PRESENT

Chetan[1], Sumanth Guptha[2] and Yuvraj Mandrah[3]

[1] 12242070, IIT Bhilai yuvrajm@iitbhilai.ac.in
[2] 12241070, IIT Bhilai murukuri@iitbhilai.ac.in
[3] 12240470, IIT Bhilai chetan@iitbhilai.ac.in

[Tho10]

Abstract. In this paper, the Present Cipher is described along with its general architecture, and the reasons for the selected key length are explained. It considers two low-level cryptanalysis techniques—Differential Cryptanalysis and Linear Cryptanalysis—and performs analysis on a round-reduced substitute. Moreover, an interesting experiment highlights that the running time of the cipher does not depend on the number of high bits in the key.

Keywords: Lightweight Cipher · Differential Cryptanalysis · Linear Cryptanalysis

1 Introduction

Algorithms such as the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES) are two of the most studied in the field of cryptanalysis. Although DES has limitations with key length, both algorithms have demonstrated resilience against various cryptanalytic attacks. The Present cipher, like AES and DES, is a block cipher, but its design takes a different approach. Despite structural similarities with AES, Present belongs to the lightweight cipher family, optimized for hardware performance in low-power devices while maintaining a satisfactory security level. Recognized as the ISO/IEC 29192-2:2019 standard, Present is commonly used in low-power applications like RFID cards and IoT nodes. However, like AES, reduced-round versions of the Present cipher are vulnerable to cryptanalysis techniques such as Linear and Differential Cryptanalysis.

2 Contributions

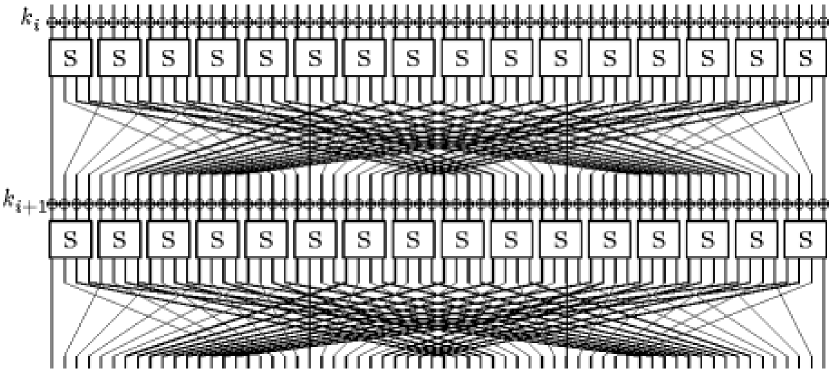
- **Yuvraj:** Yuvraj thoroughly analyzed the S-box of the PRESENT cipher, studying its structure and cryptographic properties. He then implemented a theoretical linear cryptanalysis attack by constructing a Linear Approximation Table (LAT), uncovering biases in the cipher's non-linearity. This demonstrated potential vulnerabilities in the cipher's design and highlighted weaknesses in its security.

- **Chetan:** Chetan implemented a reduced-round differential attack on the PRESENT cipher, focusing on identifying differential characteristics that maximize the number of active S-boxes. By analyzing the cipher's behavior under limited rounds, he evaluated its resistance to such attacks, providing insights into the cipher's security under reduced-round conditions.

- **Sumanth Guptha:** Sumanth conducted an integral attack on the PRESENT cipher, analyzing how certain input patterns lead to specific output structures across multiple rounds. By exploiting the properties of the cipher's transformations, he aimed to uncover information about the key, assessing the cipher's resistance to this form of cryptanalysis.

3 The PRESENT Cipher[Tho10][WW13]

The Present cipher is an ultra-lightweight block cipher with a block length of 64 bits and key sizes of 80 or 128 bits. This paper focuses on the 80-bit version, which is sufficient for applications like RFID tags and IoT security. The cipher features a public S-box, bit permutation, and key schedule.



[Tho10]

Figure 1: S-box structure of the Present cipher.

3.1 Cipher Design

[WW13] This present cipher is an instance of an SP-network that contains 31 rounds labelled as Present-80. It turns into a XORing with the round key, an S-box closing the system of 4 bits and an amount closing the system of bits in the word. In each round, S-Box of 4-bit is used 16 times for 64 bits of the input in parallel processing. Figure 2a shows the higher level of the pseudo code to perform the encryption algorithm On the other hand, figure 2b shows a high level view of the encryption process.

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

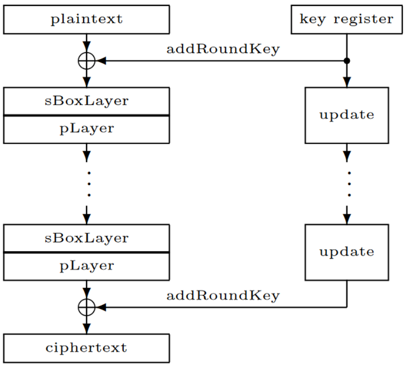


Figure 2: Algorithm description of Present

3.2 Add Round Key

Provided the round key $K_i = k_{63}, k_{62}, \ldots, k_0$ for $1 \leq i \leq 32$ and the current state $S = s_{63}, s_{62}, \ldots, s_0$, the The Provided the round key $K_i = k_{63}, k_{62}, \ldots, k_0$ for $1 \leq i \leq 32$ and the current state $S = s_{63}, s_{62}, \ldots, s_0$, the `addRoundKey` operation performs the following transformation:

$$S \to S \oplus K_i$$

$$\implies s_t \to s_t \oplus k_t$$

for $0 \leq t \leq 63$.

3.3 Substitution Layer

The substitution box (S-Box) in the PRESENT cipher is a 4-bit to 4-bit mapping. Table 1 illustrates the mapping of the S-Box in hexadecimal notation. The 4-bit S-Box is applied independently 16 times to cover the 64-bit block. The substitution box is a mapping $S : \mathbb{F}_2^4 \to \mathbb{F}_2^4$, where $\mathbb{F}$ denotes a finite field.

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

Table 1: Present S-Box

To improve the avalanche effect, the PRESENT S-Box satisfies the following conditions. Let the Fourier coefficient of the S-Box be denoted by $S_W b(a)$, defined as:

$$S_W b(a) = \sum_{x \in \mathbb{F}_2^4} (-1)^{\langle h_b, S(x) \rangle + \langle h_a, x \rangle}$$

where $\langle \cdot, \cdot \rangle$ is the dot product over $\mathbb{F}_2$. The conditions for the S-Box are as follows:

1. For any fixed input difference $\Delta I \in \mathbb{F}_2^4$, $\Delta I \neq 0$, and output difference $\Delta O \in \mathbb{F}_2^4$, $\Delta O \neq 0$, the following condition must hold:

$$\left| \{ x \in \mathbb{F}_2^4 \mid S(\Delta I + x) + S(x) = \Delta O \} \right| \leq 4.$$

2. For any fixed input difference $\Delta I \in \mathbb{F}_2^4$, $\Delta I \neq 0$, and output difference $\Delta O \in \mathbb{F}_2^4$ such that $\mathrm{wt}(\Delta O) = \mathrm{wt}(\Delta I) = 1$, the following condition must hold:

$$\{ x \in \mathbb{F}_2^4 \mid S(\Delta I + x) + S(x) = \Delta O \} = \emptyset,$$

where $\mathrm{wt}(x)$ denotes the Hamming weight of x.

3. For all $a \in \mathbb{F}_2^4$, $a \neq 0$, and $b \in \mathbb{F}_4$, the following condition must hold:

$$|S_W b(a)| \leq 8.$$

4. For all $a \in \mathbb{F}_2^4$, $a \neq 0$, and $b \in \mathbb{F}_4$ such that $\mathrm{wt}(b) = \mathrm{wt}(a) = 1$, the following condition holds:

$$S_W b(a) = \pm 4.$$

3.4 Permutation Layer

At the layer level, the permutations that occur are done at the bit level. The transformation function is $P(i)$: that maps the i-th bit of the input within the output array. Table 2 shows the mapping of $P(i)$ in tabular form and Fig. 5 illustrates the concrete arrangement of Add Round Key, Substitution Layer, and Permutation Layer in a single round of encryption.

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
Q P(i)	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

3.5 Key Schedule Algorithm

The Present cipher supports 80-bit or 128-bit long key, in this section we discuss the 80-bit key schedule algorithm.

1. The initial 80-bit key is stored in a key register K. Which is represented as $K = k_{79}k_{78}\dots k_0$.
   Round i : extracts 64-bit round key , i.e $K_i = k_{63}k_{62}\dots k_0$ from the current key. (left most 64 bits)
   $$[k_{79}k_{78}\dots k_0] = [k_{18}k_{17}\dots k_{20}k_{19}].$$

Now Key K is updated as:

1. the key register is rotated by 61 bit positions to the left
   $$[k_{79}k_{780}] = [k_{18}k_{1720}k_{19}].$$
2. The 4 left-most bits of the key register K is passed with the PRESENT S-Box:
   $$[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}].$$
3. The 5 bits of the key register is XORed with the least significant bits of the round counter value i:
   $$[k_{19}k_{18}k_{17}k_{16}k_{15}] = [k_{19}k_{18}k_{17}k_{16}k_{15}] \oplus \text{round-counter}.$$

4 Security Analysis

Here are the results of security analysis PRESENT

S - Box

[WW13]

We use a single 4-bit to 4-bit S-box $S : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ in Present. This is a direct consequence of our pursuit of hardware efficiency, with the implementation of such an S-box typically being much more compact than that of an 8-bit S-box. Since we use a bit permutation for the linear diffusion layer, AES-like diffusion techniques [WW13] are not an option for Present. Therefore, we place some additional conditions on the S-boxes to improve the so-called *avalanche of change*. More precisely, the S-box for Present fulfills the following conditions, where we denote the Fourier coefficient of S by

$$S_b^W(a) = \sum_{x \in \mathbb{F}_2^4} (-1)^{\langle b, S(x) \rangle + \langle a, x \rangle}.$$

1. For any fixed non-zero input difference $\Delta_I \in \mathbb{F}_2^4$ and any fixed non-zero output difference $\Delta_O \in \mathbb{F}_2^4$ we require
$$\#\{x \in \mathbb{F}_2^4 \mid S(x) + S(x + \Delta_I) = \Delta_O\} \leq 4.$$
2. For any fixed non-zero input difference $\Delta_I \in \mathbb{F}_2^4$ and any fixed output difference $\Delta_O \in \mathbb{F}_2^4$ such that $\mathrm{wt}(\Delta_I) = \mathrm{wt}(\Delta_O) = 1$, we have
$$\{x \in \mathbb{F}_2^4 \mid S(x) + S(x + \Delta_I) = \Delta_O\} = \emptyset.$$
3. For all non-zero $a \in \mathbb{F}_2^4$ and all non-zero $b \in \mathbb{F}_2^4$, it holds that $|S_b^W(a)| \leq 8$.
4. For all $a \in \mathbb{F}_2^4$ and all non-zero $b \in \mathbb{F}_2^4$ such that $\mathrm{wt}(a) = \mathrm{wt}(b) = 1$, it holds that $S_b^W(a) = \pm 4$.
5. For all $a \in \mathbb{F}_2^4$ and all non-zero $b \in \mathbb{F}_2^4$ such that $\mathrm{wt}(a) = \mathrm{wt}(b) = 1$, it holds that
$$S_b^W(a) = \pm 4.$$

Notable-Properties

Any five-round differential characteristic of present has a minimum of 10 active S-boxes.

[Cho10] Let ϵ_{4R} be the maximal bias of a linear approximation of four rounds of PRESENT. Then $\epsilon_{4R} \leq \frac{1}{2^7}$.

The non linear boolean functions for the sbox are:

$$\begin{aligned}
y_3 &= 1 \oplus x_0 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3,\\
y_2 &= 1 \oplus x_2 \oplus x_3 \oplus x_0x_1 \oplus x_0x_3 \oplus x_1x_3 \oplus x_0x_1x_3 \oplus x_0x_2x_3,\\
y_1 &= x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_0x_1x_2 \oplus x_0x_1x_3 \oplus x_0x_2x_3,\\
y_0 &= x_0 \oplus x_2 \oplus x_3 \oplus x_1x_2.
\end{aligned}$$

[BKL+07]

4.1 Differential Cryptanalysis

[Cho10] [Wan08] [TK09] [Nov12] [Nik17] In this section, we present an actual differential attack against reduced-round PRESENT. We use $X = x_0, x_1, \ldots, x_{15}$ to denote the XOR difference of the 16 nibbles in each step, with x_0 being the least significant. Also, we denote K_i as the subkey for round i. round.

The Difference Distribution table (DDT) of S-box in Table 3 We now make some important observations, by seeing the properties of the S-box and permutation layer. We divide the 16 S-box(s) into 4 sets (One set is shown below). We can observe the following properties :

1. The inputs of the S-box is from 4 separate S-boxes of the same set.
2. The input bits to a group of four S-boxes come from 16 different S-boxes.
3. The output from an S-box go into 4 distinct S-boxes, each of which belongs to a distinct set of S-boxes in the next round
4. The output of S-boxes from different sets go to different S-boxes. Now, form the above observations and the DDT, we conclude that one bit input difference will cause at least two bits output difference, resulting in at least two active S-boxes in the next round and the maximum differential probability of the DDT is 2^{-2}

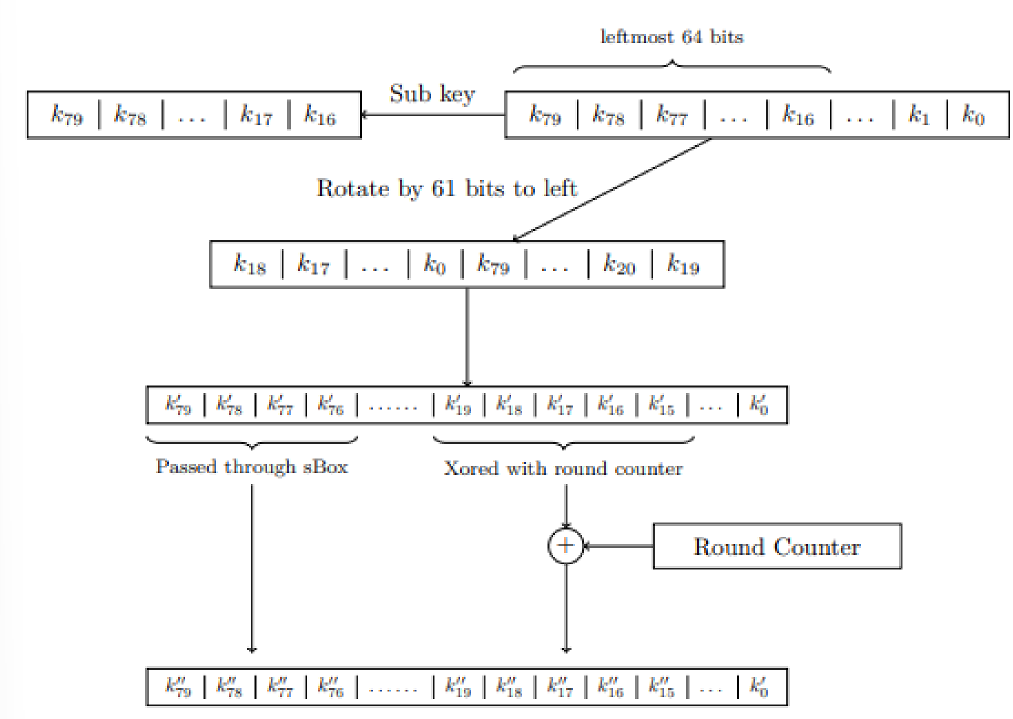

leftmost 64 bits
Sub key
k_{79} | k_{78} | ... | k_{17} | k_{16}
k_{79} | k_{78} | k_{77} | ... | k_{16} | ... | k_1 | k_0
Rotate by 61 bits to left
k_{18} | k_{17} | ... | k_0 | k_{79} | ... | k_{20} | k_{19}
k'_{79} | k'_{78} | k'_{77} | k'_{76} | ...... | k'_{19} | k'_{18} | k'_{17} | k'_{16} | k'_{15} | ... | k'_0
Passed through sBox
Xored with round counter
Round Counter
k''_{79} | k''_{78} | k''_{77} | k''_{76} | ...... | k''_{19} | k''_{18} | k''_{17} | k''_{16} | k''_{15} | ... | k'_0

Figure 3: Key schedule

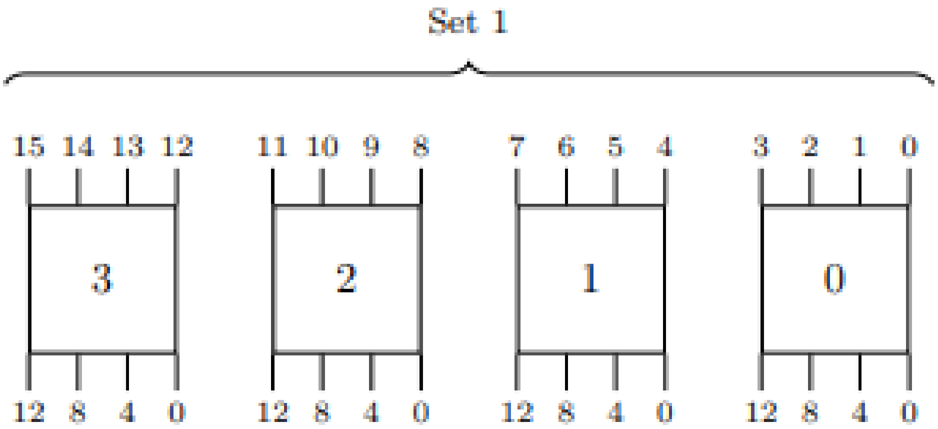


Figure 4: Figure 5

Table 1: Differential Cryptanalysis Table

X	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

4.2 Differential Characteristics

1. . We first look for 4-round iterative characteristics. The maximum number of active Number of S-boxes for 2-round to 4-round is 4, 7 and 9 and the number of possible(sprintf number of S-boxes used in each round actively is described in the table 4. According to this observation

Table 2: Possible number of active S-boxes for each round

Rounds	2	3	4
Possible number of active S-boxes	2-2	2-2-2 3-2-2 2-3-2 2-2-3	2-2-2-2 3-2-2 2-3-2 2-2-3 2-2-2-3

4-round iterative characteristics with probability 2^{-18} have been found, one of which is given in Table 5.

Table 3: Differential Probability and Variables for Each Round

Rounds	Diff. Prob.	Variables	Values
I		x0 = 4, x4 = 4	
R1	S	x0 = 5, x3 = 5	2−4
		x0 = 9, x8 = 9	1
R2	S	x0 = 4, x8 = 4	2−4
		x8 = 1, x10 = 1	1
R3	S	x8 = 9, x10 = 9	2−4
		x2 = 5, x14 = 5	1
R4	S	x2 = 1, x14 = 1	2−6
		x0 = 4, x4 = 4	1

2. We then search for iterative differential characteristics spanning 5 to 10 rounds. Using the 4-round iterative characteristics shown above, we extend this to derive 11-round to 15-round differential characteristics. The probability of the best characteristics we identified is listed in Table 6. A notable observation from Table 6 is that the number of active S-boxes increases by 2 in each round. By continuing this process, we obtained 24 distinct 14-round differential characteristics, all with the same output difference, but with different input differences. The probability of these characteristics is 2−62, as shown in Table 6. One such 14-round characteristic is depicted in Table 7

Table 4: Differential Probability and Number of Active S-boxes for Different Rounds

Rounds	**Differential Prob.**	**Number of Active S-boxes**
5	2^{-20}	10
6	2^{-24}	12
7	2^{-28}	14
8	2^{-32}	16
9	2^{-36}	18
10	2^{-42}	20
11	2^{-46}	22
12	2^{-52}	24
13	2^{-56}	26
14	2^{-62}	28
15	2^{-66}	30

Table 5: 14-round differential characteristic

Rounds	Differential Prob.	Active S-boxes
I	$x_2 = 7, x_{14} = 7$	
R1 S	$x_2 = 1, x_{14} = 1$	2
R1 P	$x_0 = 4, x_3 = 4$	1
R2 S	$x_0 = 5, x_3 = 5$	2
R2 P	$x_0 = 9, x_8 = 9$	1
R3 S	$x_0 = 4, x_8 = 4$	2
R3 P	$x_8 = 1, x_{10} = 1$	1
R4 S	$x_8 = 9, x_{10} = 9$	2
R4 P	$x_2 = 5, x_{14} = 5$	1
R5 S	$x_2 = 1, x_{14} = 1$	2
R5 P	$x_0 = 4, x_3 = 4$	1
R6 S	$x_0 = 5, x_3 = 5$	2
R6 P	$x_0 = 9, x_8 = 9$	1
R7 S	$x_0 = 4, x_8 = 4$	2
R7 P	$x_8 = 1, x_{10} = 1$	1
R8 S	$x_8 = 9, x_{10} = 9$	2
R8 P	$x_2 = 5, x_{14} = 5$	1
R9 S	$x_2 = 1, x_{14} = 1$	2
R9 P	$x_0 = 4, x_3 = 4$	1
R10 S	$x_0 = 5, x_3 = 5$	2
R10 P	$x_0 = 9, x_8 = 9$	1
R11 S	$x_0 = 4, x_8 = 4$	2
R11 P	$x_8 = 1, x_{10} = 1$	1
R12 S	$x_8 = 9, x_{10} = 9$	2
R12 P	$x_2 = 5, x_{14} = 5$	1
R13 S	$x_2 = 1, x_{14} = 1$	2
R13 P	$x_0 = 4, x_3 = 4$	1
R14 S	$x_0 = 5, x_3 = 5$	2
R14 P	$x_0 = 9, x_8 = 9$	1

4.3 Attack

In this section, we detail how the 16-round reduced PRESENT cipher is attacked. The attack is conceived with 2^{24} chosen plaintexts, where each plaintext structure is a collection of them. In the case of an 8-round PRESENT, we need 2^{40} such structures.

The differential characteristics of 14 rounds that we found above involve 2 active S-boxes in the first round at positions x_0, x_1, and $(x_2, x_{12}, x_{13}, x_{14})$. The other S-boxes in the first round do not function.

The construction of the chosen plaintexts is as follows: The 8-bit inputs to the active S-boxes take all possible values and therefore have 2^8 possible combinations. The inputs to the non-active S-boxes are assigned random values.

For each possible characteristic, the total number of structures is given by:

$$2^{40} \times 2^{16} \times 2^7 = 2^{63}$$

The probability of the desired characteristic is 2^{-32}, so the number of valid pairs satisfying one of the characteristics is:

$$2^{63} \times 2^{-62} \times 24 = 48$$

Next, we calculate the total number of pairs to be considered. In each structure, I can have

$\frac{2^{24}\times 2^{24}}{2}=2^{47}$ possible pairs. Therefore, the total number of pairs across all structures is:

$$2^{40}\times 2^{47}=2^{87}$$

As filters from the previous sections are used in the filtering procedures, we filter out incorrect field pairs in connection with Rounds 15 and 16 to focus only on the significant details. In the initial differential characteristics, we note that in Round 14 both S-boxes are active, with differences of 9 for x_0 and x_8. Thus, the input difference for Round 15 is 9 for the 0th and 8th nibbles.

From the DDT, we learn that an input difference of 9 results in an output difference of $\{2,4,6,8,12,14\}$, where the least significant bit is 0. This means that at most 6 bits of the output difference are active, which translates to 6 active S-boxes in Round 16 at positions $x_4,x_6,x_8,x_{10},x_{12},x_{14}$. We also conclude that at least 2 S-boxes must be active, based on the observations from the S-box and DDT.

In order for a pair to be labeled as the correct pair, there must be at least 10 non-active S-boxes in round 16. This means that by discarding the incorrect pairs, we are left with:

$$2^{47}\times 2^{-40}=2^{7}$$

identified possible pairs for the correct pairs from each structure. Thus, the total number of candidate pairs that can be chosen is:

$$2^{7}\times 2^{40}=2^{47}$$

Now we have the lower bound of active and non-active S-boxes in Round 16 as 2 and 10, respectively. The remaining 4 S-boxes can be active or non-active, as active S-boxes are a subset of the total S-boxes. If they are active, the input difference must be 1, and from the DDT, the output difference can only be 3, 7, 9, or 13. Using this as a filter to discard incorrect pairs, we are left with a fraction of:

$$\frac{5}{16}\times 6=6\times 2^{-10.07}$$

Thus, the number of candidates for the correct pair from each structure is:

$$2^{7}\times 2^{-10.07}=2^{-3.07}$$

This implies that the total number of candidates for the correct pair across all structures is:

$$2^{40}\times 2^{-3.07}=2^{36.93}$$

Secondly, each of the pairs is tested to check whether either of the two members meets one of the 24 differential characteristics that we discovered. Since there are about 2^{24} possible input differences, only a fraction of:

$$2^{-24}\times 20=2^{-19.68}$$

pairs remain. Therefore, the expected number of remaining pairs across all structures is:

$$2^{36.93}\times 2^{-19.68}=2^{17.25}$$

Guessing The Key

During decryption from Round 16 to Round 14 we make assumption of the bits of the subkey that is related to the active S-boxes. In detail, we expect 8 bits of K_{16} and, depending on the activity of S-boxes, from 0 to 24 bits of K_{17}. The 24 bits of subkey K_{17} are not related to the 8 least significant bits of K_{16}, therefore, for decryption of text from Round 16 to Round 14, at least 32 bits have to be guessed.

Since the biggest differential probability resulting from the DDT is 2^{-2}, the amount of subkey nibble guesses made on average per active S-box is 4. Let n denote the amount of active S-boxes in Round 16 with $2 \leq n \leq 6$. Depending on the value of n, we have 5 different cases to consider:

1. $n = 2$: The number of ciphertext pairs satisfying 2 active S-boxes is:

$$2^{17.25} \times 2^{-16} = 2^{1.25}$$

The total number of times subkeys are counted for the remaining pairs is:

$$2^{1.25} \times 4^{4} = 2^{9.25}$$

2. $n = 3$: The number of ciphertext pairs satisfying 3 active S-boxes is:

$$2^{17.25} \times (2^{-12} - 2^{-16}) = 2^{5.16}$$

The total number of times subkeys are counted for the remaining pairs is:

$$2^{5.16} \times 4^{5} = 2^{15.16}$$

3. $n = 4$: The number of ciphertext pairs satisfying 4 active S-boxes is:

$$2^{17.25} \times (2^{-8} - 2^{-12}) = 2^{9.16}$$

The total number of times subkeys are counted for the remaining pairs is:

$$2^{9.16} \times 4^{6} = 2^{21.16}$$

4. $n = 5$: The number of ciphertext pairs satisfying 5 active S-boxes is:

$$2^{17.25} \times (2^{-4} - 2^{-8}) = 2^{13.16}$$

The total number of times subkeys are counted for the remaining pairs is:

$$2^{13.16} \times 4^{7} = 2^{27.16}$$

5. $n = 6$: The number of ciphertext pairs satisfying 6 active S-boxes is:

$$2^{17.25} \times (1 - 2^{-4}) = 2^{17.16}$$

The total number of times subkeys are counted for the remaining pairs is:

$$2^{17.16} \times 4^{8} = 2^{33.16}$$

Thus, the total number of times the subkeys are counted is:

$$2^{9.25} + 2^{15.16} + 2^{21.16} + 2^{27.16} + 2^{33.16} = 2^{33.18}.$$

Hence, the average number of incorrect subkey hits is:

$$\frac{2^{33.18}}{2^{32}} = 2^{1.18},$$

which is approximately 2.27 times, compared to 48 times for the correct subkey count in the right pair. Therefore, the correct subkey can be easily identified.

To retrieve the 32 bits of information about the subkeys, we need at most $2^{33.18}$ 2-round PRESENT encryptions and 2^{32} 6-bit counters. Then, by exhaustively searching the remaining 48 bits, we can find the master key with a time complexity of 2^{48} 16-round PRESENT encryptions.

To reduce the complexity and analysis time, we follow the algorithm below:

1. For each structure:

1. The ciphertext must be added to a hash table.
2. Check for collisions. In case of a collision, verify if the resulting plaintext difference matches the characteristics.
3. Ensure that the 24 bits' difference is solely due to the output difference of the characteristics.
4. For each possible subkey of K_{17}, decrypt the rounds until the output difference of the two active S-boxes in round 15 is obtained. Check if this difference matches the output difference from the characteristics. If true, increment the counter corresponding to the 24 bits of K_{17} and 8 bits of K_{16}.

2. The correct subkey is the one with a counter greater than 48.

4.4 Attack Analysis

Complexity Analysis
Step 1 and Step 2: The time complexity of Step (a) is 2^{16} memory accesses. In Step (b), the time complexity again is, therefore, 2^8 memory access since there are approximately 2^7 pairs that get into the filtering process in Step (a). Both (c), (d), (e) and Step 2 are amenable to time complexity considerations since the number of remaining pairs is considerably small. Thus the amount of time taken up taking into account the Steps 1 and 2 is 2^{64} memory accesses.

Step 3: The time complexity of exhaustive search in Step 3 is 2^{48} 16-Round PRESENT encryptions. Thus the overall time complexity for the computation leads to 2^{64} memory accesses.

To calculate the signal-to-noise ratio (S/N) of the attack, we use the following formula:

$$\frac{S}{N} = \frac{p \times 2^k}{\alpha \times \beta} = \frac{2^{-62} \times 2^{32}}{2^{33.18-17.25} \times 2^{17.25-67.32}} = 17.63$$

The general success probability of the attack is 0.999999939, as stated in the paper by [3].

4.5 Conclusion

In this section, we attacked 16-round PRESENT using differential analysis with 2^{64} chosen plaintexts, 2^{32} 6-bit counters, and 2^{24} hash memory. The time complexity of the attack is approximately 2^{64} memory accesses.

4.6 Integral Cryptanalysis

[BKL+07] [Nov12] The basic idea of an integral attack is to analyze the propagation of sums of (many) values. Thus, it can be seen as a dual to the differential cryptanalysis. When applying an integral attack to a block cipher, an attacker first selects a d-th order integral, that is, he/she chooses a set of 2d plaintexts, where d bit positions. take on all values through the set, and the other bits are chosen to be arbitrary constants. Then, he/she traces the evolvement of sum of this set of plaintexts through the encryption algorithm and builds an integral distinguisher as long as possible. Finally, the integral distinguisher will be used to verify key guesses. In practice, a zero-sum property in specific parts of the ciphertext is often used as the integral distinguished.

Integral attacks and higher-order differential attacks also have some links in constructing distinguishers. To build a dth-order integral distinguisher with a zero-sum property is equivalent to showing that the algebraic degree of specific parts of the ciphertext is at most d1 if XOR difference is considered in the higher-order differential attack

Firstly, we analyze the algebraic properties of PRESENT's Sbox. We observe that the rightmost coordinate of the Sbox is quadratic while the other three coordinates have algebraic degree 3. Combined with the properties of the diffusion layer, we find that, for the rightmost bit of the output, the growth rate of its algebraic degree is slower than other bits.

Table 6: Summary of integral attacks on reduced-round PRESENT

Rounds	Key Size	Data	Time
5	all	$N \cdot 2^{32}$	-
5	80	$2^{6.4}$	$2^{25.7}$
6	80	$2^{22.4}$	$2^{41.7}$
7	128	$2^{24.3}$	$2^{100.1}$
7	80	$2^{8.3}$	2^{60}
8	80	$2^{10.1}$	$2^{72.6}$
9	80	$2^{20.3}$	2^{60}
10	128	$2^{22.4}$	$2^{99.3}$

Integral Distinguishers of PRESENT

In this section, we proposed two integral distinguishers of PRESENT. We denote by $X^{(i)}$ the state entering round i, $Y^{(i)}$ the state before the SBoxLayer, $Z^{(i)}$ the state after the SBoxLayer, and $K^{(i)}$ the subkey of round i. Thus, $Y^{(i)} = X^{(i)} \oplus K^{(i)}$. Each state and subkey can be represented as a vector of 64 bits, for example,

$$X^{(i)} = (x_{63}^{(i)}, x_{62}^{(i)}, \ldots, x_0^{(i)}),$$

where $x_0^{(i)}$ is the least significant (rightmost) bit of $X^{(i)}$. Additionally, let $x_{[j-k]}^{(i)}$ be the consecutive $j-k+1$ bits of $X^{(i)}$ from bit k to bit j, and $x_{[j,\ldots,k]}^{(i)}$ represents several separate bits $x_j^{(i)}, \ldots, x_k^{(i)}$ of $X^{(i)}$.

Integral Attack on Reduced-Round PRESENT

In this section, we attack reduced-round PRESENT using the 4-th order integral distinguisher and 16-th order integral distinguisher.

The general attack procedure is given as follows.

1. Choose a set of 2^n ($n = 4$ or $n = 16$) plaintexts to construct a structure, where the rightmost n bits take all possible values of $\mathbb{F}_2^n$ while other bits are chosen to be arbitrary constants over $\mathbb{F}_2$. Obtain the corresponding ciphertexts after r-round encryption.

2. For every guessing of the corresponding subkeys in the last $(r-m)$ rounds, decrypt the ciphertexts to obtain the one-bit state $y_0^{(m+1)}$ after the m-th round, where m is the length of integral distinguishers.

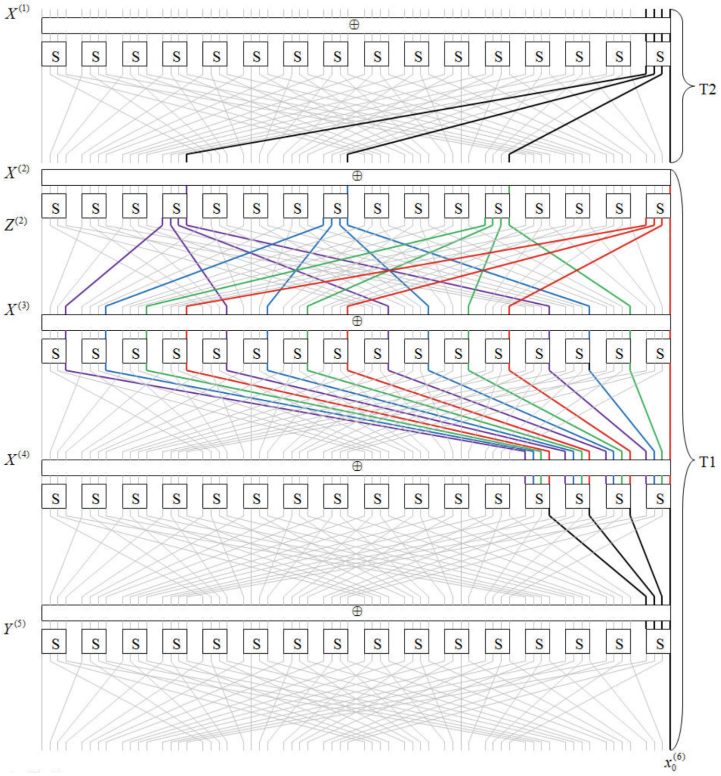


Figure 5: Integral Analysis For 4 Round PRESENT

3. Check whether $\bigoplus_\Lambda y_0^{(m+1)} \left(= \bigoplus_\Lambda x_0^{(m+1)}\right)$ is zero, where Λ with $|\Lambda| = 2^n$ is the set of chosen plaintexts. If the equation is not satisfied, we know the guessed subkey is wrong. Then, we guess another subkey and repeat until the correct subkey is found.

4. Recover the remaining key bits in the master key by the exhaustion method.

Suppose we need to guess k-bit subkey in the last $(r - m)$ rounds, the complexity of this attack can be estimated as follows. Step 1 needs about 2^n plaintexts, which requires 2^n encryptions. In step 2 and step 3, a subkey needs about $\frac{r-m}{r} \times 2^n$ encryptions. For a wrong subkey guess, equation $\bigoplus_\Lambda y_0^{(m+1)} = 0$ holds with probability $\frac{1}{2}$. Therefore, to discard all the wrong k-bit subkey guesses, we need about k plaintext structures. Suppose the master key has $|K|$ bits, then the time complexity of step 4 is about $2^{|K|-k}$ r-round encryptions.

Thus, the data complexity is about $k \times 2^n$ chosen plaintexts. The time complexity in recovering these k key bits is about

$$\sum_{i=1}^{k} \left(2^n + 2^n \times \frac{r-m}{r} 2^k \times \left(\frac{1}{2}\right)^{i-1} \right) \approx \frac{r-m}{r} \times 2^{n+k+1} \tag{7}$$

Thus, the final time complexity is $\max\{\frac{r-m}{r} \times 2^{n+k+1}, 2^{|K|-k}\}$.
A total of 2^k bits are required to keep track of possible values for the k key bits, so the memory complexity is 2^{k+3} bytes.

4.7 Linear Cryptanalysis

[Vik07] [Wan08] [Nov12] [Cho10] [Nik17] In this section, we will analyze the linear approximation of the PRESENT Cipher.
To start with linear cryptanalysis, we first present the Linear Approximation Table of PRESENT : We now make some important properties about the LAT of the PRESENT S-box.

Table 7: Linear Approximation Table

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-4	-	-4	-	-	-	-	-	-4	-	4
2	-	-	2	2	-2	-2	-	-	2	-2	-	4	-	4	-2	2
3	-	-	2	2	2	-2	-4	-	-2	2	-4	-	-	-	-2	-2
4	-	-	-2	2	-2	-2	-	4	-2	-2	-	-4	-	-	-2	2
5	-	-	-2	2	-2	2	-	-	2	2	-4	-	4	-	2	2
6	-	-	-	-4	-	-	-4	-	-	-4	-	-	4	-	-	-
7	-	-	-	4	4	-	-	-	-	-4	-	-	-	-	4	-
8	-	-	2	-2	-	-	-2	2	-2	2	-	-	-2	2	4	4
9	-	4	-2	-2	-	-	2	-2	-2	-2	-4	-	-2	2	-	-
A	-	-	4	-	2	2	2	-2	-	-	-	-4	2	2	-2	2
B	-	-4	-	-	-2	-2	2	-2	-4	-	-	-	2	2	2	-2
C	-	-	-	-	-2	-2	-2	-2	4	-	-	-4	-2	2	2	-2
D	-	4	4	-	-2	-2	2	2	-	-	-	-	2	-2	2	-2
E	-	-	2	2	-4	4	-2	-2	-2	-2	-	-	-2	-2	-	-
F	-	4	-2	2	-	-	-2	-2	-2	2	4	-	2	2	-	-

Notice that the maximum bias of all the linear approximations lies strictly below 2^{-2}, and the maximum bias contributed with respect to a single bit of the linear approximation is strictly less than 2^{-3}. For bounding the bias for four rounds of PRESENT, we first use the above observations. Recall the piling-up lemma for m independent actions (concerning m S-boxes), which states that the probability of a linear approximation is given by:

$$= \frac{1}{2} + 2^{m-1} \prod_{i=1}^{m} \left(p_i - \frac{1}{2} \right)$$

Thus, the bias is given by:

$$2^{m-1} \prod_{i=1}^{m-1} \epsilon_i$$

It is essential to notice that the number of active S-boxes can differ across the four rounds. When the number of active S-boxes is specified, we can then discuss three possible cases. Let the bias for the 4-round PRESENT cipher with i active S-boxes be denoted as $\epsilon_4^{(i)}$.

1. In the case where there are 4 active S-boxes, with one active S-box in each round, the maximum bias for the two active S-boxes in the middle rounds is at most 2^{-3}. The bias for the other two rounds, however, is at most 2^{-2}, which aligns with the model shown below.

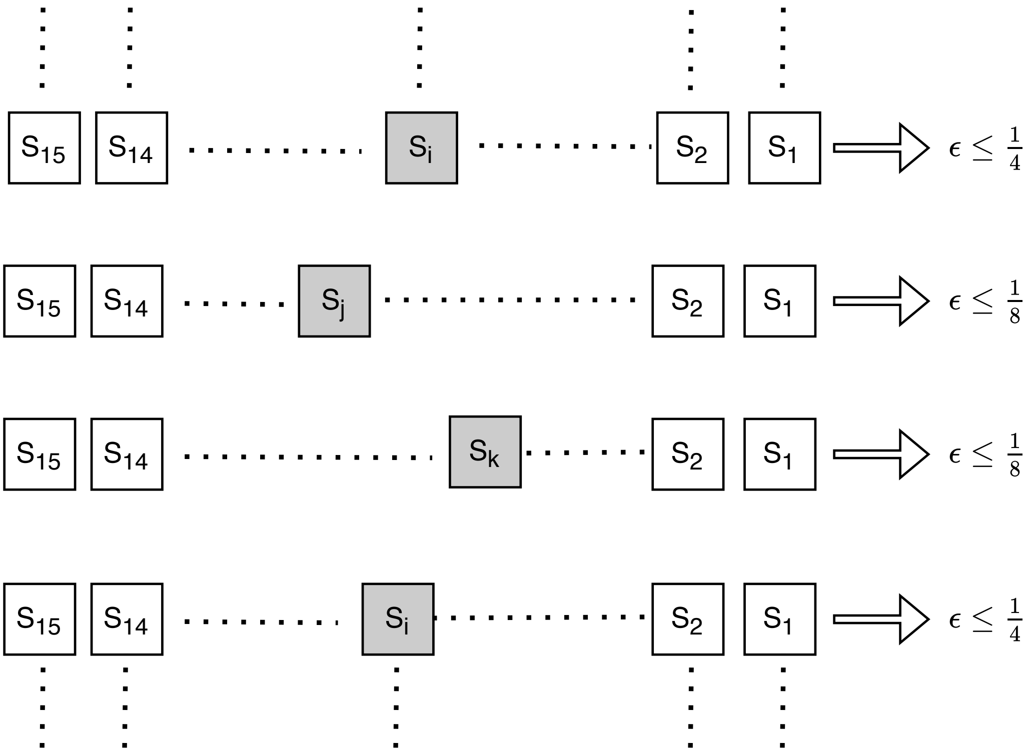

S15
S14
Si
S2
S1
$\epsilon \leq \frac{1}{4}$
Sj
$\epsilon \leq \frac{1}{8}$
Sk
$\epsilon \leq \frac{1}{8}$
Si
$\epsilon \leq \frac{1}{4}$

Figure 6: Bias Calculation

Thus, the bias for this case is bounded using the piling-up lemma as follows:

$$\epsilon_4^{(4)} \leq 2^{4-1} \times (2^{-2})^2 \times (2^{-3})^2$$
$$\epsilon_4^{(4)} \leq 2^{-7}$$

0 **Case 2:** When the number of active S-boxes across the four rounds is 5, the pattern of active S-boxes cannot be $1-2-1-1$ or $1-1-2-1$. From the above observations of the S-boxes, we see that since the two active S-boxes are initiated by the same S-box from the previous round, they must belong to two different sets. Hence, they will activate at least two S-boxes in the next round. Thus, the possible patterns for this case are $2-1-1-1$ or $1-1-1-2$, and the bias is bounded by:

$$\epsilon_4^{(5)} \leq 2^{5-1} \times (2^{-2})^4 \times (2^{-3})$$
$$\epsilon_4^{(4)} \leq 2^{-7}$$

Case 3: If the number of active S-boxes is more than 5, the maximum bias for each round is 2^{-4}. In this case, we have:

$$\epsilon_4^{(i)} \leq 2^{i-1} \times (2^{-2})^i \quad \text{for} \quad i > 5$$

Evidently, for $i = 6$, the bias is 2^{-7}, and for $i > 6$, the bias is strictly less than 2^{-7}.

From the above analysis, we can conclude that the bias for the 4-round linear approximation of PRESENT is bounded by 2^{-7}, i.e.,

$$\epsilon_4 \leq 2^{-7}$$

We can then use this result to bound the linear approximation bias for 28 rounds of PRESENT:

$$\epsilon_{28} \leq 2^6 \times \epsilon_4^{(7)} = 2^6 \times (2^{-7})^7 = 2^{-43}$$

For single-bit recovery, the number of known plaintexts (N) required for a successful attack is approximately:

$$N = c\,|\epsilon|^{-2}, \quad \text{where} \quad c \geq 2$$

Thus, to attack 31 rounds of PRESENT, the attacker will need to approximate 28 rounds, requiring approximately 2^{86} known plaintexts. This number exceeds the available space, which is 2^{64}.

5 Conclusions

This paper analysed the PRESENT cipher including the implementation design decisions of the cipher and the distinctive characteristics of the S-box within this cipher that cause the cipher to be resistant to both differential and linear attacks. We studied an academic differential attack on 16-round PRESENT cipher and even applied a built 3-round differential attack in Python. In addition, we showed that the cipher is secure against linear cryptanalysis. Lastly, we introduced an interesting experiment in which we examined the side-channel characteristics of PRESENT that could potentially impact the runtime of its encryption function. We believe this experiment warrants further investigation and strongly encourage continued analysis of the cipher

6 Difference from other ciphers

Cipher	**Design**	**Key Size**	**Focus/Key Differences from PRESENT**
PRESENT	Substitution-Permutation Network (SPN)	80 or 128 bits	Extremely lightweight with minimal hardware footprint, ideal for constrained environments like IoT and RFID systems.
Pyjamask-128	Masked cipher designed with resistance against side-channel attacks	128 bits	Emphasis on security against physical attacks compared to PRESENT's focus on minimal resources.
PRINCE	Lightweight cipher with a focus on low-latency encryption	128 bits (64-bit main key + 64-bit tweak)	Balances security and ultra-low latency encryption; more latency-focused compared to PRESENT's hardware optimization.

KLEIN	SPN-based, optimized for software and hardware implementation	64, 80, or 96 bits	Balances lightweight design with strong diffusion, while PRESENT focuses more on hardware efficiency.
LED	AES-inspired SPN	Up to 128 bits	Provides strong security for low-resource devices but uses a more complex AES-like structure compared to PRESENT's simpler SPN.
Midori	Energy-efficient block cipher	64 or 128 bits	Prioritizes energy efficiency for ultra-low-power devices; specifically optimized for power consumption, unlike PRESENT.
BAKSHEESH, PUFFIN, PRINTcipher	Application-specific lightweight ciphers	Varies	Targets niche applications, emphasizing properties like differential cryptanalysis resistance or smaller block sizes. PRESENT is generalized for lightweight use cases.
GIFT	Substitution-Permutation Network (SPN)	128 bits	Offers stronger resistance to cryptanalysis while maintaining low resource requirements, providing a modern alternative to PRESENT.
Rectangle	Feistel-like structure with a bit-slice approach	80 or 128 bits	Focuses on efficient software and hardware implementation with strong cryptographic strength. Its Feistel structure differs significantly from PRESENT's SPN design.

References

[BKL+07] Andrey Bogdanov, Lars Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: an ultra-lightweight block cipher. volume 4727, pages 450–466, 09 2007.

[Cho10] Joo Yeon Cho. Linear cryptanalysis of reduced-round present. 2010.

[Nik17] Thomas De Cnudde; Svetla Nikova. Securing the present block cipher against combined side-channel analysis and fault attacks. 2017.

[Nov12] Jan Pospíšil; Martin Novotný. Lightweight cipher resistivity against brute-force attack: Analysis of present. 2012.

[Tho10] Julia Borghoff Lars R. Knudsen Gregor Leander Soren S. Thomsen. Cryptanalysis of present-like ciphers with secret s-boxes. 2010.

[Vik07] A. BogdanovL. R. KnudsenG. LeanderC. PaarA. PoschmannM. J. B. RobshawY. SeurinC. Vikkelsoe. Present: An ultra-lightweight block cipher. 2007.

[Wan08] Meiqin Wang. Differential cryptanalysis of reduced-round present. 2008.

[WW13] Shengbao Wu and Mingsheng Wang. Integral attacks on reduced-round present. In Sihan Qing, Jianying Zhou, and Dongmei Liu, editors, *Information and Communications Security*, pages 331–345, Cham, 2013. Springer International Publishing.

[TK09] Onur ÖzenKerem VarıcıCihangir TezcanÇelebi Kocair. Lightweight block ciphers revisited: Cryptanalysis of reduced round present and hight. 2009.