SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2016, Vol. 59 032110:1–032110:14
doi: 10.1007/s11432-015-5295-9

Key recovery attack for PRESENT using slender-set linear cryptanalysis

Guoqiang LIU[1]*, Chenhui JIN[1] & Zhiyin KONG[2]

[1]*Information Engineering University, Zhengzhou* 450000*, China;*
[2]*Science and Technology on Information Assurance Laboratory, Beijing* 100072*, China*

Received October 30, 2015; accepted December 10, 2015; published online January 21, 2016

Abstract In this paper, we propose a new n-round key recovery attack using modified slender-set linear cryptanalysis on PRESENT-like cipher with public S-boxes. In our attack, an effective method for distinguishing the right key from the wrong ones is presented. We apply our attack to PRESENT-80. The experiments show that we can recover the entire 80 key bits of 12-rounds PRESENT-80 with 2^{32} data complexity, 2^{36} time complexity, and negligible memory complexity. Furthermore, we investigate an $(n+1)$-round attack by extending the n-round key recovery attack. Our method can be used in most PRESENT-like ciphers where the linear layer is a bit-wise permutation.

Keywords block cipher, linear cryptanalysis, slender-set, PRESENT cipher, S-box

Citation Liu G Q, Jin C H, Kong Z Y. Key recovery attack for PRESENT using slender-set linear cryptanalysis. Sci China Inf Sci, 2016, 59(3): 032110, doi: 10.1007/s11432-015-5295-9

1 Introduction

Block ciphers are one of the most important symmetric cryptographic algorithms and essential components in many security systems which are widely used. The cipher AES (Advanced Encryption Standard) is suitable for most of the applications. However, the hardware requirement for the AES is considered to be high for extremely constrained devices, such as smart cards, RFID (Radio Frequency IDentification) tags, mobile devices, various types of embedded systems, and IC (Integrated Circuit) printing applications. As a result, quite a few new lightweight ciphers have been proposed to provide strong security at a lower cost than standard solutions.

In recent years, various design strategies for lightweight block cipher have been proposed, such as m-Crypton [1], HIGHT [2], Hummingbird [3], SEA [4], DESL/DESX/DESXL [5], KATAN/KTANTAN [6], MIBS [7], and LED [8]. PRESENT [9] is the most remarkable representative lightweight block cipher. It is proposed by Bogdanov et al. at CHES 2007. In 2012, PRESENT cipher was adopted as ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) lightweight cryptography standard. PRESENT cipher is an iterated 31-rounds SPN (Substitution Permutation Network) block cipher with a 64-bit block size and has two variants of key. One has an 80-bit and the other has a 128-bit. Each round of PRESENT cipher has three layers. The first layer is a substitution layer,

* Corresponding author (email: liuguoqiang87@hotmail.com)

which consists of 16 parallel applications of the same 4-bit S-box. The second layer is a permutation layer, which consists of a bit-wise permutation of 64-bit. The last layer is a key addition layer, which consists of Xor addition by a round key to the text. Due to the small 4-bit S-box, bit-wise permutation, and Xor addition, PRESENT reaches the bound of around 1000 GE in a hardware implementation.

Nowadays, PRESENT cipher has attracted a lot of attention from cryptographic researchers due to its strong security, simplicity, and impressive hardware performance. In 2008, Wang [10] presented a differential cryptanalysis which could attack the 16-rounds PRESENT with 2^{64} data complexity, 2^{32} memory accesses complexity, and 2^{64} time complexity. In 2009, Collard et al. [11] investigated a statistical saturation attack against PRESENT. They can recover 16 key bits of 15-rounds PRESENT cipher with $2^{35.6}$ plaintext–ciphertext pairs in practice. There are other two papers about attacks based on linear hulls for PRESENT in the same year [12,13]. In 2011, Blondeau and Gerard [14,15] presented the multiple differential cryptanalysis on PRESENT. They used 561 differentials (including 17 input differences and 33 output differences) to attack 18-rounds PRESENT with 2^{64} data complexity, 2^{36} memory accesses complexity, and 2^{76} time complexity. In 2012, Blondeau et al. [16] proposed a multiple differential cryptanalysis based on the tools named LLR and χ^2-statistical tests and presented experiments performed on a reduced version of PRESENT. In the same year, Wang et al. [17] focused on the use of so-called structures in differential attack on PRESENT. They gave a general model and complexity analysis for structure attacks. Also, they demonstrated structure attacks for 18-rounds PRESENT-80 with 2^{64} data complexity and 2^{76} time complexity. At CT-RSA 2010, Cho [18] proposed a linear attack on 26 of the 31-rounds PRESENT cipher. This is the best known cryptanalysis attack on PRESENT cipher up to now. Cho could break the 26-rounds PRESENT with 2^{64} data complexity, 2^{32} memory accesses complexity, and 2^{72} time complexity. In 2014, Liu et al. [19] proposed a new method of recovering the secret key of PRESENT-like cipher with public S-box using a variant of the slender-set differential cryptanalysis.

Linear cryptanalysis [20, 21] is a well-known cryptanalytic technique for analyzing block ciphers. It was introduced by Matsui in 1993. An important fact about linear cryptanalysis is that it is a known plaintext attack, making it a more practical and realistic attack model than an attack based on differential cryptanalysis which requires an attack to choose plaintexts for a successful attack. In 2011 and 2013, Borghoff et al. [22, 23] introduced a slender-set linear cryptanalysis on PRESENT-like ciphers with key-dependent secret S-boxes. The work in [23] can recover the secret S-box by looking at Fourier transform for a group of output masks and every input value for a given S-box. In 2014, Liu et al. [24, 25] proposed an improved slender-set linear cryptanalysis on PRESENT-like cipher with secret S-boxes at FSE 2014. In [26], Xiaorui Sun and Xuejia Lai considered a distinguishing between the random distribution and the key-dependent distribution and used them to determine the right key of IDEA (International Data Encryption Algorithm) cipher. We take our inspiration from the key-dependent attack.

Our contributions. In this paper, we focus on the settings of PRESENT-like cipher where the P-box is a bit-wise permutation and the S-box is public fixed. Our contributions are twofold. First, we present a new n-round key recovery attack using modified slender-set linear cryptanalysis on PRESENT-like cipher with known S-boxes. Our starting point is to study the information leakage between the correct key and the wrong key using a modified slender-set linear attack. An effective distinguisher for making a distinction between the correct key and wrong key is proposed. Also, using the method of divide-and-conquer attacks, we can recover the entire 80 secret key bits of PRESENT-80 with lower time and memory complexity. To the 12-rounds PRESENT-80, the experiments show that we can break 12-rounds PRESENT-80 with 2^{32} data complexity, 2^{36} time complexity, and negligible memory complexity. If our attack uses the full plaintexts, which is 2^{64}, our experimental result deduces that at most 25-rounds PRESENT-80 can be broke with 2^{64} data complexity, 2^{68} time complexity, and negligible memory complexity. Furthermore, we propose an $(n + 1)$-round attack by extending the n-round key recovery attack. In this attack, we use the linear approximations used in n-round key recovery attack. For PRESENT cipher, compared with the n-round attack, the data complexity of $(n + 1)$-round attack is about 3.8662 times, and the time complexity has no increase for recovering the full 80 key bits at a success probability of 90%. We summarize our attacks and previous attacks in Table 1.

The paper is organized as follows. Section 2 outlines the slender-set linear attack on PRESENT-like

Table 1 Selected results of attacks on PRESENT cipher (CP: chosen plaintext; KP: known plaintext)

Round	Attack type	Data	Time	Ref.
16	Differential cryptanalysis	2^{64} CP	2^{64}	Ref. [10]
18	Multiple differential cryptanalysis	2^{64} CP	2^{76}	Refs. [14, 15]
18	Structure attack	2^{64} CP	2^{76}	Ref. [17]
24	Statistical saturation attack	2^{57} CP	2^{57}	Ref. [11]
24	Weak keys attack	$2^{63.5}$ KP	2^{40}	Ref. [13]
25	Linear (hull) cryptanalysis	2^{64} KP	$2^{96.68}$	Ref. [12]
26	Linear cryptanalysis	2^{64} KP	2^{72}	Ref. [18]
25	Modified slender-set linear attack	2^{64} KP	2^{68}	This paper[1)]

[1)] Our result deduces from low-round experimental results.

cipher described in [23]. Section 3 presents an n-round key recovery attack based on a modified slender-set linear cryptanalysis and gives experimental results for our attack on PRESENT-80. In Section 4, we outline an $(n+1)$-round key recovery attack by extending the n-round attack and discuss the complexity of attack. Finally, Section 5 concludes the paper.

2 Preliminaries

In this section, we review the Borghoff's slender-set linear attack of recovering the secret S-boxes described in [23].

First, we introduce some basic notations used in this paper. We follow the notations used in [23]. Let $a, b \in F_2^n$ and $a = (a_0, a_1, \ldots, a_{n-1})$, $b = (b_0, b_1, \ldots, b_{n-1})$. The canonical inner product of a, b on F_2^n is denoted by $\langle a, b\rangle$, that is,

$$\langle (a_0, a_1, \ldots, a_{n-1}), (b_0, b_1, \ldots, b_{n-1})\rangle = \overset{n-1}{\underset{i=0}{\oplus}} a_i b_i.$$

For a function $H : F_2^n \to F_2^m$, the Walsh or Fourier transform of H at the pair $(\alpha, \beta) \in F_2^n \times F_2^m$ is defined by

$$\hat{H}(\alpha, \beta) = \sum_{x \in F_2^n} (-1)^{\langle \beta, H(x)\rangle + \langle \alpha, x\rangle}.$$

Next, we introduce the most important equation in Borghoff's slender-set linear attack. Without loss of generality, we consider the leftmost S-box S. Assuming that the encryption function F which starts after the first layer of S-boxes is denoted as

$$F : F_2^4 \times F_2^{60} \to F_2^{64},$$

and the corresponding function with a fixed x is denoted by

$$T_x : F_2^{60} \to F_2^{64} \text{ and } T_x(y) = F(x, y).$$

The whole encryption function of PRESENT-like cipher is denoted by E as

$$E : F_2^4 \times F_2^{60} \to F_2^{64},$$

and the corresponding function with a fixed x is denoted by T'_x as

$$T'_x : F_2^{60} \to F_2^{64} \text{ and } T'_x(y) = E(x, y),$$

then we have

$$\hat{T'_x}(0, \beta) = \hat{T}_{S(x)}(0, \beta) = 2^{-4} \sum_{a \in F_2^4} (-1)^{\langle a, S(x)\rangle} \hat{F}((a, 0), \beta) \approx 2^{-4}(-1)^{\langle \alpha, S(x)\rangle} \hat{F}((\alpha, 0), \beta). \qquad (1)$$

Since the P-box in PRESENT-like cipher is a bit-wise permutation, the mask with low weight after the first layer of S-boxes should cause less linear active S-boxes through the whole cipher. In other words, the linear approximations with low-weight masks will have the larger bias. Accordingly, it is reasonable to assume that α is of weight one. According to (1), for a fixed input x and a given output mask β, we can estimate the value of $\langle \alpha, S(x)\rangle$ depending on the sign of counter $\hat{T}'_x(0,\beta)$, which can be easily done after encrypting enough plaintexts. By this means, we can partition x into two sets which are equally sized as V_0 and V_1 for a given output mask β, where $V_\gamma = \{x|\langle \alpha, S(x)\rangle = \gamma\},\ \gamma = 0, 1$. A correct partition of the set V corresponds to one coordinate function of secret S-box S. If we get all four linearly independent coordinate functions of secret S-box, such as $(\langle 2^i, S(0)\rangle, \langle 2^i, S(1)\rangle, \ldots, \langle 2^i, S(15)\rangle),\ 0 \leqslant i \leqslant 3$, we can recover the secret S-box. We summarize the main steps of Borghoff's linear attack as the following and for more details, we refer to [23]:

Step 1. Let the output mask $\beta = 0^{4j}\|b\|0^{60-4j},\ 0 \leqslant j \leqslant 15,\ 1 \leqslant b \leqslant 15$. For every leftmost input $0 \leqslant x \leqslant 15$, we estimate the value of the counter $\hat{T}'_x(0,\beta)$ by (1) after encrypting enough plaintexts. Then, we construct the vectors $W_\beta = (\hat{T}'_0(0,\beta), \hat{T}'_1(0,\beta), \ldots, \hat{T}'_{15}(0,\beta))$ for each output mask β.

Step 2. We transform the three longest vectors (using the Euclidean norm) into a binary vector, where the coordinates with eight highest counter values are set to '1' and the coordinates with eight lowest counter values are set to '0'.

Step 3. We obtain the coordinate functions of secret S-box using a majority vote among these three binary vectors.

Step 4. We recover the 4-bit secret S-boxes based on four linearly independent coordinate functions of secret S-boxes.

Borghoff et al. pointed out that they might get one or more sets with the form of $\{x|\langle 2^i, S(x)\rangle = 0\},\ 0 \leqslant i \leqslant 3$ by repeating the steps described above for other value of $\beta = 0^{4j}\|b\|0^{60-4j}$ with different $0 \leqslant j \leqslant 15,\ 1 \leqslant b \leqslant 15$. Based on Borghoff's slender-set linear attack, Liu et al. [24, 25] investigated an improved slender-set linear cryptanalysis of recovering the secret S-box uniquely at FSE 2014.

3 *n*-round key recovery attack

In this section, we explain the approach of our new n-round key recovery attack using a modified slender-set linear cryptanalysis on PRESENT-like cipher with public fixed S-boxes.

Notation. Throughout this section, assuming that α is a binary vector, the Hamming weight of α is denoted by $\mathrm{wt}(\alpha)$. The complementary vector of α is denoted by $\overline{\alpha}$.

We measure the data complexity in units which is equivalent to a known plaintext. We measure the time complexity of our attack in units which is equivalent to simple operation.

According to the attack of recovering the secret S-box using slender-set linear cryptanalysis, one naive method of recovering the round key is that we can combine the public S-box S and round key Xor addition as a new key-dependent and unknown "secret S-box" S'. First, we recover the 'secret S-box' S' completely by the slender-set linear attack. According to $S'(x) = S(x \oplus k)$, we can determine the round key k uniquely. The main steps of this method can be described as Algorithm 1.

However, to recover the secret round key, there is no need to recover the 'secret S-box' S' completely, which requires higher data complexity. In the following section, we present a new method of recovering the secret key using a modified slender-set linear cryptanalysis.

3.1 Principle of our attack

Let S-box $S(x) : \{0,1\}^m \rightarrow \{0,1\}^m$ be a bijective mapping. We have the following proposition described in Theorem 1.

Theorem 1. Let S-box $S(x) : \{0,1\}^m \rightarrow \{0,1\}^m$ be a bijective mapping and $\alpha_i, \alpha_j \in \{0,1\}^m$. Given two vectors

$$V_{\alpha_i} = (\langle \alpha_i, S(0)\rangle, \langle \alpha_i, S(1)\rangle, \ldots, \langle \alpha_i, S(2^m-1)\rangle),$$

Algorithm 1 n-round key recovery attack: method 1

```
Require: The public S-box $S$;
    The “secret S-box” $S'$ recovered by slender-set linear attack;
Ensure: The candidate key $K$;
n = 0;
for $k = 0$ to $2^m - 1$ do
    for $x = 0$ to $2^m - 1$ do
        if $S'(x) = S(x \oplus k)$ then
            $n \Leftarrow n + 1$;
            Continue;
        else
            Break;
        end if
    end for
    if $n = 2^m$ then
        $K \Leftarrow k$;
        Break;
    end if
end for
return The value of $K$.
```

Table 2 The probability distribution of $\mathrm{wt}(\overline{V \oplus U}) = k$ for the cipher PRESENT, where $V, U \in \{V_1, V_2, \ldots, V_{15}, \bar{V}_1, \bar{V}_2, \ldots, \bar{V}_{15}\}$

k	Probability (%)	k	Probability (%)	k	Probability (%)
0	3.3333	6	0	12	0
2	0	8	93.3333	14	0
4	0	10	0	16	3.3333

Table 3 The probability distribution of $\mathrm{wt}(\overline{V \oplus U}) = k$ for randomly chosen vectors V, U

k	Probability (%)	k	Probability (%)	k	Probability (%)	k	Probability (%)
2	0.1243	6	18.2751	10	30.4584	14	0.8702
4	3.0458	8	38.0730	12	9.1375	16	0.0155

$$V_{\alpha_j} = (\langle \alpha_j, S(0) \rangle, \langle \alpha_j, S(1) \rangle, \ldots, \langle \alpha_j, S(2^m - 1) \rangle),$$

it holds that

$$\mathrm{wt}(\overline{V_{\alpha_i} \oplus V_{\alpha_j}}) = \begin{cases} 2^m, & i = j, \\ 2^{m-1}, & i \neq j, \end{cases}$$

and

$$\mathrm{wt}(\overline{V_{\alpha_i} \oplus \bar{V}_{\alpha_j}}) = \begin{cases} 0, & i = j, \\ 2^{m-1}, & i \neq j. \end{cases}$$

With the notation in Theorem 1, for the cipher PRESENT, $m = 4$. We can compute the probability distribution of $\mathrm{wt}(\overline{V \oplus U})$ (see Table 2), where V, $U \in \{V_1, V_2, \ldots, V_{15}, \bar{V}_1, \bar{V}_2, \ldots, \bar{V}_{15}\}$, $V_\alpha = (\langle \alpha, S(0) \rangle, \langle \alpha, S(1) \rangle, \ldots, \langle \alpha, S(15) \rangle)$, $1 \leqslant \alpha \leqslant 15$.

Let V, U be randomly chosen vectors with $\mathrm{wt}(V) = \mathrm{wt}(U) = 8$. We can compute the probability distribution of $\mathrm{wt}(\overline{V \oplus U})$ by Lemma 1. A numerical calculation of Lemma 1 for various values of k is given in Table 3.

Lemma 1 ([24, 25])**.** With the notation above, let α, β be random vectors and $\mathrm{wt}(\alpha) = \mathrm{wt}(\beta) = 8$ with $\alpha = (0, a_1, a_2, \ldots, a_{15})$, $\beta = (0, b_1, b_2, \ldots, b_{15})$, $a_i, b_i \in \{0, 1\}$, $1 \leqslant i \leqslant 15$. Then, the probability of $\mathrm{wt}(\overline{\alpha \oplus \beta}) = k$ is equal to

$$p(\mathrm{wt}(\overline{\alpha \oplus \beta}) = k) = \frac{C_8^{(16-k)/2} C_7^{(16-k)/2}}{C_{15}^7},$$

where $k = 2, 4, 6, 8, 10, 12, 14, 16$.

According to the probability distribution shown in Tables 2 and 3, we present a distinguisher for determining whether the vector belongs to the set $\{V_1, V_2, \ldots, V_{15}, \bar{V}_1, \bar{V}_2, \ldots, \bar{V}_{15}\}$ or belongs to the set consisting of randomly chosen vectors. For a randomly chosen vector V with $\mathrm{wt}(V) = 8$, the probability distribution of $\mathrm{wt}(\overline{V \oplus V_\alpha}) = k$ will be similar to the distribution described in Table 3. For a vector $V \in \{V_1, V_2, \ldots, V_{15}, \bar{V}_1, \bar{V}_2, \ldots, \bar{V}_{15}\}$, the probability distribution of $\mathrm{wt}(\overline{V \oplus V_\alpha}) = k$ will be close to the distribution described in Table 2. This is the information leakage for recovering the round key with public S-box in our attack.

For every $0 \leqslant j \leqslant 15,\ 1 \leqslant b \leqslant 15$, we consider the output masks $\beta = 0^{4j}\|b\|0^{60-4j}$. Assuming that we get $15 \times 16 = 240$ vectors, $W_\beta = (\hat{T}'_0(0,\beta), \hat{T}'_1(0,\beta), \ldots, \hat{T}'_{15}(0,\beta))$ after encrypting enough plaintexts with the same secret key. In this paper, we transform the 240 vector W_β into the binary vectors using the method derived in Subsection 4.1 of [24, 25].

First, for every β and every $0 \leqslant i \leqslant 15$, we compute

$$\mathbb{R}_\beta^{(i)} = \hat{T}'_0(0,\beta) - \hat{T}'_i(0,\beta),$$

and we transform each of these vectors $(\mathbb{R}_\beta^{(0)}, \mathbb{R}_\beta^{(1)}, \ldots, \mathbb{R}_\beta^{(15)})$ into binary vectors $(\mathbb{B}_\beta^{(0)}, \mathbb{B}_\beta^{(1)}, \ldots, \mathbb{B}_\beta^{(15)})$, where the coordinates with eight highest counter values are set to '1' and the coordinates with eight lowest counter values are set to '0'. If $\mathbb{B}_\beta^{(0)}$ is equal to '1', then we transform the vector $(\mathbb{B}_\beta^{(0)}, \mathbb{B}_\beta^{(1)}, \ldots, \mathbb{B}_\beta^{(15)})$ into the complementary vector $B_i = (\overline{\mathbb{B}_\beta^{(0)}}, \overline{\mathbb{B}_\beta^{(1)}}, \ldots, \overline{\mathbb{B}_\beta^{(15)}})$. If $\mathbb{B}_\beta^{(0)}$ is equal to '0', then we let $B_i = (\mathbb{B}_\beta^{(0)}, \mathbb{B}_\beta^{(1)}, \ldots, \mathbb{B}_\beta^{(15)})$. These 240 binary vectors contain the reliable information about the coordinate functions of "secret S-box" S', that is, $(\langle 2^i, S'(0)\rangle, \langle 2^i, S'(1)\rangle, \ldots, \langle 2^i, S'(15)\rangle),\ 0 \leqslant i \leqslant 3$. We start with the method of partitioning 240 binary vectors B_i into four parts and for more details, we refer to [24, 25].

We define the distances between two binary vectors B_i and B_j by

$$\mathbb{D}_{B_i,B_j} = \mathrm{wt}(\overline{B_i \oplus B_j}),$$

where $1 \leqslant i, j \leqslant 240$ and $\mathrm{wt}(\overline{B_i \oplus B_j})$ is the *Hamming weight* of $\overline{B_i \oplus B_j}$.

Definition 1 ([24, 25]). Given 240 binary vectors $B_i,\ 1 \leqslant i \leqslant 240$. Let $\xi, \tau > 0$ and $\mathbb{D}_{B_i,B_j} = \mathrm{wt}(\overline{B_i \oplus B_j})$ be the Hamming distances between binary vectors B_i and B_j, where $1 \leqslant j \leqslant 240$. The *similarity degree* of B_i and B_j is defined by

$$\mathbb{S}_{B_i,B_j} = g(B_i, B_j) + \sum_{\substack{1 \leqslant k \leqslant 240 \\ k \neq i, k \neq j}} (f(B_i, B_k) + f(B_j, B_k)),$$

where the function

$$g(B_i, B_j) = \begin{cases} \xi, & \text{if } \mathrm{wt}(\overline{B_i \oplus B_j}) \geqslant t, \\ 0, & \text{others.} \end{cases}$$

and

$$f(B_i, B_j) = \begin{cases} \tau, & \text{if } \mathrm{wt}(\overline{B_i \oplus B_j}) \geqslant t, \\ 0, & \text{others.} \end{cases}$$

According to Definition 1, the higher the $\mathbb{S}_{B_i,B_j}$, the higher the possibility for two vectors B_i and B_j in the same partition. For two random binary vectors α, β with $\mathrm{wt}(\alpha) = \mathrm{wt}(\beta) = 8$, the probability of $\mathrm{wt}(\overline{\alpha \oplus \beta}) = k$ is derived in Lemma 1 by

$$p(\mathrm{wt}(\overline{\alpha \oplus \beta}) = k) = \frac{C_8^{(16-k)/2} C_7^{(16-k)/2}}{C_{15}^7},$$

where $k = 2, 4, 6, 8, 10, 12, 14, 16$. According to Table 3, one can see that the probability of $\mathrm{wt}(\overline{B_i \oplus B_j}) = 16$ is equal to 0.0155%. Such a small probability means that: If we have $\mathrm{wt}(\overline{B_i \oplus B_j}) = 16$, there must be a very strong correlation between two binary vectors B_i and B_j. That is, the vectors B_i and B_j should

Algorithm 2 Partitioning 240 binary vectors into four parts

```
Require:
    The 240 binary vectors $B_i,\ 1 \leqslant i \leqslant 240$;
    The value of $t, \xi, \tau$;
Ensure: Four partitions $\Phi_1, \Phi_2, \Phi_3, \Phi_4$;
According to the value of $t, \xi, \tau$, for every $1 \leqslant i, j \leqslant 240$, we compute the similarity degrees $\mathbb{S}_{B_i,B_j}$;
Construct $r$ priority sets $\Omega_k,\ 1 \leqslant k \leqslant r$;
Choose binary vector $w \in \Omega_k$, where the value of $|\Omega_k|$ is maximal. We mark $w$ and sort 240 binary vectors in descending
    order up to the value of $\mathbb{S}_{w,B_j}$, $1 \leqslant j \leqslant 240$;
$l = 1$;
while $l \leqslant 4$ do
    $n = 1$;
    for $j = 1$ to 240 do
        if $B_j$ is unmarked then
            Add $B_j$ into the set $\Phi_l$ and $n \Leftarrow n + 1$;
        else if $n \leqslant 60$ then
            Continue;
        else
            Break;
        end if
    end for
end while
return Four partitions $\Phi_1, \Phi_2, \Phi_3, \Phi_4$.
```

be in the same partition. Therefore, we treat the binary vectors which hold $\mathbb{D}_{B_i,B_j} = 16$ as the priority vectors. The method of partitioning 240 binary vectors can be described as Algorithm 2.

For every candidate key $k \in \{0,1\}^4$ and $\alpha \in \{0,1\}^4 \backslash \{0\}$, we compute all the vectors $V_\alpha^{(k)} = (\langle \alpha, S(0 \oplus k)\rangle, \langle \alpha, S(1 \oplus k)\rangle, \ldots, \langle \alpha, S(15 \oplus k)\rangle)$ and their complementary vectors $\bar{V}_\alpha^{(k)}$, where the $S(x)$ is known. We propose Assumption 1 for constructing the distinguisher in our attack. To keep it simple, we denote the distribution described in Table 2 as D_1 and the distribution described in Table 3 as D_0.

Assumption 1. Let $\alpha \in \{0,1\}^4 \backslash \{0\}$, $k \in \{0,1\}^4$. Let $S(x)$ be a public S-box and the binary vectors B_i. Given sets $\{V_1^{(k)}, V_2^{(k)}, \ldots, V_{15}^{(k)}, \bar{V}_1^{(k)}, \bar{V}_2^{(k)}, \ldots, \bar{V}_{15}^{(k)}\}$, where $V_\alpha^{(k)} = (\langle \alpha, S(0 \oplus k)\rangle, \langle \alpha, S(1 \oplus k)\rangle, \ldots, \langle \alpha, S(15 \oplus k)\rangle)$. The probability distribution of $\mathrm{wt}(B_i \oplus V_\alpha^{(k)})$ and $\mathrm{wt}(B_i \oplus \bar{V}_\alpha^{(k)})$ is similar to distribution D_0 for each incorrect candidate k. The distribution of $\mathrm{wt}(B_i \oplus V_\alpha^{(k)})$ and $\mathrm{wt}(B_i \oplus \bar{V}_\alpha^{(k)})$ is similar to distribution D_1 for a correct k, respectively.

We propose a new method for distinguishing the correct key from the incorrect key based on Assumption 1. Without loss of generality, we consider the vectors in the first partition Φ_1. Assuming that $\xi_i \in \Phi_1, 1 \leqslant i \leqslant 60$, we exhaust $k \in \{0,1\}^4$ and compute the set $\{V_1^{(k)}, V_2^{(k)}, \ldots, V_{15}^{(k)}, \bar{V}_1^{(k)}, \bar{V}_2^{(k)}, \ldots, \bar{V}_{15}^{(k)}\}$. If k is the correct key, the distribution of $\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)})$ and $\mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})$ should be similar to the distribution D_1. In other words, the values of $\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)})$ and $\mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})$ are approximately equal to '8'. The values of $\max_{1 \leqslant \alpha \leqslant 15}\{\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)}), \mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})\}$ and $\min_{1 \leqslant \alpha \leqslant 15}\{\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)}), \mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})\}$ are approximately equal to '16' and '0'. Respectively, if k is a wrong key, the distribution of $\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)})$ and $\mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})$ is different from the distribution D_1.

χ^2-method has already proved out to be useful, particularly in distinguisher of two distributions. We propose a method of computing the distance between the distribution $\mathrm{wt}(\xi_i \oplus V_\alpha^{(k)})$, $\mathrm{wt}(\xi_i \oplus \bar{V}_\alpha^{(k)})$ and the distribution D_1 using χ^2-statistics method by

$$D_k = \sum_{i=1}^{4} \sum_{\xi_j \in \Phi_i} \left((M_j - 16)^2 + N_j^2 + \sum_{\substack{\alpha=1, V_\alpha^{(k)} \neq M_j \\ V_\alpha^{(k)} \neq N_j}}^{15} \left((\mathrm{wt}(\xi_j \oplus V_\alpha^{(k)}) - 8)^2 + (\mathrm{wt}(\xi_j \oplus \bar{V}_\alpha^{(k)}) - 8)^2 \right) \right), \quad (2)$$

where

$$M_j = \max_{1 \leqslant \alpha \leqslant 15} \left\{ \mathrm{wt}(\xi_j \oplus V_\alpha^{(k)}), \mathrm{wt}(\xi_j \oplus \bar{V}_\alpha^{(k)}) \right\}, \quad N_j = \min_{1 \leqslant \alpha \leqslant 15} \left\{ \mathrm{wt}(\xi_j \oplus V_\alpha^{(k)}), \mathrm{wt}(\xi_j \oplus \bar{V}_\alpha^{(k)}) \right\}.$$

Table 4 The value of D_k with 4-bit round key being $(0111)_2$

Candidate k	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$
Value of D_k	18480	18592	18720	18584	18456	19360	18592	17728
Candidate k	$k=8$	$k=9$	$k=10$	$k=11$	$k=12$	$k=13$	$k=14$	$k=15$
Value of D_k	19224	19088	19360	19227	18744	18608	18448	20432

According to Assumption 1, one can see that the distribution with lowest distance D_k should correspond to the correct key. Assuming that S-box is 4-bit to 4-bit, our attack can be described as Algorithm 3. We initialize the upper bound value of D_k as $m = 2^{32}$ in Algorithm 3.

Algorithm 3 n-round key recovery attack: method 2

```
Require: The public S-box $S(x)$;
    Four partitions $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ based on the text pairs after $n$-round encryptions;
Ensure: The candidate key $K$;
$m = 2^{32}$;
Compute the set $\{V_1^{(k)}, V_2^{(k)}, \dots, V_{15}^{(k)}, \bar{V}_1^{(k)}, \bar{V}_2^{(k)}, \dots, \bar{V}_{15}^{(k)}\}$;
for $k = 0$ to 15 do
    Compute the distance $D_k$;
    if $D_k \leqslant m$ then
        $m \Leftarrow D_k$;
        $K \Leftarrow k$;
    end if
end for
return The value of $K$.
```

3.2 Application to PRESENT-80

In this section, we apply the key recovery attack on the cipher PRESENT using the 80-bit key. PRESENT is an 64-bit SPN block cipher. The round function consists of round key, S-boxes, and permutations. The number of rounds is 31. The pseudo-code of PRESENT cipher is shown in Algorithm 4. For further details, the reader is referred to [9].

(1) Round key K: 64-bit round key is Xored to the text.

(2) S-box S: 16 parallel 4-bit S-boxes.

(3) P-box P: a fixed bit permutation.

Algorithm 4 The pseudo-code of PRESENT cipher

```
Require: 64-bit plaintext $X$; main key $K$;
Ensure: 64-bit ciphertext $C = E_K(X)$;
Derive the round keys $K_i$ $(1 \leqslant i \leqslant 32)$ from the main key $K$;
$STATE = X$;
for $i = 1$ to 31 do
    Add round key $K_i$ to $STATE$;
    $STATE = S(STATE)$;
    $STATE = P(STATE)$;
end for
Add round key $K_{32}$ to $STATE$;
return
```

We assume that the 4-bit secret round key of the leftmost S-box is $(0111)_2$. In our experiment, let $\tau = 1,\ \xi = 2,\ t = 10$ (see Definition 1 and Algorithm 2).

First, we get the 240 binary vectors $B_i, 1 \leqslant i \leqslant 240$ after encrypting 2^{32} plaintexts. Then, we obtain four sets $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ using Algorithm 2. For every $0 \leqslant k \leqslant 15$, we compute the set $\{V_1^{(k)}, V_2^{(k)}, \dots, V_{15}^{(k)}, \bar{V}_1^{(k)}, \bar{V}_2^{(k)}, \dots, \bar{V}_{15}^{(k)}\}$. According to the elements $\xi_j \in \Phi_i,\ 1 \leqslant j \leqslant 60,\ 1 \leqslant i \leqslant 4$, it is easy to calculate the distance D_k, which is defined as (2). Then, we can recover the secret 4-bit key using Algorithm 3. The calculated values of $D_k,\ k = 0, 1, \dots, 15$ are shown in Table 4.

Table 5 The complexity recovering the 80 bit key of 6–12-rounds of PRESENT-80

	Round	6	7	8	9	10	11	12
Our	Data complexity	2^{17}	$2^{19.2}$	$2^{21.5}$	$2^{23.9}$	$2^{26.3}$	$2^{28.6}$	2^{32}
	Time complexity	2^{21}	$2^{23.2}$	$2^{25.5}$	$2^{27.9}$	$2^{30.3}$	$2^{32.6}$	2^{36}
	Success probability	88.5%	90.0%	88.0%	91.0%	92%	90.5%	90.5%
In [18]	Data complexity	$2^{14.8}$	$2^{17.5}$	$2^{20.1}$	$2^{22.7}$	$2^{25.3}$	$2^{27.9}$	$2^{30.5}$
	Time complexity	$2^{32.8}$	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}
	Success probability	95%	95%	95%	95%	95%	95%	95%

According to the definition of D_k, the value k with minimum D_k should correspond to the correct key. From Table 4, we can see that the value of D_k with $k = 7$ is minimal, which is equal to 17728. Thus, we can recover the 4-bit round key of the leftmost S-box as $k = (0111)_2$. Using this method, we can determine the round key of 16 S-boxes one by one. The remaining secret key bits $(80 - 64 = 16)$ can be recovered by exhaustive search. Therefore, the time complexity is equal to $16 \times 2^{32} + 2^{16} \approx 2^{36}$. However, the data complexity of recovering 80 bit key using our attack can be reduced. The plaintexts for recovering the leftmost secret key can be used in recovering remaining secret key. That is to say that the plaintext of form $x\|r$ can be treated as the plaintext of form $r_i\|x'\|r_j$. Thus, the data complexity can be approximately reduced to 2^{32} . In our linear attack, the memory complexity is equal to 60 vectors, which is approximately equal to 2^6 and may be negligible.

The best known attack to PRESENT is the linear hull cryptanalysis of 26-rounds PRESENT-80 proposed by Cho [18]. In Cho's attack on 26-rounds PRESENT, they used the 24-rounds linear characteristic holding with the capacity of $2^{-55.38}$ (see Table 1 in [18]). Due to the limitation of the full range of 2^{64} text pairs, Cho can only obtain 8 bits advantage of 32 bits candidate key. The remaining $80 - 32 = 48$ bits key is combined with the $2^{32-8} = 2^{24}$ candidate keys. Hence, the time complexity of Cho's 26-rounds attack is $2^{64} + 2^{48} \cdot 2^{24} \approx 2^{72}$.

For the low-round PRESENT, we perform Cho's attack algorithm which recovers 32 bits of the round key using the $(n - 2)$-round linear characteristic. The computational complexity of Steps 3 and 4 of the attack algorithm is equal to $2^{32} \cdot 2^{32} + 9 \cdot 2^8 \cdot 2^{32} \approx 2^{64}$, which is hardly practical due to the restriction of computational resources. To perform Cho's low-round attack in practice, they targeted to recover the 16 bits of the last round key using $(n - 1)$-round linear characteristic. According to (2) in [18], the full advantage (16 bits) of the attack with the success probability 95% is achieved by the data complexity of $N^{(r+1)} \approx 2^{9.08}/C^{(r)}$, where $C^{(r)}$ is the capacity of r rounds linear characteristic. We know that the capacity of 6–11 rounds linear characteristic, that is, $C^{(6)} = 2^{-8.42}$, $C^{(7)} = 2^{-11.00}$, $C^{(8)} = 2^{-13.61}$, $C^{(9)} = 2^{-16.22}$, $C^{(10)} = 2^{-18.82}$ and $C^{(11)} = 2^{-21.43}$ (see Table 1 in [18]). Thus, the data complexities of 7–12 rounds can be calculated as $N^{(7)} = 2^{17.5}$, $N^{(8)} = 2^{20.08}$, $N^{(9)} = 2^{22.69}$, $N^{(10)} = 2^{25.3}$, $N^{(11)} = 2^{27.9}$, and $N^{(12)} = 2^{30.5}$. The time complexity of Steps 3 and 4 of the attack algorithm requires $N \cdot 2^{16}$ operations. If the data complexity is greater than 2^{16}, this computational complexity can be reduced greatly by removing the repeated computations. Hence, Steps 3 and 4 can be done by $2^{16} \cdot 2^{16}$ operations. The total time complexity can be denoted as $\min\{2^{16}, N\} \cdot 2^{16} + 9 \cdot 2^8 \cdot 2^{16} \approx 2^{32}$. Furthermore, we can recover another $3 \times 16 = 48$ bits of round key by changing the input S-boxes and the output S-boxes. Then, the time complexity of the low-round attack is $4 \cdot (\min\{2^{16}, N\} \cdot 2^{16} + 9 \cdot 2^8 \cdot 2^{16}) + 2^{16}$. For instance, the data complexity of the Cho's 12-rounds attack is $N^{(12)} = 2^{30.5}$, and the time complexity is $4 \cdot (2^{16} \cdot 2^{16} + 9 \cdot 2^8 \cdot 2^{16}) + 2^{16} \approx 2^{34}$ in total. Table 5 compares the complexity in our attack on 6–12-rounds PRESENT-80 with that of [18]. As can be seen, our complexities are very close to that of Cho's work, which is the best known result.

In [18], Cho pointed out that 26-rounds PRESENT-80 can be broke with 2^{64} data complexity, 2^{72} time complexity, and 2^{32} memory complexity. If our attack uses the full plaintexts, which is 2^{64}, our experimental result deduces that at most 25-rounds PRESENT-80 can be broke with 2^{64} data complexity, 2^{68} time complexity, and negligible memory complexity (see Figure 1).

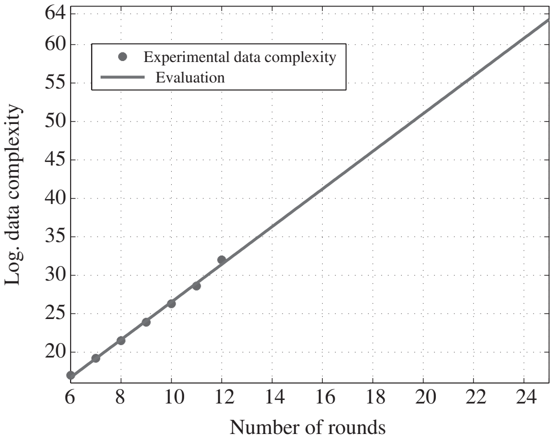

Figure 1 The experimental result and empirical evaluation of linear attack on reduced variants of PRESENT-80.

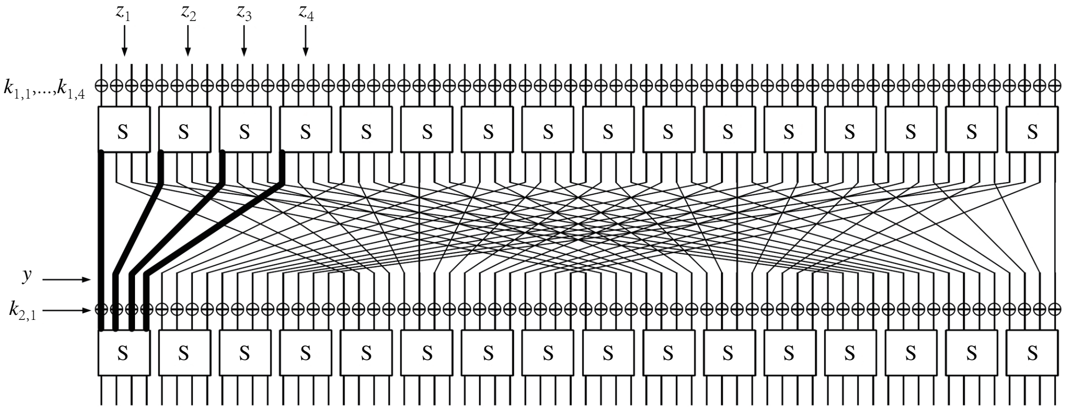

Figure 2 $(n+1)$-round linear attack.

4 $(n+1)$-round key recovery attack

In this section, we explain how to apply a key recovery attack against $(n+1)$-round PRESENT-like cipher with public S-box using slender-set linear cryptanalysis.

Notation. In this section, it is assumed that the jth 4-bit secret key in the ith round is defined by $k_{i,j}$. The symbol & represents the bit-wise AND. In this section, we measure the data complexity in units which is equivalent to a chosen plaintext/chosen ciphertext. We measure the time complexity of our attack in units which is equivalent to simple operation.

4.1 Extensions of the attack

This section describes the extended distinguisher for $(n+1)$-round key recovery attack using slender-set linear cryptanalysis presented in Section 3. For instance, we apply our method on the cipher PRESENT-80.

Next, we outline how to recover the leftmost 16-bit secret key $k_{1,1},\ k_{1,2},\ k_{1,3},\ k_{1,4}$ in the first round and the leftmost 4-bit key $k_{2,1}$ in the second round. We extend the distinguisher described in Section 3 by adding 1-round encryption of the PRESENT cipher at the top, as shown in Figure 2.

According to Figure 2, we can see that

$$y = S_k^{(y)}(z) = ((S(z_1 \oplus k_{1,1})\&8)\|(S(z_2 \oplus k_{1,2})\&8)\|(S(z_3 \oplus k_{1,3})\&8)\|(S(z_4 \oplus k_{1,4})\&8)). \tag{3}$$

In Section 3, we distinguish the correct key from the wrong ones using the information from the partition of each input x of S-box in the first round. Using a similar method, we try to use the information from the partition of each input y of S-box in the second round shown in Figure 2. Being different from the n-round key recovery attack, the partition of the values y refers to more secret round key as $k_{1,1},\ k_{1,2},\ k_{1,3},\ k_{1,4}$. In the following, we outline how to recover the secret key $k_{1,1},\ k_{1,2},\ k_{1,3},\ k_{1,4}$ and $k_{2,1}$.

We define the $(n+1)$-round encryption function as $\mathbb{E}(z,r)$. Let the plaintext be with the form of $z_1\|z_2\|z_3\|z_4\|r_i$ and every $\beta = 0^{4j}\|b\|0^{60-4j}$, $0 \leqslant j \leqslant 15$, $1 \leqslant b \leqslant 15$. To perform the $(n+1)$-round attack, we should estimate the Walsh of the $(n+1)$-round encryption function $\mathbb{E}(z,r)$ with the low-weight output masks β for every fixed input y using the text pairs. That is, we should compute the value of $w_{i,\beta}^{(z)} = (-1)^{\langle\beta,\mathbb{E}(z,r_i)\rangle}$, where each $r_i \in F_2^{48}$ is chosen uniformly at random and $z = z_1\|z_2\|z_3\|z_4$. As naive implementation, we can calculate the value of Walsh directly using the text pairs. However, $(n+1)$-round attack considers each fixed input y in the second round. For randomly chosen plaintexts, the probability of this event is relatively low (about 2^{-12}), so many plaintext pairs are wasted and the data complexity will increase. We can reduce the data complexity using precomputation.

For each $k = k_{1,1}\|k_{1,2}\|k_{1,3}\|k_{1,4}$ by exhaustive search and for every mask $\beta = 0^{4j}\|b\|0^{60-4j}$, $0 \leqslant j \leqslant 15$, $1 \leqslant b \leqslant 15$, we partition $w_{i,\beta}^{(z)}$ into the sets $\Omega_{k,\beta}^{(y)}$ according to the value of y, $0 \leqslant y \leqslant 15$ in (3). For every $0 \leqslant y \leqslant 15$, it holds that $|\Omega_{k,\beta}^{(y)}| \approx N \times 2^{-12}$, where N is the number of plaintexts.

We define the function corresponding to fixing the input $z = z_1\|z_2\|z_3\|z_4$ of $(n+1)$-round encryption function $\mathbb{E}$ as $\mathbb{T}_z$, that is,

$$\mathbb{T}_z : F_2^{48} \to F_2^{64} \ \text{ and } \ \mathbb{T}_z(r) = \mathbb{E}(z,r).$$

For a selection of masks $\beta = 0^{4j}\|b\|0^{60-4j}$, $0 \leqslant j \leqslant 15$, $1 \leqslant b \leqslant 15$, we estimate the counters $\hat{\mathbb{T}}_z(0,\beta) = \sum_{\omega\in\Omega_{k,\beta}^{(y)}} \omega \triangleq F_{k,y}(0,\beta)$. Assuming that we have got the value of $F_{k,y}(0,\beta)$, we may consider the vectors $W_\beta = (F_{k,0}(0,\beta), F_{k,1}(0,\beta), \ldots, F_{k,15}(0,\beta))$ and transform these vectors into binary vectors $B_{k,i}$ using the method described in Section 3. Our $(n+1)$-round attack is based on Assumption 2.

Assumption 2. Assuming that $k = (k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4})$ and $K = (k, k_{2,1})$. Let $\alpha \in \{0,1\}^4, \backslash\{0\}$, $S(x)$ be a public S-box and the binary vectors $B_{k,i}$ are obtained using the known plaintexts with the first 16-bit fixed to $z_1, z_2, \ldots, z_4$. Given sets $\{V_1^{(k_{2,1})}, V_2^{(k_{2,1})}, \ldots, V_{15}^{(k_{2,1})},\ \bar{V}_1^{(k_{2,1})}, \bar{V}_2^{(k_{2,1})}, \ldots, \bar{V}_{15}^{(k_{2,1})}\}$, where $V_\alpha^{(k_{2,1})} = (\langle\alpha, S(0\oplus k_{2,1})\rangle, \langle\alpha, S(1\oplus k_{2,1})\rangle, \ldots, \langle\alpha, S(15\oplus k_{2,1})\rangle)$. The probability distribution of $\mathrm{wt}(B_{k,i}\oplus V_\alpha^{(k_{2,1})})$ and $\mathrm{wt}(B_{k,i}\oplus \bar{V}_\alpha^{(k_{2,1})})$ is similar to distribution D_0 for each wrong candidate K. The distribution of $\mathrm{wt}(B_{k,i}\oplus V_\alpha^{(k_{2,1})})$ and $\mathrm{wt}(B_{k,i}\oplus \bar{V}_\alpha^{(k_{2,1})})$ is similar to the distribution D_1 for a correct K, respectively.

In $(n+1)$-round key recovery attack, we use the similar technique presented in Section 3 for computing the distance between the distribution $\mathrm{wt}(B_{k,i}\oplus V_\alpha^{(k_{2,1})}), \mathrm{wt}(B_{k,i}\oplus \bar{V}_\alpha^{(k_{2,1})})$ and the distribution D_1. If the attack works properly, the distribution with lowest distance should correspond to the correct key. We summarize the main steps of $(n+1)$-round attack as the following.

Step 1. For each plaintext of the form as $z_1\|z_2\|z_3\|z_4\|r_i$ and every $\beta = 0^{4j}\|b\|0^{60-4j}$, $0 \leqslant j \leqslant 15$, $1 \leqslant b \leqslant 15$, we compute the value of $w_{i,\beta}^{(z)} = (-1)^{\langle\beta,E(z,r_i)\rangle}$ by precomputation.

Step 2. For a candidate key $K = (k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4}, k_{2,1})$ and each mask $\beta = 0^{4j}\|b\|0^{60-4j}$, $0 \leqslant j \leqslant 15$, $1 \leqslant b \leqslant 15$, we partition $w_{i,\beta}^{(z)}$ into the sets $\Omega_{k,\beta}^{(y)}$ according to the value of y, $0 \leqslant y \leqslant 15$, which can be calculated by (3).

Step 3. After the partition of $w_{i,\beta}^{(z)}$, we estimate the Walsh $\hat{T}_z(0,\beta)$ using the counters $F_{k,y}(0,\beta) = \sum_{\omega\in\Omega_{k,\beta}^{(y)}} \omega$.

Step 4. We obtain the 240 binary vectors $B_{k,i}$ based on the vectors $W_\beta = (F_{k,0}(0,\beta), F_{k,1}(0,\beta), \ldots, F_{k,15}(0,\beta))$. Then, we partition these 240 binary vectors $B_{k,i}$ into four parts $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ using Algorithm 2.

Step 5. We compute the set $\{V_1^{(k_{2,1})}, V_2^{(k_{2,1})}, \ldots, V_{15}^{(k_{2,1})}\}$ and compute the distance between the distributions D_0 and D_1 as

$$D_K = \sum_{i=1}^{4} \sum_{\xi_j \in \Phi_i} \left((M_j - 16)^2 + N_j^2 + \sum_{\substack{\alpha=1, V_\alpha^{(k_{2,1})} \neq M_j \\ V_\alpha^{(k_{2,1})} \neq N_j}}^{15} \left((\mathrm{wt}(\overline{\xi_j \oplus V_\alpha^{(k_{2,1})}}) - 8)^2 + (\mathrm{wt}(\overline{\xi_j \oplus \bar{V}_\alpha^{(k_{2,1})}}) - 8)^2 \right) \right),$$

where

$$M_j = \max_{1 \leqslant \alpha \leqslant 15} \left\{ \mathrm{wt}(\overline{\xi_j \oplus V_\alpha^{(k_{2,1})}}), \mathrm{wt}(\overline{\xi_j \oplus \bar{V}_\alpha^{(k_{2,1})}}) \right\}, \quad N_j = \min_{1 \leqslant \alpha \leqslant 15} \left\{ \mathrm{wt}(\overline{\xi_j \oplus V_\alpha^{(k_{2,1})}}), \mathrm{wt}(\overline{\xi_j \oplus \bar{V}_\alpha^{(k_{2,1})}}) \right\},$$
$$V_\alpha^{(k_{2,1})} = (\langle \alpha, S(0 \oplus k_{2,1}) \rangle, \langle \alpha, S(1 \oplus k_{2,1}) \rangle, \ldots, \langle \alpha, S(15 \oplus k_{2,1}) \rangle),$$

and the value $k_{2,1}$ is the leftmost candidate key of the second round.

Step 6. The candidate 20-bit key $K = (k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4}, k_{2,1})$ corresponding to the lowest distance D_K is treated as the correct key. Our attack can be described as Algorithm 5.

Algorithm 5 $(n+1)$-round key recovery attack

```
Require:   The public S-box $S(x)$;
    Four partitions $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ based on the text pairs after $(n+1)$-round encryptions;
Ensure:   The candidate key $k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4}$ and $k_{2,1}$;
$m = 2^{32}$;
for $k_{1,1} = 0$ to 15, $k_{1,2} = 0$ to 15, $k_{1,3} = 0$ to 15, $k_{1,4} = 0$ to 15, $k_{2,1} = 0$ to 15 do
    Compute the distance $D_K$;
    if $D_K \leqslant m$ then
        $m \Leftarrow D_K$;
        $(k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4}, k_{2,1}) \Leftarrow K$;
    end if
end for
return  The value of $(k_{1,1}, k_{1,2}, k_{1,3}, k_{1,4}, k_{2,1})$.
```

4.2 Discussion the complexity of attack

This section estimates the data and time complexities of the $(n+1)$-round key recovery attack using slender-set linear cryptanalysis. First, we introduce a useful theorem about the success probability and the data complexity of linear cryptanalysis in [27].

Theorem 2 ([27])**.** Let P_S be the probability that a linear attack on an m-bit subkey, with a linear approximation of probability p, with N known plaintext blocks, delivers an a-bit or higher advantage. Assuming that the linear approximation's probability to hold is independent for each key tried and is equal to 1/2 for all wrong keys, we have for sufficiently large m and N:

$$N = \left(\frac{\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-a-1})}{2} \right)^2 \times |p - 1/2|^{-2}.$$

In the $(n+1)$-round attack, the linear approximations under $(n+1)$-round with fixed input $z_{1,i}, z_{2,i}, \ldots, z_{4,i}$ are same as the linear approximations under n-round with fixed input y. Let $N_{(n)}$ denote the known plaintexts of n-round key recovery attack, and $N_{(n+1)}$ denote the known plaintexts of $(n+1)$-round attack. Next, we examine the data complexity of $(n+1)$-round attack comparing with that of n-round attack.

Note that $(n+1)$-round attack can recover $a_{(n+1)} = 20$ key bits ($k_{1,1}, k_{1,2}, \ldots, k_{1,4}$ and $k_{2,1}$), while n-round attack only recovers $a_{(n)} = 4$ key bits ($k_{1,1}$). According to Theorem 2, we can calculate the data complexity $N_{(n+1)}$ and $N_{(n)}$. Then, the ratio $N_{(n+1)}/N_{(n)}$ can easily be deduced as

$$R_{n+1,n} = \frac{N_{(n+1)}}{N_{(n)}} = \left(\frac{\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-20-1})}{\Phi^{-1}(P_S) + \Phi^{-1}(1 - 2^{-4-1})} \right)^2. \tag{4}$$

Table 6 The data complexity ratio of $(n+1)$-round to n-round for various values of success probability of P_S, according to (4)

Success probability	$P_S = 0.95$	$P_S = 0.90$	$P_S = 0.85$	$P_S = 0.8$	$P_S = 0.75$	$P_S = 0.5$
Ratio	3.4827	3.8662	4.1942	4.5091	4.8288	6.9225

A numerical calculation of (4) for various values of success probability of P_S is given in Table 6.

According to Table 6, one can see that the data and time complexities of $(n+1)$-round attack which recovers extra 16 key bits are 3.8662 times comparing with that of n-round attack with success probability 90%. We repeat this method (4 times) until all the 64-bit keys of first round have been recovered. We recover the remaining secret key bits $(80 - 64 = 16)$ by exhaustive key search. Hence, compared with the n-round attack, the data complexity of $(n+1)$-round attack is about 3.8662 times, and the time complexity has no increase for recovering the full 80 key bits compared with that of n-round attack at a success probability of 90%.

However, the discussion about the complexity of $(n+1)$-round attack is a preliminary work which may not be accurate estimation. It is a possible direction of future research.

5 Conclusion

In this paper, we propose a new modified slender-set method to recover the round key to PRESENT-like cipher with public fixed S-boxes with lower time and memory complexity. We present experiments performed on reduced version of PRESENT-80, as detailed in Table 5. Our experiments suggest that 25-rounds PRESENT-80 could be broke with approximately 2^{64} data complexity, 2^{68} time complexity, and negligible memory complexity.

Furthermore, we present an $(n+1)$-round attack by extending the n-round key recovery attack. The theoretical model suggests that the data complexity of $(n+1)$-round attack is about 3.8662 times, and the time complexity is nonincreasing for recovering the full 80 key bits compared with that of n-round attack at a success probability of 90%. We hope our results to be useful in researching the new key recovery attack on PRESENT-like cipher that uses bit-wise permutation.

An interesting open question is to find a more efficient method of distinguisher for making a distinction between the correct key and wrong key. Furthermore, the theoretical model for complexity of recovering the secret round key would be a possible direction of future work.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61272488, 61402523) and Foundation of Science and Technology on Information Assurance Laboratory (Grant No. KJ-13-007).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Lim C, Korkishko T. mCrypton—a lightweight block cipher for security of low-cost RFID tags and sensors. In: Proceedings of 6th International Workshop on Information Security Applications, Jeju Island, 2005. 243–258

2 Hong D, Sung J, Hong S, et al. HIGHT: a new block cipher suitable for low-resource device. In: Proceedings of 8th International Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, 2006. 46–59

3 Engels D, Saarinen M J, Schweitzer P, et al. The hummingbird-2 lightweight authenticated encryption algorithm. In: Proceedings of 7th International Conference on RFID Security and Privacy, Amherst, 2012. 19–31

4 Standaert F X, Piret G, Gershenfeld N, et al. SEA: a scalable encryption algorithm for small embedded applications. In: Proceedings of 7th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications, Tarragona, 2006. 222–236

5 Leander G, Paar C, Poschmann A, et al. New lightweight DES variants. In: Proceedings of 14th International Workshop on Fast Software Encryption, Luxembourg, 2007. 196–210

6 Cannière C, Dunkelman O, Knežević M. KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers. In: Proceedings of 11th International Workshop on Cryptographic Hardware and Embedded Systems, Lausanne, 2009. 272–288

7 Izadi M, Sadeghiyan B, Sadeghian S, et al. MIBS: a new lightweight block cipher. In: Proceedings of 8th International Conference on Cryptology and Network Security, Kanazawa, 2009. 334–348

8 Guo J, Peyrin T, Poschmann A, et al. The LED block cipher. In: Proceedings of 13th International Workshop on Cryptographic Hardware and Embedded Systems, Nara, 2011. 326–341

9 Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, 2007. 450–466

10 Wang M. Differential cryptanalysis of reduced-round PRESENT. In: Proceedings of 1st International Conference on Cryptology in Africa, Casablanca, 2008. 40–49

11 Collard B, Standaert F X. A statistical saturation attack against the block cipher PRESENT. In: Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, 2009. 195–210

12 Nakahara J, Sepehrdad P, Zhang B, et al. Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT. In: Proceedings of 8th International Conference on Cryptology and Network Security, Kanazawa, 2009. 58–75

13 Ohkuma K. Weak keys of reduced-round PRESENT for linear cryptanalysis. In: Proceedings of 16th Annual International Workshop on Selected Areas in Cryptography, Calgary, 2009. 249–265

14 Blondeau C, Gérard B. Multiple differential cryptanalysis: theory and practice. In: Proceedings of 18th International Workshop on Fast Software Encryption, Lyngby, 2011. 35–54

15 Blondeau C, Gérard B. Multiple differential cryptanalysis: theory and practice (corrected). Cryptology ePrint Archive. Report 2011/115, 2011

16 Blondeau C, Gérard B, Nyberg K. Multiple differential cryptanalysis using LLR and χ^2 statistics. In: Proceedings of 8th International Conference on Security and Cryptography for Networks, Amalfi, 2012. 343–360

17 Wang M, Sun Y, Tischhauser E, et al. A model for structure attacks, with applications to PRESENT and Serpent. In: Proceedings of 19th International Workshop on Fast Software Encryption, Washington DC, 2012. 49–68

18 Cho J. Linear cryptanalysis of reduced-round PRESENT. In: Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, 2010. 302–317

19 Liu G Q, Jin C H. Differential cryptanalysis of PRESENT-like cipher. Designs Codes Cryptogr, 2015, 76: 385–408

20 Matsui M. The first experimental cryptanalysis of the data encryption standard. In: Proceedings of 14th Annual International Cryptology Conference on Advances in Cryptology, Santa Barbara, 1994. 1–11

21 Matsui M. Linear cryptanalysis method for DES cipher. In: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, 1994. 386–397

22 Borghoff J, Knudsen L, Leander G, et al. Cryptanalysis of PRESENT-like ciphers with secret S-boxes. In: Proceedings of 18th International Conference on Fast Software Encryption, Lyngby, 2011. 270–289

23 Borghoff J, Knudsen L, Leander G, et al. Slender-set differential cryptanalysis. J Cryptol, 2013, 26: 11–38

24 Liu G Q, Jin C H, Qi C D. Improved slender-set linear cryptanalysis. Cryptology ePrint Archive, Report 2014/100, 2014

25 Liu G Q, Jin C H, Qi C D. Improved slender-set linear cryptanalysis. In: Proceedings of 21st International Workshop on Fast Software Encryption, London, 2014. 431–450

26 Sun X R, Lai X J. The key-dependent attack on block ciphers. In: Proceedings of 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, 2009. 19–36

27 Selçuk A A. On probability of success in linear and differential cryptanalysis. J Cryptol, 2008, 21: 131–147