Impossible Differential Cryptanalysis of ARIA and Camellia

Wenling Wu, Wentao Zhang, Dengguo Feng

State Key Laboratory of Information Security,
Institute of Software, Chinese Academy of Sciences, Beijing 100080, P. R. China
{*wwl*}*@is.iscas.ac.cn*

Abstract. This paper studies the security of the block ciphers ARIA and Camellia against impossible differential cryptanalysis. Our work improves the best impossible differential cryptanalysis of ARIA and Camellia known so far. The designers of ARIA expected no impossible differentials exist for 4-round ARIA. However, we found some nontrivial 4-round impossible differentials, which may lead to a possible attack on 6-round ARIA. Moreover, we found some nontrivial 8-round impossible differentials for Camellia, whereas only 7-round impossible differentials were previously known. By using the 8-round impossible differentials, we presented an attack on 12-round Camellia without FL/FL^{-1} layers.

Key words. Block cipher, ARIA, Camellia, Data complexity, Time complexity, Impossible Differential Cryptanalysis.

1 Introduction

Both ARIA[1] and Camellia[2] support 128-bit block size and 128-,192-, and 256-bit key lengths, i.e. the same interface specifications as the Advanced Encryption Standard(AES). Camellia was jointly developed in 2000 by Nippon Telegraph and Telephone Corporation (NTT) and Mitsubishi Electric Corporation (Mitsubishi). It has now been selected as an international standard by ISO/IEC, and also been adopted by cryptographic evaluation projects such as NESSIE and CRYPTREC, as well as the standardization activities at IETF. It means Camellia gradually become one of the most worldwide used block ciphers. Therefore, a constant evaluation of its security with respect to various cryptanalytic techniques is required. Camellia was already analyzed in many papers using various attacks[3-10].

ARIA was designed by a group of Korean experts in 2003. In 2004, ARIA was established as a Korean Standard block cipher algorithm (KS X 1213) by the Ministry of Commerce, Industry and Energy. ARIA is a general-purpose involutional SPN block cipher algorithm, optimized for lightweight environments and hardware implementation. Its security was analyzed initially by the designers internally, and later by the COSIC group of K.U. Leuven, Belgium[11]. They analyzed the security of ARIA against differential and linear cryptanalysis[12,13], truncated and higher-order differential[14], impossible differential[15], slide attack[16,17], Integral attack[18], and other attacks[19−21].

Impossible differential means a differential that holds with probability 0, or a differential that does not exist. Impossible differential attacks use impossible differentials to derive the actual values of the keys, which has been used to attack AES and get very good results[22-25].

In this paper, we examine the security of ARIA and Camellia against impossible differential attacks. The initial analysis of the security of Camellia to impossible differential Cryptanalysis was given in [4]. They presented some nontrivial 7-round impossible differentials for Camellia. We found some nontrivial 8-round impossible differentials, which may lead to a possible attack of Camellia reduced to 12 rounds without FL/FL^{-1}, the attack having complexity less than that of exhaustive search to 12-round Camellia without FL/FL^{-1} layers.

As for ARIA, the designers expected that there was no impossible differentials on 4 or more rounds in [1] and [26]. In this paper, we found some 4-round impossible differentials, which lead to a possible attack of ARIA reduced to 6 rounds. The attack requires 2^{121} plaintext/ciphertext pairs and 2^{112} encryptions.

The contents of this paper are as follows: In Section 2 we give a brief description of ARIA and Camellia. In Section 3 we describe some 4-round ARIA impossible differentials and the impossible differential attack on 6-round ARIA. In section 4, we describe some 8-round Camellia impossible differentials and present the impossible differential attack on 12-round Camellia without FL/FL^{-1} layers. Finally, Section 5 summarizes this paper.

2 ARIA and Camellia

Due to space limitation, we only introduce ARIA and Camellia briefly. Details are shown in [1] and [2].

2.1 Description of ARIA

ARIA is a substitution permutation network(SPN) and uses an involutional binary 16×16 matrix in its diffusion layer. The 128-bit plaintexts are treated as byte matrices of size 4×4 as the following. Every round applies three operations

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

to the state matrix:

Round Key Addition(RKA): This is done by XORing the 128-bit round key.

Substitution Layer(SL): Applying the 8×8 S-boxes 16 times in parallel on each byte. There are two types of substitution layers to be used so as to make the cipher involution.

Diffusion Layer(DL): A linear map $A : (F_2^8)^{16} \to (F_2^8)^{16}$ is given by

$$(x_0|x_1|\ldots|x_{15}) \to (y_0|y_1|\ldots|y_{15}),$$

where

$$\begin{aligned}
y_0 &= x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14}, & y_8 &= x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15},\\
y_1 &= x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15}, & y_9 &= x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14},\\
y_2 &= x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15}, & y_{10} &= x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15},\\
y_3 &= x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14}, & y_{11} &= x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14},\\
y_4 &= x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15}, & y_{12} &= x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12},\\
y_5 &= x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15}, & y_{13} &= x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13},\\
y_6 &= x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13}, & y_{14} &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14},\\
y_7 &= x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13}, & y_{15} &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}.
\end{aligned}$$

Note that the Diffusion layer of the last round is replaced by a round key addition. We shall assume that the 6-round ARIA also has the diffusion layer replaced by a round key addition.

2.2 Description of Camellia

Camellia is based on the Feistel structure and has 18 rounds (for 128-bit keys) or 24 rounds (for 192/256-bit keys). The FL/FL^{-1}function layer is inserted at every 6 rounds. Before the first round and after the last round, there are pre- and post-whitening layers which use bitwise exclusive-or operations with 128-bit round subkeys, respectively. In this paper, we will consider camellia without FL/FL^{-1}function layer and whitening layers.
Let L_{r-1} and R_{r-1} be the left and the right halves of the r^{th} round input, and k_r be the r^{th} round subkey. Then the Feistel structure of Camellia can be written as

$$\begin{aligned}
L_r &= R_{r-1} \oplus F(L_{r-1}, k_r),\\
R_r &= L_{r-1}.
\end{aligned}$$

here F is the round function defined below:

$$\begin{aligned}
F : \{0,1\}^{64} \times \{0,1\}^{64} &\longrightarrow \{0,1\}^{64}\\
(X, k_r) &\longrightarrow Z = P(S(X \oplus k_r)).
\end{aligned}$$

where S and P are defined as follows:

$$\begin{aligned}
S : (F_2^8)^8 &\longrightarrow (F_2^8)^8\\
& x_1|x_2|x_3|x_4|x_5|x_6|x_7|x_8 \longrightarrow y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8\\
& y_1 = s_1(x_1), \quad y_2 = s_2(x_2), \quad y_3 = s_3(x_3), \quad y_4 = s_4(x_4),\\
& y_5 = s_2(x_5), \quad y_6 = s_3(x_6), \quad y_7 = s_4(x_7), \quad y_8 = s_1(x_8).
\end{aligned}$$

here s_1, s_2, s_3 and s_4 are the 8×8 boxes.

$$\begin{aligned}
P : (F_2^8)^8 &\longrightarrow (F_2^8)^8\\
& y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8 \longrightarrow z_1|z_2|z_3|z_4|z_5|z_6|z_7|z_8
\end{aligned}$$

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

The inverse of P is as follows:

$$
\begin{aligned}
P^{-1} : (F_2^8)^8 &\longrightarrow (F_2^8)^8\\
z_1|z_2|z_3|z_4|z_5|z_6|z_7|z_8 &\longrightarrow y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8
\end{aligned}
$$

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

3 Impossible Differential Cryptanalysis on Reduced-round ARIA

3.1 Some 4-Round Impossible Differentials

In this subsection, we indicate some impossible differentials on 4-round ARIA as shown in Fig.1. In this figure, we consider the 4-round impossible differential which is built in a miss-in-the-middle manner. A 2-round differential with probability 1 is *concatenated* to a 2-round differential with probability 1, in the inverse direction, where the intermediate differences contradict each other. The 4-round impossible differential is

$$(a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0) \xrightarrow{\text{4-round}} (0|h|0|0|0|0|0|0|h|h|h|0|0|0|h|0)$$

where a and h denote any non-zero value.

We use X_i^I and X_i^O to denote the input and output of round i, while X_i^S denotes the intermediate values after the application of Substitution Layer(SL) of round i. The first 2-round differential is obtained as follows:
The input difference $X_1^I = (a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0)$ is preserved through the AddRoundKey operation of round 1. This difference is in a single byte, and thus, the difference after the Substitution Layer(SL) of round 1 is still in a single byte,i.e., $X_1^S = (b|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0)$ where b is an unknown non-zero byte value. After the Diffusion Layer(DL) this difference becomes $X_2^I = (0|0|0|b|b|0|b|0|b|b|0|0|0|b|b|0)$. This difference evolves after AddRoundKey operation and the Substitution Layer(SL) of round 2 into

$$X_2^S = (0|0|0|b_3|b_4|0|b_6|0|b_8|b_9|0|0|0|b_{13}|b_{14}|0),$$

where $b_3, b_4, b_6, b_8, b_9, b_{13}$ and b_{14} are unknown non-zero byte values. Finally, after the Diffusion Layer(DL) this difference evolves to

$$X_2^O = (c_0|c_1|c_2|c_3|c_4|c_5|c_6|c_7|c_8|c_9|c_{10}|c_{11}|c_{12}|c_{13}|c_{14}|c_{15}),$$

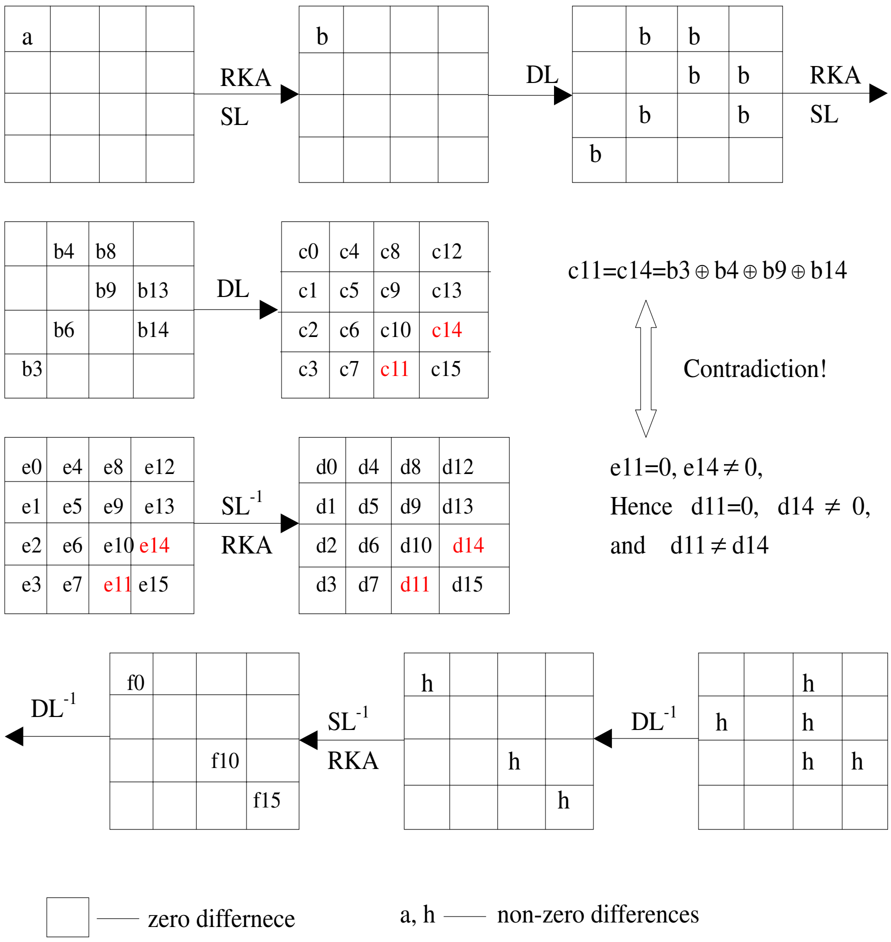

Fig.1. 4-Round Impossible Differentials of ARIA

where each byte can be expressed as:

$$\begin{aligned}
&c_0 = b_3 \oplus b_4 \oplus b_6 \oplus b_8 \oplus b_9 \oplus b_{13} \oplus b_{14}, && c_8 = b_4 \oplus b_{13},\\
&c_1 = b_8 \oplus b_9, && c_9 = b_6 \oplus b_{14},\\
&c_2 = b_4 \oplus b_6, && c_{10} = b_3 \oplus b_6 \oplus b_8 \oplus b_{13},\\
&c_3 = b_{13} \oplus b_{14}, && c_{11} = b_3 \oplus b_4 \oplus b_9 \oplus b_{14},\\
&c_4 = b_8 \oplus b_{14}, && c_{12} = b_6 \oplus b_9,\\
&c_5 = b_3 \oplus b_4 \oplus b_9 \oplus b_{14}, && c_{13} = b_3 \oplus b_6 \oplus b_8 \oplus b_{13},\\
&c_6 = b_9 \oplus b_{13}, && c_{14} = b_3 \oplus b_4 \oplus b_9 \oplus b_{14},\\
&c_7 = b_3 \oplus b_6 \oplus b_8 \oplus b_{13}, && c_{15} = b_4 \oplus b_8.
\end{aligned}$$

From the above equations, we get

$$\begin{aligned}
c_7 = c_{10} = c_{13} = b_3 \oplus b_6 \oplus b_8 \oplus b_{13},\\
c_{11} = c_{14} = b_3 \oplus b_4 \oplus b_9 \oplus b_{14}.
\end{aligned}$$

Hence, the input difference $X_1^I = (a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0)$ evolves with probability one into X_2^O which has same value in bytes 11 and 14, and X_2^O also has same value in bytes 7, 10, and 13.

The second differential ends after round 4 with difference $X_4^O = (0|h|0|0|0|0|0|0|h|h|h|0|0|0|h|0)$. When rolling back this difference through the Diffusion Layer(DL), we get the difference $X_4^S = (h|0|0|0|0|0|0|0|0|0|h|0|0|0|0|h)$. This difference has non-zero difference in bytes 0,10,and 15, thus the difference evolves after the inverse of Substitution Layer(SL) and AddRoundKey operation of round 4 into $X_4^I = (f_0|0|0|0|0|0|0|0|0|0|f_{10}|0|0|0|0|f_{15})$ where f_0, f_{10} and f_{15} are unknown non-zero byte values. When rolling back this difference through the Diffusion Layer(DL), we get the difference

$$X_3^S = (e_0|e_1|e_2|e_3|e_4|e_5|e_6|e_7|e_8|e_9|e_{10}|e_{11}|e_{12}|e_{13}|e_{14}|e_{15}),$$

where each byte can be expressed as:

$$\begin{aligned}
&e_0 = 0, && e_8 = f_0 \oplus f_{10} \oplus f_{15},\\
&e_1 = f_{15}, && e_9 = f_0,\\
&e_2 = f_{10} \oplus f_{15}, && e_{10} = f_{15},\\
&e_3 = f_0 \oplus f_{10}, && e_{11} = 0,\\
&e_4 = f_0 \oplus f_{15}, && e_{12} = 0,\\
&e_5 = f_{10} \oplus f_{15}, && e_{13} = f_0 \oplus f_{10},\\
&e_6 = f_0 \oplus f_{10}, && c_{14} = f_0,\\
&e_7 = 0, && e_{15} = f_{10} \oplus f_{15}.
\end{aligned}$$

From the above equations, we know $e_{11} = 0$ and $e_{14} = f_0 \neq 0$. Therefore, When rolling back this difference through the inverse of Substitution Layer(SL) and AddRoundKey operation of round 3, we get the difference

$$X_3^I = (d_0|d_1|d_2|d_3|d_4|d_5|d_6|d_7|d_8|d_9|d_{10}|d_{11}|d_{12}|d_{13}|d_{14}|d_{15}),$$

where $d_{11} = 0$ and $d_{14} \neq 0$.

This difference contradicts the first differential as with probability one $c_{11} = c_{14}$ while the second differential predicts $d_{11} \neq d_{14}$ with probability 1. This contradiction is emphasized in Fig.1.

Similarly, we can get other 4-round impossible differentials of ARIA, for example,

$$(a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0) \xrightarrow{\text{4-round}} (0|0|h|0|h|0|0|0|0|0|0|0|h|h|h|0),$$

$$(a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0) \xrightarrow{\text{4-round}} (0|h|0|0|0|0|0|0|0|h|0|0|0|h|h|h),$$

$$(a|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0) \xrightarrow{\text{4-round}} (0|0|0|0|h|0|0|h|h|0|0|0|h|0|h|0).$$

3.2 6-Round Impossible Differential Attack

In this subsection, we describe an impossible differential cryptanalysis of ARIA reduced to six rounds. The attack is based on the above four round impossible differentials with additional one round at each of the beginning and the end as in Fig.2. Note that the last round of ARIA does not have the diffusion layer.

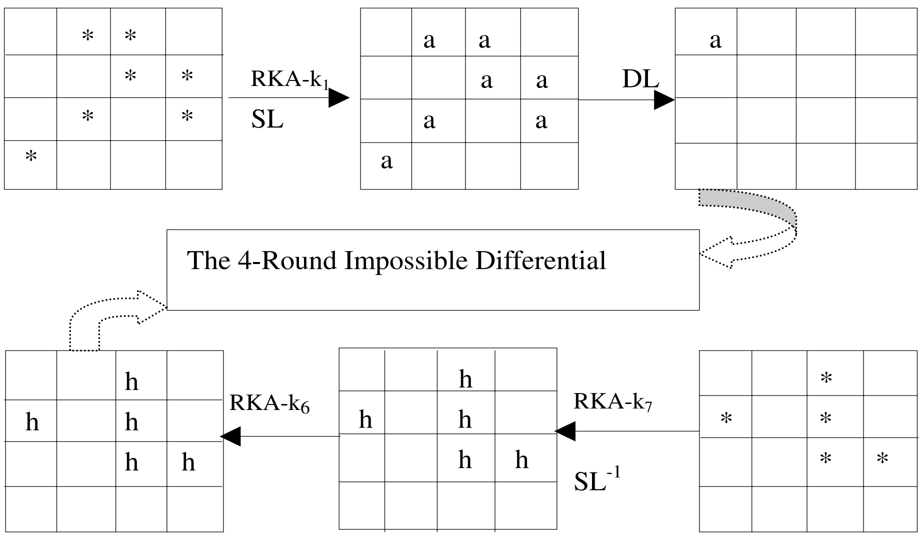

Fig.2. 6-Round Impossible Differential Attack to ARIA

The procedure is as follows:

Step 1 Choose structures of 2^{56} plaintexts which differ only at the seven bytes (3,4,6,8,9,13,14), having all possible values in these bytes. One structure proposes $2^{56} \times 2^{56} \times \frac{1}{2} = 2^{111}$ pairs of plaintexts.

Step 2 Take 2^{64} structures (2^{120} plaintexts, 2^{175} pairs of plaintexts). Choose pairs whose ciphertext pairs have zero difference at the eleven bytes (0,2,3,4,5,6,7, 11,12,13,15). The expected number of such pairs is about $2^{175} \times 2^{-88} = 2^{87}$.

Step 3 Guess the 40-bit value of the last round key k_7 at the five bytes(1,8,9,10,14), and perform the followings:

Step 3.1 For every remaining ciphertext pair (C, C^*), compute $C_5 \oplus C_5^* = SL^{-1}(C \oplus k_7) \oplus SL^{-1}(C^* \oplus k_7)$, choose pairs whose difference $C_5 \oplus C_5^*$ are same at the five bytes (1,8,9,10,14). Since the probability is about $p = (2^{-8})^4 = 2^{-32}$, the expected number of the remaining pairs is about $2^{87} \times 2^{-32} = 2^{55}$.

Step 3.2 For every remaining ciphertext pair (C, C^*) consider the corresponding plaintext pair (P, P^*), for 56-bit value at the seven bytes (3,4,6,8,9,13,14) of the subkey k_1, calculate $SL(P \oplus k_1) \oplus SL(P^* \oplus k_1)$, and check whether $SL(P \oplus k_1) \oplus SL(P^* \oplus k_1)$ are same at the seven bytes (3,4,6,8,9,13,14). If yes, discard the candidate value of the seven bytes of k_1 and the five bytes of k_7.

Since such a difference is impossible, every key that proposes such a difference is a wrong key. After analyzing 2^{55} ciphertext pairs, there remain only about $2^{56}(1-2^{-48})^{2^{55}} \approx 2^{56}e^{-2^7} \approx 2^{-128}$ wrong values of the seven bytes of k_1. Unless the initial assumption on the five bytes of k_7 is right, it is expected that we can detect the whole 56-bit value of k_1 for each 40-bit value of k_7 since the wrong value remains with the probability 2^{-88}. Hence if there remains a value of k_1, we can assume the value k_7 is right.

The time complexity of the attack is dominated by Step 3. For reducing the time complexity of Step 3.1, we first compute $C_{(5,1)} \oplus C^*_{(5,1)}$ and $C_{(5,8)} \oplus C^*_{(5,8)}$, and check whether $C_{(5,1)} \oplus C^*_{(5,1)} = C_{(5,8)} \oplus C^*_{(5,8)}$, it needs only to guess two key bytes. If yes, go on computing $C_{(5,9)} \oplus C^*_{(5,9)}$, and so on. Thus Step 3.1 requires about $4 \times 2^{103} (= 2^{16} \times 2^{87} + 2^{24} \times 2^{79} + 2^{32} \times 2^{71} + 2^{40} \times 2^{63})$ one round operations. Step 3.2 requires about $6 \times 2^{111} (= 2^{40} \times (2^{16} \times 2^{55} + 2^{24} \times 2^{47} + \cdots + 2^{56} \times 2^{15})$ one round operations.

Similarly, we can derive the other bytes of k_7 by using different impossible differentials. Consequently, this attack requires about 2^{121} chosen plaintexts and 2^{112} encryptions of 6-round ARIA.

4 Impossible Differential Cryptanalysis on Reduced-round Camellia

4.1 Some 8-Round Impossible Differentials

In [4], the authors show one impossible differential of 7-round Camellia without input/output whitening, FL, or FL^{-1}. In this subsection, we indicate one impossible differential of 8-round Camellia as shown in Fig.3.

We now show the 8-round differential

$$(0|0|0|0|0|0|0|0, a|0|0|0|0|0|0|0) \xrightarrow{\text{8-round}} (h|0|0|0|0|0|0|0, 0|0|0|0|0|0|0|0)$$

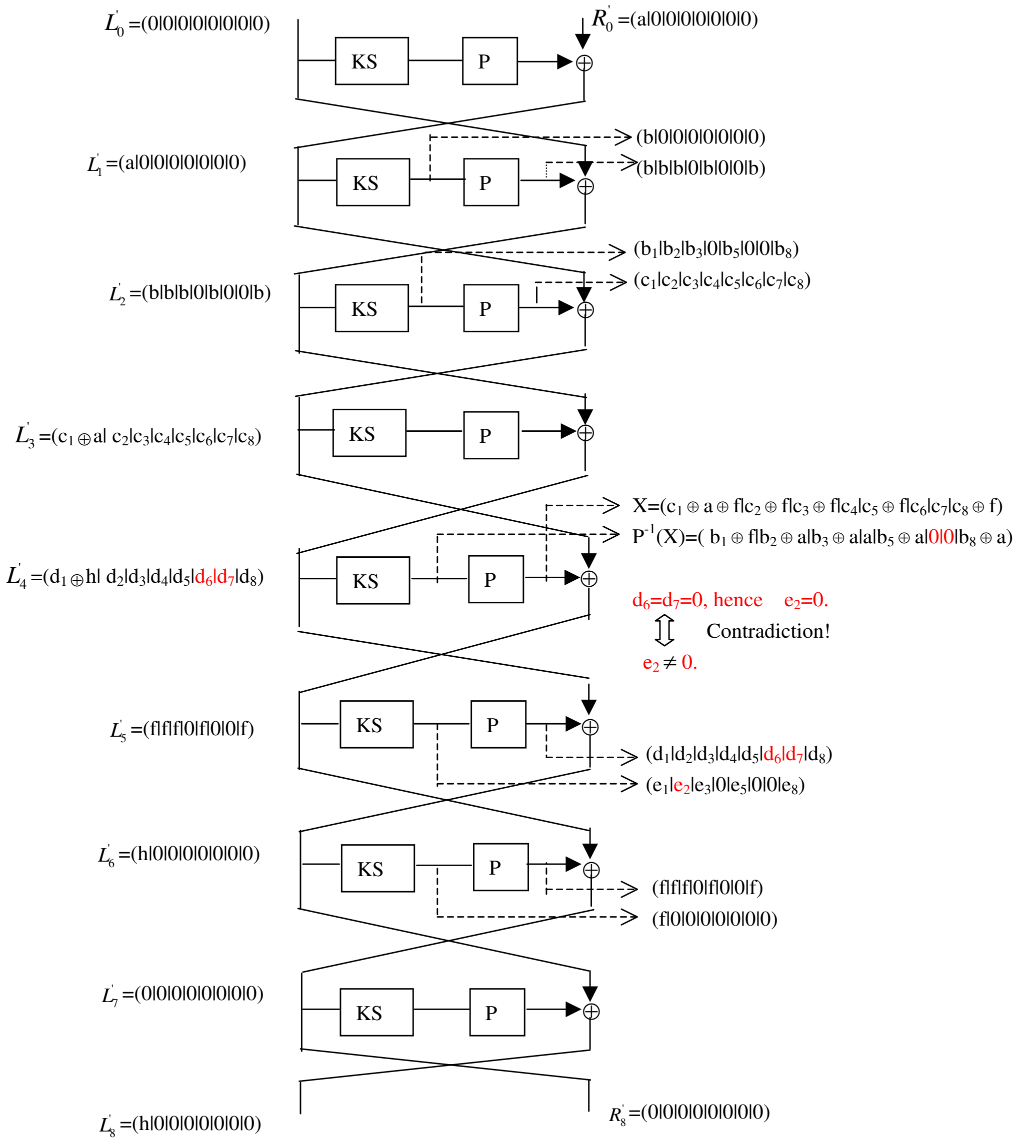

Fig.3. 8-Round Impossible Differentials of Camellia

is impossible, where a and h denote any non-zero value.

The first 3-round differential is obtained as follows:

The input difference $(L_0^{'}, R_0^{'}) = (0|0|0|0|0|0|0|0, a|0|0|0|0|0|0|0)$ becomes $(L_1^{'}, R_1^{'}) = (a|0|0|0|0|0|0|0, 0|0|0|0|0|0|0|0)$ through the first round transformation. After the subkey addition and S layer, $L_1^{'}$ becomes $(b|0|0|0|0|0|0|0)$ where b is an unknown non-zero byte value. After the linear transformation P we have $(L_2^{'}, R_2^{'}) = (b|b|b|0|b|0|0|b, a|0|0|0|0|0|0|0)$. This difference evolves after subkey addition operation and the S-box layer of round 3 into $(b_1|b_2|b_3|0|b_5|0|0|b_8)$, where b_1, b_2, b_3, b_5 and b_8 are unknown non-zero byte values. Further, after the linear transformation P this difference evolves to $(c_1|c_2|c_3|c_4|c_5|c_6|c_7|c_8)$. Thus we get

$$(L_3^{'}, R_3^{'}) = (c_1 \oplus a|c_2|c_3|c_4|c_5|c_6|c_7|c_8, b|b|b|0|b|0|0|b).$$

The second 3-round differential ends with difference $(L_8^{'}, R_8^{'}) = (h|0|0|0|0|0|0|0, 0|0|0|0|0|0|0|0)$. When rolling back this difference through 2-round transformation, we get the difference $(L_6^{'}, R_6^{'}) = (h|0|0|0|0|0|0|0, f|f|f|0|f|0|0|f)$, where f is an unknown non-zero byte value. After the subkey addition and S layer, $L_5^{'} = R_6^{'}$ becomes $(e_1|e_2|e_3|0|e_5|0|0|e_8)$, where e_1, e_2, e_3, e_5 and e_8 are unknown non-zero byte values. Further, after the linear transformation P this difference evolves to $(d_1|d_2|d_3|d_4|d_5|d_6|d_7|d_8)$. Thus we get

$$(L_5^{'}, R_5^{'}) = (f|f|f|0|f|0|0|f, d_1 \oplus h|d_2|d_3|d_4|d_5|d_6|d_7|d_8),$$

where d_6 and d_7 can be expressed as:

$$d_6 = e_2 \oplus e_3 \oplus e_5 \oplus e_8, \qquad d_7 = e_3 \oplus e_5 \oplus e_8.$$

If the first 3-round differential and second 3-round differential can build up the 8-round differential, then L_3, L_5 and R_5 must satisfy the following:

$$L_4 = R_5, \qquad P(S(R_5 \oplus k_4)) = L_3 \oplus L_5.$$

Hence we have $S(R_5 \oplus k_4) = P^{-1}(L_3 \oplus L_5)$. Because P^{-1} is a linear transformation, we have

$$\begin{aligned}
&P^{-1}(L_3^{'} \oplus L_5^{'}) = P^{-1}(L_3^{'}) \oplus P^{-1}(L_5^{'}) \\
&= P^{-1}(c_1 \oplus a|c_2|c_3|c_4|c_5|c_6|c_7|c_8) \oplus P^{-1}(f|f|f|0|f|0|0|f) \\
&= P^{-1}(c_1|c_2|c_3|c_4|c_5|c_6|c_7|c_8) \oplus P^{-1}(a|0|0|0|0|0|0|0) \oplus P^{-1}(f|f|f|0|f|0|0|f) \\
&= (b_1|b_2|b_3|0|b_5|0|0|b_8) \oplus (0|a|a|a|a|0|0|a) \oplus (f|0|0|0|0|0|0|0) \\
&= (f \oplus b_1|b_2 \oplus a|b_3 \oplus a|a|b_5 \oplus a|0|0|b_8 \oplus a).
\end{aligned}$$

The $s-$boxes of Camellia are permutations, so we can get the sixth and seventh byte difference in $R_5^{'}$ equal zero,i.e., $d_6 = d_7 = 0$. From the expression of d_6 and d_7 we have $d_6 \oplus d_7 = e_2$. This contradicts with $e_2 \neq 0$.

Similarly, we can get other 8-round impossible differentials of Camellia, for example,

$$(0|0|0|0|0|0|0|0, 0|a|0|0|0|0|0|0) \xrightarrow{\text{8-round}} (0|h|0|0|0|0|0|0, 0|0|0|0|0|0|0|0),$$

$$(0|0|0|0|0|0|0|0, 0|0|a|0|0|0|0|0) \xrightarrow{\text{8-round}} (0|0|h|0|0|0|0|0, 0|0|0|0|0|0|0|0).$$

4.2 12-Round Impossible Differential Attack

In this subsection, we describe an impossible differential attack of Camellia reduced to twelve rounds. The attack is based on the above 8-round impossible differentials with additional three rounds at the beginning and one round at the end as in Fig.4..

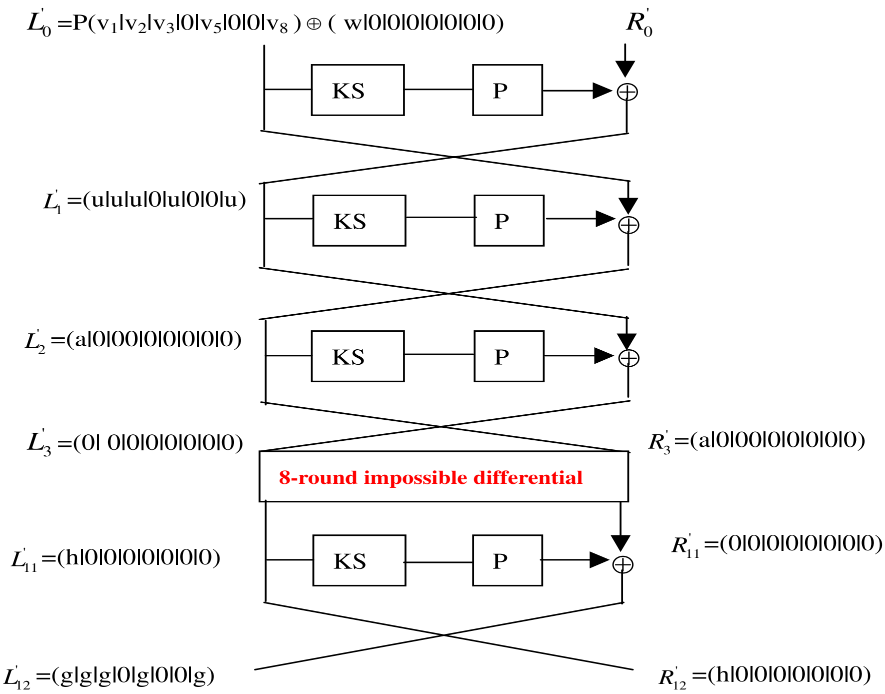

Fig.4. 12-Round Impossible Differential Attack to Camellia

The procedure is as follows:

Step 1 Choose structure of plaintexts as follows:

$$L_0 = P(x_1|x_2|x_3|\alpha_4|x_5|\alpha_6|\alpha_7|x_8) \oplus (x|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8),$$
$$R_0 = (y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8).$$

where $x_i(i = 1, 2, 3, 5, 8)$, $y_i(1 \le i \le 8)$, and x take all possible values in F_2^8, α_i, and β_i are constants in F_2^8. For each possible value of $(x_1, x_2, x_3, x_5, x_8, x, y_1, \ldots, y_8)$, we can get a unique 128-bit string $(P(x_1|x_2|x_3|\alpha_4|x_5|\alpha_6|\alpha_7|x_8)\oplus(x|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8)$, $(y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8))$. Also, for different value of $(x_1, x_2, x_3, x_5, x_8, x, y_1, \ldots, y_8)$, the corresponding 128-bit string is also different. Hence, a structure includes 2^{112} plaintexts, one structure proposes $2^{112} \times 2^{112} \times \frac{1}{2} = 2^{223}$ pairs of plaintexts.

Step 2 Take 2^8 structures (2^{120} plaintexts, 2^{231} pairs of plaintexts). Choose pairs whose ciphertext difference $(L_{12}^{'}, R_{12}^{'})$ satisfy the following:

$$L_{12}^{'} = (g|g|g|0|g|0|0|g),$$
$$R_{12}^{'} = (h|0|0|0|0|0|0|0).$$

where h and g are unknown non-zero values. There are 2^{16} $(L_{12}^{'}, R_{12}^{'})$, so the probability is about $p = 2^{16} \times 2^{-128} = 2^{-112}$. Hence, the expected number of such pairs is $2^{231} \times 2^{-112} = 2^{119}$.

Step 3 Guess the 8-bit value at the first byte of the subkey k_{12}, for every remaining pair, calculate $s_1(R_{12,1} \oplus k_{12,1}) \oplus s_1(R_{12,1}^{*} \oplus k_{12,1})$, and choose pairs which satisfy $s_1(R_{12,1} \oplus k_{12,1}) \oplus s_1(R_{12,1}^{*} \oplus k_{12,1}) = L_{12,1} \oplus L_{12,1}^{*}$. Since the probability is about $p = 2^{-8}$, the expected number of the remaining pairs is $2^{119} \times 2^{-8} = 2^{111}$.

Step 4 Guess the 64-bit value of the first round key k_1, for every remaining plaintext pair (L_0, R_0) and (L_0^{*}, R_0^{*}),

$$L_0 = P(x_1|x_2|x_3|\alpha_4|x_5|\alpha_6|\alpha_7|x_8) \oplus (x|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8),$$
$$R_0 = (y_1|y_2|y_3|y_4|y_5|y_6|y_7|y_8),$$
$$L_0^{*} = P(x_1^{*}|x_2^{*}|x_3^{*}|\alpha_4|x_5^{*}|\alpha_6|\alpha_7|x_8^{*}) \oplus (x^{*}|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8),$$
$$R_0^{*} = (y_1^{*}|y_2^{*}|y_3^{*}|y_4^{*}|y_5^{*}|y_6^{*}|y_7^{*}|y_8^{*}).$$

Compute (L_1, R_1) and (L_1^{*}, R_1^{*}), and choose pairs whose difference satisfy $L_1 \oplus L_1^{*} = (u|u|u|0|u|0|0|u)$ where u is not zero. Since the probability is about $p = 2^8 \times 2^{-64} = 2^{-56}$, the expected number of the remaining pairs is $2^{111} \times 2^{-56} = 2^{55}$.

Step 5 Guess the 40-bit value of the second round key k_2 at the five bytes(1,2,3,5,8), perform the following:

Step 5.1 For every remaining pair (L_0, R_0) and (L_0^{*}, R_0^{*}), and the corresponding output of the first round (L_1, R_1) and (L_1^{*}, R_1^{*}),

$$L_1 = (z_1|z_2|z_3|\gamma_4|z_5|\gamma_6|\gamma_7|z_8),$$
$$R_1 = P(x_1|x_2|x_3|\alpha_4|x_5|\alpha_6|\alpha_7|x_8) \oplus (x|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8),$$
$$L_1^{*} = (z_1^{*}|z_2^{*}|z_3^{*}|\gamma_4|z_5^{*}|\gamma_6|\gamma_7|z_8^{*}),$$
$$R_1^{*} = P(x_1^{*}|x_2^{*}|x_3^{*}|\alpha_4|x_5^{*}|\alpha_6|\alpha_7|x_8^{*}) \oplus (x^{*}|\beta_2|\beta_3|\beta_4|\beta_5|\beta_6|\beta_7|\beta_8).$$

Compute $s_1(z_1 \oplus k_{2,1}) \oplus s_1(z_1^{*} \oplus k_{2,1}) = v_1$, $s_2(z_2 \oplus k_{2,2}) \oplus s_2(z_2^{*} \oplus k_{2,2}) = v_2$, $s_3(z_3 \oplus k_{2,3}) \oplus s_3(z_3^{*} \oplus k_{2,3}) = v_3$, $s_2(z_5 \oplus k_{2,5}) \oplus s_2(z_5^{*} \oplus k_{2,5}) = v_5$, $s_1(z_8 \oplus k_{2,8}) \oplus s_1(z_8^{*} \oplus k_{2,8}) = v_8$. Choose pairs whose difference satisfy $(v_1|v_2|v_3|v_5|v_8) = (x_1 \oplus x_1^{*}|x_2 \oplus x_2^{*}|x_3 \oplus x_3^{*}|x_5 \oplus x_5^{*}|x_8 \oplus x_8^{*})$ and $x \neq x^{*}$. Since the probability is about $p = 2^{-40}$, the expected number of the remaining pairs is $2^{55} \times 2^{-40} = 2^{15}$.

Step 5.2 Further guess the 24-bit value of the second round key k_2 at the three bytes (4,6,7), for every remaining plaintext pair, calculate $L_{2,1}$ and $L_{2,1}^{*}$.

Step 6 For 8-bit value at the first byte of the subkey k_3, for every remaining plaintext pair, calculate $s_1(L_{2,1} \oplus k_{3,1}) \oplus s_1(L_{2,1}^{*} \oplus k_{3,1})$, and check whether $s_1(L_{2,1} \oplus k_{3,1}) \oplus s_1(L_{2,1}^{*} \oplus k_{3,1}) = L_{1,1} \oplus L_{1,1}^{*}$. If yes, discard the candidate value of $(k_1, k_2, k_{3,1}, k_{12,1})$.

Since such a difference is impossible, every key that proposes such a difference is a wrong key. After analyzing 2^{15} ciphertext pairs, there remain only about $2^{144}(1-2^{-8})^{2^{15}} \approx 2^{144}e^{-2^7} \approx 2^{-50}$ wrong candidate value of $(k_1, k_{2,1}, k_{11,1}, k_{12})$.

The time complexity of Step 3 requires about $2^{127} = 2^8 \times 2^{119}$ one round operations. Step 4 requires about $2^{183} = 2^{64} \times 2^8 \times 2^{111}$ one round operations. Step 5.1 requires about $2^{167} = 2^8 \times 2^{64} \times 2^{40} \times 2^{55}$ one round operations. Step 5.2 requires about $2^{151} = 2^8 \times 2^{64} \times 2^{64} \times 2^{15}$ one round operations. Step 6 requires about $2^{159} = 2^{72} \times 2^{72} \times 2^{15}$ one round operations.

Consequently, this attack requires about 2^{120} chosen plaintexts and less than 2^{181} encryptions of 12-round Camellia.

5 Concluding Remarks

In this paper, we examine the security of ARIA and Camellia against impossible differential attacks. The designers of ARIA expected no impossible differentials exist on 4-round ARIA. However, we found some nontrivial 4-round impossible differentials, and then presented an attack to 6-round ARIA with data complexity 2^{121} and 2^{112} encryptions. As for Camellia, we found some nontrivial 8-round impossible differentials for Camellia, whereas only 7-round impossible differentials were previously known. By using the 8-round impossible differential, we presented an attack on 12-round Camellia with data complexity 2^{120} and 2^{181} encryptions, the attack having complexity less than that of exhaustive search to 12-round Camellia without FL/FL^{-1} layers.

Since ARIA is a new cipher published in 2004, all we know about its security is limited to the designers' analysis and that of [11]. Here we only compare the complexities of our attack with those of previous works on Camellia in the following table.

Acknowledgment

This research is supported by the National Natural Science Foundation of China under Grant No. 60373047 and 90604036; the National Basic Research 973 Program of China under Grant No.2004CB318004.

References

1. Daesung Kwon, Jaesung Kim, Sangwoo Park, et al. New Block Cipher:ARIA. *Information Security and Cryptology-ICISC'03,* LNCS 2971, pp.432-445, Springer-Verlag 2004.
2. K.Aoki, T.Ichikawa, M.Kanda, M.Matsui, S.Moriai, J.Nakajima and T.Tokita. Specification of Camellia—a 128-bit Block Cipher. *Selected Areas in Cryptography - SAC'2000*, LNCS 2012, pp.183-191, Springer-Verlag 2001.

Table 1. The summary of known attacks on Camellia

Rounds	FL/FL^{-1}	Methods	Data	Time	Notes
7	$\times$	Impossible DC	–	–	Ref.[4]
8	$\times$	Truncated DC	$2^{83.6}$	$2^{55.6}$	Ref.[3](128-bit key)
9	$\vee$	Boomerang	2^{124}	2^{170}	Ref.[7](192/256-bit key)
9	$\times$	Collision Attack	2^{13}	$2^{175.6}$	Ref.[9](192/256-bit key)
9	$\vee$	Integral Attack	$2^{60.5}$	$2^{202.2}$	Ref.[8](256-bit key)
9	$\vee$	Square Attack	2^{60}	2^{202}	Ref.[6](256-bit key)
10	$\vee$	Rectangle	2^{127}	2^{241}	Ref.[7](256-bit key)
10	$\times$	Collision Attack	2^{14}	$2^{239,9}$	Ref.[9](256-bit key)
10	$\times$	Variant Square Attack	–	2^{186}	Ref.[10](192/256-bit key)
11	$\times$	DC	2^{104}	2^{232}	Ref.[7](256-bit key)
11	$\times$	Variant Square Attack	–	2^{250}	Ref.[10](256-bit key)
11	$\times$	Higher Order DC	2^{21}	2^{255}	Ref.[5](256-bit key)
11	$\vee$	Higher Order DC	2^{93}	2^{256}	Ref.[5](256-bit key)
12	$\times$	Linear Attack	2^{119}	2^{247}	Ref.[7](256-bit key)
12	$\times$	Impossible DC	2^{120}	2^{181}	This paper(192/256-bit key)

3. S. Lee, S. Hong, S. Lee, J. Lim and S. Yoon. Truncated Differential Cryptanalysis of Camellia. *Information Security and Cryptology-ICISC'01*, LNCS 2288, pp.32-38, Springer-Verlag 2002.
4. M.Sugita,K.Kobara, and H.Imai. Security of Reduced Version of the Block Cipher Camellia against Truncated and Impossible Differential Cryptanalysis. *Advances in Cryptology– Asiacrypt'01*, LNCS 2248, pp.193-207, Springer-Verlag 2001.
5. Y.Hatano,H.Sekine, and T.Kaneko. Higher Order Differential Attack of Camellia(II). *Selected Areas in Cryptography-SAC'02*, LNCS 2595, pp.39-56, Springer-Verlag 2003.
6. Y.Yeom, S.Park, and I.Kim. On the Security of Camellia against the Square Attack. *Fast Software Encryption-FSE'02*, LNCS 2356, pp.89-99, Springer-Verlag 2002.
7. T.Shirai. Differential, Linear, Boomerang and Rectangle Cryptanalysis of Reduced-Round Camellia. In Proceedings of the Third NESSIE Workshop, 2002.
8. Y.Yeom, I. Park, and I. Kim. A Study of Integral Type Cryptanalysis on Camellia. *The 2003 Symposium on Cryptography and Information Security-SCIS'03.*
9. Wenling Wu, Dengguo Feng, and Hua Chen. Collision Attack and Pseudorandomness of Reduced-round Camellia. *Selected Areas in Cryptography-SAC 2004*, LNCS 3357, pp.256-270, Springer-Verlag 2005.
10. Duo Lei, Li Chao, and Keqin Feng. New Observatiion on Camellia. *Selected Areas in Cryptography-SAC 2005*, LNCS 3897, pp.51-64, Springer-Verlag 2006.
11. Alex Biryukov, Christophe De Canniere, et al. Security and Performance Analysis of ARIA. http://homes.esat.kuleuven.be/ abiryuko/ARIA-COSICreport.pdf
12. Eli Biham, Adi Shamir. Differential Cryptanalysis of the Data Encryption Standard, Springer-Verlag, 1993.
13. M.Matsui. Linear Cryptanalysis Method for DES Cipher. *Advances in Cryptology-EUROCRYPT'93,* LNCS 765, pp.386-397, Springer-Verlag 1993.
14. L. Knudsen. Truncated and Higher Order Differentials. *Fast Software Encryption-FSE'95*, LNCS 2595, pp.196-211, Springer-Verlag 1995.

15. E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. *Advances in Cryptology-EUROCRYPT'99,* LNCS 2595, pp.12-23, Springer-Verlag 1999.
16. A. Biryukov and D. Wagner. Slide Attacks. *Fast Software Encryption-FSE'99*, LNCS 1636, pp.245-259, Springer-Verlag 1999.
17. A. Biryukov and D. Wagner. Advanced Slide Attacks. *Advances in Cryptology-EUROCRYPT'00,* LNCS 1807, pp.589-606, Springer-Verlag 2000.
18. L. Knudsen and D. Wagner. Integral Cryptanalysis(extended abstract). *Fast Software Encryption-FSE 2002*, LNCS 2595, pp.112-127, Springer-Verlag 2002.
19. D. Wagner. The Boomerang Attack. *Fast Software Encryption-FSE'99*, LNCS 1636, pp.157-170, Springer-Verlag 1999.
20. T. Jakobsen and L. Knudsen. The Interpolation Attack against Block Ciphers. *Fast Software Encryption-FSE'99*, LNCS 1267, pp.28-40, Springer-Verlag 1999.
21. N. Courtois and J. Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. *Advances in Cryptology-ASIACRYPT'02,* LNCS 2595, pp.267-287, Springer-Verlag 2002.
22. Jung Hee Cheon, Munju Kim, Kwangjo Kim, et al. Improved Impossible Differential Cryptanalysis of Rijndael and Crypton. *International Conference on Information Security and Cryptology-ICISC'01,* LNCS 2288, pp.39-49, Springer-Verlag 2002.
23. Raphael Chung-Wei Phan. Impossible Differential Cryptanalysis of 7-round AES. Information Processing Letters, Vol.91, Number 1, pp.33-38, 2004.
24. Goce Jakimoski, Yvo Desmedt. Related-Key Differential Cryptanalysis of 192-bit key AES Variants. *Selected Areas in Cryptography -SAC'2003*, LNCS 3006, pp. 208-221, Springer-Verlag 2004.
25. Eli Biham, Orr Dunkelman, and Nathan Keller. Related-Key Impossible Differential Attacks on 8-round AES-192. *CT-RSA'2003*, LNCS 3860, pp. 21-33, Springer-Verlag 2006.
26. Bon Wook Koo, Hwan Seok Jang, and Jung Hwan Song. Constructing and Cryptanalysis of a 16×16 Binary Matrix as a Diffusion Layer. *WISA'2003*, LNCS 2908,pp.489-503, Springer-Verlag 2004.