Improved Meet-in-the-Middle Attacks on Reduced-Round Camellia-192/256

Leibo Li[1] and Keting Jia[2]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education,
School of Mathematics, Shandong University, Jinan 250100, China
lileibo@mail.sdu.edu.cn

[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
ktjia@mail.tsinghua.edu.cn

Abstract. Camellia is one of the widely used block ciphers, which has been selected as an international standard by ISO/IEC. In this paper, we focus on the key-recovery attacks on reduced-round Camellia-192/256 with meet-in-the-middle methods. We utilize multiset and the differential enumeration methods which are popular to analyse AES in the recent to attack Camellia-192/256. We propose a 7-round property for Camellia-192, and achieve a 12-round attack with 2^{180} encryptions, 2^{113} chosen plaintexts and 2^{130} 128-bit memories. Furthermore, we present an 8-round property for Camellia-256, and apply it to break the 13-round Camellia-256 with $2^{232.7}$ encryptions, 2^{113} chosen ciphertexts and 2^{227} 128-bit memories.

Keywords: Block Cipher, Meet-in-the-Middle, Camellia, Truncated Differential.

1 Introduction

The block cipher Camellia is a 128-bit block cipher with variable key length of 128, 192, 256, which are denoted as Camellia-128, Camellia-192 and Camellia-256, respectively. Camellia was proposed by NTT and Mitsubishi in 2000 [2], and was selected as one of e-government recommended ciphers by CRYPTREC in 2013 [7], NESSIE block cipher portfolio in 2003 [27] and international standard by ISO/IEC 18033-3 in 2005 [13].

Many methods of cryptanalysis were applied to attack reduced-round Camellia in previous years, such as higher order differential attack [12], linear and differential attacks [28], truncated differential attack [29,16,14], collision attack [30], square attack [17,18], meet-in-the-middle attack [6,24,25], impossible differential attack [5,26,29,32,23,31,22,3,19,21,20] and zero-correlation linear cryptanalysis [4] etc. Resistance to some general cryptanalysis methods, the FL/FL^{-1} layers are inserted every 6 rounds to provide non-regularity across rounds, which are constructed by logical operations AND, OR, XOR and one bit rotation. So many previous papers proposed attacks on simplified versions of Camellia, which did not take the FL/FL^{-1} layers and the whitening layers into account. In our work, we will mainly focus on the original Camellia which is starting from the first round and includes the FL/FL^{-1} layers and whitening key. In recent years, some attacks on reduced-round Camellia have been presented under this setting [4,6,19,21,5]. Up to now, the best attacks could reach to 11-round for Camellia-128 [4], 12-round for Camellia-192 [4] and 12-round for Camellia-256 [21,5].

The meet-in-the-middle (MITM) attack on Camellia was firstly proposed by Lu et al. in [24]. Based on the integral property, they introduced 5-round and 6-round higher-order MITM properties of Camellia, and mount the attacks on 10-round Camellia-128, 11-round Camellia-192 and 12-round Camellia-256. However these attacks do not start from the first round, and exclude the whitening layers, which were further improved in [25]. Then in [6], Chen et al. applied the attack model for AES in [9] to construct a 7-round MITM property for Camellia by getting rid of the integral property. Based on this property, they launched an attack on 12-round Camellia-256 with 2^{19} chosen plaintexts and $2^{231.2}$ encryptions.

In this paper, combined with the differential enumeration technique and multisets proposed by Dunkelman et al. at ASIACRYPT 2010 [11] and some properties on FL/FL^{-1} layers and truncated differential of Camellia, we propose a new 7-round property for Camellia-192 which is used to construct the MITM attack on 12-round Camellia-192. This attack costs 2^{180} 12-round encryptions and 2^{130} 128-bit memories with 2^{113} chosen plaintexts. Furthermore, we present an 8-round property of Camellia-256, and achieve a 13-round MITM attack on Camellia-256 with 2^{113} chosen ciphertexts, $2^{232.7}$ 13-round encryptions and 2^{227} 128-bit memories. To the best our knowledge, there are the most efficient cryptanalysis of reduced-round Camellia-192/256. There are too many cryptanalysis results of Camellia. In order to compare easily, we summarize our results along with some major previously results of reduced-round Camellia-192/256 starting from the first round with FL/FL^{-1} and the whitening layers in table 1, 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-192/256

Camellia-192	10	Impossible Diff	2^{121}CP	2^{175}	$2^{155.2}$	[5]
	10	Impossible Diff	$2^{118.7}$CP	$2^{130.4}$	2^{135}	[19]
	11	Impossible Diff	$2^{114.64}$CP	2^{184}	$2^{141.64}$	[21]
	12	Impossible Diff	2^{123}CP	$2^{187.2}$	2^{160}	[21]
	12	ZC. FFT†	$2^{125.7}$CP	$2^{188.8}$	2^{112}	[4]
	12	MITM	2^{113}CP	2^{180}	2^{130}	section 3
Camellia-256	11	Impossible Diff	2^{121}CP	$2^{206.8}$	2^{166}	[5]
	11	Impossible Diff	$2^{119.6}$CP	$2^{194.5}$	2^{135}	[19]
	12	Impossible Diff	$2^{116.17}$CP/CC	2^{240}	$2^{150.17}$	[21]
	12	MITM	2^{19}CP	$2^{231.2}$	2^{229}	[6]
	13	MITM	2^{113}CC	$2^{232.7}$	2^{227}	section 4

†: Zero-correlation linear cryptanalysis with discrete fast fourier transform.

The rest of this paper is organized as follows. Section 2 provides a brief description of block cipher Camellia and some related works. Section 3 describes the 7-round MITM property for Camellia-192 and its application to 12-round attack. Then we present an 8-round property and mount an attack on 13-round Camellia-256 with two FL/FL^{-1} layers in Section 4. Finally, we conclude the paper in Section 5.

2 Preliminaries

This section first gives some notations used throughout the paper, and then introduces a brief description of Camellia. Finally, we introduce some definitions and related works of meet-in-the-middle attack.

2.1 Notations

The following notations are used in this paper:

$A_{r-1}, (B_{r-1})$: the left (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 S-box layer of the r-th round,
Z_r : the state after the diffusion layer layer of the r-th round,
$X[i]$: the $i-$th byte of a bit string X $(1 \le i \le 8)$,
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 $X(1 \le i \le 128)$, where the left-most bit is the most significant bit,
k_r : the subkey used in the $r-$th round,
kw_i : the whitening key used in the begin and end of Camellia, $i = 1, 2, 3, 4$,
$x \| y$: bit string concatenation of x and y,
$\oplus$, $\cap$, $\cup$: bitwise exclusive OR (XOR), AND, OR,
$\lll l$: bit rotation to the left by l bit.

2.2 A 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. For Camellia-192/256, the encryption procedure is as follows.

Firstly, a 128-bit plaintext M is XORed with the whitening key $(kw_1 \| kw_2)$ and get two 64-bit data A_0 and B_0. Then, for $r = 1$ to 24, expect for $r = 6$, 12 and 18, the following is carried out:

$$A_r \;=\; B_{r-1} \oplus F(A_{r-1}, k_r), \quad B_r \;=\; A_{r-1}.$$

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

$$\begin{array}{ll} A'_r = B_{r-1} \oplus F(A_{r-1}, k_r), & B'_r = A_{r-1}, \\ A_r = FL(A'_r, kf_{r/3-1}), & B_r = FL^{-1}(B'_r, kf_{r/3}), \end{array}$$

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

The round function F is composed of a key-addition layer, a substitution transformation 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. $X_r = A_{r-1} \oplus k_r$ 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:

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

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 a simple boolean function which 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.

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

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-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., $K_R = K\{129-192\}\|\overline{K\{129-192\}}$. 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 details of Camellia, we refer to [1].

2.3 Definitions and Related Works

In this paper, we focus on the meet-in-the-middle attack on Camellia, which are applied to analyse AES by Demirci, Dunkelman and Derbez et al. 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 particular property for the middle part E^m, which is used to construct a distinguish to identify the correct key (K_1, K_2). To make this paper easier to understand, some definitions and properties are given in the following, which are similar to that of the MITM attak on AES.

Definition 1. **($\delta-$set [8])** *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 [11])** *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 $\binom{511}{255} \approx 2^{506.17}$ different values.*

Property 1. **(Differential property of $S-$box)** Given the input and output differences of the S-box, there exists a pair of actual values on average to satisfy these differences. This property is also applied to the inversion of S-box operation.

Demirci and Selçuk give the first MITM attack on AES [9]. They constructed a function for the input active byte and one of the output bytes of E_m, when there is only an active byte $X[j]$ for the input X of E_m. That is to say the inputs of E_m form a $\delta-$set $(X^0, \cdots, X^{255})$, where the j-th byte is different and the other bytes are the same for $X^0, \cdots, X^{255}$. Let the $i-$th output byte of E_m be the output of the function. The outputs of function with the $\delta-$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 function 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 $2^{-506.17}$), more details seen [10]. When the number of the parameters of the constructed function is far lower than 2^{128} or 2^{256} determined by the length of master key, a precomputed table to store the multisets by traversing all the parameters for identifying the right subkey.

At ASIACRYPT 2010, Dunkelman, Keller and Shamir [11] proposed the differential enumeration technique to reduce the complexity of the attack, where they showed that if a message of the $\delta-$set belongs to a pair which conforms a special truncated differential, then the possible values of the multiset will be restricted to a small subset of the value space. Indeed, the core of this technique is to fix some values of parameters of constructed function by using the truncated differential. This attack needs enough plaintexts such that there is a pair satisfying the truncated differential, and the $\delta-$set is constructed only for such pair. Apparently, the direct advantage of this attack is reducing the memory requirement, while the data complexity is increased in a great deal. This attack was improved by Derbez *et al.* at EUROCRYPT 2013 [10]. Combined with the rebound-like view of the cipher, they showed that, the number of possible values of multiset in the precomputed table could be further reduced since many of them could not be reached, actually.

In a word, the key part of the attack 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 parameters by special truncated differential. 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** in the following, which includes precomputation phase and online phase.

1. Precomputation phase: compute all values of the output sequence of the function constructed on E_m, and store them in a hash table.
2. Online phase:
   (a) Encrypt enough chosen plaintexts such that there exists a pair satisfying the specified truncated differential.
   (b) Guess values of the related subkeys K_1 and K_2 to find a pair which coincides with the specified truncated differential.
   (c) Construct a δ-set based on the pair, and partially decrypt to get the corresponding 256 plaintexts.
   (d) Obtain the corresponding plaintext-ciphertext pairs from the collection data. Then partially decrypt the ciphertexts to get the corresponding 256-byte value of the output sequence of E_m.
   (e) If a sequence value lies in the precomputation table, the guessed related subkeys in E_1 and E_2 may be right key.
   (f) Exhaustive search the remaining subkeys to obtain the right key.

Here, we apply multiset, the differential enumeration technique and the rebound-like method in the cryptanalysis of Camellia.

3 The MITM Attack on Camellia-192

Combined with the property of FL function and the key relations, this section introduces a 7-round MITM property for Camellia-192, and applies it to attack 12-round Camellia-192.

3.1 The 7-Round Property of Camellia-192

We first list two properties which are important for the 7-round property of Camellia-192.

Property 2. ([15]) Let X, X', K be l-bit values, and $\Delta X = X \oplus X'$, then the differential properties of AND and OR operations are:

$$(X \cap K) \oplus (X' \cap K) = \Delta X \cap K,$$
$$(X \cup K) \oplus (X' \cup K) = \Delta X \oplus (\Delta X \cap K).$$

Property 3. The 128-bit subkeys kf_1, kf_2 utilized in FL/FL^{-1} layer only take 64-bit information.

Proof. According to the key schedule of Camellia-192, the inserted subkey of first FL/FL^{-1} layer is generated by 128-bit key K_R, i.e.,

$$kf_1 = (K_R \lll 30)_L,$$
$$kf_2 = (K_R \lll 30)_R.$$

Then the subkeys kf_1, kf_2 are determined by K_R, which takes 64-bit information. Thus the 128-bit subkey $kf_1\|kf_2$ only takes 64-bit information. □

The 7-round property starting from the third round and ending at the ninth round is defined in Proposition 1, which is outlined in Fig. 1, where the symbols $*$, 0 and ? represent the nonzero difference, zero difference and unknown difference (zero or nonzero), respectively. The active byte of $\delta-$set is defined at the first bytes of the input of the third round $B_2[1]$.

Proposition 1. *Considering to encrypt 2^8 values of the $\delta-$set through 7-round Camellia-192 starting from the third round, where $B_2[1]$ is the active byte, in the case of that a message of the $\delta-$set belongs to a pair which conforms to the truncated differential outlined in Fig 1, then the corresponding multiset of bytes $(P^{-1}(\Delta A_8))[6]$ only takes about 2^{128} values.*

Proof. We firstly consider the computation of the multiset of bytes $(P^{-1}(\Delta A_8))[6]$ associated with a $\delta-$set. Actually, it is determined by 36-byte intermediate variable

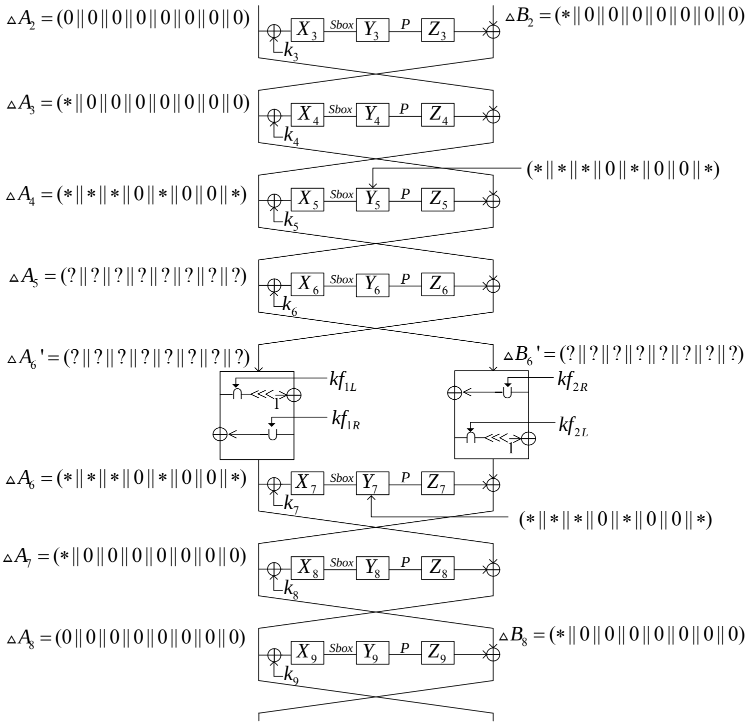
$$X_4[1]\|X_5[1,2,3,5,8]\|X_6\|kf_1\|kf_2\|X_7[2,3,5,7,8]\|X_8[6].$$

For any different value $X'_4[1]$ of the $\delta-$set, it is explicit to compute the difference $\Delta Y_4[1] = Y_4[1] \oplus Y'_4[1]$, which supports to deduce the difference $\Delta X_5[1,2,3,5,8]$. Then deduce the intermediate difference $\Delta Y_5[1,2,3,5,8]$ by the known value $X_5[1,2,3,5,8]$. Compute the difference ΔY_6 by the value X_6. Then the difference $\Delta A'_6\|\Delta B'_6$ could be deduced immediately. After that, compute the difference $\Delta A_6\|\Delta B_6$ by the Property 2. Deduce the difference $\Delta Y_7[2,3,5,7,8]$ and $\Delta Y_8[6]$ by the value $X_7[2,3,5,7,8]\|X_8[6]$. Thus, the difference $(P^{-1}(\Delta A_8))[6]$ is obtained by the equation $(P^{-1}(\Delta A_8))[6] = \Delta Y_8[6] \oplus (P^{-1}(\Delta A_6))[6]$. Therefore, the multiset of difference $(P^{-1}(\Delta A_8))[6]$ for the δ-set could be computed by exhaustively searching for all values of $X'_4[1]$ since the difference $\Delta B_2[1]$ is equal to $\Delta X_4[1]$ in such case.

Furthermore, if there exists a message of the $\delta-$set belongs to a pair which conforms the truncated differential characteristic as in Fig 1, 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, according to Property 1, the value $X_4[1]$ is deduced by 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 $\Delta A_4 = P(\Delta Y_4)$ and $\Delta A_6 = P(\Delta Y_8)$ 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 $\Delta X_7 = P(\Delta Y_8)$.

By Property 3, the total 36-byte variable is determined 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 is equal to $P^{-1}(FL^{-1}(P(\Delta Y_5) \oplus \Delta A_3)) \oplus P^{-1}(\Delta A_7)$, where the probability that $\Delta Y_7[4,6,7]$ equal to 0 is 2^{-24}. So there are only about 2^{128} possible values for 36-byte intermediate variable, actually. □



ΔA2 = (0||0||0||0||0||0||0||0)
ΔB2 = (*||0||0||0||0||0||0||0)
ΔA3 = (*||0||0||0||0||0||0||0)
ΔA4 = (*||*||*||0||*||0||0||*)
(*||*||*||0||*||0||0||*)
ΔA5 = (?||?||?||?||?||?||?||?)
ΔA6' = (?||?||?||?||?||?||?||?)
ΔB6' = (?||?||?||?||?||?||?||?)
kf1L
kf1R
kf2R
kf2L
ΔA6 = (*||*||*||0||*||0||0||*)
(*||*||*||0||*||0||0||*)
ΔA7 = (*||0||0||0||0||0||0||0)
ΔA8 = (0||0||0||0||0||0||0||0)
ΔB8 = (*||0||0||0||0||0||0||0)
X3 Sbox Y3 P Z3 k3
X4 Sbox Y4 P Z4 k4
X5 Sbox Y5 P Z5 k5
X6 Sbox Y6 P Z6 k6
X7 Sbox Y7 P Z7 k7
X8 Sbox Y8 P Z8 k8
X9 Sbox Y9 P Z9 k9

Fig. 1. The truncated differential of 7-round Camellia-192

3.2 The MITM Attack on 12-Round Camellia-192

Based on the 7-round property, we extend two rounds on the top and three rounds on the bottom to present the 12 round MITM attack on Camellia-192. We have two properties on differential characteristic for Camellia in the following which are helpful to recover the master key.

Property 4. Given the input difference of the i-th round $\varDelta A_i = (\alpha\|0\|0\|0\|0\|0\|0\|0)$, $\varDelta B_i = (0\|0\|0\|0\|0\|0\|0\|0)$, the output difference of $i+3$-th round $\varDelta B_{i+3}$ and intermediated difference $\varDelta Y_{i+2}$ satisfy the following equations:

$$P^{-1}(\varDelta B_{i+3})[4] = \varDelta A_i[1] = \alpha, \tag{1}$$

$$P^{-1}(\varDelta B_{i+3})[j] = 0, j = 6, 7 \tag{2}$$

$$P^{-1}(\varDelta B_{i+3})[1] = \varDelta Y_{i+2}[1], \tag{3}$$

$$P^{-1}(\varDelta B_{i+3})[j] = \varDelta Y_{i+2}[j] \oplus P^{-1}(\varDelta B_{i+3})[4], j = 2, 3, 4, 5, 8. \tag{4}$$

Proof. If $\alpha = 0$, this property is obvious. If $\alpha \neq 0$, we know $\varDelta Y_{i+1} = (\beta\|0\|0\|0\|0\|0\|0\|0)$, where $\beta \neq 0$. Thus we have $\varDelta A_{i+1} = (\beta\|\beta\|\beta\|0\|\beta\|0\|0\|\beta)$. By the define of round function, $Y_{i+2}[4] = 0$ are deduced. Since $P^{-1}(\varDelta A_i) = (0\|\alpha\|\alpha\|\alpha\|\alpha\|0\|0\|\alpha)$, then $P^{-1}(\varDelta A_i)[4] = \alpha$. Hence

$$P^{-1}(\varDelta A_{i+2})[4] = \varDelta Y_{i+2}[4] \oplus P^{-1}(\varDelta A_i)[4] = \alpha = \varDelta A_i[1].$$

Because

$$\begin{aligned}\varDelta Y_{i+2} &= P^{-1}(\varDelta A_i \oplus \varDelta A_{i+2}) = P^{-1}(\varDelta A_i) \oplus P^{-1}(\varDelta B_{i+3}) \\ &= (0\|\alpha\|\alpha\|\alpha\|\alpha\|0\|0\|\alpha) \oplus P^{-1}(\varDelta B_{i+3}).\end{aligned}$$

Therefore,

$$P^{-1}(\Delta B_{i+3})[j] = \Delta Y_{i+2}[j], j = 1, 6, 7,$$
$$P^{-1}(\Delta B_{i+3})[j] = \Delta Y_{i+2}[j] \oplus P^{-1}(\Delta B_{i+3})[4], j = 2, 3, 4, 5, 8.$$

Since $\Delta Y_{i+2}[j] = 0, j = 6, 7$, then $P^{-1}(\Delta B_{i+3})[j] = 0, j = 6, 7$. □

Property 5. Given the output difference of the $(i+2)$-th round $\Delta A_{i+2} = (0\|0\|0\|0\|0\|0\|0\|0)$, $\Delta B_{i+2} = (\alpha\|0\|0\|0\|0\|0\|0\|0)$, the input difference of i-th round ΔB_i and the intermediate difference ΔY_{i+1} satisfy the following equations:

$$\Delta P^{-1}(\Delta B_i)[4] = \Delta B_{i+2}[1] = \alpha, \tag{5}$$
$$\Delta P^{-1}(\Delta B_i)[j] = 0, j = 6, 7 \tag{6}$$
$$\Delta P^{-1}(\Delta B_i)[1] = \Delta Y_{i+1}[1], \tag{7}$$
$$\Delta P^{-1}(\Delta B_i)[j] = \Delta Y_{i+1}[j] \oplus P^{-1}(\Delta B_i)[4], j = 2, 3, 5, 8. \tag{8}$$

This proof of this property is similar to that of Property 4.

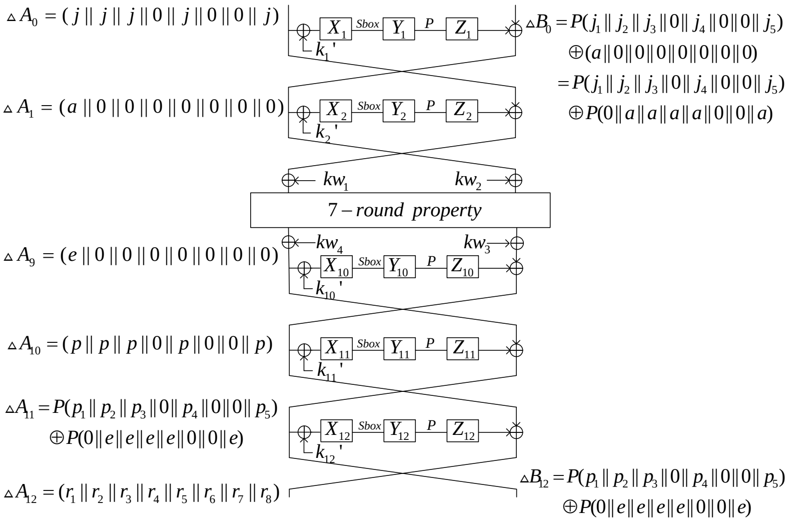
To increase efficiency of the 12-round attack on Camellia-192, we retrieve the equivalent keys k'_1, k'_2, k'_{10}, k'_{11}, k'_{12} (seen Fig. 2), and then deduce the master key. The equivalent keys are defined as $k'_1 = k_1 \oplus kw_1$, $k'_2 = k_2 \oplus kw_2$, $k'_{12} = k_{12} \oplus kw_4$, $k'_{11} = k_{11} \oplus kw_3$, 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 [5].

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 Proposition 1, and store them in a hash table $\mathcal{H}$. The attack procedure of the online phase is similar to Step 2 of Algorithm 1. However we take a balance of the time complexity of Step (b) and Step (c). 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.

In order to look for an expected pair with low time complexity for each key guess, we use the early abort technique [22] to eliminate the wrong pairs by guessing only a small fraction of the unknown subkeys every time. For example, if a pair conforms to expected truncated differential, as described in Fig. 2 the difference $\Delta Y_{12} = P^{-1}(\Delta A_{12}) \oplus P^{-1}(\Delta A_{10}) = P^{-1}(r_1\|r_2\|r_3\|r_4\|r_5\|r_6\|r_7\|r_8) \oplus (p\|0\|0\|0\|0\|0\|0\|0)$. Thus the difference $\Delta Y_{12}[2, \cdots, 8]$ is determined for every pair. That means we only need to guess 8-bit subkey $k_{12}[i]$ (for $i = 2 \cdots 8$) to delete the wrong pairs with probability 2^{-8} byte by byte.

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 $A_0 = (\alpha\|\alpha\oplus x_1\|\alpha\oplus x_2\|x_3\|\alpha\oplus x_4\|x_5\|x_6\|\alpha\oplus x_7)$, $B_0 = P(\beta_1\|\beta_2\|\beta_3\|\beta_4\|\beta_5\|y_1\|y_2\|\beta_6)$, where x_i and y_i $(i = 1, ..., 7)$ are constants, but α, β_j $(j = 1, ..., 6)$ take all the possible values. Ask for corresponding ciphertexts for each structure, compute $P^{-1}(B_{12})$ and store the plaintext-ciphertext pairs $A_0\|B_0\|A_{12}\|B_{12}$ in a hash table indexed by 16-bit value $(P^{-1}(B_{12}))[6, 7]$. Hence, there are $2^{57} \times 2^{111} \times 2^{-16} = 2^{152}$ pairs whose differences satisfy $P^{-1}(\Delta B_{12})[6, 7] = 0$ 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 $B_{12}[l]$ and keep only the pairs which satisfy $\Delta Y_{12}[l] = P^{-1}(\Delta A_{12}[l])$. 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 A_{10}.



$\triangle A_0 = (j \| j \| j \| 0 \| j \| 0 \| 0 \| j)$
X_1
Sbox
Y_1
P
Z_1
k_1'
$\triangle B_0 = P(j_1 \| j_2 \| j_3 \| 0 \| j_4 \| 0 \| 0 \| j_5)$
$\oplus (a \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$= P(j_1 \| j_2 \| j_3 \| 0 \| j_4 \| 0 \| 0 \| j_5)$
$\oplus P(0 \| a \| a \| a \| a \| 0 \| 0 \| a)$
$\triangle A_1 = (a \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
X_2
Sbox
Y_2
P
Z_2
k_2'
kw_1
kw_2
7 – round property
kw_4
kw_3
$\triangle A_9 = (e \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
X_{10}
Sbox
Y_{10}
P
Z_{10}
k_{10}'
$\triangle A_{10} = (p \| p \| p \| 0 \| p \| 0 \| 0 \| p)$
X_{11}
Sbox
Y_{11}
P
Z_{11}
k_{11}'
$\triangle A_{11} = P(p_1 \| p_2 \| p_3 \| 0 \| p_4 \| 0 \| 0 \| p_5)$
$\oplus P(0 \| e \| e \| e \| e \| 0 \| 0 \| e)$
X_{12}
Sbox
Y_{12}
P
Z_{12}
k_{12}'
$\triangle A_{12} = (r_1 \| r_2 \| r_3 \| r_4 \| r_5 \| r_6 \| r_7 \| r_8)$
$\triangle B_{12} = P(p_1 \| p_2 \| p_3 \| 0 \| p_4 \| 0 \| 0 \| p_5)$
$\oplus P(0 \| e \| e \| e \| e \| 0 \| 0 \| e)$

Fig. 2. The MITM attack on 12-round Camellia-192

   (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 $\Delta Y_{11}[l] = P^{-1}(\Delta B_{12})[l] \oplus P^{-1}(\Delta B_{12})[4]$, which should hold by Property 4. Then guess $k'_{11}[1]$ and keep the pairs making $\Delta Y_{11}[1] = P^{-1}(\Delta B_{12})[1]$ (seen Property 4) 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 $\Delta Y_1[1] = P^{-1}(\Delta B_0)[1]$ and $\Delta Y_1[l] = P^{-1}(\Delta B_0)[l] \oplus P^{-1}(\Delta B_0)[4]$ hold for $l = 2, 3, 5, 8$ (seen Property 5). 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 truncated differential described in Fig. 1, 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 for the time complexity is greater than the 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 $\Delta A_0[1]$ and $P^{-1}(\Delta B_0)[4]$, deduce the intermediate value $X_2[1] \| Y_2[1]$ of the pair by the difference distribution table of S-box s_1.
   (b) Select a message $A_0 \| B_0$ of the pair $(A_0 \| B_0, A'_0 \| B'_0)$, 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 $\Delta A'_0[1, 2, 3, 5, 8]$. Then get the left half of the plaintext $A''_0 = A_0 \oplus \Delta A'_0$.
   (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 $\Delta B'_0$ and get the right half part $B''_0 = B_0 \oplus \Delta B'_0$.
   (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 $Y_{11}[2, 3, 5, 8]$, $P^{-1}(A_{10})[6]$ 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 $(P^{-1}(\Delta A_8))[6] = \Delta Y_{10}[6] \oplus P^{-1}(\Delta A_{10})[6]$. 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.

4 The Attack on 13-Round Camellia-256

This section introduces an 8-round property of Camellia, which starts from the fifth round and ends at the twelfth round defined by Proposition 2. The truncated differential used in this section is outlined in Fig. 3 with dotted line, the active byte of the $\delta-$set is located in $A'_{12}[4]$, and the corresponding byte of multiset is defined as $P^{-1}(\Delta A_4)[1]$.

Proposition 2. *Considering to decrypt 2^8 values of the $\delta-$set through 8-round Camellia-256 starting from the 12-th round, where $A_{12}[5]$ is the active byte, in the case of that a message of the $\delta-$set belongs to a pair that conforms to the 8-round truncated differential outlined in Fig 3, then the corresponding multiset of bytes $(P^{-1}(\Delta A_4))[1]$ only takes about 2^{225} values.*

For the 8-round Camellia-256 in the dotted line of Fig. 3, we consider the computation of the multiset of bytes $(P^{-1}(\Delta A_4))[1]$ by partially decrypting (A_{12}, B_{12}), where $A_{12}[5]$ is the active bytes. A brief proof of this proposition is given as follows.

Proof. If $\Delta A_{12}[5] \neq 0$ and there is no difference on the other bytes of the input (A_{12}, B_{12}), $(P^{-1}(\Delta A_4))[1]$ 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. 3, 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 $(P^{-1}(\Delta A_4))[1]$. Hence the multiset of bytes $(P^{-1}(\Delta A_4))[1]$ 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. □

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 described in Proposition 2. We also recover the equivalent keys k'_1, k'_2, k'_3, k'_4, k'_{13} (seen Fig. 3), and then deduce the master key. The equivalent keys are defined as $k'_1 = k_1 \oplus kw_1$, $k'_2 = k_2 \oplus kw_2$, $k'_3 = k_3 \oplus kw_1$, $k'_4 = k_4 \oplus kw_2$, and $k'_{13} = k_{13} \oplus kw_4$. The attack is worked in the chosen-ciphertext model. In the precomputation phase, we traverse 225-bit $\mathcal{V}$ to compute all 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

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

   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 $P^{-1}(\varDelta A_1)$ for every pair by guessing 64-bit subkey k'_1, eliminate the pairs which do not satisfy $P^{-1}(\varDelta A_1)[6, 7] = 0$. 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 $\varDelta Y_2[l] = P^{-1}(\varDelta A_0[l])$ hold. Then guess $k'_2[1]$ to compute A_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 $\varDelta Y_3[l] = P^{-1}(\varDelta A_1)[l] \oplus P^{-1}(\varDelta A_1)[4]$. Then guess $k'_3[1]$ and keep the pairs satisfying $\varDelta Y_3[1] = P^{-1}(\varDelta A_1)[1]$. 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 $\varDelta Y_{13}[l]$. Delete the pairs which do not content $\varDelta Y_{13}[l] = P^{-1}(\varDelta A_{13}[l])$. Then guess $kf_{3R}[1]$, compute $\varDelta A'_{12}[1]$ by using Property 2, and delete the pairs when $\varDelta A'_{12}[1] \neq 0$. Hereafter, the expected number of remaining pairs is about 2^8.
6. Compute the value A_3 by guessing 24-bit subkey $k'_3[4, 6, 7]$, and then deduce the value of subkey $k'_4[1]$ for every pair. 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.
7. 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 bound 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.

5 Conclusion

In this paper, we discuss the security of reduced-round Camellia-192/256 against the meet-in-the-middle attack. Taking advantage of differential enumeration technique and multiset, we propose the 7-round and 8-round properties, and mount the attacks on 12-round Camellia-192 and 13-round Camellia-256, respectively, which improve the previous cryptanalysis results. As far as we know, there are the best results of cryptanalysis of reduced-round Camellia-192/256 in terms of the number of rounds under the original design.

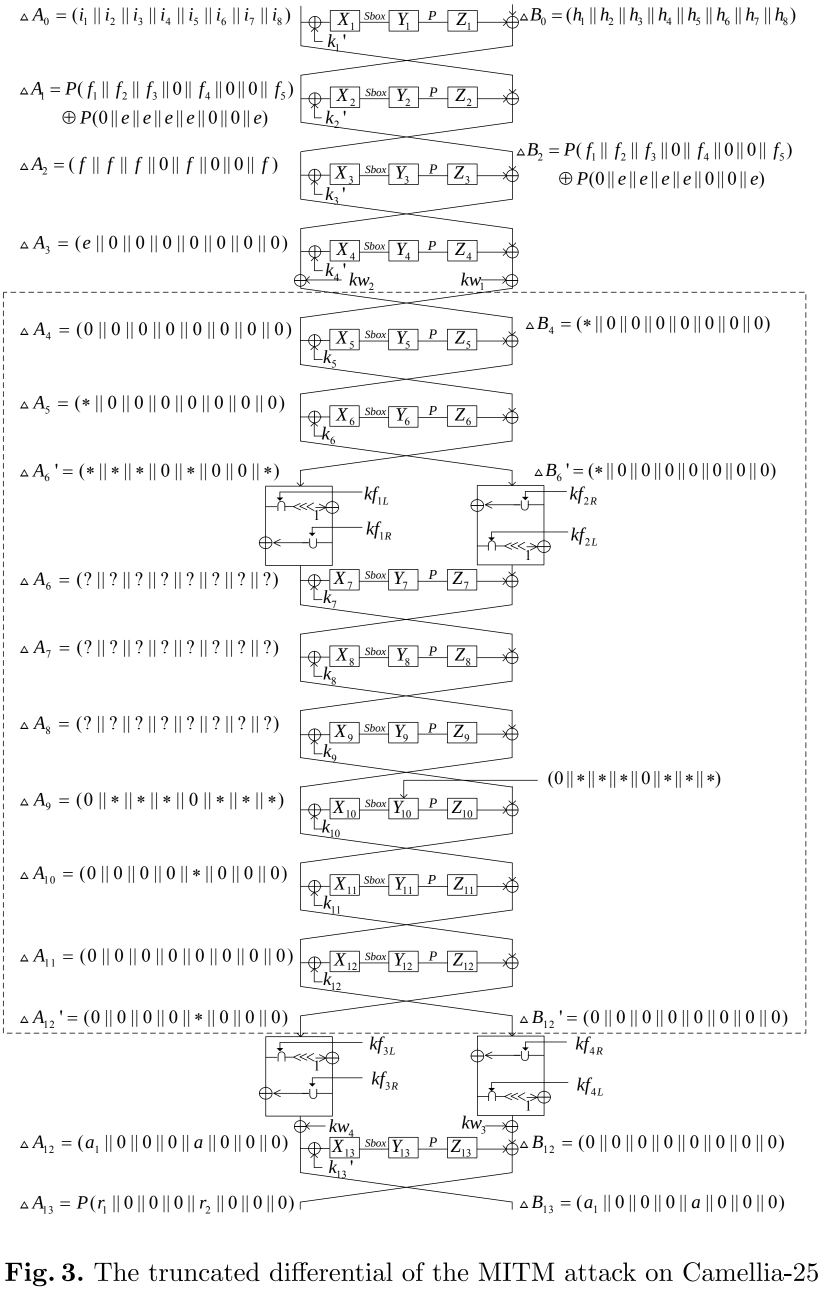
References

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. Bai, D., Li, L.: New Impossible Differential Attacks on Camellia. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. Lecture Notes in Computer Science, vol. 7232, pp. 80–96. Springer (2012)
4. 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)
5. 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)
6. 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)
7. Cryptography Research and Evaluation Committees: Http://www.cryptrec.go.jp/english/index.html
8. Daemen, J., Rijmen, V.: AES proposal: Rijndael. In: First Advanced Encryption Standard (AES) Conference (1998)
9. 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)
10. 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)
11. 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)
12. 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)
13. International Organization for Standardization(ISO): International Standard- ISO/IEC 18033-3, Information technology-Security techniques-Encryption algorithms -Part 3: Block ciphers (2010)
14. 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)
15. 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)
16. 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)
17. 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)
18. 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)
19. Li, L., Chen, J., Jia, K.: New Impossible Differential Cryptanalysis of Reduced-Round Camellia. In: Lin, D., Tsudik, G., Wang, X. (eds.) CANS 2011. Lecture Notes in Computer Science, vol. 7092, pp. 26–39. Springer (2011)

20. Liu, Y., Gu, D., Liu, Z., Li, W.: Improved results on impossible differential cryptanalysis of reduced-round camellia-192/256. Journal of Systems and Software 85(11), 2451–2458 (2012)
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. Lu, J., Wei, Y., Pasalic, E., Fouque, P.A.: Meet-in-the-Middle Attack on Reduced Versions of the Camellia Block Cipher. In: IWSEC 2012. Lecture Notes in Computer Science, vol. 7631, pp. 197–215. Springer (2012)
26. 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)
27. New European Schemes for Signatures, Integrity, and Encryption: Final Report of European project IST-1999-12324. Https://www.cosic.esat.kuleuven.be/nessie/Bookv015.pdf
28. Shirai, T.: Differential, Linear, Boomerang and Rectangle Cryptanalysis of Reduced- Round Camellia. In: the Third NESSIE Workshop (2002)
29. 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)
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, L., Zhang, W.: Improved Impossible Differential Cryptanalysis of Reduced-Round Camellia. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. Lecture Notes in Computer Science, vol. 5381, pp. 442–456. Springer (2008)
32. 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 Truncated Differential of the MITM Attack on 13-Round Camellia-256

Here we list the the truncated differential used in the MITM Attack on Camellia-256 which is on the dotted lines in 3.



$\triangle A_0 = (i_1 \| i_2 \| i_3 \| i_4 \| i_5 \| i_6 \| i_7 \| i_8)$
$\triangle B_0 = (h_1 \| h_2 \| h_3 \| h_4 \| h_5 \| h_6 \| h_7 \| h_8)$
$\triangle A_1 = P(f_1 \| f_2 \| f_3 \| 0 \| f_4 \| 0 \| 0 \| f_5) \oplus P(0 \| e \| e \| e \| e \| 0 \| 0 \| e)$
$\triangle B_2 = P(f_1 \| f_2 \| f_3 \| 0 \| f_4 \| 0 \| 0 \| f_5) \oplus P(0 \| e \| e \| e \| e \| 0 \| 0 \| e)$
$\triangle A_2 = (f \| f \| f \| 0 \| f \| 0 \| 0 \| f)$
$\triangle A_3 = (e \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
kw_2
kw_1
$\triangle A_4 = (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$\triangle B_4 = (* \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$\triangle A_5 = (* \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$\triangle A_6' = (* \| * \| * \| 0 \| * \| 0 \| 0 \| *)$
$\triangle B_6' = (* \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
kf_{1L} kf_{1R} kf_{2R} kf_{2L}
$\triangle A_6 = (? \| ? \| ? \| ? \| ? \| ? \| ? \| ?)$
$\triangle A_7 = (? \| ? \| ? \| ? \| ? \| ? \| ? \| ?)$
$\triangle A_8 = (? \| ? \| ? \| ? \| ? \| ? \| ? \| ?)$
$(0 \| * \| * \| * \| 0 \| * \| * \| *)$
$\triangle A_9 = (0 \| * \| * \| * \| 0 \| * \| * \| *)$
$\triangle A_{10} = (0 \| 0 \| 0 \| 0 \| * \| 0 \| 0 \| 0)$
$\triangle A_{11} = (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$\triangle A_{12}' = (0 \| 0 \| 0 \| 0 \| * \| 0 \| 0 \| 0)$
$\triangle B_{12}' = (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
kf_{3L} kf_{3R} kf_{4R} kf_{4L}
kw_4 kw_3
$\triangle A_{12} = (a_1 \| 0 \| 0 \| 0 \| a \| 0 \| 0 \| 0)$
$\triangle B_{12} = (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0)$
$\triangle A_{13} = P(r_1 \| 0 \| 0 \| 0 \| r_2 \| 0 \| 0 \| 0)$
$\triangle B_{13} = (a_1 \| 0 \| 0 \| 0 \| a \| 0 \| 0 \| 0)$
X_1 ... X_{13}, Sbox, Y_1 ... Y_{13}, P, Z_1 ... Z_{13}
k_1' k_2' k_3' k_4' k_5 k_6 k_7 k_8 k_9 k_{10} k_{11} k_{12} k_{13}'

Fig. 3. The truncated differential of the MITM attack on Camellia-256