to the left by l bit

2.2 Brief Description of Camellia

Camellia [2] is a Feistel structure block cipher, and the number of rounds are 18/24/24 for Camellia-128/192/256, respectively. The encryption procedure (depicted in Appendix C) for 128-bit key is as follows.

Firstly, a 128-bit plaintext M is XORed with the whitening key $(kw_1 \Vert kw_2)$ and separated into L_0 and R_0 of equal length. Then, for $r = 1$ to 18, except for $r = 6$ and 12, the following is carried out:

$$L_r \ = \ R_{r-1} \oplus F(L_{r-1}, k_r), \quad R_r \ = \ L_{r-1}.$$

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

$$\begin{array}{ll} L_r^* = R_{r-1} \oplus F(L_{r-1}, k_r), & R_r^* = L_{r-1}, \\ L_r = FL(L_r^*, kf_{r/3-1}), & R_r = FL^{-1}(R_r^*, kf_{r/3}), \end{array}$$

Lastly, the 128-bit ciphertext C is computed as: $C = (R_{18} \Vert L_{18}) \oplus (kw_3 \Vert kw_4)$.

For 192- and 256-bit keys, the 128-bit plaintext M is XORed with the whitening key $(kw_1 \Vert kw_2)$ and separated into L_0 and R_0 of equal length. Then, for $r = 1$ to 24, except for $r = 6, 12$ and 18, the following is carried out:

$$L_r \ = \ R_{r-1} \oplus F(L_{r-1}, k_r), \quad R_r \ = \ L_{r-1}.$$

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

$$\begin{aligned} L_r^* &= R_{r-1} \oplus F(L_{r-1}, k_r), & R_r^* &= L_{r-1}, \\ L_r &= FL(L_r^*, kf_{r/3-1}), & R_r &= FL^{-1}(R_r^*, kf_{r/3}), \end{aligned}$$

Lastly, the 128-bit ciphertext C is computed as: $C = (R_{24} \| L_{24}) \oplus (kw_3 \| kw_4)$.

The round function F is composed of a key-addition layer, a substitution transformation layer S and a diffusion layer P. The key-addition layer is an XOR operation of the left half input of the round function and the round key, i.e. $X_r = L_{r-1} \oplus k_r$ for the r-th round. There are four types of 8×8 S-boxes $s_1,\ s_2,\ s_3$ and s_4 in the S transformation layer. Let the input of the substitution transformation S of the r-th round be $X_r = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$, the output Y_r is computed as follows:

$$Y_r = S(X_r) = \big(s_1(x_1), s_2(x_2), s_3(x_3), s_4(x_4), s_2(x_5), s_3(x_6), s_4(x_7), s_1(x_8)\big).$$

The linear transformation P is a diffusion operation based on the bytes. Let the input of the transformation P in round r be $Y_r = (y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8)$, the output be $Z_r = (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8)$. $Z_r = P(Y_r)$ and its inverse P^{-1} are defined as follows:

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

The FL function is used every 6 rounds. FL is defined as $(a_L \| a_R, kf_L \| kf_R) \mapsto (b_L \| b_R)$, where $a_L, a_R, kf_L, kf_R, b_L$ and b_R are 32-bit words.

$$b_R = ((a_L \wedge kf_L) \lll 1) \oplus a_R, \quad b_L = (b_R \vee kf_R) \oplus a_L.$$

In accordance with the notations in [1], let the master key of Camellia be K. The subkeys $K_L,\ K_R$ are simply generated from K. For Camellia-128, $K_L = K$, $K_R = 0$. For Camellia-192, K_L is the left 128-bit of K, i.e., $K_L = K\{1-128\}$, and the concatenation of the right 64-bit of K and its complement is used as K_R, i.e., $K_R = K\{129-192\} \| \neg K\{129-192\}$. For Camellia-256, $K_L = K\{1-128\}$, and $K_R = K\{129-256\}$. Two 128-bit keys K_A and K_B are derived from K_L and K_R by a non-linear transformation. Then the whitening keys kw_i $(i = 1, ..., 4)$, round subkeys k_r $(r = 1, ..., 24)$ and kf_j $(j = 1, ..., 6)$ are generated by rotating $K_L,\ K_R,\ K_A$ or K_B. For more details of Camellia, we refer to [1].

3 Some Observations of Camellia

This section introduces some observations which help us analyze the reduced-round Camellia.

Observation 1 *([17]) Let $X,\ X',\ K$ be l-bit values, and $\Delta X = X \oplus X'$, then the differential properties of AND and OR operations are:*

$$\begin{aligned} (X \wedge K) \oplus (X' \wedge K) &= \Delta X \wedge K, \\ (X \vee K) \oplus (X' \vee K) &= \Delta X \oplus (\Delta X \wedge K). \end{aligned}$$

Observation 2 *Given the input difference of the i-th round $\Delta L_i = (\alpha, 0, 0, 0, 0, 0, 0, 0)$, $\Delta R_i = (0, 0, 0, 0, 0, 0, 0, 0)$, the output difference of $(i+3)$-th round ΔR_{i+3} and intermediate difference ΔY_{i+2} satisfy the following equations:*

$$\begin{array}{ll} P^{-1}(\Delta R_{i+3})[4] = \Delta L_i[1] = \alpha,\ P^{-1}(\Delta R_{i+3})[j] = 0,\ j = 6, 7 & \\ P^{-1}(\Delta R_{i+3})[1] = \Delta Y_{i+2}[1], \quad P^{-1}(\Delta R_{i+3})[j] = \Delta Y_{i+2}[j] \oplus P^{-1}(\Delta R_{i+3})[4],\ j = 2, 3, 5, 8. & \end{array}$$

Observation 3 *Given the output difference of the $(i+2)$-th round $\Delta L_{i+2} = (0,0,0,0,0,0,0,0)$, $\Delta R_{i+2} = (\alpha,0,0,0,0,0,0,0)$, the input difference of i-th round ΔR_i and the intermediate difference ΔY_{i+1} satisfy the following equations:*

$$\begin{array}{ll} P^{-1}(\Delta R_i)[4] = \Delta R_{i+2}[1] = \alpha, & P^{-1}(\Delta R_i)[j] = 0,\ j = 6,7 \\ P^{-1}(\Delta R_i)[1] = \Delta Y_{i+1}[1], & P^{-1}(\Delta R_i)[j] = \Delta Y_{i+1}[j] \oplus P^{-1}(\Delta R_i)[4],\ j = 2,3,5,8. \end{array}$$

Observation 4 *Let the input difference of FL^{-1} be $(\Delta a_L, 0)$. Then the output difference of FL^{-1} must be $(\Delta a_L, 0)$, when $\Delta a_L \wedge kf_{2L} = 0$.*

4 Key-Dependent Multiple Differential Attack on Reduced-Round Camellia-128

In this section, we present truncated differential based on the diffusion layer P for different key subsets. Then, 224 different types of 8-round multiple differentials for different key subsets are constructed. Finally, we launch the key-dependent multiple differential attack on 10-round Camellia-128.

4.1 Some Truncated Differentials

Observation 5 *Let the input difference of P be $(y_1, y_2, 0, 0, 0, 0, 0, 0)$,*

- *if $y_1 \neq y_2$, the output difference of P is $(y_1, y_1 \oplus y_2, y_1 \oplus y_2, y_2, y_1 \oplus y_2, y_2, 0, y_1)$.*
- *if $y_1 = y_2$, the output difference of P is $(y_1, 0, 0, y_2, 0, y_2, 0, y_1)$.*

Observation 6 *([27]) If the input difference of P is $(y_1, y_2, y_3, y_4, y_5, y_6, 0, y_8)$, then the output difference of P is $(z_1, z_2, 0, 0, 0, 0, 0, 0)$ with probability 2^{-40}. And the following equations hold: $y_1 = y_6, y_2 = y_8, y_3 = y_4 = y_5 = y_1 \oplus y_2$.*

Proof. By computing the inversion of P, we get $y_8 = z_1, y_6 = z_2, y_5 = z_1 \oplus z_2, y_4 = z_1 \oplus z_2, y_3 = z_1 \oplus z_2, y_2 = z_1, y_1 = z_2$. Then, $y_1 = y_6, y_2 = y_8, y_3 = y_4 = y_5 = y_1 \oplus y_2$. □

Using the above observations, we construct the following 4-round truncated differential with probability 2^{-56},

$$\begin{aligned} (00000000, **000000) &\xrightarrow[Pr=1]{Round} (**000000, 00000000) \xrightarrow[Pr=1]{Round} (******0*, **000000) \\ &\xrightarrow[Pr=2^{-40}]{Round} (**000000, ******0*) \xrightarrow[Pr=2^{-16}]{Round} (00000000, **000000) \end{aligned}$$

Similarly, we get another three 4-round truncated differentials with probability 2^{-56} in the last three columns of Table 2.

Table 2. 4-Round Truncated Differentials

Active S-boxes: $0 \to 2 \to 7 \to 2$			
Case-1	Case-2	Case-3	Case-4
$(00000000, **000000)$	$(00000000, 0**00000)$	$(00000000, *00*0000)$	$(00000000, 00**0000)$
$(**000000, 00000000)$	$(0**00000, 00000000)$	$(*00*0000, 00000000)$	$(00**0000, 00000000)$
$(******0*, **000000)$	$(******0, 0**00000)$	$(*****0**, *00*0000)$	$(****0***, 00**0000)$
$(**000000, ******0*)$	$(0**00000, *******0)$	$(*00*0000, *****0**)$	$(00**0000, ****0***)$
$(00000000, **000000)$	$(00000000, 0**00000)$	$(00000000, *00*0000)$	$(00000000, 00**0000)$

4.2 Key Subsets Corresponding to Truncated Differentials

In this section, we extend the 4-round truncated differentials in Table 2 by adding a FL/FL^{-1} layer at the bottom. As a result, we divide the full keyspace into different subsets corresponding to different differentials.

We denote the two nonzero input byte differences of FL^{-1} function as c_1, c_2. Then we get four types of input differences of the FL^{-1} function, which are $(c_1, c_2, 0, 0, 0, 0, 0, 0)$, $(0, c_1, c_2, 0, 0, 0, 0, 0)$, $(c_1, 0, 0, c_2, 0, 0, 0, 0)$, $(0, 0, c_1, c_2, 0, 0, 0, 0)$. To reduce the diffusion of the active S-boxes, we make the input and the output differences of the FL^{-1} function equal, which determines a key subset according to Observation 4. Therefore, a value of (c_1, c_2) corresponds to a key subset. Obviously, the lower the hamming weight of (c_1, c_2) is, the larger the size of the corresponding key subset will be. In order to reduce the complexity, we choose (c_1, c_2) to make the size of key subset as large as possible. According to Observation 5, in order to maintain the 4-round truncated differential, c_1 should be different from c_2. So we choose 56 values of (c_1, c_2) where $\text{ham}(c_1) = 1$, $\text{ham}(c_2) = 1$, and $c_1 \neq c_2$, see Table 3. Combining with 4 truncated differentials, we construct 224 key subsets, which are denoted as $KDset_i^j$, $j = 1, 2, 3, 4$ *and* $i = 1, 2 \cdots 56$.

$$\begin{aligned}
KDset_i^1 &= \{K | kf_{2L} = (\neg c_1^i \wedge *, \neg c_2^i \wedge *, *, *), * \in F_2^8\}, \\
KDset_i^2 &= \{K | kf_{2L} = (*, \neg c_1^i \wedge *, \neg c_2^i \wedge *, *), * \in F_2^8\}, \\
KDset_i^3 &= \{K | kf_{2L} = (\neg c_1^i \wedge *, *, *, \neg c_2^i \wedge *), * \in F_2^8\}, \\
KDset_i^4 &= \{K | kf_{2L} = (*, *, \neg c_1^i \wedge *, \neg c_2^i \wedge *), * \in F_2^8\}.
\end{aligned}$$

In each key subset, two bits of kf_{2L} are 0, and the other bits traverse all values. The size of a key subset is 2^{126} for Camellia-128. We denote the union of all $KDset_i^j$ as $PKSPACE$.

$$PKSPACE = \bigcup_{j=1}^{4} \bigcup_{i=1}^{56} KDset_i^j$$

Table 3. 56 Different Values of (c_1, c_2) in Hexadecimal

i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)	i	(c_1^i, c_2^i)
1	01, 02	8	02, 01	15	04, 01	22	08, 01	29	10, 01	36	20, 01	43	40, 01	50	80, 01
2	01, 04	9	02, 04	16	04, 02	23	08, 02	30	10, 02	37	20, 02	44	40, 02	51	80, 02
3	01, 08	10	02, 08	17	04, 08	24	08, 04	31	10, 04	38	20, 04	45	40, 04	52	80, 04
4	01, 10	11	02, 10	18	04, 10	25	08, 10	32	10, 08	39	20, 08	46	40, 08	53	80, 08
5	01, 20	12	02, 20	19	04, 20	26	08, 20	33	10, 20	40	20, 10	47	40, 10	54	80, 10
6	01, 40	13	02, 40	20	04, 40	27	08, 40	34	10, 40	41	20, 40	48	40, 20	55	80, 20
7	01, 80	14	02, 80	21	04, 80	28	08, 80	35	10, 80	42	20, 80	49	40, 80	56	80, 40

We collect the keys that do not belong to any one of the $KDset_i^j$ to form the remaining key set denoted as $RKset$, which is consisted of two classes:

Class 1 The pattern of kf_{2L} is $(*, \neg 0, *, \neg 0)$ or $(\neg 0, *, \neg 0, *)$, where '*' is a random byte. There are $2 \times (2^8)^2 - 1 = 2^{17} - 1$ possible kf_{2L}.

Class 2 The remaining keys are not included in Class 1.

- If zero(kf_{2L})=2, the number of possible kf_{2L} is $8 \times 4 = 48$.
- If zero(kf_{2L})=3, the number of possible kf_{2L} is $8C_4^3 = 32$.
- If zero(kf_{2L})=4, the number of possible kf_{2L} is $8C_4^4 = 8$.

Totally, there are $48 + 32 + 8 = 88$ possible kf_{2L}.

So the size of remaining key set is $2^{96} \times (88 + 2^{17} - 1) \approx 2^{113}$.

The $PKSPACE$ and remaining key set $RKset$ form the full keyspace $KSPACE$:

$$KSPACE = \left(\bigcup_{j=1}^{4} \bigcup_{i=1}^{56} KDset_i^j \right) \bigcup RKset.$$

Let the input difference of FL^{-1} function be $(c_1, c_2, 0, 0)$, which corresponds a key subset $KDset_i^1$. Therefore, for the key subset $KDset_i^1$, the probability for 4-round truncated differential of the case-1 appending a FL/FL^{-1} layer with output difference $(00000000, c_1c_2000000)$ is $2^{-56} \times 2^{-16} = 2^{-72}$.

4.3 Searching 8-Round Multiple Differentials for Every Key Subset

We use 4-round truncated differentials in Table 2 to construct 8-round differentials with FL/FL^{-1} functions. We extend the 4-round truncated differential by adding two rounds forward and appending a FL/FL^{-1} layer and two rounds at the bottom to obtain 8-round differentials. We get four types of 8-round differential patterns, named as type-1/-2/-3/-4 which are constructed by case-1/-2/-3/-4, respectively.

Property 1. For each $KDset_i^j$, $i = 1, 2, \cdots, 56$, $j = 1, 2, 3, 4$, we construct a family of 8-round multiple differentials.

1. There are 2^{31} input differences and 2^6 output differences which produce $2^{31+6} = 2^{37}$ 8-round differentials with the probability 2^{-125}.
2. 2^{38} input differences and 2^6 output differences produce $2^{38+6} = 2^{44}$ 8-round differentials with probability 2^{-126}.
3. 2^{45} input differences and 2^6 output differences produce $2^{45+6} = 2^{51}$ 8-round differentials with probability 2^{-127}.

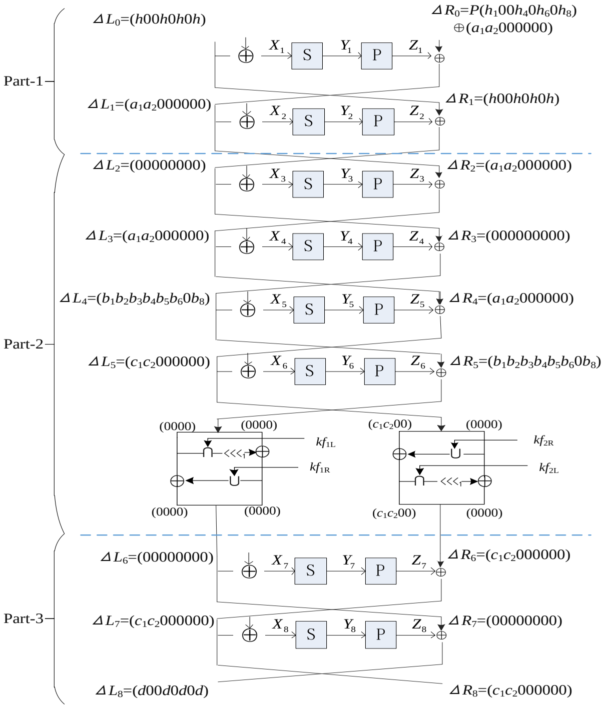

Fig. 1. Type-1: 8-Round Differential Pattern with FL/FL^{-1} Layer

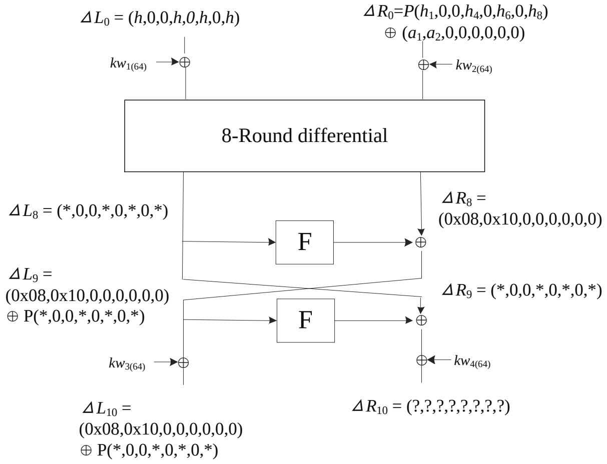

Fig. 2. Multiple Differential Attack on 10-Round Camellia-128

Proof. We prove the Property 1 by type-1 differential pattern illustrated in Fig. 1.

For the top two rounds, we apply the following 2-round differential

$$(\varDelta L_0, \varDelta R_0) \xrightarrow[Pr_1]{Round} (a_1a_2000000,\quad h00h0h0h) \xrightarrow[Pr=2^{-14}]{Round} (00000000,\quad a_1a_2000000),$$

where $\varDelta L_0 = (h, 0, 0, h, 0, h, 0, h)$, $\varDelta R_0 = P(h_1, 0, 0, h_4, 0, h_6, 0, h_8) \oplus (a_1, a_2, 0, 0, 0, 0, 0, 0)$.

By the 2-round differential, we know $\varDelta Y_1 = (h_1, 0, 0, h_4, 0, h_6, 0, h_8)$, $\varDelta Y_2 = (h, h, 0, 0, 0, 0, 0, 0)$. Obviously, there are $(2^8 - 1)$ $\varDelta L_0$. For each $\varDelta L_0$, there are $2^7 \times 2^7 = 2^{14}$ possible $\varDelta L_1$ with probability 2^{-14} as a result of two active S-boxes in round 2. Considering the 4 active S-boxes in the first round to compute Pr_1 and number of $\varDelta Y_1$ values, there are $C_4^3 \cdot 2^7 = 2^9$ possible values of $\varDelta Y_1$ with probability $2^{-6\times 3} \times 2^{-7} = 2^{-25}$, $C_4^2 \cdot 2^{14} = 2^{16}$ possible values of $\varDelta Y_1$ with probability $2^{-6\times 2} \times 2^{-7\times 2} = 2^{-26}$, $C_4^1 \cdot 2^{21} = 2^{23}$ possible values of $\varDelta Y_1$ with probability $2^{-6} \times 2^{-7\times 3} = 2^{-27}$, and 2^{28} possible values of $\varDelta Y_1$ with probability 2^{-28}.

So, for the 2-round differential, there are $2^8 \times 2^9 \times 2^{14} = 2^{31}$ values of $(\varDelta L_0, \varDelta R_0)$ with probability $2^{-25} \times 2^{-14} = 2^{-39}$, $2^8 \times 2^{16} \times 2^{14} = 2^{38}$ values of $(\varDelta L_0, \varDelta R_0)$ with probability $2^{-26} \times 2^{-14} = 2^{-40}$, $2^8 \times 2^{23} \times 2^{14} = 2^{45}$ values of $(\varDelta L_0, \varDelta R_0)$ with probability $2^{-27} \times 2^{-14} = 2^{-41}$, and $2^8 \times 2^{28} \times 2^{14} = 2^{50}$ values of $(\varDelta L_0, \varDelta R_0)$ with probability $2^{-28} \times 2^{-14} = 2^{-42}$.

The last 2-round differential with the input difference $(00000000, c_1c_2000000)$ is

$$(00000000,\quad c_1c_2000000) \xrightarrow[Pr=1]{Round} (c_1c_2000000,\quad 00000000) \xrightarrow[Pr=2^{-14}]{Round} (d00d0d0d,\quad c_1c_2000000).$$

There are about 2^6 $\varDelta L_8$. The probability of each $(\varDelta L_7, \varDelta R_7) \Rightarrow (\varDelta L_8, \varDelta R_8)$ is 2^{-13} or 2^{-14}.

Totally, there are 2^{31} input differences and 2^6 output differences which form $2^{31+6} = 2^{37}$ 8-round differentials, and the probability of each differential is $2^{-72-39-14} = 2^{-125}$; there are 2^{38} input differences and 2^6 output differences which form $2^{38+6} = 2^{44}$ 8-round differentials with probability $2^{-72-40-14} = 2^{-126}$; there are 2^{45} input differences and 2^6 output differences which form $2^{45+6} = 2^{51}$ 8-round differentials with probability $2^{-72-41-14} = 2^{-127}$. □

Without loss of generality, we search type-1 differentials as an example to verify the correctness of Property 1 experimentally. The search procedure is as follows.

1. We exhaustively search differentials which match 4-round truncated differential with appending a FL/FL^{-1} layer depicted in Part-2 of Fig 1. Let $(00000000, a_1a_2000000)$ be input difference, and $(00000000, c_1c_2000000)$ be the input difference of the FL/FL^{-1} layer, where (c_1, c_2) is chosen in Table 3. Store the 4-round differential and its corresponding probability in a 56×2^{16} table, where "row" is indexed by (c_1, c_2), "column" is indexed by (a_1, a_2), and the elements are the corresponding probability Pr of the differential, which is calculated by the following equations. We denote $Y_4 = (a_1^{'}a_2^{'}000000)$.

$$\begin{aligned} &Pr_1 = Pr((a_1a_2000000) \xrightarrow{S} (a_1^{'}a_2^{'}000000)), Pr_2 = Pr((c_1c_2000000) \xrightarrow{S} (a_1^{'}a_2^{'}000000)),\\ &Pr_3 = Pr(P(a_1^{'}, a_2^{'}, 0, 0, 0, 0, 0, 0) \xrightarrow{S} P^{-1}(a_1 \oplus c_1, a_2 \oplus c_2, 0, 0, 0, 0, 0, 0))\\ &Pr = \sum_{a_1^{'}, a_2^{'} \in F_2^8} Pr_1 \cdot Pr_2 \cdot Pr_3 \end{aligned}$$

2. For each row indexed by (c_1, c_2), calculate the output differences $(d_100d_40d_60d_8, c_1c_2000000)$ of the 8-round differential, whose values form the output differences set, denoted as $\varDelta OUTset$. And then for each column indexed by (a_1, a_2), collect the input differences of 8-round differential that could result in $(00000000, a_1a_2000000)$ differences after two rounds of encryption, to produce the input differences set, denoted as $\varDelta INset$.

When $c_1 = 0x08$, $c_2 = 0x10$, we search type-1 differentials by PC, and obtain $|\varDelta OUTset| = 57 \approx 2^6$. If the probability of each differential is larger than 2^{-125}, the $|\varDelta INset|$ is $2^{31.1}$. If the probability of each differential is larger than 2^{-126}, the $|\varDelta INset|$ is $2^{37.9}$. If the probability of each differential is larger than 2^{-127}, the $|\varDelta INset|$ is $2^{44.8}$. Therefore, the experimental data reveals correctness of Property 1.

4.4 Key-Dependent Multiple Differential Attack on 10-Round Camellia-128

For every $KDset_i^j$, $i = 1, 2 \cdots 56$, $j = 1, 2, 3, 4$, we choose 2^{37} input differences from $\Delta INset$ where the probabilities are all larger than 2^{-126} and pick all the 2^6 output differences of $\Delta OUTset$. We launch multiple differential attack using these differentials. We repeat 224 times multiple differential attacks, if one of the attacks succeeds, the right key can be recovered. Otherwise the right key belongs to $RKset$. The following is one of the 224 attacks.

We choose type-1 differentials and $c_1 = 0x08$ $c_2 = 0x10$ to launch an attack, whose corresponding key subset is $KDset_{32}^1$. As the Fig. 2 shows, we add two rounds after the 8-round differentials distinguisher to analyse 10-round Camellia-128.

In [4], there is a strong condition that the set of input differences are "admissible". However, paper [29] proves the condition is not necessary when applying structure technique. Here, we take advantage of the structure attack model to implement multiple differential attack displayed as follows:

1. Choose 2^x structures of plaintexts, and each structure contains 2^{56} plaintexts with $L_0 = (\alpha_1, x_1, x_2, \alpha_1, x_3, \alpha_1, x_4, \alpha_1)$, $R_0 = P(\alpha_2, x_5, x_6, \alpha_3, x_7, \alpha_4, x_8, \alpha_5) \oplus (\alpha_6, \alpha_7, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14})$, where x_i are fixed values and α_j take all the possible values in each structure.
2. For each structure, ask for the encryptions of the plaintexts P and store the 2^{56} ciphertexts C, indexed by $P^{-1}(C_L)[1,4,6,8]$. When choosing one ciphertext indexed by $P^{-1}(C_L)[1,4,6,8]$ and another ciphertext indexed by $P^{-1}(C_L)[1,4,6,8] \oplus P^{-1}(0x08, 0x10, 0, 0, 0, 0, 0, 0,)[1,4,6,8]$, we get a pair whose difference matches ΔL_{10}. Totally, we get 2^{79+x} pairs.
3. For each pair, check whether the input difference is one of the 2^{37} input differences. There are about $2^{79+x} \times 2^{37} \times 2^{-56} = 2^{60+x}$ pairs left.
4. For each pair and each possible ΔR_9, where $|\Delta R_9| = |\Delta OUTset| = 2^6$, do the following substeps.
   (a) In the 10th round, we know the input difference and output difference of the F function, so we deduce 64-bit key $kw_3 \oplus k_{10}$ by the difference distribution table of S-boxes.
   (b) We calculate the output value of the F function in 10th round by the values of $kw_3 \oplus k_{10}$. In the 9th round, deduce 32-bit key $(kw_4 \oplus k_9)[1,4,6,8]$ by the difference distribution table of S-boxes.
   (c) Increase the corresponding counter of 96-bit subkey $kw_3 \oplus k_{10}, (kw_4 \oplus k_9)[1,4,6,8]$, and then we obtain 2^6 subkeys for every pair.
5. Check all counters and generate a list L of the l candidate subkeys whose counters are the highest l values.

We choose $x = 33$, then there are $2^{111+33} \times 2^{37-56} = 2^{125}$ pairs, and each matches one of the 2^{37} input differences. The counter expectation for right key is $2^{125} \times 2^6 \times 2^{-126} = 2^5$, and the expectation of the counter for wrong key is about $2^{125} \times 2^6 \times 2^{-128} = 2^3$. We use the Blondeau et al.'s method [4] to compute the success rate. We know the number of differentials is $|\Delta| = 2^{37} \times 2^6 = 2^{43}$, the sum of the probability of all differentials is $\sum_{i=1}^{|\Delta|} Pr_i = 2^{-83}$, the number of pairs is $N_s = 2^{125}$, the bit number of guessed subkey is $n_k = 96$, and $l = 2^{40}$, then the success probability is:

$$Ps \approx 1 - G_*[G^{-1}(1 - \frac{l-1}{2^{n_k} - 2}) - 1] = 99.9\%,$$

where the definitions of functions $G_*()$ and $G^{-1}()$ refer to Appendix B.

Key-Dependent Multiple Differential Attack on the $PKSPACE$. If the key belongs to the $PKSPACE$, obviously this happens with significantly high probability of $1 - \frac{1}{2^{15}} \approx 99.99\%$, then 224 multiple differential attacks can recover the key. For a particular j of $KDset_i^j, i = 1, 2, \cdots 56$, the 56 multiple differential attacks use the differentials which have the common input truncated difference, the structures can be shared in the 56 times multiple differential attacks. So the data complexity of the attack is about $2^{56+33} \times 4 = 2^{91}$ chosen plaintexts. The time complexity is $2^{93+6} \times \frac{2}{10} \times 224 = 2^{104.5}$ 10-round encryptions. The memory complexity is 2^{96} which is used to store the counters for each of the 224 multiple differential attacks.

Key-Dependent Multiple Differential Attack on the Full $KSPACE$. For each one of $KDset_i^j$, $i = 1, 2 \cdots, 56, j = 1, 2, 3, 4$, we launch the above multiple differential attack. If one of the attack succeeds, the right key will be recovered; if all fail, we exhaustively search all the subkeys in the $RKset$.

Success Rate. If the correct key belongs to the remaining keyspace, then we will definitely recover the key when traversing the remaining keyspace. If the correct key does not belong to the remaining keyspace, then one of the 224 multiple differential attacks recovers the correct key with the probability of Ps. So the success rate of the is the minimum of 224 Ps, which is about 99.9%.

Complexity Analysis. The data complexity of the attack is about $2^{56+33} \times 4 = 2^{91}$ chosen plaintexts. The whole attack procedure includes 224 multiple differential attacks and traversing the remaining key set. The time complexity is $2^{60+33+6} \times \frac{2}{10} \times 224 + 2^{113} = 2^{104.5} + 2^{113} \approx 2^{113}$. The memory complexity is 2^{96} which is used to store the counters for each of the 224 multiple differential attacks.

The key-dependent multiple differential attack is also available to 11-round Camellia-192 and 12-round Camellia-256. However, we find that it is more efficient for the meet-in-the-middle attack on Camellia-192/256.

5 MITM Attacks on Reduced-Round Camellia-192/256

In this section, we first present a brief description of meet-in-the-middle attack, and then give the meet-in-the-middle attack on reduced-round Camellia combining with multiset, the differential enumeration technique, the relations of intermediate variables and subkeys etc.

5.1 Description of Meet-in-the-Middle Attack

For the meet-in-the-middle attack, the encryption cipher E_K is divided into three parts $E_K = E^2_{K_2} \circ E^m \circ E^1_{K_1}$, and there exists a specific property for the middle part E^m, which is used to construct a distinguisher and identify the correct key (K_1, K_2). The meet-in-the-middle methods we applied are similar to the MITM attaks on AES [12,11]. Therefore we introduce some definitions of $\delta-$set and multiset.

Definition 1. **($\delta-$set)** *The $\delta-$set is a set of 256 intermediate states of Camellia that one byte traverses all values (the active byte) and the other bytes are constants (the inactive bytes).*

Definition 2. **(Multiset of bytes [12])** *A multiset generalizes the set concept by allowing elements to appear more than once. Here, a multiset of 256 bytes can take as many as $\binom{511}{255} \approx 2^{506.7}$ different values.*

We explain the multiset with more details. Let a $\delta-$set $(X^0, \cdots, X^{255})$ be the inputs of E_m, where the j-th byte is a variable and the other bytes are kept constant. Let the i-th output byte of E_m be the output of the function. The outputs of function with the δ-set as inputs form a 2048-bit vector $E_K(X^0)[i] \| \cdots \| E_K(X^{255})[i]$ with ordered arrangement. However, if we don't consider the ordering of the output bytes, the 256-byte value will form a multiset $[E_K(X^0)[i] \oplus E_K(X^0)[i], E_K(X^0)[i] \oplus E_K(X^1)[i], \cdots, E_K(X^0)[i] \oplus E_K(X^{255})[i]]$. However, given two random functions f, g: $\mathbb{F}_{256} \to \mathbb{F}_{256}$, the multisets $(f(X^0), \cdots, f(X^{255}))$ and $(g(X^0), \cdots, g(X^{255}))$ are equal with a probability smaller than $2^{-467.6}$ (but not $2^{-506.17}$). For more details, we refer to [11].

The key part of the meet-in-the-middle attack on AES is to construct a function for the input active byte and one of the output bytes of E_m, and reduce the number of the function parameters by specific truncated differential, which decides the size of the multiset. Based on the subcipher E_m, a few rounds is extended at the top and bottom of E_m, i.e. the cipher $E_K = E^2_{K_2} \circ E^m \circ E^1_{K_1}$. The attack procedure is described in Algorithm 1.

It is noticed that the number of values for a good multiset is much less than $2^{467.6}$. The precomputation phase is to compute all the values of multiset in a table.

5.2 MITM Attack on 12-Round Camellia-192

This section introduces a 7-round property starting from the third round and ending at the ninth round which is described in Property 2 outlined in Fig. 3. The active byte of $\delta-$set is defined at the first byte of the input of the third round $R_2[1]$.

Algorithm 1 The Main Procedure of Meet-in-the-Middle Attack

```
Precomputation phase: compute all values of the output sequence of the function constructed on $E_m$, and
    store them in a hash table.
Online phase:
1: Encrypt enough chosen plaintexts such that there exists a pair satisfying the specific differential.
2: Guess values of the subkeys $K_1$ and $K_2$ to find a pair satisfying the specific truncated differential.
3: Construct a $\delta$-set based on the pair, and partially decrypt to get the corresponding 256 plaintexts.
4: Obtain the corresponding 256 plaintext-ciphertext pairs from the collected data. Then partially decrypt the
    ciphertexts to get the corresponding 256-byte value of the output sequence of $E_m$.
5: If a sequence value lies in the precomputation table, the guessed $K_1$ and $K_2$ may be right key.
6: Exhaustively search the remaining subkeys to obtain the right key.
```

Property 2. Encrypt 2^8 values of the $\delta-$set through 7-round Camellia-192 starting from the third round, where $R_2[1]$ is the active byte, in the case that a pair of the $\delta-$set conforms to the truncated differential outlined in Fig 3, then the corresponding multiset of bytes $(P^{-1}(\Delta L_8))[6]$ only takes about 2^{128} instead of $2^{467.6}$ values on average.

It is obvious that, the computation of the multiset of bytes $(P^{-1}(\Delta L_8))[6]$ associated with a $\delta-$set is determined by a 36-byte intermediate variable

$$X_4[1]\|X_5[1,2,3,5,8]\|X_6\|kf_1\|kf_2\|X_7[2,3,5,7,8]\|X_8[6].$$

The main work is to prove that there are only 16 byte variables needed to compute the multiset.

Proof. If a pair of the δ-set conforms the truncated differential as in Fig. 3, the 18-byte variable $X_4[1]\|X_5[1,2,3,5,8]\|X_6\|X_7[2,3,5,8]$ is determined by the 9-byte difference $\Delta X_4[1]\|\Delta Y_4[1]\|\Delta Y_5[1,2,3,5,8]\|\Delta X_8[1]\|\Delta Y_8[1]$ and 128-bit subkey $kf_1\|kf_2$. Here, the value $X_4[1]$ is deduced from the differences $\Delta X_4[1]$ and $\Delta Y_4[1]$. Similarly, the value $X_5[1,2,3,5,8]$ is obtained by the differences $\Delta Y_4[1], \Delta Y_5[1,2,3,5,8]$. In the backward direction, the difference ΔY_6 is computed by $\Delta Y_4[1], \Delta Y_8[1]$ and kf_1 since $\Delta L_4 = P(\Delta Y_4)$ and $\Delta L_6 = P(\Delta Y_8)$ in this case. The difference ΔX_6 is computed by $\Delta X_4[1], \Delta Y_5[1,2,3,5,8]$, which is used to deduce the value X_6. Similarly, the difference ΔY_7 is computed by the difference $\Delta X_4[1], \Delta Y_5[1,2,3,5,8]$, $\Delta X_8[1]$ and kf_2, which helps us deduce $X_7[2,3,5,8]$ owing to $\Delta X_7 = P(\Delta Y_8)$.

Since $kf_1\|kf_2$ has only 64-bit information by key schedule, the total 36-byte variable is computed by 19-byte variable $\Delta X_4[1]\|\Delta Y_4[1]\|\Delta Y_5[1,2,3,5,8]\|\Delta X_8[1]\|\Delta Y_8[1]\|X_7[7]\|X_8[6]\|kf_1$ in such case.

However, for every 19-byte variable, we find that the difference ΔY_7 equals to $P^{-1}(FL^{-1}(P(\Delta Y_5) \oplus \Delta L_3)) \oplus P^{-1}(\Delta L_7)$, where the probability that $\Delta Y_7[4,6,7]$ equals to 0 is 2^{-24}. So there are only about 2^{128} possible values for 36-byte intermediate variable, actually. □

Based on the 7-round property, we extend two rounds on the top and three rounds on the bottom to attack the 12-round Camellia-192, see Fig.4. To reduce the computation complexity of the 12-round attack on Camellia-192, we retrieve the equivalent keys k'_1, k'_2, k'_{10}, k'_{11}, k'_{12}, and then deduce the master key. The equivalent keys are defined as $k'_1 = k_1 \oplus kw_1$, $k'_2 = k_2 \oplus kw_2$, $k'_{12} = k_{12} \oplus kw_4$, $k'_{11} = k_{11} \oplus kw_3$, and $k'_{10} = k_{10} \oplus kw_4$. Note that the master key could be deduced by the equivalent key using the method introduced in [7].

The key recovery is also composed of two phases: precomputation phase and online phase. In the precomputation phase, we get 2^{128} possible values of multiset as described in Property 2, and store them in a hash table $\mathcal{H}$. The attack procedure of the online phase is similar to Algorithm 1. However we take a balance of the time complexity of Step 2 and Step 3. We guess some related subkeys to find the possible pairs which may satisfy the truncated differential, and then construct the $\delta-$set to get their plaintexts.

The attack procedure of online phase is described as follows.

1. Choose 2^{57} structures of plaintexts, and each structure contains 2^{56} plaintexts that satisfy $L_0 = (\alpha, \alpha \oplus x_1, \alpha \oplus x_2, x_3, \alpha \oplus x_4, x_5, x_6, \alpha \oplus x_7)$, $R_0 = P(\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, y_1, y_2, \beta_6)$, where $x_i (i = 1, ..., 7)$, y_1 and

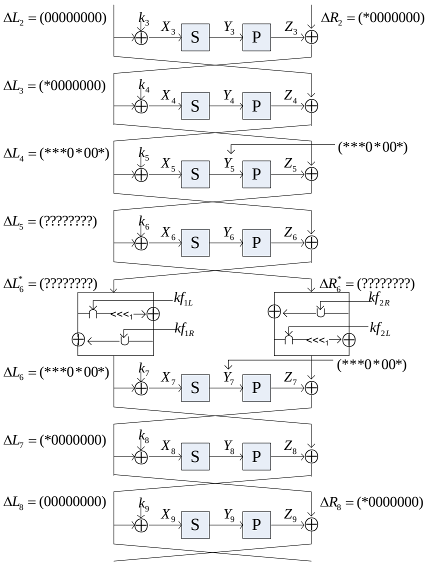
$\Delta L_2 = (00000000)$
$\Delta R_2 = (*0000000)$
$\Delta L_3 = (*0000000)$
$\Delta L_4 = (***0*00*)$
$(***0*00*)$
$\Delta L_5 = (????????)$
$\Delta L_6^* = (????????)$
$\Delta R_6^* = (????????)$
kf_{1L}
kf_{1R}
kf_{2R}
kf_{2L}
$<<<_1$
$\Delta L_6 = (***0*00*)$
$(***0*00*)$
$\Delta L_7 = (*0000000)$
$\Delta L_8 = (00000000)$
$\Delta R_8 = (*0000000)$
k_3 X_3 S Y_3 P Z_3
k_4 X_4 S Y_4 P Z_4
k_5 X_5 S Y_5 P Z_5
k_6 X_6 S Y_6 P Z_6
k_7 X_7 S Y_7 P Z_7
k_8 X_8 S Y_8 P Z_8
k_9 X_9 S Y_9 P Z_9

Fig. 3. The Truncated Differential of 7-round Camellia-192

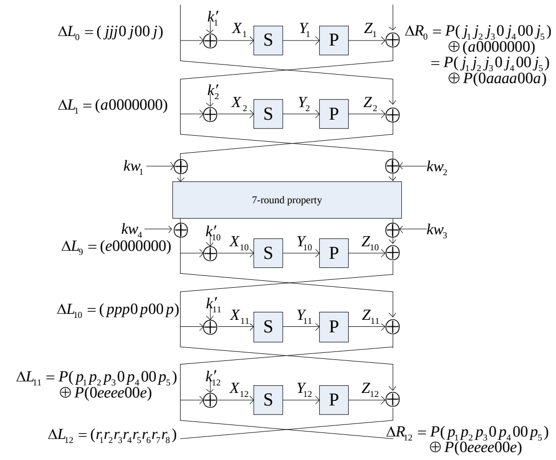
$\Delta L_0 = (jjj0j00j)$
$\Delta R_0 = P(j_1 j_2 j_3 0 j_4 00 j_5)$
$\oplus (a0000000)$
$= P(j_1 j_2 j_3 0 j_4 00 j_5)$
$\oplus P(0aaaa00a)$
$\Delta L_1 = (a0000000)$
kw_1
kw_2
7-round property
kw_4
kw_3
$\Delta L_9 = (e0000000)$
$\Delta L_{10} = (ppp0p00p)$
$\Delta L_{11} = P(p_1 p_2 p_3 0 p_4 00 p_5)$
$\oplus P(0eeee00e)$
$\Delta L_{12} = (r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8)$
$\Delta R_{12} = P(p_1 p_2 p_3 0 p_4 00 p_5)$
$\oplus P(0eeee00e)$
k'_1 X_1 S Y_1 P Z_1
k'_2 X_2 S Y_2 P Z_2
k'_{10} X_{10} S Y_{10} P Z_{10}
k'_{11} X_{11} S Y_{11} P Z_{11}
k'_{12} X_{12} S Y_{12} P Z_{12}

Fig. 4. The MITM Attack on 12-round Camellia-192

y_2 are constants, but α, β_j $(j = 1, ..., 6)$ take all possible values. Ask for corresponding ciphertexts for each structure, compute $P^{-1}(R_{12})$ and store the plaintext-ciphertext pairs $L_0 \| R_0 \| L_{12} \| R_{12}$ in a hash table indexed by 16-bit value $(P^{-1}(R_{12}))[6, 7]$. Hence, there are $2^{57} \times 2^{111} \times 2^{-16} = 2^{152}$ pairs whose differences satisfy $P^{-1}(\Delta R_{12})[6, 7] = 0$ on average.

2. For every pair, do the following substeps to find a pair with corresponding subkeys conforming the truncated differential.

   (a) For $l = 2, 3, 4, 5, 6, 7, 8$, guess the 8-bit value of $k'_{12}[l]$ one by one. Partially decrypt the ciphertext $R_{12}[l]$ and keep only the pairs which satisfy $\Delta Y_{12}[l] = P^{-1}(\Delta L_{12}[l])$. The expected number of pairs left is about $2^{152} \times 2^{7\times(-8)} = 2^{96}$. After that guess $k'_{12}[1]$, partially decrypt the remaining pairs to get the value L_{10}.

   (b) For $l = 2, 3, 5, 8$, guess the 8-bit value of $k'_{11}[l]$. Compute the intermediate value $Y_{11}[l]$ and eliminate the pairs whose intermediate values do not satisfy $\Delta Y_{11}[l] = P^{-1}(\Delta R_{12})[l] \oplus P^{-1}(\Delta R_{12})[4]$(see Observation 2). Then guess $k'_{11}[1]$ and keep the pairs making $\Delta Y_{11}[1] = P^{-1}(\Delta R_{12})[1]$ hold. The expected number of remaining pairs is $2^{96} \times 2^{-40} = 2^{56}$.

   (c) Similarly, for $l = 1, 2, 3, 5, 8$, guess $k'_1[l]$ and discard the pairs which do not make the equations $\Delta Y_1[1] = P^{-1}(\Delta R_0)[1]$ and $\Delta Y_1[l] = P^{-1}(\Delta R_0)[l] \oplus P^{-1}(\Delta R_0)[4]$(see Observation 3) hold for $l = 2, 3, 5, 8$. Then the expected number of remaining pairs is $2^{56} \times 2^{-40} = 2^{16}$.

3. For the 2^{16} remaining pairs, if we want to find the pair in content with the 7-round truncated differential, we have to guess 64-bit equivalent key $k'_1[4, 6, 7] \| k'_2[1] \| k'_{11}[4, 6, 7] \| k'_{10}[1]$ under each 144-bit subkey guess. Obviously, it is infeasible, since the time complexity is greater than exhaustively searching in such case. However, there are about a pair satisfying the truncated differential, for the probability of the truncated differential occuring is about 2^{-16} for the remaining pairs. Therefore we construct the $\delta-$set for all 2^{16} pairs. If the guessed 144-bit key information is correct, then there should exist a pair to conform the truncated differential, and the corresponding value of the multiset should exist in the table $\mathcal{H}$. We construct a $\delta-$set for every remaining pair under 144-bit key guesses in the following.

   (a) According to the differences $\Delta L_0[1]$ and $P^{-1}(\Delta R_0)[4]$, deduce the intermediate value $X_2[1] \| Y_2[1]$ of the pair by the difference distribution table of S-box s_1.

(b) For the pair $(L_0\|R_0, L'_0\|R'_0)$ corresponding to $(X_2[1], X'_2[1])$, change the value $X'_2[1]$ to a different value $X''_2[1]$, compute $\Delta Y'_2[1] = s_1(X''_2[1]) \oplus s_1(X_2[1])$, and get the difference $\Delta L'_0[1,2,3,5,8]$. Then get the left half of the plaintext $L''_0 = L_0 \oplus \Delta L'_0$.

(c) Compute the difference $\Delta Y'_1[1,2,3,5,8]$ by the guessed subkey $k'_1[1,2,3,5,8]$. Then obtain the difference $\Delta R'_0$ and get the right half part $R''_0 = R_0 \oplus \Delta R'_0$. Here we get a new plaintext (L''_0, R''_0) of the $\delta-$set.

(d) Compute all left 253 values of $X_2[1]$ to obtain all plaintexts of the $\delta-$set, and identify the corresponding ciphertexts.

4. For each $\delta-$set under 144-bit key guesses, compute the intermediate value $Y_{11}[2,3,5,8]$, $P^{-1}(L_{10})[6]$ for every plaintext-ciphertext pairs by above guessed subkey. Guess 8-bit key $k'_{11}[7]$ to compute the value $X_{10}[6]$.
5. Guess 8-bit key $k'_{10}[6]$ to compute the multiset of byte $(P^{-1}(\Delta L_8))[6] = \Delta Y_{10}[6] \oplus P^{-1}(\Delta L_{10})[6]$. Detect whether it belongs to $\mathcal{H}$. Here, we need to detect 2^{16} values of multiset for every 160-bit guessed key. Then find the correct subkey if one of 2^{16} values belongs to $\mathcal{H}$. Note that the probability that a wrong value of multiset could pass the check is about $2^{128} \times 2^{-467.6} = 2^{-339.6}$.
6. Compute the related part of the master key by the equivalent keys k'_1, k'_2, k'_{10}, k'_{11}, k'_{12}, and search the unknown part.

Complexity Analysis. The precomputation phase needs about $2^{128} \times 2^8$ computations and 2^{130} 128-bit memories. Step 1 needs about 2^{113} encryptions. We also need 2^{113} 128-bit memories to store all plaintext-ciphertext pairs. The complexity of step 2 is dominated by substep 2.(c), which needs about 2^{168} computations. Step 3 needs about 2^{168} simple computations to construct 2^{16} δ-for every 144-bit key guess. Step 4 needs about $2^{160} \times 2^8 \times 2^8 \times 2^{-3} = 2^{173}$ 12-round encryptions. The time complexity of step 5 is equivalent to $2^{176} \times 2^8 \times 2^{-4} = 2^{180}$ 12-round encryptions. In total, the time complexity of the attack is about 2^{180} encryptions, the data complexity is about 2^{113} chosen plaintexts, the memory complexity is about 2^{130} 128-bit.

5.3 The Attack on 13-Round Camellia-256

This section introduces an 8-round property of Camellia-256, which starts from the fifth round and ends at the twelfth round introduced by Property 3. The truncated differential used in this section is outlined in Fig. 5 of Appendix A, the active byte of the $\delta-$set is located at $L'_{12}[5]$, and the corresponding byte of multiset is defined as $P^{-1}(\Delta L_4)[1]$.

Property 3. Decrypt 2^8 values of the $\delta-$set through 8-round Camellia-256 starting from the 12-th round, where $L_{12}[5]$ is the active byte, in the case that a pair of the $\delta-$set conforms to the 8-round truncated differential outlined in Fig 5 of Appendix A, then the corresponding multiset of bytes $(P^{-1}(\Delta L_4))[1]$ only takes about 2^{225} instead of $2^{467.6}$ values on average.

The sketch of Property 3 is similar to Property 2, we give the proof in Appendix A.

We mount a 13-round attack on Camellia-256 by adding four rounds in the forward and one round in the backward of the 8-round Camellia (see Fig. 6 in Appendix A). We also recover the equivalent keys k'_1, k'_2, k'_3, k'_4, k'_{13}, and then deduce the master key, where the equivalent keys are defined as $k'_1 = k_1 \oplus kw_1$, $k'_2 = k_2 \oplus kw_2$, $k'_3 = k_3 \oplus kw_1$, $k'_4 = k_4 \oplus kw_2$, and $k'_{13} = k_{13} \oplus kw_4$. The attack is worked in the chosen-ciphertext model. In the precomputation phase, we compute all 2^{225} possible values of multiset, and store them in a hash table. The attack procedure of the online phase is described as follows.

1. Select 2^{81} structures of ciphertexts, and each structure contains 2^{32} ciphertexts

$$L_{13} = P(\alpha_1, x_1, x_2, x_3, \alpha_2, x_4, x_5, x_6), R_{13} = (\beta_1, y_1, y_2, y_3, \beta_2, y_4, y_5, y_6),$$

where x_i and y_i $(i = 1, ..., 6)$ are fixed values, and α_j, β_j $(j = 1, 2)$ take all the possible values. Decrypt and obtain the corresponding plaintexts. There are 2^{144} pairs totally.

2. Compute $P^{-1}(\Delta L_1)$ for every pair by guessing 64-bit subkey k'_1, eliminate the pairs which do not satisfy $P^{-1}(\Delta L_1)[6,7] = 0$. There are $2^{144-16} = 2^{128}$ pairs left on average.
3. For $l = 2,3,4,5,6,7,8$, guess the 8-bit value of $k'_2[l]$ one by one, compute the value $Y_2[l]$, and keep the pairs which make $\Delta Y_2[l] = P^{-1}(\Delta L_0[l])$ hold. Then guess $k'_2[1]$ to compute L_2. The number of pairs kept about $2^{128-7*8} = 2^{72}$.
4. For $l = 2,3,5,8$, guess the 8-bit value of $k'_3[l]$. Compute $Y_3[l]$ and discard the pairs which do not conform $\Delta Y_3[l] = P^{-1}(\Delta L_1)[l] \oplus P^{-1}(\Delta L_1)[4]$(see Observation 3). Then guess $k'_3[1]$ and keep the pairs satisfying $\Delta Y_3[1] = P^{-1}(\Delta L_1)[1]$. There are 2^{32} pairs remain for every 168-bit guessed key after this step.
5. For $l = 1,5$, guess the 8-bit value of $k'_{13}[l]$, and compute the value $\Delta Y_{13}[l]$. Delete the pairs which do not content $\Delta Y_{13}[l] = P^{-1}(\Delta L_{13}[l])$. Then guess $kf_{3R}[1]$, compute $\Delta L^*_{12}[1]$ by using Observation 1, and delete the pairs when $\Delta L^*_{12}[1] \neq 0$. Hereafter, the expected number of remaining pairs is about 2^8.
6. Compute the value L_3 by guessing 24-bit subkey $k'_3[4,6,7]$, and then deduce the value of subkey $k'_4[1]$ for every pair.
7. Construct the $\delta-$set for every pair, and compute corresponding value of multiset. Detect whether it belongs to the precomputed table and find the possible correct key.
8. Compute the related part of the master key by the correct equivalent keys k'_1, k'_2, k'_3, k'_4, k'_{13}, and search the unknown part.

Complexity Analysis. The time complexity of precomputation phase is about $2^{225} \times 2^8 \times 2^{-1} = 2^{232}$ 13-round encryptions. The memory complexity is about $2^{225} \times 2^2 = 2^{227}$ 128-bit. The time complexity of online phase is bounded to that of Step 6, which costs $2^{224} \times 2^8 \times 2^{-2} = 2^{230}$ 13-round encryptions, which also needs 2^{113} chosen ciphertexts to find the correct pairs. In total, the data, time and memory complexities of the attack, including the precomputation phase, are 2^{113} chosen ciphertexts, $2^{232.3}$ encryptions and 2^{227} 128-bit memories, respectively.

6 Conclusion

In this paper, we give the key-dependent multiple differential attack and meet-in-the-middle attacks on reduced-round Camellia-128/192/256. For key-dependent multiple differential attack, we divide the keyspace into 224+1 subsets to ensure the input and output difference of FL^{-1} function same, and then produce 224 types of corresponding 8-round differentials, and each type of differentials include 2^{43} differentials. Based on 8-round multiple differentials, we attack 10-round Camellia-128 for every key subsets, which works for about 99.99% of the keys, and exhaustively search for the remaining fraction of $1/2^{15}$ of the keyspace. This attack is more efficient than previous 10-round attack on Camellia-128.

Furthermore, we also discuss the security of reduced-round Camellia-192/256 against the meet-in-the-middle attack. Considering differential enumeration technique, multisets, intermediate variable relations and key relations etc, we mount the attacks on 12-round Camellia-192 and 13-round Camellia-256 with non-marginal complexities.

7 Acknowledgments

We would like to thank anonymous reviewers for their very helpful comments on the paper. This work is supported by the National Natural Science Foundation of China (No. 61133013) and 973 Program (No.2013CB834205), and the National Natural Science Foundation of China (No. 61402256 and 61272035).

References

1. 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

2. 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.E. (eds.) SAC 2000. Lecture Notes in Computer Science, vol. 2012, pp. 39–56. Springer (2001)
3. Ben-Aroya, I., Biham, E.: Differential cryptanalysis of lucifer. In: Advances in CryptologyCRYPTO93. pp. 187–199. Springer (1994)
4. Blondeau, C., Gérard, B.: Multiple Differential Cryptanalysis: Theory and Practice. In: Joux, A. (ed.) Fast Software Encryption - FSE 2011. Lecture Notes in Computer Science, vol. 6733, pp. 35–54. Springer (2011)
5. Bogdanov, A., Geng, H., Wang, M., Wen, L., Collard, B.: Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA. In: Lange, T., Lauter, K., Lisonek, P. (eds.) SAC 2013 to appear (2013)
6. Boura, C., Naya-Plasencia, M., Suder, V.: Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. pp. 179–199 (2014), http://dx.doi.org/10.1007/978-3-662-45611-8_10
7. Chen, J., Jia, K., Yu, H., Wang, X.: New Impossible Differential Attacks of Reduced-Round Camellia-192 and Camellia-256. In: Parampalli, U., Hawkes, P. (eds.) ACISP 2011. Lecture Notes in Computer Science, vol. 6812, pp. 16–33. Springer (2011)
8. Chen, J., Li, L.: Low Data Complexity Attack on Reduced Camellia-256. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. Lecture Notes in Computer Science, vol. 7372, pp. 101–114. Springer (2012)
9. Cryptography Research and Evaluation Committees: Http://www.cryptrec.go.jp/english/index.html
10. Demirci, H., Selçuk, A.A.: A Meet-in-the-Middle Attack on 8-Round AES. In: Nyberg, K. (ed.) FSE 2008. Lecture Notes in Computer Science, vol. 5086, pp. 116–126. Springer (2008)
11. Derbez, P., Fouque, P.A., Jean, J.: Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 7881, pp. 371–387. Springer (2013)
12. Dunkelman, O., Keller, N., Shamir, A.: Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 158–176. Springer (2010)
13. Hatano, Y., Sekine, H., Kaneko, T.: Higher Order Differential Attack of Camellia (II). In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. Lecture Notes in Computer Science, vol. 2595, pp. 129–146. Springer (2003)
14. International Organization for Standardization(ISO): International Standard- ISO/IEC 18033-3, Information technology-Security techniques-Encryption algorithms -Part 3: Block ciphers (2010)
15. Kanda, M., Matsumoto, T.: Security of Camellia against Truncated Differential Cryptanalysis. In: Matsui, M. (ed.) Fast Software Encryption - FSE 2002. Lecture Notes in Computer Science, vol. 2355, pp. 286–299. Springer (2001)
16. Knudsen, L.R., Rijmen, V.: On the decorrelated fast cipher (dfc) and its theory. In: Fast Software Encryption. pp. 81–94. Springer (1999)
17. Kühn, U.: Improved Cryptanalysis of MISTY1. In: Daemen, J., Rijmen, V. (eds.) Fast Software Encryption - FSE 2002. Lecture Notes in Computer Science, vol. 2365, pp. 61–75. Springer (2002)
18. Lee, S., Hong, S., Lee, S., Lim, J., Yoon, S.: Truncated Differential Cryptanalysis of Camellia. In: Kim, K. (ed.) ICISC 2001. Lecture Notes in Computer Science, vol. 2288, pp. 32–38. Springer (2002)
19. Lei, D., Li, C., Feng, K.: New Observation on Camellia. In: Preneel, B., Tavares, S.E. (eds.) SAC 2005. Lecture Notes in Computer Science, vol. 3897, pp. 51–64. Springer (2006)
20. Lei, D., Li, C., Feng, K.: Square Like Attack on Camellia. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. Lecture Notes in Computer Science, vol. 4861, pp. 269–283. Springer (2007)
21. Liu, Y., Li, L., Gu, D., Wang, X., Liu, Z., Chen, J., Li, W.: New Observations on Impossible Differential Cryptanalysis of Reduced-Round Camellia. In: Canteaut, A. (ed.) Fast Software Encryption 2012. Lecture Notes in Computer Science, vol. 7549, pp. 90–109. Springer (2012)
22. Lu, J., Kim, J., Keller, N., Dunkelman, O.: Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1. In: Malkin, T. (ed.) CT-RSA 2008. Lecture Notes in Computer Science, vol. 4964, pp. 370–386. Springer (2008)
23. Lu, J., Wei, Y., Fouque, P.A., Kim, J.: Cryptanalysis of reduced versions of the Camellia block cipher. IET Information Security 6(3), 228–238 (2012)
24. Lu, J., Wei, Y., Kim, J., Pasalic, E.: The Higher-Order Meet-in-the-Middle Attack and Its Application to the Camellia Block Cipher. In: Galbraith, S.D., Nandi, M. (eds.) Progress in Cryptology - INDOCRYPT 2012. Lecture Notes in Computer Science, vol. 7668, pp. 244–264. Springer (2012)

25. Mala, H., Shakiba, M., Dakhilalian, M., Bagherikaram, G.: New Results on Impossible Differential Cryptanalysis of Reduced-Round Camellia-128. In: Jacobson, M., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. Lecture Notes in Computer Science, vol. 5867, pp. 281–294. Springer (2009)
26. Shirai, T.: Differential, Linear, Boomerang and Rectangle Cryptanalysis of Reduced- Round Camellia. In: the Third NESSIE Workshop (2002)
27. 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.) Advances in Cryptology - ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 193–207. Springer (2001)
28. Sun, X., Lai, X.: The key-dependent attack on block ciphers. In: Advances in Cryptology–ASIACRYPT 2009, pp. 19–36. Springer (2009)
29. Wang, M., Sun, Y., Tischhauser, E., Preneel, B.: A model for structure attacks, with applications to present and serpent. In: Fast Software Encryption. pp. 49–68. Springer (2012)
30. Wu, W., Feng, D., Chen, H.: Collision Attack and Pseudorandomness of Reduced-Round Camellia. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. Lecture Notes in Computer Science, vol. 3357, pp. 252–266. Springer (2004)
31. Wu, W., Zhang, W., Feng, D.: Impossible Differential Cryptanalysis of Reduced-Round ARIA and Camellia. J. Comput. Sci. Technol. 22(3), 449–456 (2007)

A The Proof of Property 3

By Property 3, the 8-round property starts from the fifth round and ends at the twelfth round. The active byte of $\delta-$set is defined at the first bytes of the input of the third round $L_{12}[5]$, i.e., $L_{12}[5]$ is the active byte. Considering to decrypt 2^8 values of the $\delta-$set through 8-round Camellia-256, in the case of that a pair of $\delta-$set conforms to the 8-round truncated differential outlined in Fig. 5, we prove the corresponding multiset of bytes $P^{-1}(\Delta L_4)[1]$ has 2^{225} values.

Proof. If $\Delta L_{12}[5] \neq 0$ and there is no difference on the other bytes of the input (L_{12}, R_{12}), $(P^{-1}(\Delta L_4))[1]$ is determined by 321-bit intermediate variable

$$X_{11}[5]\|X_{10}[2,3,4,6,7,8]\|X_9\|X_8\|X_7\|kf_1\{9-33,42-64\}\|kf_{2L}[1]\|kf_{2R}[1]\|kf_{2L}\{9\}\|X_6[1].$$

However, if there exists a pair satisfying the truncated differential as described in Fig. 6, the 312-bit intermediate variable

$$X_{11}[5]\|X_{10}[2,3,4,6,7,8]\|X_9\|X_8\|X_7\|X_6[1]\|kf_1\{9-33,42-64\}\|kf_{2L}[1]$$

is determined by 216-bit variable

$$\Delta X_{11}[5]\|\Delta Y_{11}[5]\|\Delta Y_{10}[2,3,4,6,7,8]\|\Delta Y_9\|\Delta X_6[1]\|\Delta Y_6[1]\|kf_1\|kf_{2L}[1].$$

Besides, 9-bit value $kf_{2R}[1]\|kf_{2L}\{9\}$ are also necessary to compute $(P^{-1}(\Delta L_4))[1]$. Hence the multiset of bytes $(P^{-1}(\Delta L_4))[1]$ could be computed by traversing all the 225-bit intermediate variable

$$\mathcal{V} = \Delta X_{11}[5]\|\Delta Y_{11}[5]\|\Delta Y_{10}[2,3,4,6,7,8]\|\Delta Y_9\|\Delta X_6[1]\|\Delta Y_6[1]\|kf_1\|kf_{2L}[1]\|kf_{2R}[1]\|kf_{2L}\{9\}.$$

That is to say there are about 2^{225} possible values of multiset totally. □

B Blondeau *et al.'s* Multiple Differential Cryptanalysis

Blondeau *et al.'s* propose multiple differential cryptanalysis in 2011. A precise analytical model as well as formulas to compute success rate has been given. The success rate of a multiple differential attack can be calculated as follows:

$$P_S \approx 1 - G_*[G^{-1}(1 - \frac{l-1}{2^{n_k}-2}) - 1/N_s], \tag{1}$$

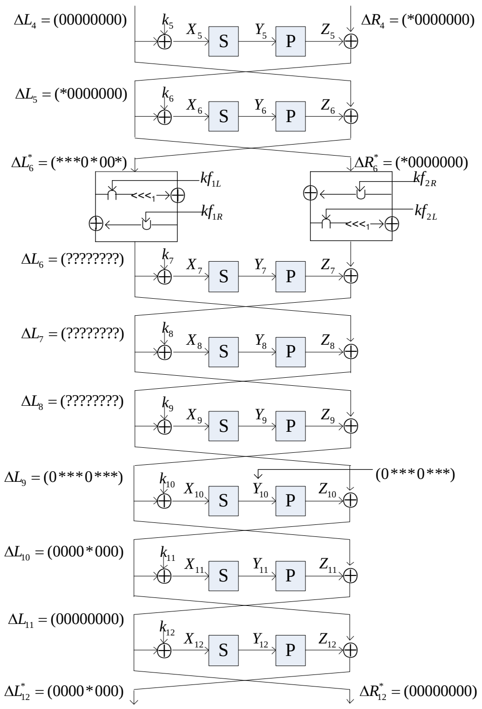

Fig. 5. The 8-round Truncated Differential of Camellia-256

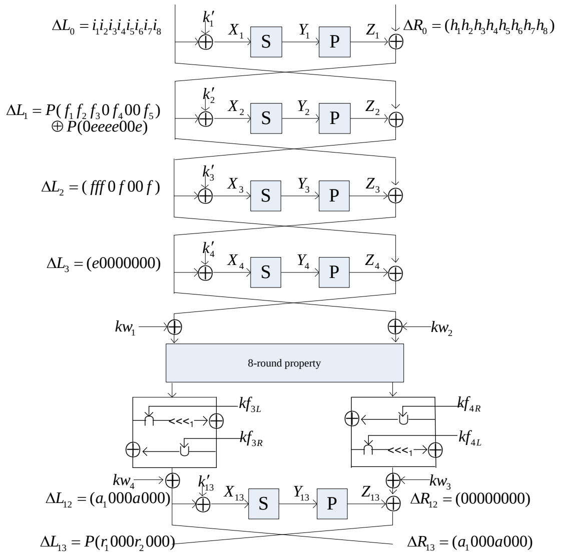

Fig. 6. The Meet-in-the-Middle Attack on 13-round Camellia-256

where n_k is the number of key candidates, l is the size of list to keep and N_s is the number of samples. The function G and G^* are defined as follows:

$$\begin{aligned} G_*(\tau) &\overset{def}{=} G(\tau, p_*) \\ G(\tau) &\overset{def}{=} G(\tau, p) \end{aligned} \tag{2}$$

where $p_* = \Sigma_{i,j} p_*^(i,j)$ and $p = \frac{|\Delta|}{2^m |\Delta_0|}$. $\Sigma_{i,j}$ is the sum of probability of all differential characters and m is the block size. $|\Delta|$ denotes the number of input difference values while $|\Delta_0|$ is the number of differentials. $G^(- 1)$ is defined by $G^(- 1)(y) = \min x | G(x) > y$. $G(\tau, p_*)$ and $G(\tau, p)$ can be calculated as follows:

$$G(\tau, q) = \begin{cases} G_-(\tau, q) & if \qquad \tau < q - 3\sqrt{q/N_s}, \\ 1 - G_+(\tau, q) & if \qquad \tau > q + 3\sqrt{q/N_s}, \\ G_p(\tau, q) & otherwise, \end{cases} \tag{3}$$

where $G_p(\tau, q)$ is the cumulative distribution function of the Poisson distribution with parameter qN_s. $G_-(\tau, q)$ and $G_+(\tau, q)$ are defined as follows:

$$G_-(\tau, q) = e^{(-N_s D(\tau \|, q))}[\frac{q\sqrt{1-\tau}}{(q-\tau)\sqrt{2\pi\tau N_s}} + \frac{1}{\sqrt{8\pi\tau N_s}}] \tag{4}$$

$$G_+(\tau, q) = e^{(-N_s D(\tau \|, q))}[\frac{(1-q)\sqrt{\tau}}{(q-\tau)\sqrt{2\pi\tau N_s}} + \frac{1}{\sqrt{8\pi\tau N_s}}] \tag{5}$$

where $D(\tau \parallel q)$ is the Kullback-Leibler divergence defined by:

$$D(\tau \parallel q) = \tau ln(\frac{\tau}{q}) + (1+\tau) ln(\frac{1-\tau}{1-q}) \tag{6}$$

C Figure of the Camellia Algorithm

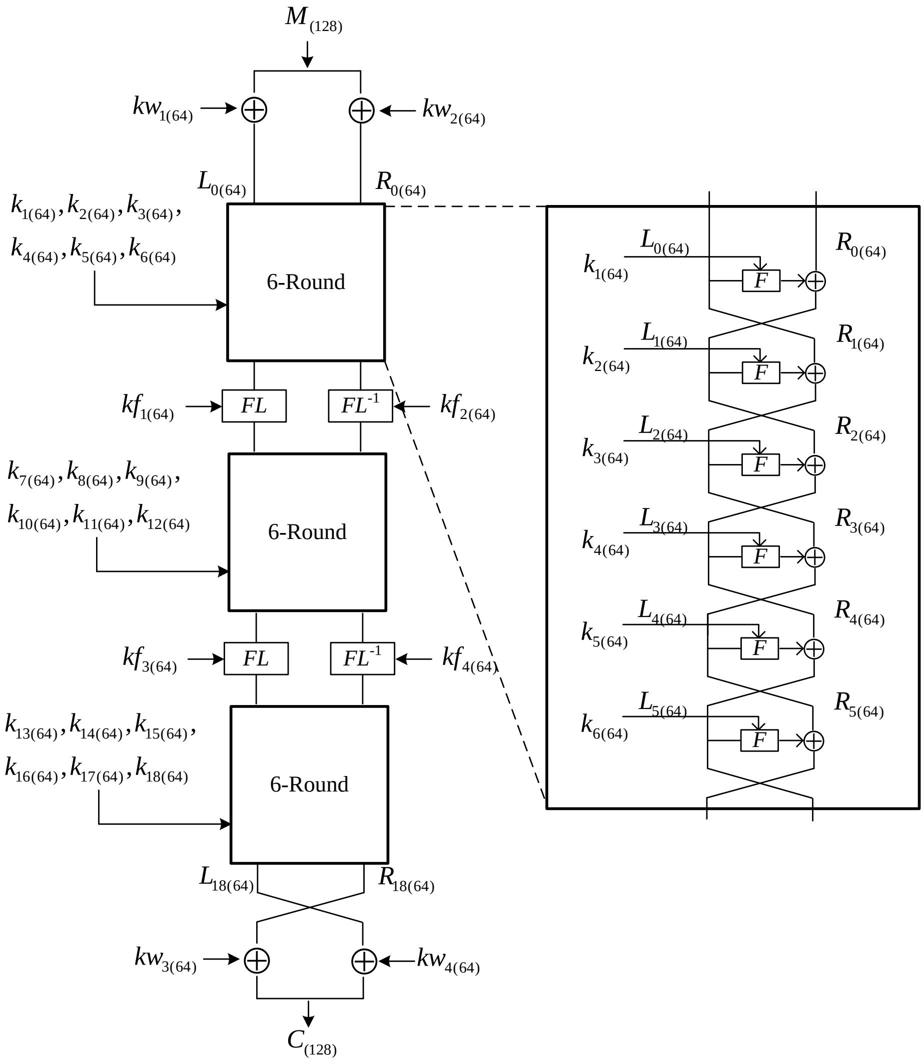

Fig. 7. : Encryption procedure of Camellia for 128-bit keys