J Supercomput (2014) 70:95–103
DOI 10.1007/s11227-014-1103-3

Biclique cryptanalysis of PRESENT-80 and PRESENT-128

Changhoon Lee

Published online: 28 January 2014

Abstract In this paper, we evaluate the security of lightweight block ciphers PRESENT-80 and PRESENT-128 applicable to hybrid information systems against biclique cryptanalysis. To recover the secret key of PRESENT-80/128, our attacks require $2^{79.76}$ full PRESENT-80 encryptions and $2^{127.91}$ full PRESENT-128 encryptions, respectively. These results are superior to known biclique cryptanalytic results on them.

Keywords Block cipher · PRESENT · Biclique · Cryptanalysis

1 Introduction

In Asiacrypt 2011, biclique cryptanalysis of AES was proposed [2]. To recover the secret keys of the full AES-128/192/256, the authors applied this technique to them in [2]. This is a kind of meet-in-the-middle attack such that bicliques improve an efficiency. After the proposal of biclique cryptanalysis, it brings new cryptanalytic techniques on block ciphers, which were known mainly in cryptanalysis of hash functions. The most attractive point is that this approach does not use related keys. Because of this property, many biclique cryptanalytic results on block ciphers were proposed [4,6–8].

In this paper, we apply biclique cryptanalysis to the most popular lightweight block ciphers PRESENT [3] applicable to hybrid information systems. In [2], two concepts of bicliques for AES were considered. One is the long biclique and the other is the independent biclique. We use the concept of independent biclique. This is composed of constructing bicliques from independent related-key differentials and matching with

C. Lee (✉)
Seoul National University of Science and Technology, Seoul, Korea
e-mail: chlee@seoultech.ac.kr

Table 1 Summary of biclique cryptanalytic results on PRESENT

Target algorithm	Rounds	Data complexity	Time complexity	Reference
PRESENT-80	Full(31)	2^{60}	$2^{79.46}$	[1]
	Full(31)	2^{23}	$2^{79.76}$	This paper
PRESENT-128	Full(31)	2^{44}	$2^{127.37}$	[1]
	Full(31)	2^{19}	$2^{127.81}$	This paper

precomputations. We find that a slow and limited diffusion of the key schedule and encryption process in the target algorithm leads to relatively long bicliques with high dimension and an efficient matching check with precomputations. As a result, our attacks can recover the secret key of target algorithms with time complexities smaller than an exhaustive search.

Our results are summarized in Table 1. On the other hand, in [1], biclique cryptanalysis of PRESENT was proposed. In detail, the attack on the full PRESENT-80 requires 2^{60} chosen plaintexts and a time complexity of $2^{79.46}$, and the attack on the full PRESENT-128 requires 2^{44} chosen plaintexts and a time complexity of $2^{127.37}$. Compared with these results, our attacks on PRESENT need the smaller data complexity.

This paper is organized as follows. In Sect. 2, we describe the structures of PRESENT. Biclique cryptanalysis on PRESENT is proposed in Sect. 3. We give our conclusion in Sect. 4.

2 Description of PRESENT

PRESENT is a 64-bit block cipher with 80/128-bit secret keys and 31 iterative rounds. According to the length of secret keys, we call this algorithm PRESENT-80/128, respectively. PRESENT has the SPN structure and it is composed of the round function and the post-whitening. Both versions of PRESENT have the similar structure except the key schedule. In detail, PRESENT-80 takes a 64-bit plaintext $P = (P_{15}, P_{14}, \ldots, P_0)$ and the 80-bit secret key $K = (k_{79}, k_{78}, \ldots, k_0)$ as input values and generates a 64-bit ciphertext $C = (C_{15}, C_{14}, \ldots, C_0)$. Similarly, PRESENT-128 takes a 64-bit plaintext P and the 128-bit secret key $K = (k_{127}, k_{126}, \ldots, k_0)$ as input values and generates a 64-bit ciphertext C.

As depicted in Fig. 1, there are three subfunctions involved in the round function. The first subfunction is addRoundKey. At the beginning of round r, a 64-bit input

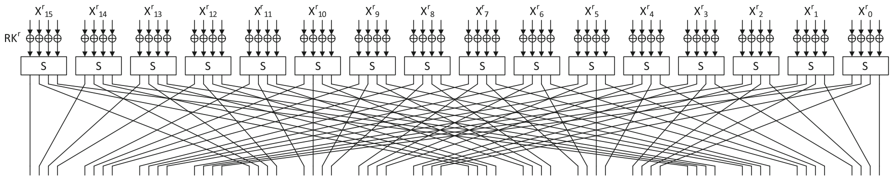

Fig. 1 Round function of PRESENT

value X^r is XORed with a round key $RK^r = (RK_{15}^r, \ldots, RK_0^r)$ $(r = 0, \ldots, 30)$. The second subfunction is sBoxLayer. Sixteen identical 4×4 S-boxes are used in parallel as a nonlinear substitution layer. In the third subfunction, pLayer, a bit permutation is performed to provide diffusion. See [3] for the detailed description of the round function.

The key schedule of PRESENT takes the 80/128-bit secret key and generates thirty two 64-bit round keys RK^r and RK^{31} $(r = 0, \ldots, 30)$. Note that RK^{31} is used for post-whitening. To generate 32 round keys, the key schedule of PRESENT-80 conducts the following procedure. First, the 80-bit secret key $K = (k_{79}, \ldots, k_0)$ is loaded to a 80-bit register $SK = (sk_{79}, \ldots, sk_0)$: $sk_i = k_i$ $(i = 0, \ldots, 79)$. Then, SK is updated as follows.

1. $(sk_{79}, \ldots, sk_0) = (sk_{18}, \ldots, sk_0, sk_{79}, \ldots, sk_{19})$.
2. $(sk_{79}, sk_{78}, sk_{77}, sk_{76}) = Sbox[(sk_{79}, \ldots, sk_{76})]$.
3. $(sk_{19}, sk_{18}, sk_{17}, sk_{16}, sk_{15}) = (sk_{19}, \ldots, sk_{15}) \oplus (r + 1)$.

Applying the above procedure repeatedly, a 64-bit round key RK^r consists of the 64 leftmost bits of SK. That is, at round r, RK^r is computed as follows.

$$RK^r = (sk_{79}, sk_{78}, \ldots, sk_{16}).$$

The key schedule of PRESENT-128 is similar to that of PRESENT-80. After loading the 128-bit secret key $K = (k_{127}, \ldots, k_0)$, a 128-bit register $SK = (sk_{127}, sk_{126}, \ldots, sk_0)$ is updated as follows.

1. $(sk_{127}, \ldots, sk_0) = (sk_{66}, \ldots, sk_0, sk_{127}, \ldots, sk_{67})$.
2. $(sk_{127}, \ldots, sk_{124}) = Sbox[(sk_{127}, \ldots, sk_{124})]$.
3. $(sk_{123}, \ldots, sk_{120}) = Sbox[(sk_{123}, \ldots, sk_{120})]$.
4. $(sk_{66}, sk_{65}, sk_{64}, sk_{63}, sk_{62}) = (sk_{66}, \ldots, sk_{62}) \oplus (r + 1)$.

Applying the above procedure repeatedly, RK^r is computed as follows.

$$RK^r = (sk_{127}, sk_{126}, \ldots, sk_{64}).$$

3 Biclique cryptanalysis of PRESENT

In this section, we propose biclique cryptanalysis of PRESENT-80/128. For the detailed attack procedure based on biclique cryptanlysis, see [2].

3.1 Biclique cryptanalysis of PRESENT-80

First, we explain how to construct a 4-dimensional biclique for round 28–30 of PRESENT-80. The partial secret keys used in $(RK^{28}, RK^{29}, RK^{30}, RK^{31})$ are as follows.

- $RK^{28} : (k_{51}, k_{50}, \ldots, k_0, k_{79}, k_{78}, \ldots, k_{68})$.
- $RK^{29} : (k_{70}, k_{69}, \ldots, k_7)$.

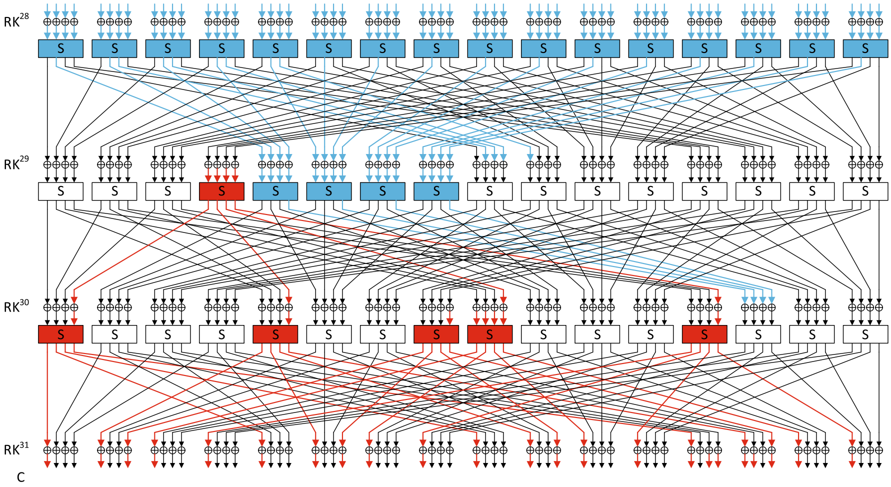

Fig. 2 Four-dimensional biclique for PRESENT-80

- RK^{30} : $(k_9, k_8, \ldots, k_0, k_{79}, k_{78}, \ldots, k_{26})$.
- RK^{31} : $(k_{28}, k_{27}, \ldots, k_0, k_{79}, k_{78}, \ldots, k_{45})$.

From the above relation, we found that varying $(k_{58}, k_{57}, k_{56}, k_{55})$ and $(k_{37}, k_{36}, k_{35}, k_{34})$ gives bicliques for the attack on the full PRESENT-80. In detail, to construct the Δ_i-differential and the ∇_j-differential, we consider $(k_{58}, k_{57}, k_{56}, k_{55})$ and $(k_{37}, k_{36}, k_{35}, k_{34})$, respectively. Let f be a subcipher from round 28 to round 30 (see Fig. 2). An attacker fixes $C_0 = 0$ and derives $S_0 = f^{-1}_{K_{<0,0>}}(C_0)$. The Δ_i-differentials are based on the difference Δ_i^K where the difference of $(k_{58}, k_{57}, k_{56}, k_{55})$ is i and the other bits have zero differences. Similarly, the ∇_j-differentials are based on the difference ∇_j^K where the difference of $(k_{37}, k_{36}, k_{35}, k_{34})$ is j and the other bits have zero differences. Since the Δ_i-differential affects only 23 bits of the ciphertext from Fig. 2, all ciphertexts can be forced to share the same values in other bits. As a result, the data complexity does not exceed 2^{23}.

Now, we are ready to describe our attack on the full PRESENT-80. We rewrite the full PRESENT-80 as follows. Here, g_1, g_2 and f are subciphers for round 0–14, round 15–27 and round 28–30, respectively.

$$E : P \underset{g_1}{\longrightarrow} V \underset{g_2}{\longrightarrow} S \underset{f}{\rightarrow} C.$$

We assume that the plaintext set $\{P_i\}$ corresponding to a 3-round biclique is obtained through the decryption oracle. Then, an attacker detects the right secret key by computing an intermediate variable v in both directions.

$$P_i \xrightarrow[g_1]{K_{<i,j>}} \overrightarrow{v} \overset{?}{=} \overleftarrow{v} \xleftarrow[g_2^{-1}]{K_{<i,j>}} S_j. \tag{1}$$

The attack procedure on the full PRESENT-80 is as follows.

1. *Precomputation* For all $i = 0, \ldots, 2^4 - 1 (d = 4)$, an attacker computes the most significant 4 bits of the output value of round 15 from P_i and $K_{<i,0>}$ in the forward direction, and store it as $\overrightarrow{v}_i$, together with intermediate states and round keys in memory (see Fig. 3). For all $j = 0, \ldots, 2^4 - 1$, an attacker computes the most significant 4 bits of the input value of round 16 from S_i and $K_{<0,j>}$ in the backward direction, and store it as $\overleftarrow{v}_j$, together with intermediate states and round keys in memory.
2. *Computation in the backward direction* In the backward direction, an attacker should compute $\overleftarrow{v}_j$ from S_j and $K_{<i,j>}$ for all i and j, and store them in memory. Recomputations are performed according to red/blue lines in Fig. 3, and the values on the other lines are reused from the precomputation table.
3. *Computation in the forward direction* In the forward direction, an attacker should compute $\overrightarrow{v}_i$ from P_i and $K_{<i,j>}$. Recomputations are performed according to red/blue lines in Fig. 3, and the values on the other lines are reused from the precomputation table.

For each computed $\overrightarrow{v}$, an attacker checks whether the corresponding key candidate $K_{<i,j>}$ satisfies Eq. (1). If he finds such one, he should check the matching on the whole input value of round 16 for $K_{<i,j>}$, P_i and S_j. This matching step yields the right secret key K with a high probability. If a biclique does not give the right secret key, an attacker should choose another biclique and repeat the above procedure until the right secret key is found.

A total time complexity of our attack on PRESENT-80 is computed as follows.

$$C_{\text{total}} = 2^{k-2d} \left(C_{\text{biclique}} + C_{\text{precomp}} + C_{\text{recomp}} + C_{\text{falsepos}}\right). \tag{2}$$

- $k = 80$ and $d = 4$.
- C_{biclique} is a time complexity of constructing a single biclique. In our attack, it is $2^{1.63} \left(\approx 2^{4+1} \cdot (3/31)\right)$ full PRESENT-80 encryptions.
- C_{precomp} is a time complexity of preparing the precomputation for the matching check in Eq. (1). Applying it to our attack, it is $2^{3.85} \left(\approx 2^4 \cdot (28/31)\right)$ full PRESENT-80 encryptions.
- C_{recomp} is a time complexity of recomputing the internal variable v $2^{2d} \left(= 2^8\right)$ times. In our attack, it is $2^{7.53} \left(\approx 2^{2\cdot 4} \cdot (22.31/31)\right)$ full PRESENT-80 encryptions.
- C_{falsepos} is a time complexity caused by false positives, which have to be matched on other bit positions. Since the matching check is performed on four bits in our attack, C_{falsepos} is $2^4 \left(\approx 2^{2\cdot 4-4}\right)$ full PRESENT-80 encryptions.

Hence, a time complexity of our attack on the full PRESENT-80 is computed as follows.

$$C_{\text{total}} = 2^{79.86} \left(\approx 2^{80-2\cdot 4} \left(2^{1.63} + 2^{3.85} + 2^{7.53} + 2^4\right)\right).$$

3.2 Biclique cryptanalysis of PRESENT-128

Since biclique cryptanalysis of the full PRESENT-128 is similar to that of the full PRESENT-80, we briefly introduce our attack on the full PRESENT-128.

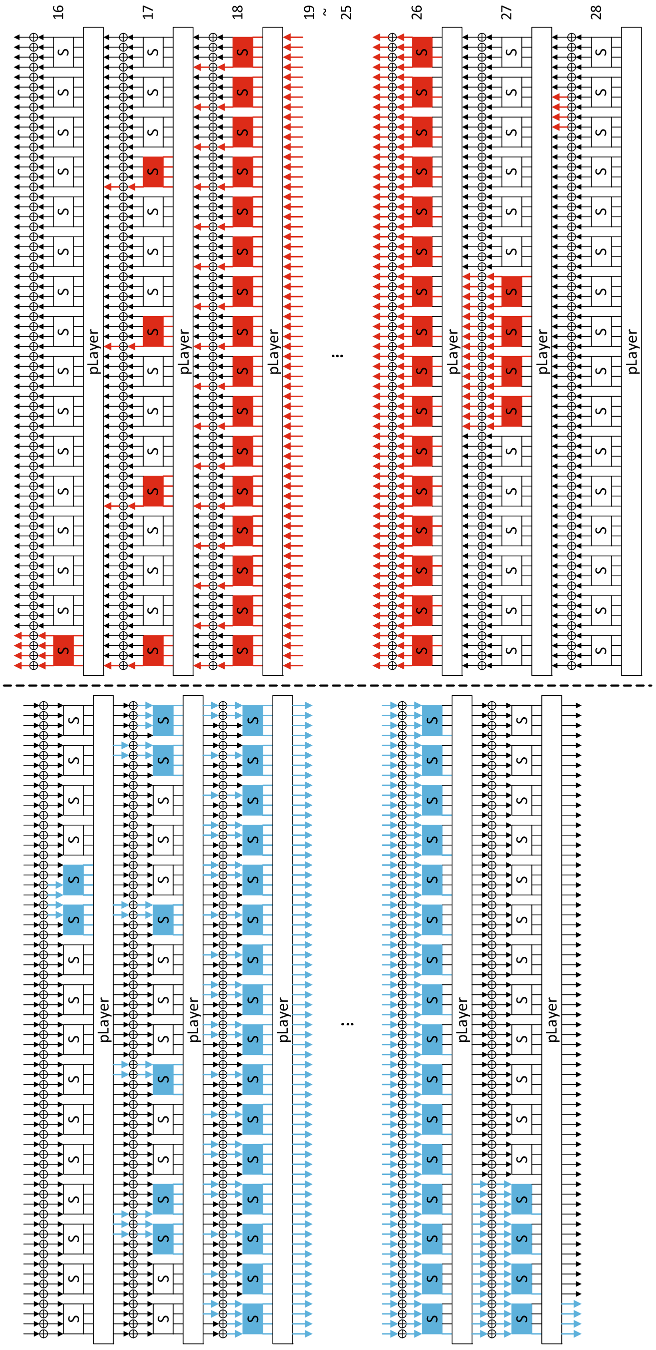

Fig. 3 Recomputations in forward and backward directions for PRESENT-80

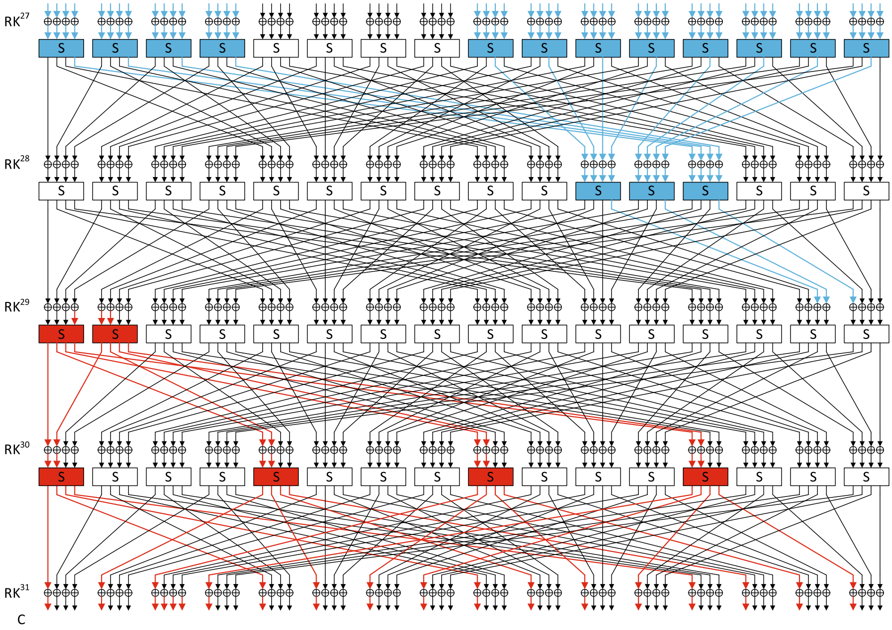

Fig. 4 Four-dimensional biclique for PRESENT-128

To recover the 128-bit secret key, we construct a 3-dimensional biclique for round 27–31 of PRESENT-128 as shown in Fig. 4. The partial secret keys used in $(RK^{27}, RK^{28}, RK^{29}, RK^{30}, RK^{31})$ are as follows.

- RK^{27}: $(k_{16}, k_{15}, \ldots, k_0, k_{127}, k_{126}, \ldots, k_{81})$.
- RK^{28}: $(k_{83}, k_{82}, \ldots, k_{20})$.
- RK^{29}: $(k_{22}, k_{21}, \ldots, k_0, k_{127}, k_{126}, \ldots, k_{87})$.
- RK^{30}: $(k_{89}, k_{88}, \ldots, k_{26})$.
- RK^{31}: $(k_{28}, k_{27}, \ldots, k_0, k_{127}, k_{126}, \ldots, k_{93})$.

From the above relation, to construct the Δ_i-differential and the ∇_j-differential, we consider (k_{19}, k_{18}, k_{17}) and (k_{80}, k_{79}, k_{78}), respectively. The Δ_i-differential affects only 19 bits of the ciphertext from Fig. 4. As a result, the data complexity does not exceed 2^{19}.

We rewrite the full PRESENT-128 as follows. Here, g_1, g_2 and f are subciphers for round 0–13, round 14–26 and round 27–30, respectively.

$$E : P \underset{g_1}{\longrightarrow} V \underset{g_2}{\longrightarrow} S \underset{f}{\rightarrow} C.$$

As depicted in Fig. 5, the matching variable v is the least significant 4 bits of the output value of round 13 (or the input value of round 14). Then, the complexities of our attack are computed as follows (see Eq. (2)).

- Time complexity: $2^{127.81}$ full PRESENT-128 encryptions.

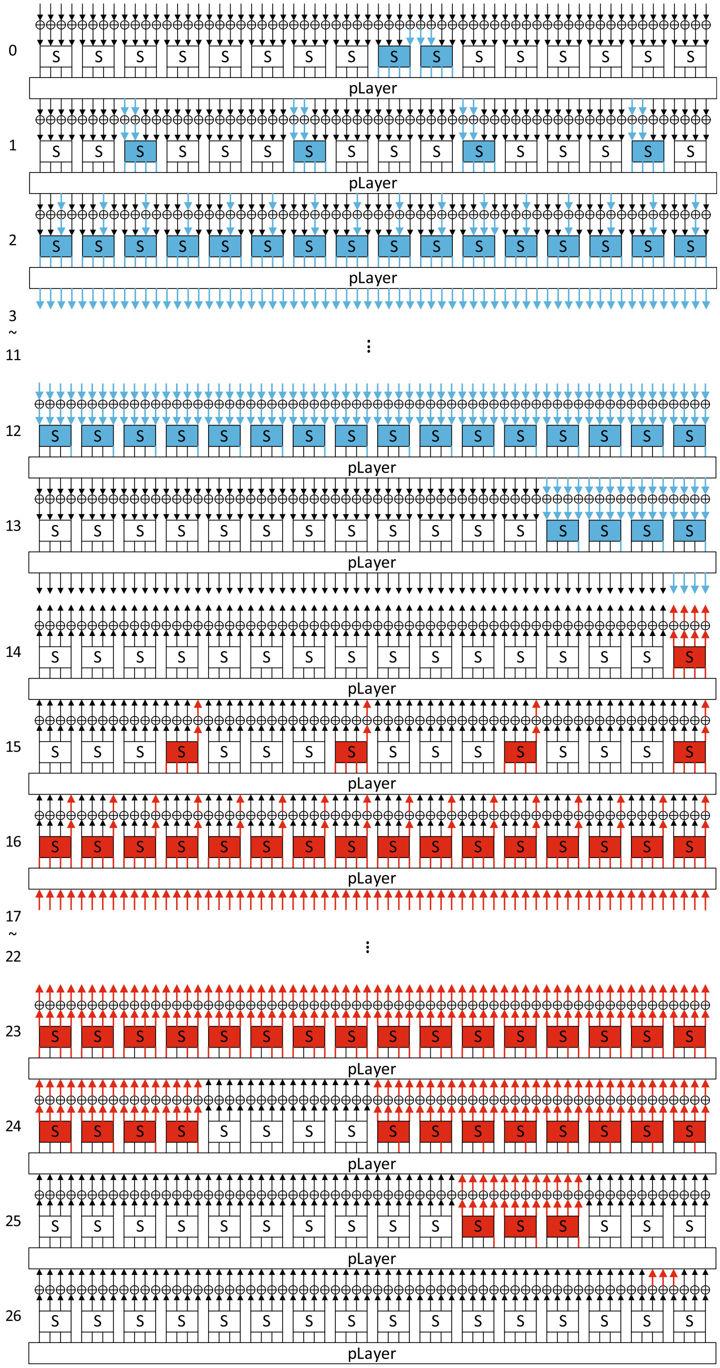

Fig. 5 Recomputations in forward and backward directions for PRESENT-128

- $k = 128$ and $d = 3$.
- $C_{\text{biclique}} = 2^{1.05}\left(\approx 2^{3+1} \cdot (4/31)\right)$ full PRESENT-128 encryptions.
- $C_{\text{precomp}} = 2^{2.8}\left(2^{3} \cdot (27/31)\right)$ full PRESENT-128 encryptions.
- $C_{\text{recomp}} = 2^{5.43}\left(2^{2 \cdot 3} \cdot (20.88/31)\right)$ full PRESENT-128 encryptions.
- $C_{\text{falsepos}} = 2^{2}\left(2^{2 \cdot 3-4}\right)$ full PRESENT-128 encryptions.

4 Conclusion

In this paper, we proposed biclique cryptanalysis of lightweight block ciphers PRESENT-80 and PRESENT-128 applicable to hybrid information systems. Our attack results are summarized in Table 1. To recover the secret key, our attacks require $2^{79.76}$ full PRESENT-80 encryptions and $2^{127.91}$ full PRESENT-128 encryptions, respectively. This means that our results are superior to known biclique cryptanalytic results on PRESENT-80 and PRESENT-128.

References

1. Abed F, Forler C, List E, Lucks S, Wenzel J (2012) Biclique cryptanalysis of the PRESENT and LED lightweight ciphers. Cryptology ePrint Archive, Report 2012/591
2. Bogdanov A, Khovratovich D, Rechberger C (2011) Biclique cryptanalysis of the full AES. In: ASIACRYPT 2011. LNCS, vol 7073. IACR, Lyon, pp 344–371
3. Bogdanov A, Knudsen L, Leander G, Paar C, Poschmann A, Robshaw M, Seurin Y, Vikkelsoe C (2007) PRESENT: an ultra-lightweight block cipher. In: CHES 2007. LNCS, vol 4727. Springer, Berlin, pp 450–466
4. Chen S (2012) Biclique attack of the full ARIA-256. Cryptology ePrint Archive, Report 2012/011
5. Jeong K, Kang H, Lee C, Sung J, Hong S, Lim J (2013) Weakness of lightweight block ciphers mCrypton and LED against biclique cryptanalysis. Peer-to-peer networking and applications. Springer, USA
6. Çoban M, Karakoç F, Biztaş Ö (2012) Biclique Cryptanalysis of TWINE. Cryptology ePrint Archive, Report 2012/422
7. Hong D, Koo B, Kwon D (2012) Biclique attack on the full HIGHT. In: ICISC 2011. LNCS, vol 7259. Springer, Berlin, pp 365–374
8. Khovratovich D, Leurent G, Rechberger C (2012) Narrow-Bicliques: cryptanalysis of Full IDEA. In: EUROCRYPT 2012. LNCS, vol 7237. IACR, Lyon, pp 392–410
9. Lee S, Kim D, Yi J, Ro W (2013) An efficient block cipher implementation on many-core graphics processing units. J Inf Process Syst 8(1):159–174
10. Nakahara Jr J, Sepehrdad P, Zhang B, Wang M (2009) Linear (Hull) and algebraic cryptanalysis of the block cipher PRESENT. In: CANS 2009. LNCS, vol 5888. Springer, Berlin, pp 58–75