Symbolic computation in block cipher with application to PRESENT

Changyong Peng [a] Chuangying Zhu [b] Yuefei Zhu [a] Fei Kang [a]

[a] *Zhengzhou Information Science and Technology Institute,Zhengzhou, China,*
Email: pengchangyong@tom.com,39463021@qq.com,zyf0136@sina.com,k_fei@sina.com

[b] *School of Computer and Control, Guillin University of Electronic Technology, Guilin, China*

Abstract

In this paper,we give an example of how symbolic computation are used to analyze the block cipher PRESENT,an ultra-lightweight block cipher proposed by Bogdanov et al. at CHES'07.The block size is 64 bits and the key size can be 80 bit or 128 bit.Using Mathematica 7.0,this paper obtains the unexpanded polynomial expressions of the output of round 1-6 of PRESENT-80(80-bit key variant).The time complexity of getting these expressions is 4 minutes on a PC with a 2.6GHz CPU and 8G RAM.Then we expand the expressions of the output of round 1-2 and the LSB(least significant bit) of the output of round 3 and obtain the ANFs(Algebraic Normal Form) of these 129(=2*64+1) expressions. The time complexity of getting these ANFs is 22 minutes.It it known that the time complexity of the classical method of computing the ANF of an n-ary Boolean function from its truth table is $O(n2^n)$,with total time complexity of obtaining these 129 ANFs $O(129 * 144 * 2^{144}) = O(2^{158})$(each of the 129 ANFs can be viewed as a 144-ary Boolean function,where 144=64+80,the sum of the block size and the key size).As an application,we give a side channel attack on PRESENT-80 under the single bit leakage model proposed by Dinur and Shamir.If the LSB bit of the output of the 3rd round can be obtained without error,then with 200 known plaintexts,we can set up an equation system in terms of the master key bits and can recover 43 bits key by the Gröbner Basis method.Compared with the previous side channel attack on PRESENT,such as Yang et al. in CANS 2009,Abdul-Latip et al. in ASIACCS 2011 and Zhao et al. in 2011,each of which needs at least 2^{13} chosen plaintexts,the data complexity of our attack is the best.

Keywords: PRESENT, symbolic computation, Gröbner Basis, side channel attack, block cipher

1. Introduction

As ubiquitous computing becomes a reality, sensitive information is increasingly processed and transmitted by smart cards, mobile devices and various types of embedded systems. This has led to the requirement of a new class of lightweight cryptographic algorithm to ensure security in these resource constrained environments.In January 2012,the International Organization for Standardization (ISO) has standardized two low-cost block ciphers [1] for this purpose,CLEFFIA and PRESENT.

PRESENT [2] is an ultra-lightweight block cipher proposed by Bogdanov et al. at CHES'07.The block size is 64 bits and the key size can be 80 bit or 128 bit.In this paper,we focus on the al-

gebraic expression of PRESENT.The standard algebraic expression of a Boolean functions is called its algebraic normal form(ANF). Every Boolean function has a unique ANF.A block cipher can be viewed as a vector-valued Boolean function and each ciphertext bit can be viewed as a Boolean function with the plaintext bits and key bits as variables. The ANF form of a block cipher is important in the cryptanalysis of the block cipher.For example, the idea of the method of formal coding [3] is to find the algebraic expression of each bit in the ciphertext as an XOR sum of products of the bits of the plaintext and the master key.The XOR-sum form of the ciphertext is just the algebraic normal form.The formal manipulations of these expressions may decrease the key search effort.The ANF form of a cipher is also useful in the realization of a cipher.For example,the author of [4],in order to obtain a minimal hardware realization of DES,optimized Boolean expressions that represents DES S-boxes.

The classical method for obtaining the ANF of an n-variable Boolean function is the algorithm [5] of computing the ANF from truth table,with time complexity $O(n2^n)$.For the PRESENT-80 block cipher,each bit of the middle state or the final ciphertext can be viewed as a 144-ary Boolean function.Here 144 is the sum of the block size(64) and the key size(80).So if we use the algorithm in [5] to find the ANF,the time complexity is $O(144 * 2^{144}) = O(2^{151})$,which is computationally infeasible.

Using Mathematica 7.0,this paper obtains the unexpanded polynomial expressions of the output of round 1-6 of PRESENT-80(80-bit key variant).The time complexity of getting these expressions is 4 minutes on a PC with a 2.6GHz CPU and 8G RAM.Then we expand the expressions of the output of round 1-2 and the LSB(least significant bit) of the output of round 3 and obtain the ANFs(Algebraic Normal Form) of these 129(=2*64+1) expressions. The time complexity of getting these ANFs is 22 minutes.Note that the time complexity of computing these 129 ANFs by the classical method in [5] is $O(129 * 144 * 2^{144}) = O(2^{158})$(each of the 129 ANFs can be viewed as a 144-ary Boolean function,where 144=64+80,the sum of the block size and the key size).As an application,we give a side channel attack on PRESENT-80 under the single bit leakage model proposed by Dinur and Shamir.If the LSB bit of the output of the 3rd round can be obtained without error,then with 200 known plaintexts,we can set up an equation system in terms of the master key bits and can recover 43 bits key by the Gröbner Basis method.Compared with the previous side channel attack on PRESENT,such as Yang et al. in CANS 2009,Abdul-Latip et al. in ASIACCS 2011 and Zhao et al. in 2011,each of which needs at least 2^{13} chosen plaintexts,the data complexity of our attack is the best.

2. Description of PRESENT

PRESENT is a 31-round SPN structure block cipher with block size of 64 bits, the cipher is described in Figure 1 and 2. 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 31st round output Xored with the subkey K_{32} is the ciphertext.

Each round of PRESENT consists of the following 3 steps.

(1)**addRoundKey**. Given round key K_i=$\kappa^i_{63}...\kappa^i_0$ for $1 \leq i \leq 31$ and current state $b_{63}...b_0$, addRoundKey consists of the operation for $0 \leq i \leq 63$,

$$b_j \rightarrow b_j \oplus \kappa^i_j.$$

Note:In this paper we number bits from zero with bit zero(the least significant bit) on the right of a block or word.

generateRoundKeys()
for $i = 1$ to 31 **do**
  addRoundKey(STATE,K_i)
  sBoxLayer(STATE)
  pLayer(STATE)
end for
addRoundKey(STATE,K_{32})

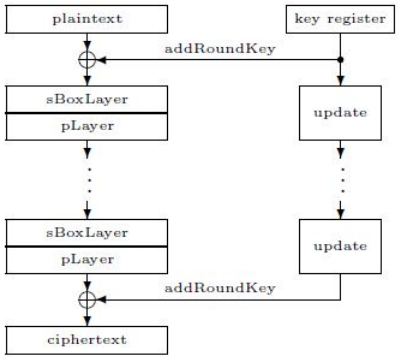

Figure 1: A top-level description of 31-round PRESENT encryption.

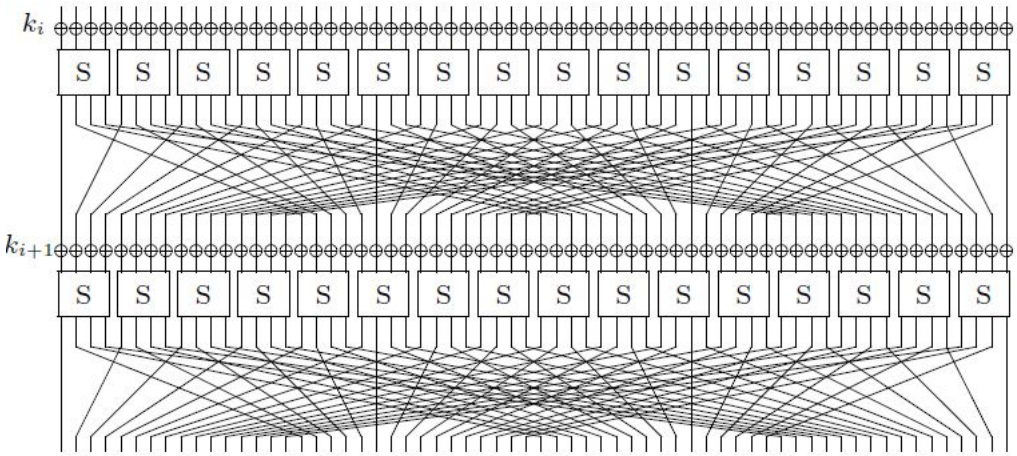

Figure 2: The S/P network for PRESENT

Table 1: **Specification of the PRESENT S-box.**

i	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[i]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 2: **Specification of the PRESENT permutation layer.**

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P[i]	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P[i]	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P[i]	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P[i]	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

(2)**sBoxlayer**. PRESENT uses a single 4-bit S-box S as shown in Table 1, which is applied 16 times in parallel in each round. The 4-bit nibble i at the input of an S-box is substituted by the 4-bit $S[i]$ in output, i.e. $S : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$.

Let $x = x_3x_2x_1x_0$ be the input of Sbox and $y = y_3y_2y_1y_0$ be the output.Then we have

$y_3 = x_1x_2x_0 + x_1x_3x_0 + x_2x_3x_0 + x_0 + x_1 + x_1x_2 + x_3 + 1$,

$y_2 = x_0x_1 + x_0x_3x_1 + x_3x_1 + x_2 + x_0x_3 + x_0x_2x_3 + x_3 + 1$,

$y_1 = x_0x_2x_1 + x_0x_3x_1 + x_3x_1 + x_1 + x_0x_2x_3 + x_2x_3 + x_3$,

$y_0 = x_0 + x_1x_2 + x_2 + x_3$.

(3)**pLayer**.The pLayer which provide linear bitwise permutation is shown in Table 2.Bit i of state is moved to bit position $P(i)$.

The key schedule. The difference between the 80-bit key and the 128-bit key variants of PRESENT is on the key schedule. For the 80-bit key variant, the user-supplied key that is stored in key register K is represented as $k_{79}k_{78}...k_0$. The 64-bit round key K_i=$\kappa_{63}...\kappa_0$ consists of the leftmost bits of the current contents (i.e. at round i) of register K . Thus the round key at round i can be depicted as:

$$K_i=\kappa_{63}...\kappa_0=k_{79}k_{78}...k_{16}.$$

After extracting the round key K_i, the key register $K = k_{79}k_{78}...k_0$ is updated as follows.

1. $[k_{79}k_{78}...k_1k_0]$=$[k_{18}k_{17}...k_{20}k_{19}]$

2. $[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$

3. $[k_{19}k_{18}k_{17}k_{16}k_{15}]$=$[k_{19}k_{18}k_{17}k_{16}k_{15}]\oplus round\_counter$

Thus, the key register is rotated by 61 bit positions to the left, the left-most four bits are passed through the present S-box, and the *round_counter* value i 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.For more information of the key-schedule of PRESENT-128,see [2].

3. Obtaining the algebraic expression of the middle state of PRESENT

In this section,we give the polynomial expressions of the output of round 1-6 of PRESENT-80(It's similar for PRESENT-128,we omit the details here).Of these 64*6=384 expressions,we

expand 129 expressions and obtain the standard algebraic expression i.e. the algebraic normal form(ANF).These 129 expressions are the output of the 1st two rounds and the least significant bit of the output of the 3rd round,with the plaintext bits and master key bits as variables.

Notation:

$K = k_{79}k_{78}...k_0$: master key of PRESENT,

$P = p_{63}p_{62}...p_0$: plaintext of PRESENT,

$B^i = b^i_{63}b^i_{62}...b^i_0$: output of the ith($i = 1, ..., 6$)round of PRESENT.

We use the powerful symbolic computation software Mathematica [6] to obtain the result.Let $\mathbb{F}_2 = GF(2)$ and $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ be a mapping from n binary input bits into one output bit,then f is called a Boolean function. The ANF of the Boolean function f is the polynomial

$$f(x_1, x_2, ..., x_n) = a_0 \oplus a_1x_1 \oplus ... \oplus a_nx_n \oplus a_{n+1}x_1x_2 \oplus ... \oplus a_{2^n-1}x_1x_2...x_n$$

with unique a_i's in $\mathbb{F}_2$. The number of nonzero a_i's is called the number of terms in the ANF of f.

First we give the ANF of the Sbox of PRESENT.Let $x = x_3x_2x_1x_0$ be the input of Sbox and $y = y_3y_2y_1y_0$ be the output.Then we have

$y_3 = x_1x_2x_0 + x_1x_3x_0 + x_2x_3x_0 + x_0 + x_1 + x_1x_2 + x_3 + 1$,

$y_2 = x_0x_1 + x_0x_3x_1 + x_3x_1 + x_2 + x_0x_3 + x_0x_2x_3 + x_3 + 1$,

$y_1 = x_0x_2x_1 + x_0x_3x_1 + x_3x_1 + x_1 + x_0x_2x_3 + x_2x_3 + x_3$,

$y_0 = x_0 + x_1x_2 + x_2 + x_3$.

With the above algebraic expressions of the Sbox of PRESENT,we can give the Mathematica code of the key schedule generation and the pure symbolic encryption procedure of PRESENT-80.The source code was given in appendix A.We have tested the correctness of our symbolic encryption code by using the 4 test vectors given in [2].The symbolic encryption code gets the correct answer each time.Then we set the plaintext and master key as pure symbols,i.e. let $K = k_{79}k_{78}...k_0$ be the master key,$P = p_{63}p_{62}...p_0$ be the plaintext and $B^i = b^i_{63}b^i_{62}...b^i_0$ be the output of the ith round of PRESENT-80.Then we run the symbolic encryption code and obtain the ANF of each bit of $B^i = b^i_{63}b^i_{62}...b^i_0$ for $i = 1, 2$ and the ANF of b^3_0.

3.1. Algebraic normal form of the output of the 1st round

Let $B^1 = b^1_{63}b^1_{62}...b^1_0$ be the output of the 1st round of PRESENT-80.Then the number of terms of $B^1 = b^1_{63}b^1_{62}...b^1_0$ are as follows:35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 36, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10.

We list only the most significant bit b^1_{63} and the 4 least significant bits $b^1_3, b^1_2, b^1_1, b^1_0$ here.

b^1_{63}=$k_{78}k_{76}p_{61} + k_{79}k_{76}p_{61} + k_{77}k_{76}p_{62} + k_{79}k_{76}p_{62} + k_{76}p_{61}p_{62} + k_{77}k_{76}p_{63} + k_{78}k_{76}p_{63} + k_{76}p_{61}p_{63} + k_{76}p_{62}p_{63} + k_{77}k_{78}p_{60} + k_{77}k_{79}p_{60} + k_{78}k_{79}p_{60} + k_{78}p_{61} + k_{78}p_{60}p_{61} + k_{79}p_{60}p_{61} + k_{77}p_{62} + k_{77}p_{60}p_{62} + k_{79}p_{60}p_{62} + k_{77}p_{60}p_{63} + k_{78}p_{60}p_{63} + k_{77}k_{78}k_{76} + k_{77}k_{79}k_{76} + k_{78}k_{79}k_{76} + k_{76} + k_{77} + k_{77}k_{78} + k_{79} + p_{60} + p_{61} + p_{60}p_{61}p_{62} + p_{61}p_{62} + p_{60}p_{61}p_{63} + p_{60}p_{62}p_{63} + p_{63} + 1$

$=(k_{76} + p_{60})(k_{77} + p_{61})(k_{78} + p_{62}) + (k_{77} + p_{61})(k_{78} + p_{62}) + (k_{76} + p_{60})(k_{77} + p_{61})(k_{79} + p_{63}) + (k_{76} + p_{60})(k_{78} + p_{62})(k_{79} + p_{63}) + k_{76} + k_{77} + k_{79} + p_{60} + p_{61} + p_{63} + 1$,

b^1_0=$k_{18}p_1 + k_{17}p_2 + k_{16} + k_{17}k_{18} + k_{18} + k_{19} + p_0 + p_1p_2 + p_2 + p_3$

$=(k_{17} + p_1)(k_{18} + p_2) + k_{16} + k_{18} + k_{19} + p_0 + p_2 + p_3$.

b^1_1=$(k_{21} + p_5)(k_{22} + p_6) + k_{20} + k_{22} + k_{23} + p_4 + p_6 + p_7$

$=k_{22}p_5 + k_{21}p_6 + k_{20} + k_{21}k_{22} + k_{22} + k_{23} + p_4 + p_5p_6 + p_6 + p_7$.

$b_2^1=(k_{25} + p_9)\,(k_{26} + p_{10}) + k_{24} + k_{26} + k_{27} + p_8 + p_{10} + p_{11}$
$=k_{26}p_9 + k_{25}p_{10} + k_{24} + k_{25}k_{26} + k_{26} + k_{27} + p_8 + p_9p_{10} + p_{10} + p_{11}$.

$b_3^1=(k_{29} + p_{13})\,(k_{30} + p_{14}) + k_{28} + k_{30} + k_{31} + p_{12} + p_{14} + p_{15}$
$=k_{30}p_{13} + k_{29}p_{14} + k_{28} + k_{29}k_{30} + k_{30} + k_{31} + p_{12} + p_{13}p_{14} + p_{14} + p_{15}$.

The correctness of these expressions can easily be checked manually from the description of PRESENT.

3.2. Algebraic normal form of the output of the 2nd round

Let $B^2 = b_{63}^2 b_{62}^2 ... b_0^2$ be the output of the 2nd round of PRESENT-80.Then the number of terms of $B^2 = b_{63}^2 b_{62}^2 ... b_0^2$ are as follows:158482, 131145, 131145, 131145, 108339, 110055, 110055, 110055, 150702, 148686, 153440, 153440, 4148, 4148, 3638, 4148, 114455, 89495, 89495, 89495, 74059, 75207, 75207, 75207, 103030, 103704, 105488, 105488, 3048, 3048, 2738, 3048, 157359, 129956, 129956, 129956, 108470, 109000, 109000, 109000, 152045, 149993, 154771, 154771, 4257, 4257, 3721, 4257, 1711, 1330, 1330, 1330, 1186, 1188, 1188, 1188, 1478, 1476, 1480, 1480, 154, 154, 150, 154.

First,we give the algebraic expression of the least significant bit b_0^2 here.

Since $b_0^2=\left(b_1^1 + \kappa_1\right)\left(b_2^1 + \kappa_2\right) + b_0^1 + b_2^1 + b_3^1 + \kappa_0 + \kappa_2 + \kappa_3$, where $\kappa_3\kappa_2\kappa_1\kappa_0$ are the 4 least significant bits of the second round subkey.From the key schedule algorithm,we have

$\kappa_3 = k_{38}, \kappa_2 = k_{37}, \kappa_1 = k_{36}, \kappa_0 = k_{35}$.

Replace $b_3^1, b_2^1, b_1^1, b_0^1$ with their expressions in section 3.1 and expand it we have

$b_0^2=b_0^2(k_{16}, k_{17}, ..., k_{30}, k_{31}, k_{35}, k_{36}, k_{37}, k_{38}, p_0, p_1, ..., p_{14}, p_{15})=k_{16}+k_{17}k_{18}+k_{18}+k_{19}+k_{20}k_{24}+k_{21}k_{22}k_{24}+k_{22}k_{24}+k_{23}k_{24}+k_{24}+k_{20}k_{26}+k_{21}k_{22}k_{26}+k_{22}k_{26}+k_{23}k_{26}+k_{20}k_{25}k_{26}+k_{21}k_{22}k_{25}k_{26}+k_{22}k_{25}k_{26}+k_{23}k_{25}k_{26}+k_{25}k_{26}+k_{26}+k_{20}k_{27}+k_{21}k_{22}k_{27}+k_{22}k_{27}+k_{23}k_{27}+k_{27}+k_{28}+k_{29}k_{30}+k_{30}+k_{31}+k_{35}+k_{24}k_{36}+k_{25}k_{26}k_{36}+k_{26}k_{36}+k_{27}k_{36}+k_{20}k_{37}+k_{21}k_{22}k_{37}+k_{22}k_{37}+k_{23}k_{37}+k_{36}k_{37}+k_{37}+k_{38}+p_0+k_{18}p_1+k_{17}p_2+p_1p_2+p_2+p_3+k_{24}p_4+k_{25}k_{26}p_4+k_{26}p_4+k_{27}p_4+k_{37}p_4+k_{22}k_{24}p_5+k_{22}k_{26}p_5+k_{22}k_{25}k_{26}p_5+k_{22}k_{27}p_5+k_{22}k_{37}p_5+k_{21}k_{24}p_6+k_{24}p_6+k_{21}k_{26}p_6+k_{21}k_{25}k_{26}p_6+k_{25}k_{26}p_6+k_{26}p_6+k_{21}k_{27}p_6+k_{27}p_6+k_{21}k_{37}p_6+k_{37}p_6+k_{24}p_5p_6+k_{25}k_{26}p_5p_6+k_{26}p_5p_6+k_{27}p_5p_6+k_{37}p_5p_6+k_{24}p_7+k_{25}k_{26}p_7+k_{26}p_7+k_{27}p_7+k_{37}p_7+k_{20}p_8+k_{21}k_{22}p_8+k_{22}p_8+k_{23}p_8+k_{36}p_8+p_4p_8+k_{22}p_5p_8+k_{21}p_6p_8+p_5p_6p_8+p_6p_8+p_7p_8+p_8+k_{20}k_{26}p_9+k_{21}k_{22}k_{26}p_9+k_{22}k_{26}p_9+k_{23}k_{26}p_9+k_{26}p_9+k_{26}k_{36}p_9+k_{26}p_4p_9+k_{22}k_{26}p_5p_9+k_{21}k_{26}p_6p_9+k_{26}p_6p_9+k_{26}p_5p_6p_9+k_{26}p_7p_9+k_{20}p_{10}+k_{21}k_{22}p_{10}+k_{22}p_{10}+k_{23}p_{10}+k_{20}k_{25}p_{10}+k_{21}k_{22}k_{25}p_{10}+k_{22}k_{25}p_{10}+k_{23}k_{25}p_{10}+k_{25}p_{10}+k_{25}k_{36}p_{10}+k_{36}p_{10}+k_{25}p_4p_{10}+p_4p_{10}+k_{22}p_5p_{10}+k_{22}k_{25}p_5p_{10}+k_{21}p_6p_{10}+k_{21}k_{25}p_6p_{10}+k_{25}p_6p_{10}+k_{25}p_5p_6p_{10}+p_5p_6p_{10}+p_6p_{10}+k_{25}p_7p_{10}+p_7p_{10}+k_{20}p_9p_{10}+k_{21}k_{22}p_9p_{10}+k_{22}p_9p_{10}+k_{23}p_9p_{10}+k_{36}p_9p_{10}+p_4p_9p_{10}+k_{22}p_5p_9p_{10}+k_{21}p_6p_9p_{10}+p_5p_6p_9p_{10}+p_6p_9p_{10}+p_7p_9p_{10}+p_9p_{10}+p_{10}+k_{20}p_{11}+k_{21}k_{22}p_{11}+k_{22}p_{11}+k_{23}p_{11}+k_{36}p_{11}+p_4p_{11}+k_{22}p_5p_{11}+k_{21}p_6p_{11}+p_5p_6p_{11}+p_6p_{11}+p_7p_{11}+p_{11}+p_{12}+k_{30}p_{13}+k_{29}p_{14}+p_{13}p_{14}+p_{14}+p_{15}$.

Similarly,we have

$b_1^2=b_1^2(k_{32}, k_{33}, ..., k_{46}, k_{47}, p_{16}, p_{17}, ..., p_{30}, p_{31})=(k_{33}+p_{17})(k_{34}+p_{18})+(k_{41}+p_{25})(k_{42}+p_{26})+((k_{37}+p_{21})(k_{38}+p_{22})+k_{36}+k_{38}+k_{39}+k_{40}+p_{20}+p_{22}+p_{23})((k_{41}+p_{25})(k_{42}+p_{26})+k_{40}+k_{41}+k_{42}+k_{43}+p_{24}+p_{26}+p_{27})+(k_{45}+p_{29})(k_{46}+p_{30})+k_{32}+k_{34}+k_{35}+k_{39}+k_{40}+k_{41}+k_{43}+k_{44}+k_{46}+k_{47}+p_{16}+p_{18}+p_{19}+p_{24}+p_{26}+p_{27}+p_{28}+p_{30}+p_{31}$.

b_2^2=$b_2^2(k_{43}, k_{44}, k_{45}, k_{46}, k_{48}, k_{49}, ..., k_{62}, k_{63}, p_{32}, p_{33}, ..., p_{45}, p_{46}, p_{47})$=$(k_{49} + p_{33})(k_{50} + p_{34}) + (k_{57} + p_{41})(k_{58} + p_{42}) + ((k_{53} + p_{37})(k_{54} + p_{38}) + k_{44} + k_{52} + k_{54} + k_{55} + p_{36} + p_{38} + p_{39})((k_{57} + p_{41})(k_{58} + p_{42}) + k_{45} + k_{56} + k_{58} + k_{59} + p_{40} + p_{42} + p_{43}) + (k_{61} + p_{45})(k_{62} + p_{46}) + k_{43} + k_{45} + k_{46} + k_{48} + k_{50} + k_{51} + k_{56} + k_{58} + k_{59} + k_{60} + k_{62} + k_{63} + p_{32} + p_{34} + p_{35} + p_{40} + p_{42} + p_{43} + p_{44} + p_{46} + p_{47}$.

b_3^2=$b_3^2(k_{47}, k_{48}, k_{49}, k_{50}, k_{64}, k_{65}, ..., k_{78}, k_{79}, p_{48}, p_{49}, ..., p_{62}, p_{63})$ =$(k_{65} + p_{49})(k_{66} + p_{50}) + (k_{73} + p_{57})(k_{74} + p_{58}) + ((k_{69} + p_{53})(k_{70} + p_{54}) + k_{48} + k_{68} + k_{70} + k_{71} + p_{52} + p_{54} + p_{55})((k_{73} + p_{57})(k_{74} + p_{58}) + k_{49} + k_{72} + k_{74} + k_{75} + p_{56} + p_{58} + p_{59}) + (k_{77} + p_{61})(k_{78} + p_{62}) + k_{47} + k_{49} + k_{50} + k_{64} + k_{66} + k_{67} + k_{72} + k_{74} + k_{75} + k_{76} + k_{78} + k_{79} + p_{48} + p_{50} + p_{51} + p_{56} + p_{58} + p_{59} + p_{60} + p_{62} + p_{63}$.

The correctness of these expressions can also be easily checked manually from the description of PRESENT.

3.3. Algebraic expressions of the output of round 3-6

The algebraic expressions after round 3 are quite quite complicated.We give only the simplest one,i.e. b_0^3,the least significant bit of the output of the 3rd round of PRESENT-80. First we give the unexpanded form of b_0^3.

Since b_0^3=$\left(b_1^2 + \kappa_1\right)\left(b_2^2 + \kappa_2\right) + b_0^2 + b_2^2 + b_3^2 + \kappa_0 + \kappa_2 + \kappa_3$, where $\kappa_3\kappa_2\kappa_1\kappa_0$ are the 4 least significant bits of the third round subkey.From the key schedule algorithm,we have

$\kappa_3 = k_{57}, \kappa_2 = k_{56}, \kappa_1 = k_{55}, \kappa_0 = k_{54} + 1$.

Replace $b_3^2, b_2^2, b_1^2, b_0^2$ with their expressions in section 3.2 and we can obtain the unexpanded form of b_0^3,which is

b_0^3=$b_0^3(k_{16}, k_{17}, ..., k_{78}, k_{79}, p_0, p_1, ..., p_{62}, p_{63})$

$= k_{16}+k_{18}+k_{19}+k_{24}+k_{26}+k_{27}+k_{28}+k_{30}+k_{31}+k_{35}+k_{37}+k_{38}+k_{43}+k_{45}+k_{46}+k_{47}+k_{48}+k_{49}+k_{51}+k_{54}+k_{57}+k_{58}+k_{59}+k_{60}+k_{62}+k_{63}+k_{64}+k_{66}+k_{67}+k_{72}+k_{74}+k_{75}+k_{76}+k_{78}+k_{79}+p_0+p_2+(k_{17}+p_1)(k_{18}+p_2)+p_3+p_8+p_{10}+(k_{25}+p_9)(k_{26}+p_{10})+p_{11}+\Big(k_{20}+k_{22}+k_{23}+k_{36}+p_4+p_6+(k_{21}+p_5)(k_{22}+p_6)+p_7\Big)\Big(k_{24}+k_{26}+k_{27}+k_{37}+p_8+p_{10}+(k_{25}+p_9)(k_{26}+p_{10})+p_{11}\Big)+p_{12}+p_{14}+(k_{29}+p_{13})(k_{30}+p_{14})+p_{15}+p_{32}+p_{34}+(k_{49}+p_{33})(k_{50}+p_{34})+p_{35}+p_{40}+p_{42}+(k_{57}+p_{41})(k_{58}+p_{42})+p_{43}+\Big(k_{44}+k_{52}+k_{54}+k_{55}+p_{36}+p_{38}+(k_{53}+p_{37})(k_{54}+p_{38})+p_{39}\Big)\Big(k_{45}+k_{56}+k_{58}+k_{59}+p_{40}+p_{42}+(k_{57}+p_{41})(k_{58}+p_{42})+p_{43}\Big)+p_{44}+p_{46}+(k_{61}+p_{45})(k_{62}+p_{46})+p_{47}+\Big(k_{32}+k_{34}+k_{35}+k_{39}+k_{40}+k_{41}+k_{43}+k_{44}+k_{46}+k_{47}+k_{55}+p_{16}+p_{18}+(k_{33}+p_{17})(k_{34}+p_{18})+p_{19}+p_{24}+p_{26}+(k_{41}+p_{25})(k_{42}+p_{26})+p_{27}+\Big[k_{36}+k_{38}+k_{39}+k_{40}+p_{20}+p_{22}+(k_{37}+p_{21})(k_{38}+p_{22})+p_{23}\Big]\Big[k_{40}+k_{41}+k_{42}+k_{43}+p_{24}+p_{26}+(k_{41}+p_{25})(k_{42}+p_{26})+p_{27}\Big]+p_{28}+p_{30}+(k_{45}+p_{29})(k_{46}+p_{30})+p_{31}\Big)\Big(k_{43}+k_{45}+k_{46}+k_{48}+k_{50}+k_{51}+k_{58}+k_{59}+k_{60}+k_{62}+k_{63}+p_{32}+p_{34}+(k_{49}+p_{33})(k_{50}+p_{34})+p_{35}+p_{40}+p_{42}+(k_{57}+p_{41})(k_{58}+p_{42})+p_{43}+\Big[k_{44}+k_{52}+k_{54}+k_{55}+p_{36}+p_{38}+(k_{53}+p_{37})(k_{54}+p_{38})+p_{39}\Big]\Big[k_{45}+k_{56}+k_{58}+k_{59}+p_{40}+p_{42}+(k_{57}+p_{41})(k_{58}+p_{42})+p_{43}\Big]+p_{44}+p_{46}+(k_{61}+p_{45})(k_{62}+p_{46})+p_{47}\Big)+p_{48}+p_{50}+(k_{65}+p_{49})(k_{66}+p_{50})+p_{51}+p_{56}+$

$$p_{58}+(k_{73}+p_{57})(k_{74}+p_{58})+p_{59}+\Big(k_{48}+k_{68}+k_{70}+k_{71}+p_{52}+p_{54}+(k_{69}+p_{53})(k_{70}+p_{54})+p_{55}\Big)\Big(k_{49}+$$

$$k_{72}+k_{74}+k_{75}+p_{56}+p_{58}+(k_{73}+p_{57})(k_{74}+p_{58})+p_{59}\Big)+p_{60}+p_{62}+(k_{77}+p_{61})(k_{78}+p_{62})+p_{63}+1.$$

Then we use the Mathematica command

$$b_0^3\text{=(Expand}[b_0^3]\text{/.Power}[a\_, b\_]- > a)/.a\_\text{Integer}- >\text{Mod}[a, 2]$$

to expand the above expression.The expanded expression contains terms like x^2yz^3,which is replaced with xyz after the command /.Power$[a_, b_]- > a$.This is a rule-based command in Mathematica,which replaces each term of the form a^b with a. The Mathematica command $f/.a\_\text{Integer}- >\text{Mod}[a, 2]$ replace each integer in the expression of f with the integer mod2.With these computation,we can obtain the algebraic normal form of b_0^3.The number of terms of b_0^3 is 23443.So we can not list its ANF here. The degree of b_0^3 is 8.Of the 23443 terms of b_0^3,there are 256 terms of degree of 8, 1792 terms of degree 7, 5072 terms of degree 6, 7392 terms of degree 5, 5850 terms of degree 4, 2470 terms of degree 3, 550 terms of degree 2, 60 terms of degree 1 and the constant term 1.

For the other bits of the output of round 3,we list only the number of key bits and plaintext bits here.Each bit contains 64 plaintext bits.The number of key bits contained in the expressions from the most significant bit to the least significant bit are as follows:77, 67, 64, 64, 77, 69, 68, 68, 78, 67, 64, 64, 77, 67, 64, 64, 77, 67, 64, 64, 77, 69, 68, 68, 78, 67, 64, 64, 77, 67, 64, 64, 77, 67, 64, 64, 77, 69, 68, 68, 78, 67, 64, 64, 77, 67, 64, 64, 77, 67, 64, 64, 77, 69, 68, 68, 78, 67, 64, 64, 77, 67, 64, 64.

This shows that PRESENT-80 is not complete after 3 rounds.

Each algebraic expression of the bits of the output of round 4-6 contains all the 64 plaintext bits and 80 key bits.It's too complicated to be listed here.

3.4. The correctness test of the algebraic expressions of PRESENT-80

The correctness of the algebraic expressions of the output of the first two rounds in section 3.1 and 3.2 can be checked manually.For the expressions of the output after round 3,it's almost impossible to check it manually. The correctness of these expressions are guaranteed by the correctness of the symbolic encryption code on which we have made a correctness test using the test vector in [2] .Moreover,we have also made the following correctness test.

Let's take b_0^3(see section 3.3) as an example.Do the following test 100 times:

(1)Random choose plaintext and key,encrypt the plaintext with the key,and output the value of b_0^3,denoted by A.

(2)Substitute the above chosen plaintext and key for the corresponding plaintext variables and key variables in the algebraic expression of b_0^3 in section 3.3.Denote the value we obtain now by B.If $A = B$,output *Yes*,otherwise output *No*.

We have obtained 100 *Yes*.This shows the correctness of the algebraic expression of b_0^3(If the algebraic expression is not correct,then the probability we obtain *Yes* in step (2) is 1/2 and the probability of obtaining 100 *Yes* is $1/2^{100}$). The correctness of the other algebraic expressions can also be verified in just the same way.

4. Application of the algebraic expression of b_0^3– A side channel attack on PRESENT-80

In this section,we give a side channel attack on PRESENT-80 based on the algebraic expressions of b_0^3, the least significant bit of the output of the 3rd round.We choose the single bit leakage

model proposed by Dinur and Shamir in [7].In this paper we do not consider how information leakage can be measured. Rather, we assume the single-bit-leakage side channel model as an abstract attack model, and concentrate on investigating the security of the PRESENT block cipher against this attack model. So we suppose that the least significant bit of the output of the 3rd round of PRESENT can be obtained without error.

From the algebraic expression of b_0^3 in section 3.3,if the value of b_0^3 is leaked and the plaintext is known,then an equation with the 64 master key bits $k_{16}, k_{17}, ..., k_{78}, k_{79}$ as variables can be set up.Suppose we have obtained n leaked bits $v_1, ..., v_n$.The corresponding n known plaintexts are $P^i = p^i_{63}p^i_{62}...p^i_0, i = 1, ..., n$. Then we have n equations

$$v_i = b_0^3(k_{16}, k_{17}, ..., k_{78}, k_{79}, p^i_0, p^i_1, ..., p^i_{62}, p^i_{63}), i = 1, ..., n. \quad (1)$$

With enough leaked bits and known plaintexts,the equation system can be solved via SAT solver [12] or Gröbner Basis method [13].Magma [11] is one of the best software of computing Gröbner Basis.In this paper we choose the Magma online caculator v2.18-8 to solve the equation.Our experiment shows that with 200 known plaintexts,we can recover 43 bits key in an average time 70 seconds.The key bits which can not be determined in the experiments are $k_{19}, k_{23}, k_{27}, k_{28}, k_{31}, k_{35}, k_{39}, k_{47}, k_{51}, k_{52}, k_{55}, k_{59}, k_{60}, k_{63}, k_{64}, k_{67}, k_{68}, k_{71}, k_{75}, k_{76}, k_{79}$.

We do our experiment in Mathematica 7.0.The experiment proceeds as follows.

(1)Use the Mathematica command Table[RandomInteger[],{80}] to generate a random key.

(2)Use the Mathematica command Table[Table[RandomInteger[],{64}],{200}] to generate 200 known plaintexts.

(3)Encrypt each plaintext in (2) with the key in (1),denote the value of the least significant bit of round 2 by **A**.Substitute the plaintext for the corresponding plaintext variables in the algebraic expression of b_0^3 in section 3.3.Denote the algebraic expression we obtain now by **B**.Then we obtain an equation **A** = **B**.With 200 known plaintexts,we can obtain 200 equations.

(4)Output the equation system obtained in (3) as a text file,which is suitable for Magma to solve.

(5)Copy the content of the text file to Magma online calculator [11] and find the solution.

We have made the above test 100 times.Each time Magma can only recover 43 bits key in less than 120 seconds and the average time is 70 seconds.Compared with the previous side channel attack on PRESENT,such as Yang et al. [8] in CANS 2009,Abdul-Latip et al. [9] in ASIACCS 2011 and Zhao et al. [10] in 2011,each of which needs at least 2^{13} chosen plaintexts,the data complexity of our attack is the best.Moreover,ours is a known plaintexts model,which is better than the chosen plaintexts model.

We summarize our work and the known side channel attack on PRESENT in Table 3.

5. Conclusion

Using the symbolic computation technique,this paper obtains the algebraic expressions of the output of round 1-6 of PRESENT-80.As far as we know,this is the first time,the explicit algebraic expressions of a block cipher up to 6 rounds are obtained.As an application of these expression,we give a side channel attack on PRESENT-80 under the single bit leakage model.These expressions can also be used in other leakage model.

```
sbox[{x3_, x2_, x1_, x0_}] := {1 + x0 + x1 + x1 x2 + x0 x1 x2 + x3 + x0 x1 x3 + x0 x2 x3,
   1 + x0 x1 + x2 + x3 + x0 x3 + x1 x3 + x0 x1 x3 + x0 x2 x3, x1 + x0 x1 x2 + x3 + x1 x3 + x0 x1 x3 +
    x2 x3 + x0 x2 x3, x0 + x2 + x1 x2 + x3};
sBoxLayer[state_] := Table[sbox[Partition[state, 4][[i]]], {i, 1, 16}] // Flatten;
keySchedule[masterKey_, round_, symbolFlag_] := Module[{r, roundKey, keyRegister},
   keyRegister = masterKey;
   roundKey = Table[0, {round + 1}]; roundKey[[1]] = masterKey[[1 ;; 64]];
   For[r = 1, r ≤ round, r++, keyRegister = RotateLeft[keyRegister, 61];
    keyRegister[[1 ;; 4]] = sbox[keyRegister[[1 ;; 4]]];
    keyRegister[[61 ;; 65]] = keyRegister[[61 ;; 65]] + IntegerDigits[r, 2, 5];
    If[symbolFlag == 0, keyRegister = Mod[keyRegister, 2]];
    roundKey[[r + 1]] = keyRegister[[1 ;; 64]]];
   Return[roundKey]];
encryption[masterKey_, plainText_, round_, outputMiddleFlag_, symbolFlag_] :=
  Module[{roundKey, state, i}, state = plainText;
   If[symbolFlag == 1, roundKey = keySchedule[masterKey, round, 1],
    roundKey = keySchedule[masterKey, round, 0]];
   For[i = 1, i ≤ round, i++, state = state + roundKey[[i]];
    state = sBoxLayer[state]; If[symbolFlag == 0, state = Mod[state, 2]];
    state = state[[{1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61, 2, 6, 10, 14,
       18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43,
       47, 51, 55, 59, 63, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64}]];
   If[outputMiddleFlag == 1, Return[state]];
   state = state + roundKey[[round + 1]]; If[symbolFlag == 0, state = Mod[state, 2]];
   Return[state]];
key = Table[ToExpression["k" <> ToString[i]], {i, 79, 0, -1}];
pText = Table[ToExpression["p" <> ToString[i]], {i, 63, 0, -1}];
outPutOfRound1 = encryption[key, pText, 1, 1, 1] //. a_Integer → Mod[a, 2];
outPutOfRound2 = encryption[key, pText, 2, 1, 1] //. a_Integer → Mod[a, 2];
outPutOfRound3 = encryption[key, pText, 3, 1, 1] //. a_Integer → Mod[a, 2];
outPutOfRound4 = encryption[key, pText, 4, 1, 1] //. a_Integer → Mod[a, 2];
outPutOfRound5 = encryption[key, pText, 5, 1, 1] //. a_Integer → Mod[a, 2];
outPutOfRound6 = encryption[key, pText, 6, 1, 1] //. a_Integer → Mod[a, 2];
```

Figure A.3: Mathematica Symbolic encryption code of PRESENT-80.By running the above code in Mathematica we can obtain the algebraic expressions of the output of round 1-6 of PRESENT-80. Note:The outputMiddleFlag in the encryption function is used to test whether to output the middle state. The symbolFlag is used to test whether we do a symbol encryption or a numerical encryption.

Table 3: **Summary of side channel attack on PRESENT**

PRESENT	Source	Leakage model	Data Complexity	Recovered Key
PRESENT-80	[8]	3rd round single bit leakage	2^{15} CP	48 bits
PRESENT-80	[9]	Hamming weight leakage after the 1st round	2^{13} CP	64 bits
PRESENT-80	this paper	3rd round single bit leakage	200 KP	43 bits
PRESENT-80	[10]	3rd round single bit leakage	2^{12} CP	48 bits
PRESENT-80	[10]	4th round single bit leakage	2^{15} CP	72 bits
PRESENT-128	[9]	Hamming weight leakage after the 1st round	2^{13} CP	64 bits
PRESENT-128	[10]	4th round single bit leakage	2^{15} CP	85 bits

Note:CP:Chosen Plaintext;KP:Known Plaintext;

Appendix A. Symbolic encryption code of PRESENT-80 in Mathematica 7.0

[1] ISO/IEC 29192-2, Information technology-Security techniques-Lightweight cryptography-Part 2: Block ciphers, January 2012.

[2] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, CHES, volume 4727 of Lecture Notes in Computer Science, pages 450-466. Springer, 2007.

[3] Ingrid Schaumuller-Bichl, Cryptanalysis of the Data Encryption Standard by the Method of Formal Coding, Lecture Notes in Computer Science, Cryptography, proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March 29,April 2, pp. 235-255, 1982.

[4] M.Kwan,Reducing the Gate Count of Bitslice DES,http://eprint.iacr.org/2000/051

[5] Hakan Englund,Thomas Johansson,Meltem Sonmez Turan,A Framework for Chosen IV Statistical Analysis of Stream Ciphers,INDOCRYPT 2007,pp.268-281.See also Tools for Cryptoanalysis 2007.

[6] http://www.wolfram.com/mathematica/

[7] Dinur, I., Shamir, A.: Side Channel Cube Attacks on Block Ciphers. Cryptology ePrint Archive, Report 2009/127 (2009), http://eprint.iacr.org/2009/127

[8] Yang, L., Wang, M., Qiao, S.: Side Channel Cube Attack on PRESENT. In: Garay, J.A., Miyaji, A., Otsuka, A. (Eds.) CANS 2009. LNCS, vol. 5888, pp. 379-391. Springer, Heidelberg (2009)

[9] 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. In Bruce Cheung, Lucas Chi Kwong Hui, Ravi Sandhu, Duncan S.Wong (Eds.): ASIACCS 2011, pp.296-305,2011.

[10] XinJie Zhao, Tao Wang and ShiZe Guo: Improved Side Channel Cube Attacks on PRESENT. Cryptology ePrint Archive, Report 2011/165 (2011), http://eprint.iacr.org/2011/165

[11] http://magma.maths.usyd.edu.au/calc/

[12] G. Bard, N. Courtois, C. Jefferson, Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers, Cryptology ePrint Archive, Report 2007/024.

[13] G. Ars, J.-C. Faugre, H. Imai, M. Kawazoe, M. Sugita, Comparison Between XL and Gröbner Basis Algorithms, in the proceedings of ASIACRYPT 2004, LNCS, vol 3329, pp 338-353, Jeju Island, Korea, December 2004.