Improved Side Channel Cube Attacks on PRESENT

XinJie Zhao[1], Tao Wang[1], ShiZe Guo[2,3]

[1](Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003, China)

[2](The Institute of North Electronic Equipment, Beijing 100083, China)

[3](School of Information Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The paper presents several improved side channel cube attacks on PRESENT based on single bit leakage model. Compared with the previous study of Yang et al in CANS 2009 [30], based on the same model of single bit leakage in the 3rd round, we show that: if the PRESENT cipher structure is unknown, for the leakage bit 0, 32-bit key can be recovered within $2^{7.17}$ chosen plaintexts; if the cipher structure is known, for the leakage bit 4,8,12, 48-bit key can be extracted by $2^{11.92}$ chosen plaintexts, which is less than 2^{15} in [30]; then, we extend the single bit leakage model to the 4th round, based on the two level "divide and conquer" analysis strategy, we propose a sliding window side channel cube attack on PRESENT, for the leakage bit 0, about $2^{15.14}$ chosen plaintexts can obtain 60-bit key; in order to obtain more key bits, we propose an iterated side channel cube attack on PRESENT, about $2^{8.15}$ chosen plaintexts can obtain extra 12 equivalent key bits, so overall $2^{15.154}$ chosen plaintexts can reduce the PRESENT-80 key searching space to 2^8; finally, we extend the attack to PRESENT-128, about $2^{15.156}$ chosen plaintexts can extract 85 bits key, and reduce the PRESENT-128 key searching space to 2^{43}. Compared with the previous study of Abdul-Latip et al in ASIACCS 2011 [31] based on the Hamming weight leakage model, which can extract 64-bit key of PRESENT-80/128 by 2^{13} chosen plaintexts, our attacks can extract more key bits, and have certain advantages over [31].

Keywords: Side channel attacks, Cube attack, black box attack, divide and conquer, sliding window; iterated attack, PRESENT-80/128

0 Introduction

PRESENT[1] is an ultra-lightweight block cipher designed by Bogdanov et al. aimed at restricted applications such as RFID tags and sensor networks in CHES 2007. Due to its impressive hardware performance and the strong security, PRESENT has drawn a lot of attentions from the lightweight cryptographic community. On the other hand, the cryptanalysis on PRESENT has been also actively performed so far, it can be mainly divided into three types: mathematics based cryptanalysis [2], leakage based cryptanalysis [3], combining based cryptanalysis [4].

Mathematics based cryptanalysis treats PRESENT cipher implementation as a black box, combing the cipher input, output and design structure, uses differential cryptanalysis[5][6][7], linear cryptanalysis[8][9][10], saturation cryptanalysis[11], related-key cryptanalysis[12], algebraic cryptanalysis[13] etc mathematical methods to deduce the key. Due to the strong security of PRESENT design, such attacks mainly focus on reduced-round variants of PRESENT, thus pose no real threat to its security.

Leakage based cryptanalysis, which is also named as implementation attack or side channel attack. It treats cipher implementation as a gray box, assuming that: besides the plaintexts or ciphetexts, the extra physical information leakage (such as executing timing[14], power consumption[15], electromagnetic emission[16], sound[17], fault[18]) during the cipher implementation can be measured and used to deduce the cipher intermediate states. Since more information is available to the attacker, leakage based attacks are potentially much easier than mathematics based attacks. For now, there are several published papers on power[19] and fault[20][21][22] based attacks on PRESENT. Due to the limitation of its analysis method, current leakage based cryptanalysis mainly focuses on the first or last few rounds of ciphers, it can not make full use of the information leakages in deeper rounds, and different information leakage usually adopt different analysis strategy.

Combining based cryptanalysis bands the above two methods together. It can make full use of the information leakage during cipher implementation, reduce the sample size, and it is not limited by the types of information leakage, and has the potential of becoming the generic distinguisher for ciphers. Typical methods are two

recently emerged algebraic side channel attack[23][24] (ASCA, combined algebraic attack[25] and side channel attack[3]) and side channel cube attack[26] (SCCA, combined cube attack[27][28] and side channel attack[3]). What's more, it can be extended to the more complicated attacking scenarios. The most interesting part of the ASCA method is that it can be applied to unknown plaintext and ciphertext scenarios [24], only one sample is enough to obtain the full bits of the secret key at the extreme cases, and the SCCA method can be applied to unknown cipher inner structure scenarios [29], limited chosen plaintexts are enough to obtain the secret key. This related research is a hot spot of cryptographic research in recent years. Until now, there are limited papers of ASCA [24] and SCCA [30] on PRESENT. This paper mainly focuses on improving the SCCA methods on PRESENT cipher.

In CANS 2009, Yang et al [30] proposed the first side channel cube attack on PRESENT-80. The attack assumes that the PRESENT structure is known and any output bit of the third round is leaked, for any bit leakage of the first S-box lookup 4-bit output in the third PRESENT-80 round, at most 32-bit key can be recovered, and especially for the first bit leakage of the 2^{nd}, 3^{rd}, 4^{th} S-box lookup output in the third round, 2^{15} chosen plaintexts analysis can recover 48-bit of master key, and reduce the master key searching space to 2^{32}.

In ASIACCS 2011, Abdul-Latip et al[31] proposed the an extended side channel cube attack by extracting low degree non-linear equations, based on the Hamming weight leakage model, they applied the attack to PRESENT-80/128, and showed that about 2^{13} chosen plaintexts can recover 64-bit of master key for both PRESENT/80 and PRESENT./128.

In the pioneering side channel cube attack idea proposed by Dinur et al. in [26], they presented many open problems for future research. An important one is how to find the best maxterms of the multivariate polynomial for ciphers, and improve the attack efficiency at the least costs. Motivated by the ideas above, this paper tries to find some more efficient side channel cube attacks on ciphers, and apply it to PRESENT [30]. Be different with Abdul-Latip et al [31] based on the Hamming weight leakage model to improve the attacks by extracting low degree non-linear equations of PRESENT. This paper takes another approach, also based on the same single bit leakage model as [30], we try to find the best maxterms of PRESENT under the 3^{rd} round leakage model to reduce the sample size, and then extend the attack to the 4^{th} round leakage model to extract more key bits, and find out a more efficient attack on PRESENT-128.

This paper presents several improved side channel cube attacks on PRESENT. Table 1 demonstrates the improvements of the attacks in this paper over several previous attacks.

Table 1. Comparison with previous SCCA on PRESENT

PRESENT	Reference	Leakage model	Sample size	Recovered key
PRESENT-80	[30]	3^{rd} round single bit leakage	2^{15}	48-bit
PRESENT-80	[31]	Hamming weight leakage after the 1^{st} round	2^{13}	64-bit
PRESENT-80	Section 4	3^{rd} round single bit leakage	2^{12}	48-bit
PRESENT-80	Section 5,6	4^{th} round single bit leakage	$2^{15.154}$	72-bit
PRESENT-128	[31]	Hamming weight leakage after the 1^{st} round	2^{13}	64-bit
PRESENT-128	Section 7	4^{th} round single bit leakage	$2^{15.156}$	85-bit

The main contributions of this paper are listed as follows:

(1) Based on the same leakage model of [30], assuming that single bit information of the 3^{rd} round S-box lookup output can be leaked, we propose a black box side channel cube attack PRESENT-80. Under the assumption that the internal cipher design of PRESENT is unknown, and the attacker can only observe the 1^{st} bit of the first S-box lookup output in the 3^{rd} PRESENT round, we show that about 32 bits key can be obtained by $2^{7.17}$ chosen plaintexts.

(2) Based on the 3^{rd} round leakage model of [30], we get some new results of side channel cube attack on PRESENT-80. Experiment shows that: when the first bit of the first S-box lookup output is leaked, $2^{7.17}$ chosen plaintexts can obtain 32-bit key; when the 2^{nd}, 3^{rd} or 4^{th} bit of the first S-box lookup output is leaked, $2^{8.59}$ chosen plaintexts can obtain 32-bit key; when the first bit of the 2^{nd}, 3^{rd}, 4^{th} S-box lookup output is leaked, approximately $2^{11.92}$ chosen plaintexts can obtain 48-bit key, which is smaller than 2^{15} in [30]. And we provide concrete cube indexes and key bit related linear equations.

(3) Based on the two level "divide and conquer"

analysis strategy, we propose a new sliding window side channel cube attack on PRESENT, and extend the attack to the 4th round under the single bit leakage model. About $2^{15.14}$ chosen plaintexts analysis can obtain 60-bit PRESENT-80 key.

(4) Based on the 4th round leakage model, combined the obtained 60-bit key, we propose a new iterated side channel cube attack on PRESENT. We iterate the recovered key bits into the first round polynomial, and about $2^{8.15}$ chosen plaintexts can obtain the extra 12 equivalent key bits, so overall $2^{15.154}$ chosen plaintexts can obtain 72-bit key, and reduce the PRESENT-80 master key searching space to 2^8.

(5) Based on the 4th round leakage model, we extend the above attacks to PRESENT-128. About $2^{15.156}$ chosen plaintexts can obtain 85-bit key, and reduce the PRESENT-128 master key searching space to 2^{43}.

Organization of the Paper. This paper is organized as follows: A review on the cube attack and the side channel cube attack is briefly presented in Section 1, and the PRESENT cipher is described in Section 2. Then a black box side channel cube attack on PRESENT-80 is proposed in Section 3, the improved side channel cube attack on PRESENT-80 based on the 3rd round single bit leakage model is presented in Section 4, and the two extended attacks based on the 4th round single bit leakage model are proposed in Section 5 and Section 6, then the extended attack on PRESENT-128 is proposed in Section 7. Finally, the conclusions and future directions are given in section 8.

1 A Review on the Cube Attack and Side Channel Cube Attack

1.1. Cube Attack

Cube attack was announced by Dinur and Shamir in 2008[27], and published at EUROCRYPT 2009[28]. The ideas behind cube attack can be found in several pervious works [32][33]. Cube attack is a generic key-recovery attack that can be applied to cryptosystems under a black-box setting, that is, the internal structure of the target cipher is unknown. It can be used to attack cryptosystem in which the output can be represented as a low-degree decomposition multivariate polynomial by the public variables and the key variables.

As to the m-bit public variables $V = \{v_1, .., v_m\}$ and the n-bit secret key $K = \{k_1, ..., k_n\}$ of a cipher, let $X = V \cup K$, then the 1 bit output of a cryptosystem can be described by a multivariate master polynomial, which is also defined as function f. Suppose I is a subset of V. The output function f can be written as

$$f(X)=f(v_1, .., v_m, k_1, ..., k_n)=t_I \bullet p_{S(I)} + q_I(X)$$

I is called the cube, the index of the subset I is defined as cube index. t_I is the multiple of all variables whose indexes are in I, $pS(I)$ is called the superpoly, q_I contains any and all terms that are not divisible by t_I.

For example, considering a polynomial of degree 3 in 6 variables

$$f(v_1, v_2, v_3, k_1, k_2, k_3) = v_1v_2k_1 + v_1v_2k_3 + v_1v_3k_2 + v_1v_2 + k_1k_2 + v_3 + 1 \quad (1)$$

Let $I = \{1,2\}$ be an index subset of size 2. We can represent f as:

$$f(v_1, v_2, v_3, k_1, k_2, k_3) = v_1v_2(k_1 + k_3+1) + (v_1v_3k_2 + k_1k_2 + v_3 + 1) \quad (2)$$

And $t_I = v_1v_2$, $p_{S(I)} = k_1 + k_3+1$, $q_I = v_1v_3k_2 + k_1k_2 + v_3 + 1$

If the subset of variables in the term t_I are assigned by all the possible 0/1 values, and iterated it into f to compute the polynomial output, then the symbolic sum over $GF(2)$ of all the derived polynomials f is exactly $p_{S(I)}$, which is the superpoly of t_I in $f(X)$. As to the equation above, the result of this summation is the superpoly of t_I, $p_{S(I)} = k_1 + k_3+1$. t_I is named as a maxterm of $f(X)$, a linear polynomial which is not a constant. Applying the above method, a series of linear equations on secret key can be found, and combining the equation solver, the secret key can be recovered.

The cube attack can be divided into two phases: the preprocessing phase and the online phase. In the first phase, the attacker finds maxterms of the master polynomial, the main challenge of the attacker in this phase is how to find many maxterms with linearly independent superpolys sufficiently. The attacker randomly chooses a subset I of public variables and uses efficient linearity tests to check whether its superpoly is linear. In case the subset I is too small, the superpoly is likely to be nonlinear and less superpolys can be found, then the attacker adds a public variable to I and repeats the process. In case I is too large, the sum will be a constant function and the required test timing is quite long, in this case the attacker drops one of the public variables from I and repeats the process. While finding such linear superpolys can be a challenging preprocessing task, once they are found for a particular cryptosystem, we can repeatedly use them to find any secret key easily during the online phase by summing the outputs of the cryptosystem for every possible assignment to the public variables V which correspond to one of its maxterms and solving the resultant system of linear equations to obtain

K.

1.2. Side Channel Cube Attack

With the increase of the rounds, the degree of the multivariate polynomial grows exponentially. It's quite difficult to express and store such huge polynomial, how to find sufficient maxterms in a short time is also a tough problem. For now, standard pure cube attacks are only effective to the reduced round variants of stream cipher and block cipher. With the introduction of the side channel attack, the attacker can obtain either the plaintext/ciphertext or the internal state for any intermediate round of block cipher, which is far more information than standard cube attack. However, as for attacks on block cipher, traditional side channel attacks usually focus on the first and last few rounds of block cipher, many information leakages of the deeper intermediate round are not explored in the cryptanalysis, which limits the power of the side channel attack.

Side channel cube attack[26] combines the cube attack and side channel attack together, besides holding the advantages of the two attacks above, it can use the intermediate round information leakage to satisfy the precondition of the standard cube attack against full rounds cryptosystem and extend the attack to the deeper round of the side channel attack, thus pose real threats to many block ciphers, such as PRESENT[30][31], NOEKEON[35] and KATAN[36].

2 A Brief Description of the PRESENT Block Cipher

PRESENT is a 31-round SPN structure block cipher with block size of 64 bits, the cipher is described in Figure 1. It supports 80 and 128-bit secret key. Firstly, the plaintext Xored subkey K^1 as the input of the 1st round, after 31 rounds iterations, the 31th round output Xored with the subkey K^{32} is the ciphertext.

Encryption procedure:

Each encryption round consists of the following 3 steps:

(1) addRoundKey—*AK*: At the beginning of each round, 64 bits output of the last round function is Xored with the subkey.

(2) sBoxlayer—*SL*. The *SL* function $\{0,1\}^4 \rightarrow \{0,1\}^4$ maps input (x_0, x_1, x_2, x_3) to output (y_0, y_1, y_2, y_3), 16 identical 4-bit to 4-bit S-boxes are used in parallel. The boolean function of S-box is

$$\begin{aligned} y_0 &= x_0 + x_2 + x_3 + x_1x_2 \\ y_1 &= x_1 + x_3 + x_1x_3 + x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 \\ y_2 &= 1 + x_2 + x_3 + x_0x_1 + x_0x_3 + x_1x_3 + x_0x_1x_3 + x_0x_2x_3 \\ y_3 &= 1 + x_0 + x_1 + x_3 + x_1x_2 + x_0x_1x_2 + x_0x_1x_3 + x_0x_2x_3 \end{aligned} \quad (3)$$

(3) pLayer—*PL*: the i^{th} bit is moved to bit position $P(i)$ by a constant permutation table P.

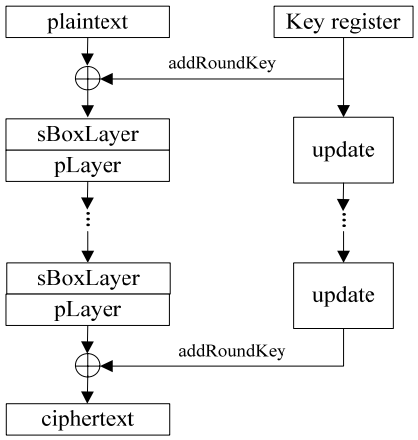

Figure 1. Overview of PRESENT Encryption Algorithm

Key schedule:

PRESENT can take keys of either 80 or 128 bits. Below is the key schedule algorithm of 80-bit version, more information about PRESENT-128 can be found in [1]. The 80-bit key is stored in a register $K=k_{79}||k_{78}||\ldots||k_0$. At round r ($1 \le r \le 31$) the 64-bit round key K^r consists of the 64 leftmost bits of K. After K^r is extracted, K is rotated by 61 bit positions to the left, then S-box is applied to the left-most 4 bits of K and finally the round-counter value r is exclusive-ored with bits $k_{19}||k_{18}||k_{17}||k_{16}||k_{15}$ of K with the least significant bit of round_counter on the right.

Some notations:

Here we introduce some notations in order to make our discussion conveniently.

P_{SI}: cube index set of the plaintext

N_{SI}: numbers of the plaintext cube indexes

S_{SI}: summations of the plaintext cube indexes

K_i: the i^{th} bit of the master key

K_{ME}: superpoly of P_{SI}, key bit related linear equation

I_i^j: the target j^{th} bit of the i^{h} round S-box lookup output in cube attack

N_{kb}: the number of the recovered key bit

+: boolean xor

3 Black box Side channel Cube Attack on PRESENT

As is pointed in [27], [28] and [29], cube attacks can recover a secret key through querying a black box polynomial with tweakable public variables (e.g. chosen

plaintext bits for block ciphers) and solving a linear system of equations on the secret key variables. Inspired by this idea, under the assumption that the internal cipher design is unknown and the attacker can observe its input and only single bit information leakage in the 3rd round S-box lookup output, we apply a black box side channel cube attack on PRESENT-80.

Based on the 3rd round single bit leakage model, we randomly choose m bits (m=1, 2 or 3) of plaintext as the cube indexes, n bits (n=1 or n=2) of PRESENT first round key as the target key bits, and select the bit index of the plaintext, key and ciphertext randomly, then test the linearity of the output secret key equations by Blum-Luby Rubinfeld (BLR) method [37]. So there are ${C_1}^{64}$, ${C_2}^{64}$ and ${C_3}^{64}$ plaintext index combinations for m=1, 2 and m=3, $2\cdot{C_1}^{64}$ (k_i or $1+k_i$) and $2\cdot{C_2}^{64}$ (k_i+k_j or $1+k_i+k_j$) combinations of linear key equations for n=1 and n=2, 64 target ciphertext indexes. If the 1st bit of the 1st S-box look up is leaked in the 3rd round of PRESENT, the attack results are shown in Table 2.

It's clear to see that about 32-bit of the PRESENT-80 key can be recovered, and the PRESENT-80 master key searching space can be reduced to 2^{48}.

Table 2. The 3rd round attack results of ${I_3}^0$

P_{SI}	K_{ME}	P_{SI}	K_{ME}
1	k_{18}	16,34	$1+k_{49}$
2	$1+k_{17}$	16,33	k_{50}
14	$1+k_{29}$	16,46	$1+k_{61}$
13	k_{30}	16,45	k_{62}
49	k_{66}	54,56	$1+k_{69}$
50	$1+k_{65}$	53,56	k_{70}
61	k_{78}	52,58	$1+k_{73}$
62	$1+k_{77}$	52,57	k_{74}
6,8	$1+k_{21}$	22,24,32	$1+k_{37}$
5,8	k_{22}	21,24,32	k_{38}
4,10	$1+k_{25}$	20,26,32	$1+k_{41}$
4,9	k_{26}	20,25,32	k_{42}
18,32	$1+k_{33}$	16,38,40	$1+k_{53}$
17,32	k_{34}	16,37,40	k_{54}
30,32	$1+k_{45}$	16,36,42	$1+k_{57}$
29,32	k_{46}	16,36,41	k_{58}

4 Improved Side Channel Cube Attack on PRESENT

4.1. Complexity Analysis of the Attack Round and Bit Position

In cube attacks on PRESENT, how to choose the attack round r and bit index b is very important. If r is quite small, such as r=1 and r=2, the complexity of chosen plaintexts is minimized, however the number of recovered key bits is very small, and the master key exhaustive searching complexity would be quite high. If r is quite big, such as $r\geq4$, the maxterms will involve more key bits, but both the degree and number of polynomial will grow exponentially, and the attack complexity would be much higher.

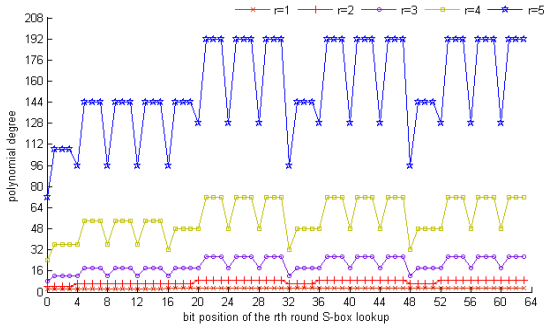

Figure 2. polynomial degree for 64 positions of the r^{th} round S-box lookup output

Figure 2 is the polynomial degree for 64 bit positions of the r^{th} round PRESENT S-box lookup. It's clear to see that with the extending of PRESENT rounds, the degree of the polynomial grows exponentially.

Due to the boolean functions of PRESENT S-box, after the 1st round S-box lookup, each S-box output bit is computed by 4 distinct plaintext bits and 4 distinct key bits, and the permutation layer only changes the position of the state bit without changing its value, after the 2nd round S-box lookup, each S-box output bit is computed by 16 plaintext bits and 16 distinct key bits, and until the 3rd round S-box lookup, each S-box output bit is begin to be computed by full 64 plaintext bits and 64 distinct key bits firstly. So considering the recovered key bits number and polynomial complexity, the 3rd round would be the best choice for the attacker.

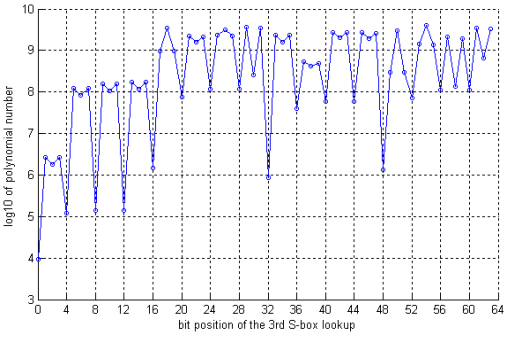

Figure 3. polynomial number for 64-bit positions of the 3rd round S-box lookup

Figure 3 is the sub-polynomial number for 64 bit positions of the 3rd PRESENT round S-box lookup, it's clear to see that different bit positions have different sub-polynomial numbers. According to Figure 2 and

Figure 3, attacking the 1st bit of the S-box 0,1,2,3 would have quite low complexity.

4.2. The Attack Procedure and Results

In order to reduce the scale of polynomial with the increase of rounds, we adopt the same strategy in [30]. For each round, we reserve these terms involving a key variable and the terms only involving public variables, and discard the terms involving more than one key variable.

According to Figure 2 and Figure 3, we can classify the side channel cube attack on the 3rd round of PRESENT-80 into 6 classes; the required sample size and recovered key bit number are shown in Table 3.

Table 3. The 3rd round attack results in this paper

Class	Sample size	N_{kb}
Class 1: I_3^0	$8\cdot2+16\cdot2^2+8\cdot2^3=2^{7.17}$	32
Class 2: I_3^1, I_3^2, I_3^3	$16\cdot2^2+16\cdot2^4\approx2^{8.59}$	32
Class 3: I_3^4, I_3^8, I_3^{12}	$12\cdot2^2+24\cdot2^5+12\cdot2^8\approx2^{11.92}$	48
Class 4: $I_3^{5\text{-}7,9\text{-}11,13\text{-}15}$	$24\cdot2^8+24\cdot2^{11}\approx2^{16}$	48
Class 5: $I_3^{16,20,..60}$	$24\cdot2^8+24\cdot2^{17}\approx2^{22}$	48
Class 6: $I_3^{17\text{-}19,...,61\text{-}63}$	$48\cdot2^{26}=2^{32}\approx2^{32}$	48

Table 4. The 3rd round attack results in [30]

Class	Sample size	N_{kb}
Class 1:I_3^0,I_3^1,I_3^2,I_3^3		32
Class 2: I_3^4, I_3^8, I_3^{12}	$12\cdot(2^2+2^5+2^8+2^{11})\approx2^{15}$	48
Class 3: $I_3^{5\text{-}7,9\text{-}11,13\text{-}15}$	$24\cdot2^8+24\cdot2^{11}\approx2^{16}$	48
Class 4: $I_3^{16,20,..60}$	$24\cdot2^8+24\cdot2^{17}\approx2^{22}$	48
Class 5: $I_3^{17\text{-}19,...,61\text{-}63}$	$48\cdot2^{26}\approx2^{32}$	48

Our classifications and attack results are different with [30] as follows:

(1) Be different from [30] by sorting the attack bits into 5 classes, we sort the bits into 6 classes by different S-box index and S-box output bit position. We discover that the sample size of attacking the 1st bit of S-box 0 (Table 2) are different with the 2nd, 3rd, and 4th bit of S-box 1 (Appendix Table A.1). When the first bit of the first S-box lookup output is leaked, $2^{7.17}$ chosen plaintexts can obtain 32-bit key; when the 2nd, 3rd or 4th bit of the first S-box lookup output is leaked, $2^{8.59}$ chosen plaintexts can obtain 32-bit key.

(2) We show that the least sample size of attacking the 1st bit of S-box 1, 2, 3 is about $2^{11.92}$, which is much smaller than 2^{15} in [30], the results of attacking the 1st bit of S-box 1 is shown in Table 5, the results of attacking the 1st bit of S-box 2 and 3 are shown in Appendix Table A.2 and A.3.

Table 5. The 3rd round attack results of I_3^4

P_{SI}	K_{ME}	P_{SI}	K_{ME}
1,3	$1+k_{16}$	16,17,18,33,35	$1+k_{48}$
0,1	$k_{18}+k_{19}$	16,17,18,32,33	$k_{50}+k_{51}$
0,2	$k_{17}+k_{19}$	16,17,18,32,34	$k_{49}+k_{51}$
13,15	$1+k_{28}$	16,17,18,45,47	$1+k_{60}$
12,13	$k_{30}+k_{31}$	16,17,18,44,45	$k_{62}+k_{63}$
12,14	$k_{29}+k_{31}$	16,17,18,44,46	$k_{61}+k_{63}$
49,51	$1+k_{64}$	53,55,56,57,58	$1+k_{68}$
48,49	$k_{66}+k_{67}$	52,53,56,57,58	$k_{70}+k_{71}$
48,50	$k_{65}+k_{67}$	52,54,56,57,58	$k_{69}+k_{71}$
61,63	$1+k_{76}$	52,53,54,57,59	$1+k_{72}$
60,61	$k_{78}+k_{79}$	52,53,54,56,57	$k_{74}+k_{75}$
60,62	$k_{77}+k_{79}$	52,53,54,56,58	$k_{73}+k_{75}$
5,7,8,9,10	$1+k_{20}$	21,23,24,25,26,32,33,34	$1+k_{36}$
4,5,8,9,10	$k_{22}+k_{23}$	20,21,24,25,26,32,33,34	$k_{38}+k_{39}$
4,6,8,9,10	$k_{21}+k_{23}$	20,22,24,25,26,32,33,34	$k_{37}+k_{39}$
4,5,6,9,11	$1+k_{24}$	20,21,22,25,27,32,33,34	$1+k_{40}$
4,5,6,8,9	$k_{26}+k_{27}$	20,21,22,24,25,32,33,34	$k_{42}+k_{43}$
4,5,6,8,10	$k_{25}+k_{27}$	20,21,22,24,26,32,33,34	$k_{41}+k_{43}$
17,19,32,33,34	$1+k_{32}$	16,17,18,37,39,40,41,42	$1+k_{52}$
16,17,32,33,34	$k_{34}+k_{35}$	16,17,18,36,37,40,41,42	$k_{54}+k_{55}$
16,18,32,33,34	$k_{33}+k_{35}$	16,17,18,36,38,40,41,42	$k_{53}+k_{55}$
29,31,32,33,34	$1+k_{44}$	16,17,18,36,37,38,41,43	$1+k_{56}$
28,30,32,33,34	$k_{45}+k_{47}$	16,17,18,36,37,38,40,41	$k_{58}+k_{59}$
28,29,44,46,47	$k_{46}+k_{47}$	16,17,18,36,37,38,40,42	$k_{57}+k_{59}$

5 Sliding Window Side Channel Cube Attack on PRESENT

5.1. Sliding Window Attack Idea

From above, we can see that, based on the model of single bit information leakage in the 3rd PRESENT round, at most 48-bit key can be obtained and reduce the PRESENT-80 master key search space to 2^{32}. In order to obtain more key bits, we try to extend the attack to the 4th round.

However, as discussed in Section 4.1, both the degree and number of the polynomial grow exponentially with the extending of the rounds. From Figure 2 and Figure 3, we can see that I_4^0 has the lowest polynomial degree, which is 24, and the number of its related 4 bit of the 4th round S-box lookup input (almost the bit 0,4,8,12 of the 3rd round S-box lookup output in Figure 3) is much smaller than others. So we choose I_4^0 as the target bit.

In order to reduce the complexity of the attack, we adopt a two level "divide and conquer" strategy, and propose a sliding window side channel cube attack on PRESENT. The main idea of sliding window side channel cube attack is to use the cube index number as the sliding window during the full target bit polynomial building procedure to reduce the polynomial scale, and use the cube index sum as a sliding window for the final

construction of the target bit polynomial, and extract the related maxterms separately within less complexity.

As the scale of the polynomial is reduced rapidly, sliding window side channel cube attack is quite efficient under deeper rounds leakge model, and can extract the maxterms more efficiently, the same idea is also proposed in [34].

(1) First level divide and conquer strategy

According to the cube plaintext index number N_{SI}, we divide the attack into 23 cases for $N_{SI}=i(i=[1,23]$. And for each N_{SI} candidate, for each encryption round, we reserve these terms involving at most one key variable and public variables number smaller than or equal to N_{SI}, and discard the terms involving more than one key variable or public variables number bigger than N_{SI}.

(2) Second level divide and conquer strategy

For each candidate of $N_{SI}=i$, according to all the possible plaintext cube index sum $S_{SI}=i(i=[0,63\cdot N_{SI}]$, we propose sliding window size L as one time processing S_{SI} candidate number, and divide the attack into $N=(1+63\cdot N_{SI})/L$ cases, compute all the possible cube index set P_{SI}, and verify P_{SI} for about 100 random generated keys and P_{SI} related tweakable plaintext encryptions by BLR tests[37].

5.2. Proposed Attack and Results

Based on the 4th round single bit leakage model, applying the sliding window side channel cube attack, we set $L=16$, compute the 11 cases for $N_{SI}=i(i=[1,11]$, and then execute the attack. The cube index set P_{SI} and related linear key equations are shown in Table 6.

Table 6. The 4th round attack results of I_4^0

P_{SI}	K_{ME}	P_{SI}	K_{ME}
2,3,62,63	$k_{16}+k_{76}$	0,1,3,28,29,32,33,34	$k_{46}+k_{47}$
14,15,62,63	$k_{28}+k_{76}$	0,1,2,28,29,32,33,35	$1+k_{47}$
50,51,62,63	$k_{64}+k_{76}$	0,1,3,16,17,18,32,34	$k_{49}+k_{51}$
0,3,12,13,14	$1+k_{17}+k_{18}$	0,1,3,16,17,18,32,33	$k_{50}+k_{51}$
0,1,12,13,14	$1+k_{19}$	0,1,2,16,17,19,32,33	$1+k_{51}$
0,1,2,12,15	$1+k_{29}+k_{30}$	0,1,3,16,17,18,44,46	$k_{61}+k_{63}$
0,1,2,12,13	$1+k_{31}$	0,2,3,16,17,18,44,45	$k_{62}+k_{63}$
0,1,2,48,51	$1+k_{65}+k_{66}$	0,1,2,16,17,19,44,45	$1+k_{63}$
0,1,2,48,49	$1+k_{67}$	0,1,3,52,54,56,57,58	$k_{69}+k_{71}$
0,1,2,61,63	$1+k_{76}$	0,2,3,52,53,56,57,58	$k_{70}+k_{71}$
48,49,50,60,63	$1+k_{77}+k_{78}$	0,1,2,52,53,56,57,59	$1+k_{71}$
0,1,2,60,61	$1+k_{79}$	0,1,3,52,53,54,56,58	$k_{73}+k_{75}$
2,3,6,7,8,9,11	$k_{16}+k_{20}$	0,2,3,52,53,54,56,57	$k_{74}+k_{75}$
6,7,10,11,60,62,63	$1+k_{20}+k_{24}$	0,1,2,52,53,55,56,57	$1+k_{75}$
0,1,3,18,19,34,35	$1+k_{48}+k_{32}$	0,1,3,22,23,24,25,27,34,35	$1+k_{48}+k_{36}$
0,1,3,30,31,34,35	$1+k_{48}+k_{44}$	0,1,3,20,21,23,26,27,34,35	$1+k_{48}+k_{40}$
2,3,16,17,19,34,35	$k_{16}+k_{48}$	0,1,3,16,17,19,38,39,42,43	$1+k_{52}+k_{56}$
0,1,3,18,19,46,47	$1+k_{32}+k_{60}$	0,1,3,18,19,36,37,39,42,43	$1+k_{32}+k_{56}$
0,1,3,54,55,58,59	$1+k_{72}+k_{68}$	0,1,3,20,22,24,25,26,32,33,34	$k_{37}+k_{39}$
2,3,52,53,55,58,59	$k_{16}+k_{72}$	0,1,3,20,21,24,25,26,32,33,34	$k_{38}+k_{39}$
0,1,3,4,6,8,9,10	$k_{21}+k_{23}$	0,1,2,20,21,24,25,27,32,33,35	$1+k_{39}$
0,1,3,4,5,8,9,10	$k_{22}+k_{23}$	0,1,3,20,21,22,24,26,32,33,34	$k_{41}+k_{43}$
0,1,2,4,5,8,9,11	$1+k_{23}$	0,1,3,20,21,22,24,25,32,33,34	$k_{42}+k_{43}$
0,1,3,4,5,6,8,10	$k_{25}+k_{27}$	0,1,2,20,21,23,24,26,32,34,35	k_{43}
0,1,3,4,5,6,8,9	$k_{26}+k_{27}$	0,1,3,16,17,18,36,38,40,41,42	$k_{53}+k_{55}$
0,1,2,4,5,7,8,9	$1+k_{27}$	0,1,3,16,17,18,36,37,40,41,42	$k_{54}+k_{55}$
0,1,3,16,18,32,33,34	$k_{33}+k_{35}$	0,1,2,16,17,19,36,37,40,41,43	$1+k_{55}$
0,1,3,16,17,32,33,34	$k_{34}+k_{35}$	0,1,3,16,17,18,36,37,38,40,42	$k_{57}+k_{59}$
0,1,2,16,17,32,33,35	$1+k_{35}$	0,1,3,16,17,18,36,37,38,40,41	$k_{58}+k_{59}$
0,1,3,28,30,32,33,34	$k_{45}+k_{47}$	0,1,2,16,17,19,36,37,39,40,41	$1+k_{59}$

It's clear to see that when $N_{SI}=4,5,7,8,10,11$, we can obtain 60 linear key equations, and recover 60-bit of the master key within $3\cdot2^4+9\cdot2^5+8\cdot2^7+24\cdot2^8+4\cdot2^{10}+12\cdot2^{11}\approx2^{15.14}$ chosen plaintexts. Note that we can only obtain $k_{17}+k_{18}$, $k_{29}+k_{30}$, $k_{65}+k_{66}$, $k_{77}+k_{78}$, but without known the exact value of k_{17}, k_{18}, k_{29}, k_{30}, k_{65}, k_{66}, k_{77}, k_{78}.

6 Iterated Side Channel Cube Attack on PRESENT

6.1. Iterated Attack Idea

From section 5, we can obtain 60-bit of the PRESENT master key, in order to recover the exact value of k_i ($i \in$ A, and $A=\{17,18,29,30,65,66,77,78\}$), we propose a new iterated side channel cube attack on PRESENT.

The main idea of iterated side channel cube attack on PRESENT is to iterate the recovered key bits into the former polynomial, and apply a extended side channel cube attack, it can reduce the degree and number of the polynomial, and deduce more key bits related equations, And the chosen plaintext cube variable can make a small change here, as to k_i , if $i \in A$, we still choose P_i as the tweakable cube variable, but if $i \notin A$, we can choose $P_i \oplus k_{16+i}$ as the equivalent extended tweakable cube variable.

6.2. Proposed Attack and Results

(1) Iteration based attack 1

During the attack on PRESENT-80, we iterate the 56 determinate key bits into the first 4 rounds polynomial of

PRESENT, choose related $P_i \oplus k_{16+i}$ as the equivalent tweakable cube, and apply the cube attack on the 4th round of PRESENT. The k_{17}, k_{18}, k_{29}, k_{30}, k_{65}, k_{66}, k_{77}, k_{78} can be recovered by following cube indexes, as shown in Table 7.

Table 7. Results of iterated attack 1 on I_4^0

P_{SI}	K_{ME}	P_{SI}	K_{ME}
0,2,12,13,15	k_{17}	0,1,3,48,50	k_{65}
0,3,12,13,14	$1+k_{17}+k_{18}$	0,1,3,48,49	$1+k_{66}$
0,1,3,12,14	k_{29}	0,1,3,60,62	k_{77}
0,1,3,12,13	$1+k_{30}$	0,1,3,60,61	$1+k_{78}$

(2) Iteration based attack 2

After k_{17}, k_{18}, k_{29}, k_{30}, k_{65}, k_{66}, k_{77}, k_{78} are recovered, in order to obtain k_i, $i \in [0,15]$, we can iterate all 64-bit k_i, $i \in [16,79]$ into the first 4 rounds polynomial of PRESENT, and choose the first round output($PL(SL(AK(P,K^1)))$) bit as the equivalent tweakable cube variable, then apply the attack. As shown in Table 8, we can recover 8 extra key bits: k_0,k_1,k_4,k_5,k_8,k_9,k_{12},k_{14}.

Table 8. Results of iterated attack 2 on I_4^0

P_{SI}	K_{ME}	P_{SI}	K_{ME}
50	$1+k_4$	54,56	$1+k_8$
49	k_5	53,56	k_9
16,46	$1+k_0$	52,58	$1+k_{12}$
16,45	k_1	52,57	k_{13}

So after 2 iteration based attacks, using about $282 \approx 2^{8.14}$ choosing plaintexts, we can obtain 12-bit key. Combing the 60-bit key extracted in Section 5, totally about $2^{15.154}$ chosen plaintexts can reduce the PRESENT-80 master key searching space from 2^{80} to 2^8.

7 Side Channel Cube attack on PRESENT-128

Note that the two attack methods above can be easily extended to PRESENT-128. Based on the 4th round leakage model, applying the sliding window side channel cube attack of Section 5, we extend the attack to PRESENT-128 and recover 60-bit key, as is shown in Table 9.

Table 9. The 4th round attack results of I_4^0 (PRESENT-128)

P_{SI}	K_{ME}	P_{SI}	K_{ME}
2,3,14,15	$k_{64}+k_{76}$	0,2,3,28,29,32,33,34	$k_{94}+k_{95}$
2,3,50,51	$k_{64}+k_{112}$	0,1,2,28,29,32,33,35	$1+k_{95}$
2,3,62,63	$k_{64}+k_{124}$	0,1,3,16,17,18,32,34	$k_{97}+k_{99}$
1,2,12,13,15	k_{64}	0,2,3,16,17,18,32,33	$k_{98}+k_{99}$
0,3,12,13,14	$1+k_{65}+k_{66}$	0,1,2,16,17,19,32,33	$1+k_{99}$
0,2,12,13,14	K_{67}	0,1,3,16,17,18,44,46	$k_{109}+k_{111}$
0,1,2,12,15	$1+k_{77}+k_{78}$	0,2,3,16,17,18,44,45	$k_{110}+k_{111}$
0,1,2,12,13	$1+k_{79}$	0,1,2,16,17,19,44,45	$1+k_{111}$
0,1,2,48,51	$1+k_{113}+k_{114}$	0,1,3,52,54,56,57,58	$k_{117}+k_{119}$
0,1,2,48,49	$1+k_{115}$	0,2,3,52,53,56,57,58	$k_{118}+k_{119}$
0,1,2,60,63	$1+k_{125}+k_{126}$	0,1,2,52,53,56,57,59	$1+k_{119}$
0,1,2,60,61	$1+k_{127}$	0,1,3,52,53,45,56,58	$k_{121}+k_{123}$
2,3,6,7,8,9,11	$k_{64}+k_{68}$	0,2,3,52,53,45,56,57	$k_{122}+k_{123}$
0,1,3,6,7,10,11	$1+k_{68}+k_{72}$	0,1,2,52,53,55,56,57	$1+k_{123}$
0,1,3,18,19,34,35	$1+k_{80}+k_{96}$	22,23,24,25,27,34,35, 48,49,51	$1+k_{84}+k_{96}$
0,1,3,30,31,34,35	$1+k_{92}+k_{96}$	20,22,23,26,27,34,35, 48,50,51	$1+k_{88}+k_{96}$
2,3,16,17,19,34,35	$k_{64}+k_{96}$	16,17,19,38,39,40,41, 43,50,51	$k_{100}+k_{112}$
0,1,3,18,19,46,47	$1+k_{80}+k_{108}$	16,17,19,36,37,39,42, 43,50,51	$k_{104}+k_{112}$
0,1,3,54,55,58,59	$1+k_{116}+k_{120}$	0,1,3,20,22,24,25,26, 32,33,34	$k_{85}+k_{87}$
2,3,52,53,55,58,59	$k_{64}+k_{120}$	0,1,3,20,21,24,25,27, 32,33,34	$k_{86}+k_{87}$
0,1,3,4,6,8,9,10	$k_{69}+k_{71}$	0,1,2,20,21,24,25,27, 32,33,35	$1+k_{87}$
0,1,3,4,5,8,9,10	$k_{70}+k_{71}$	0,1,3,20,21,22,24,26, 32,33,34	$k_{89}+k_{91}$
0,1,2,4,5,8,9,11	$1+k_{71}$	0,1,3,20,21,22,25,26, 32,33,35	$k_{90}+k_{91}$
0,1,3,4,5,6,8,10	$k_{73}+k_{75}$	0,1,2,20,21,23,24,25, 32,34,35	$1+k_{91}$
0,2,3,4,5,6,8,9	$k_{74}+k_{75}$	0,1,3,16,17,18,36,38, 40,41,42	$k_{101}+k_{103}$
0,1,2,4,5,7,8,9	$1+k_{75}$	0,1,3,16,17,18,36,37, 40,41,43	$k_{102}+k_{103}$
0,1,3,16,18,32,33,34	$k_{81}+k_{83}$	0,1,2,16,17,19,36,37, 40,41,43	$1+k_{103}$
0,2,3,16,17,32,33,34	$k_{82}+k_{83}$	0,1,3,16,17,18,36,37, 38,40,42	$k_{105}+k_{107}$
0,1,2,16,17,32,33,35	$1+k_{83}$	0,2,3,16,17,18,36,37, 38,40,41	$k_{106}+k_{107}$
0,1,3,28,30,32,33,34	$k_{93}+k_{95}$	0,1,2,16,17,19,36,37, 39,40,41	$1+k_{107}$

During the attack ,we choose the cube plaintext index number N_{SI} =4,5,7,8,10,11, and obtain 60 linear key equations, finally recover 60-bit of the master key within about $2^{15.14}$ chosen plaintexts. Note that we can only obtain $k_{65}+k_{66}$, $k_{77}+k_{78}$, $k_{113}+k_{114}$, $k_{125}+k_{126}$, but without known the exact value of k_{65},k_{66},k_{77},k_{78},k_{113},k_{114},k_{125},k_{126}.

Then we iterate the recovered 56 determined key bits into the first 4 rounds polynomial of PRESENT,

apply the iterated attack 1 in Section 6, and obtain the other 8 undetermined key bits, as is shown in Table 10.

Table 10. Results of iterated attack 1 on I_4^0 (PRESENT-128)

P_{SI}	K_{ME}	P_{SI}	K_{ME}
0,2,12,13,15	k_{65}	0,1,3,48,50	k_{113}
0,3,12,13,14	$k_{128}+k_{65}+k_{66}$	0,1,3,48,49	$k_{128}+k_{114}$
0,1,3,12,13	$k_{128}+k_{78}$	0,1,3,60,62	k_{125}
0,1,2,12,15	$k_{128}+k_{77}+k_{78}$	0,1,3,60,61	$k_{128}+k_{126}$

Then we apply the iterated attack 2 in Section 6, and obtain other 21 key bits, as is shown in Table 11. It's clear to see that we can extract extra 21 key bits, which is more than 8 key bits of attack on PRESENT-80. This is mainly caused by the subtle difference between the key schedule of PRESENT-80 and PRESENT-128.

Table 11. Results of iterated attack 2 on I_4^0 (PRESENT-128)

P_{SI}	K_{ME}	P_{SI}	K_{ME}	P_{SI}	K_{ME}
1	k_5	6,8	$1+k_8$	16,33	k_{37}
2	$1+k_4$	4,9	k_{13}	16,34	$1+k_{36}$
13	k_{17}	4,10	$1+k_{12}$	16,45	k_{49}
14	$1+k_{16}$	17,32	k_{21}	16,46	$1+k_{48}$
49	k_{53}	18,32	$1+k_{20}$	53,56	k_{57}
50	$1+k_{52}$	29,32	k_{33}	54,56	$1+k_{56}$
5,8	k_9	30,32	$1+k_{32}$	16,37,40	K_{41}

So after 2 iteration based attacks, using about $326 \approx 2^{8.35}$ choosing plaintexts, we can obtain 25-bit key. Combing the 60-bit key above, totally about $2^{15.156}$ chosen plaintexts can reduce the PRESENT-128 master key searching space from 2^{128} to 2^{43}.

8 Conclusion and Future Research

Under the novel precondition of obtaining accurate single bit information leakage model, this paper proposes some improved side channel cube attacks on PRESENT. Our best result is that: based on the 4^{th} round single bit leakage model, about $2^{11.92}$ chosen plaintexts can extract 48-bit PRESENT-80; based on the 4^{th} round single bit leakage model, by applying the sliding window and iteration based attack strategies of this paper, about $2^{15.154}$ and $2^{15.156}$ chosen plaintexts can extract 72-bit PRESENT-80 and 85-bit PRESENT-128 key. As far as we know, this is the most efficient side channel cube attack on PRESENT-80/128.

The further research can be planned as follows:

(1) Identify resistant S-boxes against side channel cube attacks

(2) Physical side channel cube attacks based on different leakage models

(3) Nonlinear equation based side channel cube attacks

(4) Error resistant side channel cube attacks

References

[1] Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., et al. PRESENT: An Ultra-Lightweight Block Cipher[A]. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007[C]. LNCS, vol. 4727, 2007, pp. 450–466.

[2] Bruce Schneier. Applied Cryptography[B], 1996.

[3] Jean-Jacques Quisquater, Math RiZK. Side channel attacks-State-of-the-art[R]. October 2002.

[4] YongBin Zhou, DengGuo Feng. Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing. Cryptology ePrint Archive, Report 2005/388, 2005.

[5] Wang, M.Q.: Differential Cryptanalysis of Reduced-Round PRESENT[A]. In: Vaudenay, S. (ed.) AFRICACRYPT 2008[C]. LNCS, vol. 5023, 2008, pp. 40–49.

[6] Albrecht, M., Cid, C. Algebraic Techniques in Diferential Cryptanalysis[A]. In: Dunkelman, O. (ed.) FSE 2009[C]. LNCS, vol. 5665, 2009, pp. 193–208.

[7] Manoj Kumar, Pratibha Yadav, Meena Kumari. Flaws in Differential Cryptanalysis of Reduced Round PRESENT [EB/OL]. Cryptology ePrint Archive, Report 2010/407, 2010.

[8] Ohkuma, K. Weak keys of reduced-round PRESENT for linear cryptanalysis[A]. In: Preproceeding of SAC 2009[C], 2009.

[9] Jorge Nakahara Jr., Pouyan Sepehrdad, Bingsheng Zhang, and Meiqin Wang. Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT[A]. J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009[C], LNCS 5888, 2009, pp. 58–75.

[10] Cho, J. Linear Cryptanalysis of Reduced-Round PRESENT. In: Topics in Cryptology-CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010, Springer (2010)

[11] Collard, B., Standaert, F.X. A Statistical Saturation Attack against the Block Cipher PRESENT. In: CT-RSA 2009. LNCS, vol. 5473, 2009, pp. 195–210.

[12] Onur Ozen, Kerem Var, Cihangir Tezcan, and C¸ elebi Kocair. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT[A]. C. Boyd and J. Gonz´alez Nieto (Eds.): ACISP 2009[C], LNCS 5594, 2009, pp. 90–107.

[13] BU Fan, JIN Chen-hui. Algebraic Attack on Low—round PRESENT(In Chinese). Computer Engineering, 2010, 36(6):128-130.

[14] Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems [A]. CRYPTO 96[C], LNCS 1109, 1996, pp.104–113.

[15] P. Kocher, J.Jaffe, B. Jun. Differential power analysis[A]. CRYPTO '99[C] (M. Wiener, ed.), LNCS 1666, 1999, pp. 388-397.

[16] J. J. Quisquater, D. Samyde. A new tool for non-intrusive analysis of smart cards based on electro-magnetic emissions: the SEMA and DEMA methods[EB/OL]. Eurocrypt rump session, 2000.

[17] Shamir, A. and Tromer, E. Acoustic cryptanalysis: On nosy people and noisy machines[EB/OL]. Avaiable at http://www.wisdom.weizmann.ac.il/~tromer/acoustic/, 2004.

[18] D.Boneh, R.A.DeMillo, R.J.Lipton. On the Importance of Checking Cryptographic Protocols for Faults[J]. LNCS 1233, 1997: 37-51.

[19] Carsten Rolfes. Side-Channel Analysis Aspects of Lightweight Block Ciphers. Diploma Thesis of Ruhr-University Bochum, February 28, 2009.

[20] Li, J.R., Gu, D.W. Differential Fault Analysis on PRESENT[A]. CHINACRYPT 2009[C](in chinese), 2009, pp.3-13.

[21] Wang, G.L., Wang, S.S. Differential Fault Analysis on PRESENT Key Schedule[A]. In Proc of International Conference on Computational Intelligence and Security (CIS 2010[C]), 2010, pp.362-366.

[22] XinJjie Zhao, Tao Wang, ShiZe Guo. Fault Propagate Pattern Based DFA on SPN Structure Block Ciphers using Bitwise Permutation, with Application to PRESENT and PRINTcipher[EB/OL]. Cryptology ePrint Archive, Report 2011/089, 2011.

[23] Andrey Bogdanov, Andrey Pyshkin. Algebraic Side-Channel Collision Attacks on AES[EB/OL], Cryptology ePrint Archive, Report 2007/477, 2007.

[24] M. Renauld, F.-X. Standaert. Algebraic Side-Channel Attacks[EB/OL], Cryptology ePrint Archive, Report 2009/279, 2009.

[25] N.T.Courtois, J.Pieprzyk. Cryptanalysis of block ciphers with over-defined systems of equations[A]. In Y. Zheng (Ed.): ASIACRYPT 2002[C], LNCS 2501, 2002, pp.267–287.

[26] Dinur, I., Shamir, A. Side Channel Cube Attacks on Block Ciphers[EB/OL]. Cryptology ePrint Archive. Report 2009/127.

[27] Dinur, I., Shamir, A. Cube Attacks on Tweakable Black Box Polynomials. Cryptology ePrint Archive, Report 2008/385, 2008.

[28] Dinur, I., Shamir, A. Cube Attacks on Tweakable Black Box Polynomials[A]. In: Joux, A. (ed.) EUROCRYPT 2009[C]. LNCS, vol. 5479, 2009, pp. 278–299.

[29] Aumasson, J.-P., Dinur, I., Meier, W., Shamir, A. Cube Testers and Key Recovery attacks on Reduced-Round MD6 and Trivium[A]. In: Dunkelman, O. (ed.) FSE 2009[C]. LNCS, vol. 5665, 2009, pp. 1–22.

[30] Lin Yang, Meiqin Wang, and Siyuan Qiao. Side Channel Cube Attack on PRESENT[A]. In J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009[C], LNCS 5888, 2009, pp. 379–391.

[31] Shekh Faisal Abdul-Latip, Mohammad Reza Reyhanitabar, Willy Susilo and Jennifer Seberry.Extended Cubes: Enhancing the cube attack by Extracting Low-Degree Non-linear Equations [A]. In Bruce Cheung, Lucas Chi Kwong Hui, Ravi Sandhu, Duncan S.Wong (Eds.): ASIACCS 2011[C], 2011, pp.296–305.

[32] Vielhaber, M. Breaking ONE.FIVIUM by AIDA - an Algebraic IV Differential Attack. Cryptology ePrint Archive, Report 2007/413 (2007) http://eprint.iacr.org/.

[33] Lai, X. Higher Order Derivatives and Differential Cryptanalysis. Communications and Cryptography: Two Sides of One Tapestry (1994) pp. 227

[34] Aileen Zhang, Chu-Wee Lim, Khoongming Khoo, Wei Lei, and Josef Pieprzyk. Extensions of the Cube Attack based on Low Degree Annihilators[EB/OL]. Cryptology ePrint Archive. Report 2009/049.

[35] Shekh Faisal Abdul-Latip, Mohammad Reza Reyhanitabar, Willy Susilo, and Jennifer Seberry. On the Security of NOEKEON against Side Channel Cube Attacks[A]. ISPEC 2010[C], LNCS 6047, 2010, pp. 45–55.

[36] Gregory V. Bard, Nicolas T. Courtois, Jorge Nakahara Jr, Pouyan Sepehrdad, and Bingsheng Zhang. Algebraic, AIDA/Cube and Side Channel Analysis of KATAN Family of Block Ciphers[A].

[37] Blum, M., Luby, M., Rubinfeld, R. Self-testing/correcting with applications to numerical problems. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing. STOC '90, New York, NY, USA, ACM (1990) 73-83.

Appendix

Table A.1. The 3rd round attack results of I_3^1

P_{SI}	K_{ME}	P_{SI}	K_{ME}
2,16,32	$1+k_{17}$	6,8,16,32	$1+k_{21}$
1,16,32	k_{18}	5,8,16,32	k_{22}
14,16,32	$1+k_{29}$	4,10,16,32	$1+k_{25}$
13,16,32	k_{30}	4,9,16,32	k_{26}
0,18,32	$1+k_{33}$	0,22,24,32	$1+k_{37}$
0,17,32	k_{34}	0,21,24,32	k_{38}
0,30,32	$1+k_{45}$	0,20,26,32	$1+k_{41}$
0,29,32	k_{46}	0,20,25,32	k_{42}
0,16,34	$1+k_{49}$	0,16,38,40	$1+k_{53}$
0,16,33	k_{50}	0,16,37,40	k_{54}
0,16,46	$1+k_{61}$	0,16,36,42	$1+k_{57}$
0,16,45	K_{62}	0,16,36,41	k_{58}
0,16,50	$1+k_{65}$	0,16,54,56	$1+k_{69}$
0,16,49	k_{66}	0,16,53,56	k_{70}
0,16,62	$1+k_{77}$	0,16,52,58	$1+k_{73}$
0,16,61	k_{78}	0,16,52,57	k_{74}

Table A.2. The 3rd round attack results of I_3^8

P_{SI}	K_{ME}	P_{SI}	K_{ME}
1,3	$1+k_{16}$	28,29,31,33,35	$1+k_{48}$
0,3	$1+k_{17}+k_{18}$	16,17,19,32,35	$1+k_{49}+k_{50}$
0,2	k_{19}	16,17,19,32,34	k_{51}
13,15	$1+k_{28}$	16,17,19,45,47	$1+k_{60}$
12,15	$1+k_{29}+k_{30}$	16,17,19,44,47	$1+k_{61}+k_{62}$
12,13	$1+k_{31}$	16,17,19,44,45	$1+k_{63}$
49,51	$1+k_{64}$	53,55,56,57,59	$1+k_{68}$
48,51	$1+k_{65}+k_{66}$	52,55,56,57,59	$1+k_{69}+k_{70}$
48,49	$1+k_{67}$	52,53,56,57,59	$1+k_{71}$
61,63	$1+k_{76}$	52,53,55,57,59	$1+k_{72}$
60,63	$1+k_{77}+k_{78}$	52,53,55,56,59	$1+k_{73}+k_{74}$
60,61	$1+k_{79}$	52,53,55,56,57	$1+k_{75}$
5,7,8,9,11	$1+k_{20}$	21,23,24,25,27,32,33,35	$1+k_{36}$
4,7,8,9,10	$1+k_{21}+k_{22}$	20,23,24,25,27,32,33,35	$1+k_{37}+k_{38}$
4,5,8,9,11	$1+k_{23}$	20,21,24,25,27,32,33,35	$1+k_{39}$
4,5,7,9,11	$1+k_{24}$	20,21,23,25,27,32,33,35	$1+k_{40}$
4,5,7,8,11	$1+k_{25}+k_{26}$	20,22,23,24,27,32,33,35	$1+k_{41}+k_{42}$
4,5,7,8,9	$1+k_{27}$	20,22,23,24,26,32,33,35	k_{43}
17,19,32,33,35	$1+k_{32}$	16,17,19,37,39,40,41,43	$1+k_{52}$
16,19,32,33,35	$1+k_{33}+k_{34}$	16,17,19,36,39,40,41,43	$1+k_{53}+k_{54}$
16,17,44,46,47	$1+k_{35}$	16,17,19,36,37,40,41,43	$1+k_{55}$
29,31,32,33,35	$1+k_{44}$	16,17,19,36,37,39,41,43	$1+k_{56}$
28,31,32,33,35	$1+k_{45}+k_{46}$	16,17,19,36,37,39,40,43	$1+k_{57}+k_{58}$
28,29,32,33,35	$1+k_{47}$	16,17,19,36,37,39,40,41	$1+k_{59}$

Table A.3. The 3rd round attack results of I_3^{12}

P_{SI}	K_{ME}	P_{SI}	K_{ME}
1,2	$1+k_{16}$	16,17,18,33,34	$1+k_{48}$
0,2	$k_{17}+k_{19}$	16,17,18,32,34	$k_{49}+k_{51}$
0,1	$k_{18}+k_{19}$	16,17,18,32,33	$k_{50}+k_{51}$
13,14	$1+k_{28}$	16,17,18,45,46	$1+k_{60}$
12,14	$k_{29}+k_{31}$	16,17,18,44,46	$k_{61}+k_{63}$
12,13	$k_{30}+k_{31}$	16,17,18,44,45	$k_{62}+k_{63}$
49,50	$1+k_{64}$	53,54,56,57,58	$1+k_{68}$
48,50	$K_{65}+k_{67}$	52,54,56,57,58	$k_{69}+k_{71}$
48,49	$k_{66}+k_{67}$	52,53,56,58,59	$k_{70}+k_{71}$
61,62	$1+k_{76}$	52,53,54,57,58	$1+k_{72}$
60,62	$k_{77}+k_{79}$	52,53,54,56,58	$k_{73}+k_{75}$
60,61	$k_{78}+k_{79}$	52,53,54,56,57	$k_{74}+k_{75}$
5,6,8,9,10	$1+k_{20}$	21,22,24,25,26,32,33,34	$1+K_{36}$
4,6,8,9,10	$k_{21}+k_{23}$	20,22,24,25,26,32,33,34	$k_{37}+k_{39}$
4,5,8,9,10	$k_{22}+k_{23}$	20,21,24,25,26,32,33,34	$k_{38}+k_{39}$
4,5,6,9,10	$1+k_{24}$	20,21,22,25,26,32,33,34	$1+k_{40}$
4,5,6,8,10	$k_{25}+k_{27}$	20,21,22,24,26,32,33,34	$k_{41}+k_{43}$
4,5,6,8,9	$k_{26}+k_{27}$	20,21,22,24,25,32,33,34	$k_{42}+k_{43}$
17,18,32,33,34	$1+k_{32}$	16,17,18,37,38,40,41,42	$1+k_{52}$
16,18,32,33,34	$k_{33}+k_{35}$	16,17,18,36,38,40,41,42	$k_{53}+k_{55}$
16,17,32,33,34	$k_{34}+k_{35}$	16,17,18,36,37,40,41,42	$k_{54}+k_{55}$
29,30,32,33,34	$1+k_{44}$	16,17,18,36,37,38,41,42	$1+k_{56}$
28,30,32,33,34	$k_{45}+k_{47}$	16,17,18,36,37,38,40,42	$k_{57}+k_{59}$
28,29,32,33,34	$k_{46}+k_{47}$	16,17,18,36,37,38,40,41	$K_{58}+K_{59}$