JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1041-1053 (2017)
DOI: 10.6688/JISE.2017.33.4.12

Impossible Differential Analysis on Round-Reduced PRINCE

YAO-LING DING[1], JING-YUAN ZHAO[2], LEI-BO LI[3] AND HONG-BO YU[1,4,+]
[1]*Department of Computer Science and Technology*
Tsinghua University
Beijing, 100084 P.R. China
[2]*State Key Laboratory of Information Security, Institute of Information Engineering*
Chinese Academy of Sciences
Beijing, 100084 P.R. China
[3]*Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education*
Shandong University
Jinan, 250100 P.R. China
[4]*Science and Technology on Communication Security Laboratory*
Chengdu, 610041 P.R. China
E-mail: dyl13@mails.tsinghua.edu.cn; zhaojingyuan@iie.ac.cn;
lileibo@mail.sdu.edu.cn; yuhongbo@mail.tsinghua.edu.cn

PRINCE is a lightweight block cipher proposed at ASIACRYPT 2012, which is composed of a 12-round core cipher referred to as PRINCE$_{core}$ and two key whitening layers. The security of the cipher mainly depends on PRINCE$_{core}$. In this paper, we give some observations on M' operation, a part of the linear layer in the round function, to construct a 4-round impossible differential distinguisher. Based on the distinguisher, impossible differential attacks on 6-round and 7-round PRINCE$_{core}$ are launched. Moreover, we extend them to analysis 6- and 7-round PRINCE by guessing equivalent keys. The complexity of our attacks meets the security claims stated by the designers.

Keywords: PRINCE$_{core}$, impossible differential, M' operation, cryptanalysis, light-weight block cipher

1. INTRODUCTION

Since many countries work on building smart cities, the security of communication is drawing much more attentions. Many corresponding applications should be supported by cryptographic techniques, such as CBIR scheme [1] in cloud computing, search over encrypted outsourced data [2, 3], cloud storage [4] and the internet of things. However, in some cases, the hardware and energy provided for security are limited, such as the smart sensors used in internet of things. Therefore, the resources that can be used for security are few. In order to use a minimum of resource to provide required security in some extreme application, some designers proposed the concept of lightweight block cipher, which can guarantee the privacy and occupy fewer resources than the classic block cipher at the same time. Several lightweight block ciphers have been proposed in the last decade, and PRINCE is one of them.

Received August 15, 2016; revised September 28, 2016; accepted November 5, 2016.
Communicated by Zhe Liu.
[+] The corresponding author.
[*] This work was supported by 973 Program (No. 2013CB834205) and the National Natural Science Foundation of China (No. 61133013 and No. 61373142).

PRINCE, proposed by Borghoff *et al.*, is a low-latency block cipher [5]. In order to reduce the latency in hardware implementation, the designers employ the FX construction [6, 7] and a property called α-reflection. Based on the FX construction, a 12-round core cipher, called PRINCE$_{core}$ is used to holds the major encryption process. Due to the α-reflection property, one can reuse the encryption process with $k_1 \oplus \alpha$ to perform decryption, where k_1 is the subkey used in the round function. Obviously, the α-reflection simplifies the implementation of the compatibility of encryption and decryption while it reduces the security of the cipher. The designers claimed that the operations of encryption or decryption should not be more than 2^{127-n} when 2^n queries were made with a single fixed unknown key.

Several cryptanalytic results on PRINCE, and the underlying PRINCE$_{core}$, have been presented. In [7], Jean *et al.* did a lot of investigations and presented their results at FSE 2013. They attacked the full cipher with related keys using boomerang cryptanalysis, and derived an attack in the single-key model for PRINCE$_{core}$ with several instances of the α parameter (neither random nor the one chosen by the designers). And they gave SQUARE attacks on 4, 5 and 6-round PRINCE$_{core}$ as well as PRINCE with the original α. Also at FSE 2013, Soleimany *et al.* introduced new generic distinguishers in [9], namely reflection cryptanalysis, on PRINCE-like ciphers and investigated many classes of α to launch attacks on the full cipher. However, the complexity violated the security claims of the designers. Then a 6-round reflection attack was claimed but not detailed in [9]. In[10], Abed *et al.* presented an independent-biclique attack for the full version of PRINCE$_{core}$ and also a differential cryptanalysis on a reduced version of PRINCE$_{core}$. Besides, various cryptanalytic techniques have been employed to evaluate PRINCE and got some new results, such as meet-in-the-middle attack [11-13], sieve-in-the-middle attack [14], integral cryptanalysis [15] and differential attack [11, 15].

The impossible differential attack was independently proposed by Biham *et al.* and Knudsen in [16, 17]. The attacker finds an impossible differential firstly. Then, several rounds are added before and/or after the impossible differential path. After collecting sufficient certain plaintext-ciphertext pairs, the attacker partly encrypts and decrypts the pairs with guessed keys. The keys that lead to the impossible differential will be eliminated. The attacker finally obtains the correct key by discarding the candidates until there remains only one.

In this paper, we find a property of M' operation which is the central layer of PRINCE and a part of the linear layer of the round function. The property shows that, for every column of the input and output states of M' operation (regarding the state as a 4×4 nibble matrix), if there are only two active nibbles in the input, there are at least two active nibbles in the output. Moreover, if there are only two active nibbles in the output, and the two input active nibbles are adjacent, the two output active nibbles must be adjacent, and vice versa. Based on the property, a 4-round impossible differential distinguisher of PRINCE is constructed, and we use it to launch impossible differential attacks on 6-round and 7-round PRINCE$_{core}$ which are further extended to attacks on PRINCE.

The rest of the paper is organized as follows. We give a brief description of PRINCE in Section 2. Section 3 identifies the impossible differential property for PRINCE. Impossible differential attacks on 6-round and 7-round PRINCE$_{core}$ and PRINCE are proposed in Section 4. We conclude the paper in Section 5.

2. BRIEF DESCRIPTION OF PRINCE

PRINCE [6] is a 64-bit block cipher with a 128-bit key. We denote the 128-bit master key by k, which is divided into two 64-bit subkeys, k_0 and k_1 in the key schedule algorithm. The key expansion derives k_0' from k_0 employing shift and rotation. k_0 and k_0' are taken as the pre-whitening and post-whitening keys respectively, and k_1 is used in the round function of PRINCE$_{core}$. The whole key expansion routine is as Eq. (1).

$$k = (k_0 \| k_1) \rightarrow (k_0 \| k_0' \| k_1) := (k_0 \| (k_0 >>> 1) \oplus (k_0 >> 63) \| k_1), \tag{1}$$

where || denotes the concatenation, >>> denotes right rotation and >> denotes right shift.

PRINCE$_{core}$, a 12-round block cipher, is the core encryption (decryption) logic of PRINCE. We denote the round function by R_i ($0 \le i \le 11$). The six forward rounds, excepting for R_5, are composed of a round constant addition layer, a key addition layer, an Sbox layer and a linear layer. The linear layer can be divided into two parts which are AES-like shift rows and M' operation. R_5 is identical to the other five forward rounds except its lack of the linear layer. The six afterward rounds, excepting for R_6, are composed of an inverse linear layer, an inverse Sbox layer, a key addition layer and a round constant addition layer. Similarly, R_6 has no linear layer. Besides, the operations of the afterward rounds are in the reverse order of the forward ones. An M' operation is inserted in the middle to connect the two symmetric parts. The whole encryption process of PRINCE is depicted in Fig. 1.

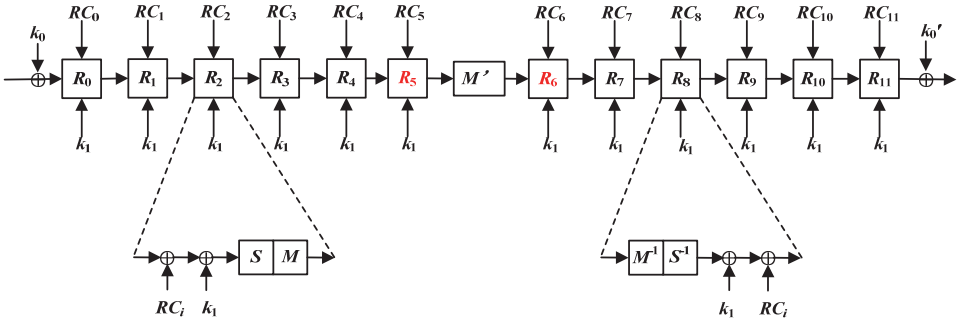

Fig. 1. Schematic view of the PRINCE cipher.

Key and Constant Addition Layers: The states of PRINCE$_{core}$, as well as k_0 and round constants, can be regarded as 4×4 nibble matrices. The 64-bit key k_1 and round constant are xored with the state in the key addition layer and constant addition layer respectively. Note that the round constants satisfy the condition

$$RC_i \oplus RC_{11-i} = \alpha = c0ac29b7c97c50dd, 0 \le i \le 11.$$

Sbox Layer: The cipher employs a 4-bit Sbox that is given in hexadecimal format as $S[x]$={0xB, 0xF, 0x3, 0x2, 0xA, 0xC, 0x9, 0x1, 0x6, 0x7, 0x8, 0x0, 0xE, 0x5, 0xD, 0x4}.

Linear Layer: The linear layer is defined by the authors as $M = SR \bullet M'$, in which SR denotes shift rows that behaves like AES and M' denotes a matrix multiplication. M' also indicates the matrix for simplification. In order to explain the property of M' clearly in the following section, we describe the block diagonal matrix $M' = diag(\hat{M}^{(0)}, \hat{M}^{(1)}, \hat{M}^{(1)}, \hat{M}^{(0)})$ in detail as Eq. (2).

$$\hat{M}^{(0)} = \begin{pmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{pmatrix}, \hat{M}^{(1)} = \begin{pmatrix} M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{pmatrix} \tag{2}$$

and the four 4×4 building matrices are as Eq. (3).

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

3. IMPOSSIBLE DIFFERENTIAL PROPERTY OF PRINCE

In this section, we identify the impossible differential property of the M' operation. Based on it, a 4-round impossible differential distinguisher is presented.

3.1 Notations

In this paper, we use the following notations.
x_i: The (i+1)th nibble of the input of M' operation, $0 \le i \le 15$
y_i: The (i+1)th nibble of the output of M' operation, $0 \le i \le 15$
$x_{(i,j)}$: The (j+1)th bit of x_i, $0 \le j \le 3$
$y_{(i,j)}$: The (j+1)th bit of y_i, $0 \le j \le 3$
x^i: The input of R_i, $0 \le i \le 11$
y^i: The output of R_i, $0 \le i \le 11$
$x^i_{col(j)}$: The (j+1)th column of x^i, $0 \le j \le 3$
$y^i_{col(j)}$: The (j+1)th column of y^i, $0 \le j \le 3$
$k_{s,i}$: The (i+1)th nibble of k_s, $0 \le i \le 15$, $k_s \in \{k_0, k_1, k'_0\}$
$k_{s,col(j)}$: The (j+1)th column of k_s, $0 \le i \le 3$, $k_s \in \{k_0, k_1, k'_0\}$

3.2 Matrix Multiplication Property

Since M' is constructed by $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$, we study the properties of $\hat{M}^{(0)}$ and $\hat{M}^{(1)}$ instead of M'. Due to its diagonal property, M' operation can be viewed as four 16×16 bit matrix multiplications in four columns of the state respectively. The first and fourth columns of the state are multiplied by $\hat{M}^{(0)}$, while the other two columns are multiplied by $\hat{M}^{(1)}$. Owing to the linearity of $\hat{M}^{(0)}$, the analysis of its input and output differentials can be regarded as the analysis of its input and output text. We give the property of $\hat{M}^{(0)}$

as follows. We use active nibbles to demonstrate nonzero nibbles in order to consistent with the following.

Property 1: For the situation of there are only two active nibbles both in the input and the output of $\hat{M}^{(0)}$, adjacent active nibbles in the input lead to adjacent active nibbles in the output, while non-adjacent active nibbles in the input lead to non-adjacent active nibbles in the output. That is to say, adjacent active nibbles and non-adjacent active nibbles cannot lead to each other through $\hat{M}^{(0)}$.

***Proof*:** Take the first column as an example. The multiplication can be expressed as Eq. (4).

$$\left(y_0, y_1, y_2, y_3\right)^T = \hat{M}^{(0)}\left(x_0, x_1, x_2, x_3\right)^T. \tag{4}$$

We transform the matrix-level expressions to bit-level expressions and get 16-bit equations as in Eq. (5).

$$\begin{cases} y_{(0,0)} = x_{(1,0)} \oplus x_{(2,0)} \oplus x_{(3,0)} \\ y_{(0,1)} = x_{(0,1)} \oplus x_{(2,1)} \oplus x_{(3,1)} \\ y_{(0,2)} = x_{(0,2)} \oplus x_{(1,2)} \oplus x_{(3,2)} \\ y_{(0,3)} = x_{(0,3)} \oplus x_{(1,3)} \oplus x_{(2,3)} \end{cases} \begin{cases} y_{(1,0)} = x_{(0,0)} \oplus x_{(1,0)} \oplus x_{(2,0)} \\ y_{(1,1)} = x_{(1,1)} \oplus x_{(2,1)} \oplus x_{(3,1)} \\ y_{(1,2)} = x_{(0,2)} \oplus x_{(2,2)} \oplus x_{(3,2)} \\ y_{(1,3)} = x_{(0,3)} \oplus x_{(1,3)} \oplus x_{(3,3)} \end{cases}$$
$$\begin{cases} y_{(2,0)} = x_{(0,0)} \oplus x_{(1,0)} \oplus x_{(3,0)} \\ y_{(2,1)} = x_{(0,1)} \oplus x_{(1,1)} \oplus x_{(2,1)} \\ y_{(2,2)} = x_{(1,2)} \oplus x_{(2,2)} \oplus x_{(3,2)} \\ y_{(2,3)} = x_{(0,3)} \oplus x_{(2,3)} \oplus x_{(3,3)} \end{cases} \begin{cases} y_{(3,0)} = x_{(0,0)} \oplus x_{(2,0)} \oplus x_{(3,0)} \\ y_{(3,1)} = x_{(0,1)} \oplus x_{(1,1)} \oplus x_{(3,1)} \\ y_{(3,2)} = x_{(0,2)} \oplus x_{(1,2)} \oplus x_{(2,2)} \\ y_{(3,3)} = x_{(1,3)} \oplus x_{(2,3)} \oplus x_{(3,3)} \end{cases} \tag{5}$$

Assuming that x_0 and x_1 are zero and the other two variables are non-zero, we get the expressions of Eq. (6).

$$\begin{cases} y_{(0,0)} = x_{(2,0)} \oplus x_{(3,0)} \\ y_{(0,1)} = x_{(2,1)} \oplus x_{(3,1)} \\ y_{(0,2)} = x_{(3,2)} \\ y_{(0,3)} = x_{(2,3)} \end{cases} \begin{cases} y_{(1,0)} = x_{(2,0)} \\ y_{(1,1)} = x_{(2,1)} \oplus x_{(3,1)} \\ y_{(1,2)} = x_{(2,2)} \oplus x_{(3,2)} \\ y_{(1,3)} = x_{(3,3)} \end{cases}$$
$$\begin{cases} y_{(2,0)} = x_{(3,0)} \\ y_{(2,1)} = x_{(2,1)} \\ y_{(2,2)} = x_{(2,2)} \oplus x_{(3,2)} \\ y_{(2,3)} = x_{(2,3)} \oplus x_{(3,3)} \end{cases} \begin{cases} y_{(3,0)} = x_{(2,0)} \oplus x_{(3,0)} \\ y_{(3,1)} = x_{(3,1)} \\ y_{(3,2)} = x_{(2,2)} \\ y_{(3,3)} = x_{(2,3)} \oplus x_{(3,3)} \end{cases} \tag{6}$$

If there are two non-adjacent zero nibbles or more than two zero nibbles in (y_0, y_1, y_2, y_3), we can deduce that x_2 and x_3 equal to zero. That is contradicting to our assump-

tion. However, if there are two adjacent or less than two zero nibbles in (y_0, y_1, y_2, y_3), x_2 and x_3 cannot be deduced to equal to zero. Actually, we find all the four input-output pairs that satisfy the assumption and lead to two zero nibbles in the output equal to zero. They are (0x0011, 0x1100), (0x0022, 0x2002), (0x0044, 0x0044) and (0x0088, 0x0880) in hexadecimal format. The same results can be obtained when any other two adjacent nibbles in (x_0, x_1, x_2, x_3) are assumed to be zero. Similarly, we can proof that two non-adjacent zero nibbles in the input lead to two non-adjacent zero nibbles or less than two zero nibbles in the output. All pairs satisfying Property 1 are listed in Table 1.

Table 1. Pairs satisfying Property 1.

active nibbles in the input	input-output pairs in hexadecimal format (input, output)
x_0, x_1	(1100,0011),(2200,0220),(4400,4400),(8800,8008)
x_1, x_2	(0110,0110),(0220,2200),(4400,4004),(0880,0088)
x_2, x_3	(0011,1100),(0022,2002),(0044,0044),(0088,0880)
x_0, x_3	(1001,1001),(2002,0022),(4004,0440),(8008,8800)
x_0, x_2	(1010,0101),(2020,2020),(4040,0404),(5050,0505),(8080,8080),(a0a0,a0a0)
x_1, x_3	(0101,1010),(0202,0202),(0404,4040),(0505,5050),(0808,0808),(0a0a,0a0a)

Although $\hat{M}^{(1)}$ and $\hat{M}^{(0)}$ are different in the order of rows, they have the same property which can be proof similarly, we omit it here.

3.3 Impossible Differential Path of PRINCE$_{core}$

In this section, we present a 4-round impossible differential path of PRINCE$_{core}$ which is defined as follows:

Property 2: (4-round Impossible Differential of PRINCE$_{core}$) For a 4-round PRINCE$_{core}$ with an M' layer in the middle, Given a pair of inputs whose difference is non-zero in two adjacent columns (*i.e.* at least one active nibble in the non-zero column) but zero in the other two, then after 4-round PRINCE$_{core}$ encryption, the corresponding output difference cannot be non-zero in two non-adjacent columns but zero in the other two. Here, the first and the fourth columns are considered to be adjacent.

***Proof*:** We call a nibble that has non-zero difference an active nibble. As shown in Fig. 2. Let the first and the fourth columns of the input be non-zero columns, then after 2-round encryption, there must be at most two active nibbles in every column of y^1 and the two active nibbles must be adjacent.

While, for any output differences of which two non-adjacent columns are zero but the other two are not, after 2-round decryption from the ciphertexts, there must be at most two active nibbles in every column of x^2 and the two active nibbles must be non-adjacent.

y^1 and x^2 are the input and the output of M'. According to the property we give in Section 3.2, a contradiction appears.

Here, we should consider a degenerate situation. Since the branch number of $\hat{M}^{(0)}$ ($\hat{M}^{(1)}$) is four, the input with active nibbles will result in an output with at least one ac-

tive nibble. So after the shift rows operation, there may be only one or none active nibble in some (not all) columns in y^1 and x^2. However, the input and the output of the middle M' still produce contradictions in columns with active nibbles because of the branch number. Our property still holds in this situation.

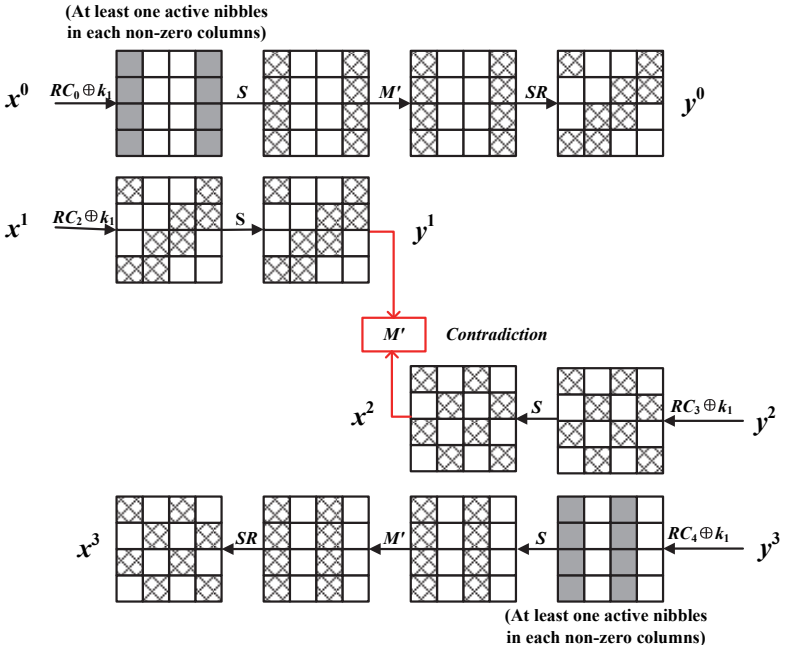

Fig. 2. Impossible differential path of PRINCE$_{core}$.

When extra rounds are added to the impossible differential path to launch impossible differential attacks, the position and number of the active nibbles in non-zero columns should be taken into consideration. Fig. 2 illustrates the impossible differential property as an instance. The boxes with shadow refer to uncertain nibbles, whereas the white ones denote the nibbles with zero difference. The gray columns denotes the non-zero columns which has one active nibble at least. Arrows labeled $RC_i \oplus K_1$, S, M and SR denote the addition of constants and subkey, Sbox, M' operation and shift rows, respectively. The box with M' refers to the middle M' operation.

4. IMPOSSIBLE DIFFERENTIAL ATTACKS ON PRINCE

In this section, we show our impossible differential attacks on 6-round and 7-round PRINCE$_{core}$ using the 4-round impossible differential property presented in Section 3. Then, we show how to extend them to impossible differential attacks on PRINCE.

4.1 Impossible Differential Attack on 6-round PRINCE$_{core}$

We define a structure as a set of 2^{16} x^0 which have fixed values in all nibbles but

nibbles of $x^0_{col(0)}$. Such a structure proposes 2^{31} pairs of x^0. We demonstrate the 6-round attack as follows, see also Fig. 3.

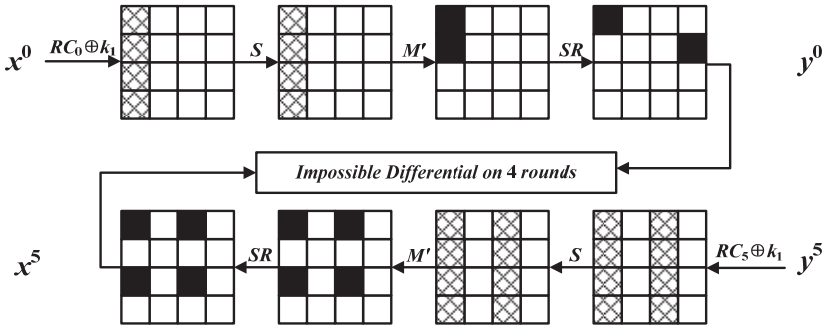

Fig. 3. 6-round impossible differential attack.

Step 1: Take 2^n structures of x^0. For each structure, query the corresponding x^5. Insert $(x^0_{col(0)}, y^5_{col(0)}, y^5_{col(2)})$ into a hash table indexed by $(y^5_{col(1)}, y^5_{col(3)})$. Save every pair of the elements in the row with more than one element. We expect to have $2^{n+31-32} = 2^{n-1}$ pairs on average at the end of this step.

Step 2: Guess the values of $k_{1,col(0)}$ and partially encrypt the corresponding nibbles in x^0. Choose pairs of which the differences after M' operation are non-zero at two adjacent nibbles and zero at the other two nibbles. In this step, we expect to have $2^{n-1} \times 2^{-8} \times 4 = 2^{n-7}$ pairs. The probability of two active nibbles is 2^{-8} and we have 4 choices to locate the two adjacent non-zero adjacent nibbles.

Step 3: For every $k_{1,col(0)}$ guessed in step 2, partially decrypt the associated nibbles in y^5. Choose pairs whose difference after M' operation is non-zero at two non-adjacent nibbles and zero at the other two nibbles in the first column (to ensure that after the shift row operation the active nibbles are in non-adjacent columns). In this step, we expect to have $2^{n-7} \times 2^{-8} \times 4 = 2^{n-14}$ pairs. The probability of having two active nibbles is 2^{-8} and we have two choices to locate the active nibbles.

Step 4: Guess the values of $k_{1,col(2)}$ and partially decrypt these nibbles in y^5. Choose pairs whose difference after M' operation are non-zero at the nibbles in the same row to the two in step 3 and zero at the other two nibbles(to avoid active nibbles in every column after shift row operation). In this step, we expect to have $2^{n-14} \times 2^{-8} \times 2 = 2^{n-22}$ pairs. 2^{-8} is the probability of having the active nibbles.

Step 5: If there are pairs left after step 4, we discard the corresponding $(k_{1,col(0)}, k_{1,col(2)})$ from the list of all the 2^{32} possible partial values of k_1.

Attack Complexity: The probability of a candidate for $(k_{1,col(0)}, k_{1,col(2)})$ being a wrong key is 2^{-21} from the average 2^{-6} for step 2, 2^{-7} for step 3 and 2^{-8} for step 4. After analyzing N pairs, we expect $2^{32} \times (1-2^{-21})^N$ key candidates remain. To single out the correct key, we

must guarantee $N \geq 25.6$. Here, the pairs are those remained after step 1. Thus, the attack requires $2^n=2^{26.6}$ structures. Consequently, the data complexity of the attack is $2^{26.6}\times2^{16}=2^{42.6}$ chosen plaintexts.

The time complexity of our attack consists of three parts. Step 2 requires $2\times2^{16}\times2^{25.6}=2^{42.6}$ one round encryptions from checking each of the $2^{25.6}$ pairs for the 2^{16} guessed keys. Step 3 requires $2\times2^{16}\times2^{19.6}=2^{36.6}$ one round decryptions from checking each of the $2^{19.6}$ remaining pairs for the 2^{16} keys guessed in step 2. And step 4 requires $2\times2^{16}\times2^{16}\times2^{12.6}=2^{45.6}$ one round decryptions from checking each of the $2^{12.6}$ remaining pairs for the 2^{16} keys guessed in step 2 and the 2^{16} keys guessed in this step. Consequently, we get the overall time complexity of the attack which is $(2^{42.6}+2^{35.6}+2^{45.6})/6=2^{43}$ full encryptions. We can obtain the other two columns of k_1 by exhaustive search with 2^{32} encryptions, so $2^{43}+2^{32}\approx2^{43}$ encryptions are required to recover the whole k_1.

The memory occupied by the attack can be viewed as two parts which are the key table and the pairs table for $(x^0_{col(0)}, y^5_{col(0)}, y^5_{col(2)})$. The key table, if indexed by the key values, requires about 2^{32} bits, *i.e.* 2^{29} bytes memory. Meanwhile, $2^{25.6}\times(16+32)\times2/8\approx2^{30}$ bytes of memory are needed to store the pairs remained after step 1. Accordingly, 2^{30} bytes of memory are necessary to launch the attack.

4.2 Impossible Differential Attack on 7-round PRINCE$_{core}$

In this section, we extend the impossible differential attack on 6-round PRINCE$_{core}$ described in Section 4.1 to a 7-round attack by adding one round at the bottom. The procedure of this attack is as follows in which we describe the similar step briefly, see also Fig. 4.

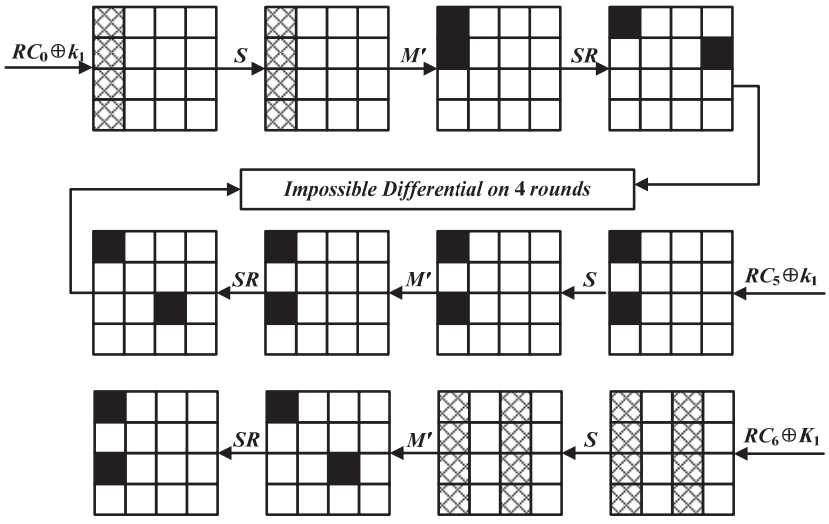

Fig. 4. 7-round impossible differential attack.

Step 1: Take 2^n structures of x^0. For each structure, insert $(x^0_{col(0)}, y^6_{col(0)}, y^6_{col(2)})$ into a hash table indexed by $(y^6_{col(1)}, y^6_{col(3)})$ and save pairs in the row with more than one element. We expect to get 2^{n-1} pairs in the step.

Step 2: Guess the values of $k_{1,col(0)}$ and collect pairs whose difference after M' operation of R_1 is non-zero at two adjacent nibbles and zero at the other two nibbles. We expect to get 2^{n-7} pairs.

Step 3: For every $k_{1,col(0)}$ guessed in step 2, partially decrypt the associated nibbles in y^6. Choose pairs whose difference after M' operation of R_6 is non-zero at only one nibble in $x^6_{col(0)}$. We have two choices to locate the non-zero nibble to guarantee that after shift rows operation its location is covered by the key guessed in steps 2 or 4 to avoid guessing two more key nibbles. We expect to get $2\times2^{n-7}\times2^{-12}\times2 = 2^{n-18}$ pairs.

Step 4: Guess the values of $k_{1,col(2)}$ and partially decrypt these nibbles in y^6. Choose pairs whose difference after M' operation is non-zero at the nibble in the non-adjacent row to the one in step 3 and zero at the other three nibbles. We keep these pairs to ensure that they are in the same column after shift rows operation. We expect to get 2^{n-30} pairs on average.

Step 5: For every $(k_{1,0}, k_{1,2})$ guessed in step 2 (decided by the locations of the active nibbles at the end of step 4) partially decrypt these nibbles in y^5. Choose pairs whose difference after M' operation is non-zero at the nibbles in non-adjacent rows and zero at the other two nibbles. In this step, we expect to have $2\times2^{n-30}\times(6/2^8)=2^{n-35.5}$ pairs. According to Table 1, the probability of the input difference with two non-adjacent active nibbles resulting in an output difference with two non-adjacent active nibbles is $6/2^8$.

Step 6: If there are pairs left after step 5, we discard the corresponding $(k_{1,col(0)}, k_{1,col(2)})$ from the list of all the 2^{32} possible values of k_1.

Attack Complexity The probability of a candidate for $(k_{1,col(0)}, k_{1,col(2)})$ being a wrong key is $2^{-34.5}$ (2^{-6} for step 2, 2^{-11} for step 3, 2^{-12} for step 4 and $2^{-5.5}$ for step 5). 2^{40} structures are needed to get the correct key, so the data complexity of the attack is 2^{56} chosen plaintexts.

The time complexity of our attack is dominated by step 2 and step 4 in which 2^{56} one round encryptions and 2^{55} one round decryptions are processed respectively. Hence, the overall time complexity is $2^{53.8}$ encryptions.

The memory occupied by the attack is decided by the pairs table for $(x^0_{col(0)}, y^6_{col(0)}, y^6_{col(2)})$. Accordingly, the amount of memory required for the attack is $2^{39}\times(16+32)\times2/8 \approx 2^{43}$ bytes.

4.3 Extending the Attacks to 6-round and 7-round PRINCE

Both of the previous attacks on PRINCE_{core} can be extended to attacks on PRINCE by guessing the equivalent keys $k_1\oplus k_0$ and $k_1\oplus k'_0$ instead of k_1. Therefore, the amount of keys to be guessed increases and leads to higher complexity.

As for 6-round PRINCE, we reuse the attack procedure of PRINCE_{core} except guessing $(k_1\oplus k_0)_{col(0)}$ in step 2, $(k_1\oplus k'_0)_{col(0)}$ in step 3 and $(k_1\oplus k'_0)_{col(2)}$ in step 4. As a result, we guess 2^{48} keys in total which make the memory occupied by the attack increase to 2^{45} bytes. From $2^{48}(1-2^{-21})^N=1$ we find that the data complexity is $2^{43.1}$ chosen plaintexts.

The time complexity is dominated by step 4 which needs $2^{62.1}$ one round decryptions, *i.e.* $2^{59.5}$ full encryptions. Then, we can apply a similar attack with columns 1 and 3 of $(k_1 \oplus k'_0)$. Since we can use the same data structures, the data complexity remains $2^{43.1}$ and time complexity is duplicated to $2^{60.5}$. Moreover, an extra step should be added to recover the whole 128-bit master key by exhausting the 2^{64} values of either k_0 or k_1. Hence, the time complexity of the attack is 2^{64} full encryptions.

For 7-round PRINCE, we guess $(k_1 \oplus k_0)_{col(0)}$ in step 2, $(k_1 \oplus k'_0)_{col(0)}$ in step 3 and $(k_1 \oplus k'_0)_{col(0)}$ in step 4 instead of $(k_{1,col(0)}, k_{1,col(2)})$. Besides, $(k_{1,0}, k_{1,2})$ have to be guessed in step 5. Therefore, 2^{53} bytes of memory are needed to store the keys. For this attack, we have four (instead of two in the attack on 7-round PRINCE$_{core}$) choices to locate the non-zero nibble in step 3, since we guess two key nibbles of k_1 anyway and do not have to ensure the two nibbles covered by the key nibbles guessed before. Hence, the probability of a candidate being a wrong key is $2^{-33.4}$. From $2^{56}(1-2^{-33.4})^N=1$, we get the data complexity to be $2^{-55.7}$ plaintexts. Similarly, we can apply an attack with column 1 and column3 of $(k_1 \oplus k'_0)$. And the data complexity remains $2^{55.7}$ but time complexity is duplicated to $2^{68.9}$. Although an exhausting step is added, the time complexity of the attack is $2^{68.9}$ full encryptions determined by step 4.

5. CONCLUSIONS

In this paper we introduce impossible differential attacks on light weight block cipher PRINCE, and the underlying PRINCE$_{core}$. We find a property of M', based on witch an impossible differential distinguisher is constructed. Using the distinguisher, we launch impossible differential attacks on 6-round and 7-round PRINCE$_{core}$. Furthermore, the attacks on PRINCE$_{core}$ are extended to attacks on PRINCE with the same number of rounds.

REFERENCES

1. Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, Vol. 11, 2016, pp. 2594-2608.
2. Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, 2015, pp. 2546-2559.
3. Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, Vol. E98-B, 2015, pp. 190-200.
4. Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," *Journal of Internet Technology*, Vol. 16, 2015, pp. 317-323.
5. J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. Thomsen, and T. Yalçın,

"PRINCE – A low-latency block cipher for pervasive computing applications," in *Proceedings of ASIACRYPT*, Vol. 7658, 2012, pp. 208-225.
6. P. Baretto and V. Rijmen, "The KHAZAD legacy-level block cipher," Submission to NESSIE project, http://www.cosic.esat.kuleuven.ac.be/nessie/.
7. J. H. van Tilborg, *Encyclopedia of Cryptography and Security*, Springer, Heidelberg, 2005.
8. J. Jean, I. Nikolić, T. Peyrin, L. Wang, and S. Wu, "Security analysis of PRINCE," in *Proceedings of the 22nd International Workshop on Fast Software Encryption*, Vol. 8424, 2013, pp. 92-111.
9. H. Soleimany, C. Blondeau, X. Yu, W. Wu, K. Nyberg, H. Zhang, L. Zhang, and Y. Wang, "Reflection cryptanalysis of PRINCE-like ciphers," in *Proceedings of the 22nd International Workshop on Fast Software Encryption*, Vol. 8424, 2013, pp. 71-91.
10. F. Abed, E. List, and S. Lucks, "On the security of the core of PRINCE against Biclique and differential cryptanalysis," *Cryptology ePrint Archive*, Report 2012/712, http://eprint. iacr.org/.
11. P. Derbez and L. Perrin, "Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE," in *Proceedings of the 24th International Workshop on Fast Software Encryption*, Vol. 9054, 2015, pp. 190-216.
12. L. Li, K. Jia, and X. Wang, "Improved meet-in-the-middle attacks on AES-192 and PRINCE," *IACR Cryptology ePrint Archive*, Report /573, 2013.
13. S. Rasoolzadeh and H. Raddum, "Cryptanalysis of PRINCE with minimal data," in *Proceedings of International Conference on Cryptology in Africa*, Vol. 9646, 2016, pp. 109-126.
14. A. Canteaut, M. Naya-Plasencia, and B. Vayssière, "Sieve-in-the-middle: improved MITM attacks," in *Proceedings of CRYPTO*, Vol. 8042, 2013, pp. 222-240.
15. A. Canteaut, T. Fuhr, H. Gilbert, M. Naya-Plasencia, and J.-R. Reinhard, "Multiple differential cryptanalysis of round-reduced PRINCE," in *Proceedings of the 23rd International Workshop on Fast Software Encryption*, Vol. 8540, 2014, pp. 591-610.
16. E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," *Lecture Notes in Computer Science*, Vol. 1592, 1999, pp. 12-23.
17. L. R. Knudsen, "DEAL – A 128-bit block cipher," Technical Report 151, Department of Informatics, University of Bergen, Norway, February 1998, Submitted as an AES candidate by Richard Outerbridge.

Yao-Ling Ding (丁瑶玲) was born in 1987. She received her B.S. degree in Computer Science and Technology from Jilin University in 2011. She is currently studying in Tsinghua University. Her research interests include cryptanalysis, as well as side-channel analysis and countermeasures.

Jing-Yuan Zhao (赵静远) gained her B.E. and Ph.D. degrees from Shandong University. She is an Assistant Researcher of State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences. Her research interest is the symmetric cipher.

Lei-Bo Li (李雷波) was born in 1980. He received his Ph.D. degree in Shangdong University in 2014. He currently works in China army. His main research interests include symmetric cipher and hash function.

Hong-Bo Yu (于红波) was born in 1980. She received her Ph.D. degree in Shangdong University in 2007. She currently works in Department of Computer Science and Technology in Tsinghua University. Her main research interest is analysis and design of cryptographic algorithms.