Biclique cryptanalysis of full round PRESENT with reduced data complexity

Abstract
Biclique cryptanalysis is a generic attack on block cipher to recovering the secret key faster than brute force. In this paper we use asymmetric independent biclique attack on full round PRESENT. The data complexities of our attacks are less than before, for which a full round attack on PRESENT-80 and PRESENT-128 are $2^{17}$ plain text-ciphertext pairs. The computational complexity is slightly improved.

Keywords: PRESENT, Lightweight block cipher, asymmetric independent biclique, key recovery

Introduction
Biclique cryptanalysis, introduced first in 2011 for hash cryptanalysis. After that, Bogdanov in used this technique and introduced the first biclique attack on full round AES. We can see lots of cryptanalysis results on the other block cipher, such as IDEA, Piccolo, LBlock, SQUARE, KLINE, ARIA, TWINE and HIGHT where proposed.

Biclique attack is a kind of meet in the middle (MITM) attack that is improved for cryptanalysis block cipher to find the unknown secret key. In a biclique attack; first, all possible secret keys are partitioned into a set of groups of keys. For each group of keys, a bipartite graph is constructed. After that, a partial matching is used to filter the wrong key and get the candidate key for the correct keys. Finally, the valid pair (p,c) is used to get the correct key.

PRESENT has a simple structure and belong to a family of light weight 64-bit block cipher proposed in 2007. It supports two key sizes of 80 and 128 bits and denoted by PRESENT-80 and PRESENT-128. A number of biclique attacks on full round PRESENT have been published in. In this paper, we use an asymmetric independent biclique attack on full round PRESENT to reduce the data complexity. The data complexity for a full round attack on PRESENT-80 and PRESENT-128 are $2^{17}$ plaintext-ciphertext pairs. By minimizing the number of active sboxes, the computational complexity is slightly improved. A summary of previous works and our result is given in Table 1.

<table>
<thead>
<tr>
<th>Table 1 Summary of biclique cryptanalytic results on PRESENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>PRESENT-80</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>PRESENT-128</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Volume 2 Issue 2 - 2018
M Hadian Dehkordi, Roghayeh Taghizadeh
Iran University of Science and Technology (IUST), Iran

Correspondence: M Hadian Dehkordi, Iran University of Science and Technology (IUST), Iran, Email mhadian@iust.ac.ir

Received: February 25, 2018 | Published: September 26, 2018

Outline: First in section 2, the structure of PRESENT is described. We overview biclique cryptanalysis in section 3. Section 4 contains a detailed description of our attack and compute complexity. Finally, we conclude our work in section 5.

Description of PRESENT
PRESENT is a 64-bit lightweight cipher that consists of 31 rounds with 80 or 128 bits secret key. After the final round, the state is added XORed with round key to generate the ciphertext. Both versions of PRESENT, have the same structure as shown in Figure 1. But the key schedule is slightly different. See for detail description of round function. The key schedule expands a secret key 80 or 128 bits to 32 round keys $RK_r$, $r = 0, ..., 31$ with 64 bits each. In PRESENT-80, the 80 bits secret key $K = (k_{31}, ..., k_0)$, is stored in 80 bits register $SK = (sk_{79}, ..., sk_0)$ with $sk_0 = k_0$. After that, SK is updated as follow:

- $sk_0, sk_79, sk_{78}, sk_{77}, sk_{76}$ = $\text{shbox}(sk_{79}, sk_{78}, sk_{77}, sk_{76})$
- $sk_{19}, sk_{18}, sk_{17}, sk_{16}, sk_{15}$ = $(sk_{19}, sk_{18}, sk_{17}, sk_{16}, sk_{15}) \oplus (r + 1)$.

The 64 leftmost bits of SK selected as 64 bits round keys $RK_r$, for PRESENT-80. The above procedure is repeated until all round keys are constructed. The key schedule of PRESENT-128 is similar and is as follow:

- $(sk_{127}, ..., sk_0)$ = $(sk_{126}, ..., sk_0)$
- $(sk_{123}, sk_{122}, sk_{121}, sk_{120}) = \text{shbox}(sk_{123}, sk_{122}, sk_{121}, sk_{120})$
- $(sk_{127}, sk_{126}, sk_{125}, sk_{124}) = \text{shbox}(sk_{127}, sk_{126}, sk_{125}, sk_{124})$
- $(sk_{16}, sk_{65}, sk_{64}, sk_{63}, sk_{62}) = (sk_{16}, sk_{65}, sk_{64}, sk_{63}, sk_{62}) \oplus (r + 1)$.

The above procedure is repeated, and 64 bits round key $RK_r$, for PRESENT-128 are constructed as 64 leftmost bits of $SK_r$. 

Keywords: PRESENT, Lightweight block cipher, asymmetric independent biclique, key recovery

Creative Commons Attribution License
Biclique cryptanalysis

In this section, we give a brief overview on biclique cryptanalysis. See the cited references for complete detail of this. In a biclique attack, the biclique can be constructed in plaintext or ciphertext side. Here, we construct a biclique in ciphertext side so the attack is a chosen ciphertext attack, and we need the decryption oracle (for the biclique in plaintext side, the attack is a chosen plaintext attack, and we need the encryption oracle.)

(d1, d2) -dimensional asymmetric biclique: Let f be subcipher of cipher E that maps an intermediate state S to the ciphertext C; f ∗(S) = C. The 3-tuple [c1, c2, K[i, j]] is called a (d1, d2) -dimensional asymmetric biclique if for all i = 0, 1, ..., 2d1 -1 and j = 0, 1, ..., 2d2 -1, s = f ∗K[i, j](c). The structure (d1, d2) -dimensional asymmetric biclique is shown in Figure 2.

Figure 1 round function of Present

Biclique cryptanalysis

In this section, we give a brief overview on biclique cryptanalysis. Since some candidate keys remain to be chosen by varying k after g, we need the decryption oracle (for the biclique in plaintext side, the attack is a chosen plaintext attack, and we need the encryption oracle.)

(d1, d2) -dimensional asymmetric biclique: Let f be subcipher of cipher E that maps an intermediate state S to the ciphertext C; f ∗(S) = C. The 3-tuple [c1, c2, K[i, j]] is called a (d1, d2) -dimensional asymmetric biclique if for all i = 0, 1, ..., 2d1 -1 and j = 0, 1, ..., 2d2 -1, s = f ∗K[i, j](c). The structure (d1, d2) -dimensional asymmetric biclique is shown in Figure 2.

Figure 2 (d1, d2) dimensional asymmetric biclique in the ciphertext side.

The four main steps of the biclique attack are:

Key partitioning: An adversary chooses a partition of the key space into groups of key of cardinality 2d1(d2) each. A key in a group is indexed as an element of 2d1×2d2 matrices K[i, j].

Constructing a biclique: Independent biclique is an efficient tool for making a biclique. The procedure of constructing the biclique is as follow:

1. Choose a random ciphertext c0 and compute S0 as S0 = f ∗K[0,0](c0).
2. Compute s j as s j = f ∗K[j,0](c0) for i = 1, ..., 2d2 -1.
3. Compute c j as c j = f ∗K[0,j](s j) for j = 1, ..., 2d1 -1.

Partial matching: Partial matching is an efficient way that all keys in a group are tested. By using this method the computational complexity is reduced significantly. Since the biclique is constructed in the ciphertext side, partial matching is performed at the plaintext side. The remaining parts of cipher E are split into the subcipher g1 and g2. E = g1g2. f ∗. We choose the matching variable v after subcipher before g1 subcipher g2. The value of matching variable v is calculated in both directions to find the correct key. First, we called on the decryption oracle to obtain plaintext p, j = 0, 1, ..., 2d1 -1.

In forward direction, p, j is partially encrypted under key k = 0, j) to get v(0, j), j = 0, 1, ..., 2d1 -1 and store 2d1 values. After that, in backward direction, s is partially decrypted under key k[i,0] to get v(i, j), i = 0, 1, ..., 2d2 -1 and store 2d2 values. For checking the matching, v(i, j) = v(i, j), we recomputed only those parts that differ from the stored values. Note that by using partial matching, we can reduce the computational complexity significantly.

Rechecking the candidate keys: Since some candidate keys remain in each group of keys, to filter the wrong keys, the valid pair { p, c } is used to get the correct key. The structure of (d1, d2) -dimensional asymmetric biclique cryptanalysis is shown in Figure 3.

Independent asymmetric biclique attack on full round PRESENT-80

In this section, we first construct a (1, 3) -dimensional asymmetric biclique on the ciphertext side for round 29-31 of PRESENT-80. The partial secret key used in RK39, RK38, RK36 and RK35 are as fallow:

- RK28: (k27, k26, k25, k24, k23)
- RK29: (k27, k26, k25, k24, k23, k22)
- RK30: (k27, k26, k25, k24, k23, k22, k21)
- RK31: (k27, k26, k25, k24, k23, k22, k21, k20)

The attack consists of four steps:

Key partitioning: The 80 bits key space is partitioned into 28 groups of 28 key each. The base keys K[0,0] take all 80 bits of the secret key with 4 bits fixed to zero and the remaining 76 bits take all possible values. In our attack, the differences Δi are chosen by varying k20 in forward direction, and the key differences Δi are chosen by varying (k36, k35, k34) in backward direction.

Constructing a biclique: We use these steps to construct a biclique for each group of keys in the ciphertext side. Let f be subcipher for round 29-31 of PRESENT-80.

Step1. Fix c0 = 0 and compute s0 = f ∗K[0,0](c0).

Step2. Decrypt c j under different keys K[i, 0] to get corresponding states, s j, i = 1, 2, 3. The active sboxes are visualized by the blue trail in forward direction in Figure 4. These sboxes should be calculated 2 times, while the other ones are computed only once; already done in step1.

Step3. Encrypt s j under key K[0, 1] to get the corresponding state c j. The active sboxes are visualized by the red trail in forward direction in Figure 5. These sboxes should be calculated 2 times, while the other ones have already been computed.
Matching: We use partial matching by applying matching with precomputation and recomputation over the remaining round of the cipher to filter the wrong key. Let \( g_r \) and \( g_j \) be subciphers for rounds 1-15 and rounds 16-28 respectively. We take matching variable \( v \) in the bits \( v_{25}, \ldots, v_{26} \) of the state \( v(v_{25}, \ldots, v_{26}) \) after round 15. Then we detect the right key by computing \( v \) in both directions. The correct key is the one that meets in both directions.

Complexity

The cost of a biclique is dominated by the number of sboxes to be calculated, so we count the number of sbox operations in the round transformation and key schedule. Each round of PRESENT-80 together with one round of key schedule takes 17 sbox computations, so the complexity of single encryption equals calculation of \( 31 \times 17 = 527 \) sboxes.

Biclique complexity: For each \( 2^5 \) groups of keys, the following computations should be calculated.

Forward direction complexity: In forward direction, 5 sboxes should be calculated 2 times. The active sboxes are visualized by the red trail in forward direction in Figure 4.

Backward direction complexity: In backward direction, 17 sboxes should be calculated \( 2^{17} \times 8 \) times. The active sboxes are visualized by the blue trail in backward direction in Figure 4. The remaining 26 sboxes are calculated once. In total, \( 5 \times 2 + 17 \times 18 + 26 = 162 \) sboxes should be calculated to construct a biclique.

Matching complexity: For each \( 2^6 \) groups of keys the following computations should be calculated.

Forward direction complexity: In forward direction, \( 2 + 5 + 12 \times 16 + 4 = 203 \) sboxes in round transformation and 2 sboxes in key schedule should be calculated \( 2^1 \) times, and 25 sboxes are calculated only once. The active sboxes are visualized by the blue trail in forward direction in Figure 5. The remaining 26 sboxes are calculated once. In total, \( 5 \times 2 + 17 \times 18 + 26 = 162 \) sboxes should be calculated to construct a biclique.

Backward direction complexity: In backward direction, 42 sboxes are calculated once, and \( 1 + 5 + 8 \times 16 + 4 + 1 = 139 \) sboxes in round transformation and one sbox in key schedule should be calculated 2 times. The active sboxes are visualized by the red trail in backward direction in Figure 5. This procedure is repeated \( 2^1 \) for all \( s_j \)'s, so in backward direction, \( 2^2 \times (2 \times 140 + 42) = 2576 \) sboxes should be calculated.

Rechecking the candidate keys complexity: Since in each group, \( 2^5 \) keys should be checked and the probability of matching is \( 2^{-5} \), so \( 2^5 \times 2^{-5} = 1 \) remaining key candidate should be tested to get the correct key.

In total, the computational complexity of our attack is given by:

\[
2^{36} \left( \frac{162 + 3330 + 2756}{527} + 1 \right) = 2^{70.64}
\]

Data complexity: We need at most \( 2^{17} \) chosen ciphertexts for performing a biclique attack.

Independent asymmetric biclique attack on full round PRESENT-128: In this section, we describe our attack on PRESENT-128. Since a biclique attack on full round PRESENT-128 is similar to PRESENT-80, we only state our results.

The partial keys used in \( RK_{27} \), \( RK_{28} \), \( RK_{29} \) and \( RK_{30} \) are as follow:

\[
\begin{align*}
RK_{27}: \; &k_{16}, \ldots, k_{0}, k_{127}, \ldots, k_{81} \\
RK_{28}: \; &k_{43}, \ldots, k_{20} \\
RK_{29}: \; &k_{23}, \ldots, k_{17}, k_{127}, \ldots, k_{97} \\
RK_{30}: \; &k_{89}, \ldots, k_{26} \\
RK_{31}: \; &k_{28}, \ldots, k_{0}, k_{127}, \ldots, k_{93}
\end{align*}
\]

We construct \((1, 3)\)-dimensional asymmetric biclique on the ciphertext side for round 27-31. The key differences

\[
\Delta^i \text{ are chosen by varying } k_{i} \text{ in forward direction and the key differences } \nabla^j \text{ are chosen by varying } \left( k_{i2}, k_{j1}, k_{90} \right) \text{ in backward direction. The biclique construction is illustrated in Figure 6. Here, } f, g, \text{ and } s \text{ are subciphers for rounds } 28-31, \text{ rounds } 1-14 \text{ and rounds } 15-27 \text{ respectively. As we see, the } \Delta^i \text{ affects only 17 bits of ciphertext, and since we use the same value for } c_{0} \text{ in each biclique, the data complexity does not exceed } 2^{17}. \text{ We take matching variable } v; \text{ the least significant 4 bits of the output value after round 14 as shown in Figure 7.}
\]

Citation: Dehkordi MH, Taghzideh R. Biclique cryptanalysis of full round PRESENT with reduced data complexity. Electric Electron Tech Open Acc J. 2018;2(2):251–255. DOI: 10.15406/eetoaj.2018.02.00021

Copyright: ©2018 Dehkordi et al.
Since in each group,

In forward direction, 5 sboxes in round

remaining key candidate should be tested to reach the

js

I n

's, so in

sboxes. The

times and 26 sboxes are calculated once only. The

sboxes in round transformation should be

sboxes,

's, so in backward direction,

in Figure 7. This procedure is repeated 2

active sboxes are visualized by the red trail in forward direction

calculated to construct a biclique.

Biclique complexity: In forward direction, 5 sboxes in round transformation and one sbox in key schedule should be calculated 2 times. The active sboxes are visualized by the red trail in forward direction in Figure 6. In backward direction, 15 sboxes should be calculated 2\(^8\) times. The active sboxes are visualized by the blue trail in backward direction in Figure 6. The remaining 44 sboxes are calculated once. In total, 6 \times 2 + 15 \times 8 + 44 = 176 sboxes should be calculated to construct a biclique.

Matching complexity: In forward direction, 2 \times 4 + 11 \times 16 + 4 = 186 sboxes in round transformation and one sbox in key schedule should be calculated. The active sboxes are visualized by the blue trail in forward direction in Figure 7. This procedure is repeated 2 times for all \(p \neq s\) 's, so in forward direction \(2 \times (2 \times 187 + 26) = 3044\) sboxes should be calculated. In backward direction, 43 sboxes are calculated once and 1 + 4 + 8 \times 16 + 4 + 1 = 138 sboxes in round transformation should be calculated 2 times. The active sboxes are visualized by the red trail in backward direction in Figure 7. This procedure is repeated 2\(^7\) times for all \(s \neq s\) 's, so in backward direction, \(2 \times (2 \times 138 + 43) = 2552\) sboxes should be calculated.

Rechecking the candidate keys complexity: Since in each group, 2\(^n\) keys should be checked and the probability of matching is 2\(^{-n}\), so 2\(^{2 \times 2^{17}}\) = 1 remaining key candidate should be tested to reach the correct key.

In total, the computational complexity of our attack is given by:

\[
2^{124} \times \left( \frac{176 + 3044 + 2552}{558} + 1 \right) = 2^{127.50}
\]

Data complexity: We need at most 2\(^{17}\) chosen ciphertexts for performing a biclique attack.

Conclusion

In this paper, we present a Biclique cryptanalysis of full round PRESENT with reduced data complexity. In this paper we use asymmetric independent biclique attack on full round PRESENT. We construct a asymmetric biclique in ciphertext side for round 29-31 of PRESENT-80 and round 27-31 of PRESENT-128.

The data complexities of our attacks are less than before, which for a full round attack on PRESENT-80 and PRESENT-128 are 2\(^{17}\) plain text-ciphertext pairs. The computational complexity is slightly improved.

Acknowledgements

None.

Conflicts of interests

The author declares there are no conflicts of interests.

References

10. Deukjo Hong, Bonwook Koo, Daesung Kwon. Biclique Attack on the
Biclique cryptanalysis of full round PRESENT with reduced data complexity


