Two Kinds of Biclique Attacks on Lightweight Block Cipher PRINCE

Zheng Yuan[1,2],✉ Zhen Peng[1,2],✉ Haiwen Ou[1,2]

1. Beijing Electronic Science &Technology Institute, Beijing 100070,China
2. Xidian University, Xi'an 710071, China

zyuan@tsinghua.edu.cn; pengzhen0822@126.com

Abstract: Inspired by the paper [10], using better differential characteristics in the biclique construction, we give another balanced biclique attack on full rounds PRINCE with the lower complexity in this paper. Our balanced biclique attack has $2^{62.67}$ computational complexity and 2^{32} data complexity. Furthermore, we first illustrate a star-based biclique attack on full rounds PRINCE cipher in this paper. Our star-based biclique attack has $2^{63.02}$ computational complexity and the required data can be reduced to only a single plaintext-ciphertext pair, this is the optimal data complexity among the existing results of full round attack on PRINCE.

Keywords: balanced biclique attack, star-based biclique, PRINCE, PRINCEcore, data complexity

1 Introduction

Lightweight Block Cipher PRINCE. Lightweight block ciphers are suitable for extremely constrained environment with short block length and key length. The best studied lightweight block ciphers are mCrypton[1],CLEFIA[2],Piccolo[3],PRESENT[4],KLEIN[5],PRINCE[6]. PRINCE is a modern involutive lightweight block cipher proposed by Rechberger et al. at Asiacrypt 2012[6]. Since then, it has been widely used in many constrained devices. PRINCE with 64-bit block and 128-bit key, uses so-called FX-construction. It is consisted of a 64-bit core cipher, named as PRINCEcore, and two whitenings before and after the PRINCEcore. PRINCEcore holds the major encryption logic, so the security of PRINCE mainly depends on the properties of the PRINCEcore. After being proposed in 2012, there are some cryptanalysis of PRINCE, such as differential cryptanalysis[10], algebraic cryptanalysis[11], Reflection Cryptanalysis[9] and biclique cryptanalysis[10]. Farzaech et al. gave the security evaluations of PRINCEcore against biclique and differential cryptanalysis, respectively[10]. They presented an independent-biclique attack on the full version PRINCEcore with $2^{62.72}$ computational complexity and 2^{40} data complexity[10], they also presented upon 2-round attack of differential cryptanalysis of PRINCEcore with $2^{32.44}$ computational complexity and 2^{32} data complexity, and upon 4-round attack of differential cryptanalysis with $2^{56.26}$ computational complexity and 2^{48} data complexity. Lilang gave algebraic attack on PRINCE[11], in which all the key bits of 5-round PRINCEcore could be obtained based on the different known plaintexts and all the key bits of 6-round PRINCE can be successful recovered under the chosen plaintext. Anne used multiple differentials and exploited some properties of PRINCE for recovering the whole key[12]. Their attack could be extended up to 11 rounds with $2^{62.43}$ computational complexity and $2^{59.81}$ data complexity.

Biclique Cryptanalysis. Biclique cryptanalysis was first proposed by Khovratovich et al. in 2011[7] and they demonstrated the first single-key attacks on full rounds of three variants of AES with a significant advantage over exhaustive search. Bogdanov et al. proposed a star-based biclique[8] with just one state in one vertex set and 2^{2d} states in the other ones, and implemented a star-based biclique attack on AES-128/192/256 , they achieved the theoretically minimal data complexity. In 2015, Yuan et al. gave a star-based independent biclique attack on full rounds SQUARE [13], which is the second application of star-based biclique attack to a block cipher.

Our contribution. Stimuliated by the balanced independent biclique cryptanalysis of PRINCEcore[10] and star-based independent biclique crypanalysis [8,13], in this paper we present another balanced independent biclique attack on PRINCEcore, Especially, we first give a star-based independent biclique attack on PRINCEcore. Both of them are full rounds attack on PRINCEcore cipher. Our balanced independent biclique attack is superior to the previous balanced ones with $2^{62.67}$ computational complexity and 2^{32} data complexity. Our star-based independent biclique attack, with $2^{63.02}$ computational complexity and required data can be reduced to a single plaintext-ciphertext pair, is first to use this kind of attack on PRINCEcore. To be the best of our knowledge, this is the optimal data complexity. We can note that the computational complexity and data complexity is influenced by the biclique construction.

Outline. This paper is organized as follows: Section 2 describes the lightweight block cipher PRINCE. Section 3 and section 4 presents the balanced independent biclique attack and star-based independent biclique attack on full rounds PRINCEcore, respectively. Seciton 5 summarizes the whole paper.

2 Description of Lightweight Block Cipher PRINCE

PRICNE cipher is a 64-bit block cipher with a 128-bit key. It uses the FX construction (See Fig.1), and has three parts, one part is considered as the core cipher, named as PRICNEcore, and remaining parts are used for whitenings before and after the core.

Fig1. The construction of PRINCE

2.1 The Key Schedule

The key schedule of PRINCE is not so complicated. Firstly, the 128-bit key k is split into two 64-bit words

$$k = k_0 \| k_1$$

Then, it is extended to192-bit by a linear mapping of the form

$$k = (k_0 \| k_1) \longrightarrow (k_0 \| k_0' \| k_1) := (k_0 \| (k_0 >>> 1) \oplus (k_0 >>> 63) \| k_1)$$

where the 64-bit k_1 is used for the PRINCEcore, the 64-bit k_0 and k_0' are used to wrap the core with two additions, the pre- and post- whitening.

2.2 Round Transformation

As described in Fig.2, we can see that PRINCEcore consists of the first five rounds, the middle rounds, and the last five rounds. Except the middle rounds, each round has two additional XORs with the key k_1 and a different round constant RC_i. Round operation in the last five rounds is the inverse of the first five rounds. Every round in PRINCEcore constains following five steps.

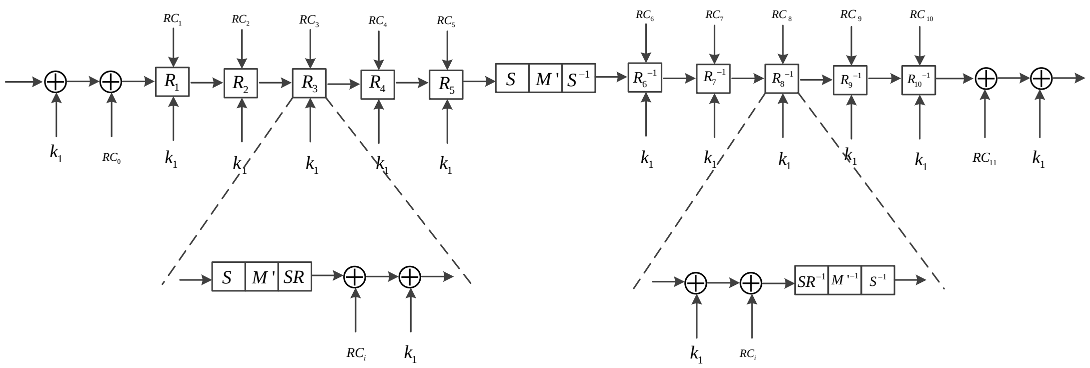

Fig2. PRINCEcore

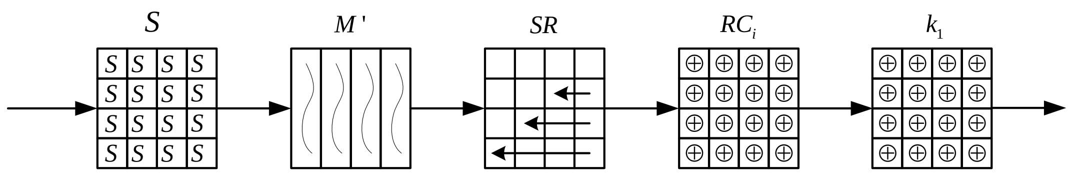

Fig3. Single round of PRINCEcore

SB **-layer.**In the SB -layer step,PRINCEcore uses one 4-bit Sbox. i.e. a nibble Sbox. Every nibble in the state is replaced by the nibble generated after using the following S -box.

M/M' **-layer.** M/M' layer is a linear layer. M' is a 64×64 block diagonal matrix. M_0 and M_1 are two 16×16 submatrices which are placed on the diagonal of M'.

$$M'(x) = M' \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} M_0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 \\ 0 & 0 & M_1 & 0 \\ 0 & 0 & 0 & M_0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = M_0(x_1) \| M_1(x_2) \| M_1(x_3) \| M_0(x_4)$$

However, M' is only used in the middle round. To ensure the α -reflection property, M' need to be an involution. In the first five rounds, linear layer uses matrix M which can be derived from M'. i.e., $M = SR \circ M'$, SR is a shift row operation.

SR **-layer.** SR operation in the PRINCEcore is as same as the one in the AES. Row i of the state is rotated i cell positions to the left, $i = 0,1,2,3$.

k_i **-add.** In the k_i -add step, a 64-bit state is Xor with the 64-bit subkey k_1.

RC_i **-add.** In the RC_i -add step, the 64-bit round constants RC_i is Xor with the 64-bit state.

3 Balanced Independent Biclique Attack on PRINCEcore

Inspired by paper [10], we give another balanced biclique attack on the full rounds PRINCEcore using the better differential characteristics in the biclique construction. We construct a biclique over the first round of PRINCE and match with precomputations technique on the remaining rounds.

3.1 Key Partitioning

We divide the 64-bit key space into 2^{48} 16-nibble key groups. The base keys $K[0,0]$ are all 2^{48} 16-nibble values with four nibbles fixed to 0 and all other 12 nibbles in the state taking on all possible values. The 2^{16} keys in a set $K[i,j]$ are defined relative to the base key $K[0,0]$,and two difference Δ_i^K and ∇_j^K, where $i, j \in \{0,...,2^8-1\}$ and $i = (i_1 \| i_2)$ and $j = (j_1 \| j_2)$.

$K[0,0] =$

0		0	
0		0	

$\Delta_i^K(k_1) =$

i_1			
i_2			

$\nabla_j^K(k_1) =$

		j_1	
		j_2	

Fig4. Key Partitioning

3.2 Constructing a Single Round Independent-Biclique of Dimension 8

Here, we construct an independent biclique on the first round of PRINCEcore. We consider the block cipher as a composition of three subciphers: $e = g_2 \circ g_1 \circ f$.

$$P \xrightarrow{f} S \xrightarrow{g_1} V \xrightarrow{g_2} C$$

According to section 3.1, Δ_i-trail actives nibble 0 and nibble 1, while ∇_j-trail actives nibble 8 and nibble 9. we determine 2^8 plaintexts P_i and 2^8 internal states S_j to satisfy the definition of the biclique $P_i \xrightarrow[f]{K[i,j]} S_j$ ($i, j \in \{0,...,2^8-1\}$). P_i is the plaintext and S_j is the internal state, this paper refers to the output of round 1 encryption. Fig5 illustrates the 1-round independent biclique on PRINCEcore, including base computation, Δ_i-differentials and ∇_j-differentials.

Step1. Fix $P_0 = 0_{(64)}$,and derive $S_0 = f_{K[0,0]}(P_0)$ with key $K[0,0]$ .This process is called as base computation (Fig5, left).

Step2. Encrypt P_0 under different keys $K[0, j]$ ($j \in \{0,...,2^8-1\}$) and derive $P_0 \xrightarrow[f]{K[0,j]} S_j$ (Fig5,middle). This process, is called Δ_i-differentials computation, has the same starting point and ending point as the base computation, so the computation complexity of this process is determined by the difference between $K[0, j]$ and $K[0,0]$.

Step3. Decrypt S_0 under different keys $K[i,0]$ ($i \in \{0,...,2^8-1\}$) and derive $P_i \xleftarrow[f^{-1}]{K[i,0]} S_0$ (Fig5,right). This process, i.e. ∇_j-differentials computations, are from over the same part of the cipher, so its computation complexity is determined by the difference between $K[i,0]$ and $K[0,0]$.

From Fig5, we can see that the forward differential trails and the backward differential trails do not share any non-linear components during the first round. Therefore, it is easy to find that $P_i \xrightarrow[f]{K[i,j]} S_j$ ($i, j \in \{0,...,2^8-1\}$) is true. So we can construct a 8-dimensional balanced independent biclique for every key group. In forward differential, Δ_i-differences are injected into nibble 0 and nibble 1 of key k_1 and it affects five nibbles after one round. In backward differential, ∇_j-differences are injected into nibble 8 and nibble 9 of key k_1 and it influences eight nibbles plaintexts after one round.

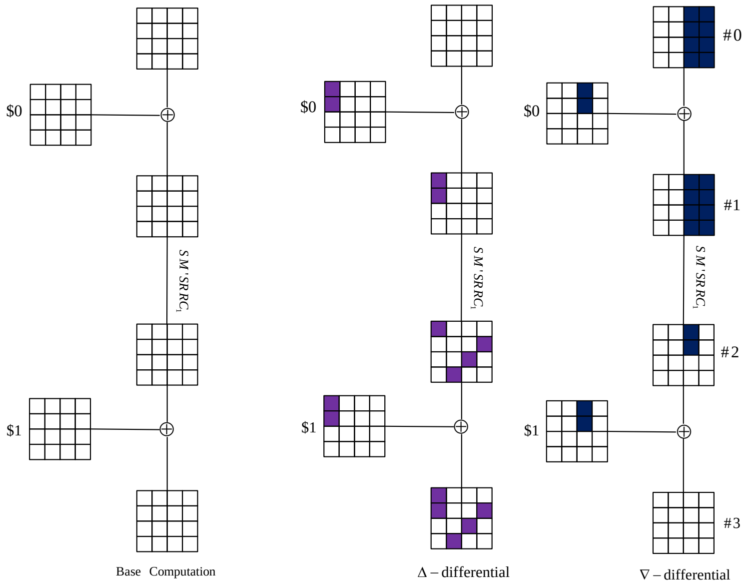

Fig5.Balanced independent biclique over the first round of PRINCEcore

3.3 Matching with Precomputation

In this section, we apply matching with precomputation technique on the remaining rounds to reduce computational complexity. Because PRINCEcore is an involutive structure, we choose the two nibbles of the state in the middle round, i.e, before 6-th round, as the matching values.

Forward computation. Firstly, we encrypt S_j under key $K[0,j]$ and store 2^8 precomputation values of $S_j \xrightarrow[g_1]{K[0,j]} v_{0,j}$. Then, we encrypt the same S_j under keys $K[i,j]$, $S_j \xrightarrow[g_1]{K[i,j]} v_{i,j}$. Because both the processes have the same starting point, the recomputation complexity is determined by the difference between $K[0,j]$ and $K[i,j]$.

Backward computation. Firstly ,we should ask the oracle to encrypt plaintext P_i with the secret key K_{secret} and obtain ciphertext C_i .Then, we decrypt C_i under key $K[i,0]$ and store 2^8 precomputation values of $v_{i,0} \xleftarrow[g_2^{-1}]{K[i,0]} C_i$. Lastly, we decrypt the same C_i under keys $K[i,j]$, $v_{i,j} \xleftarrow[g_2^{-1}]{K[i,j]} C_i$. Because both the processes have the same ending point, the recomputation complexity is determined by the difference between $K[i,0]$ and $K[i,j]$.

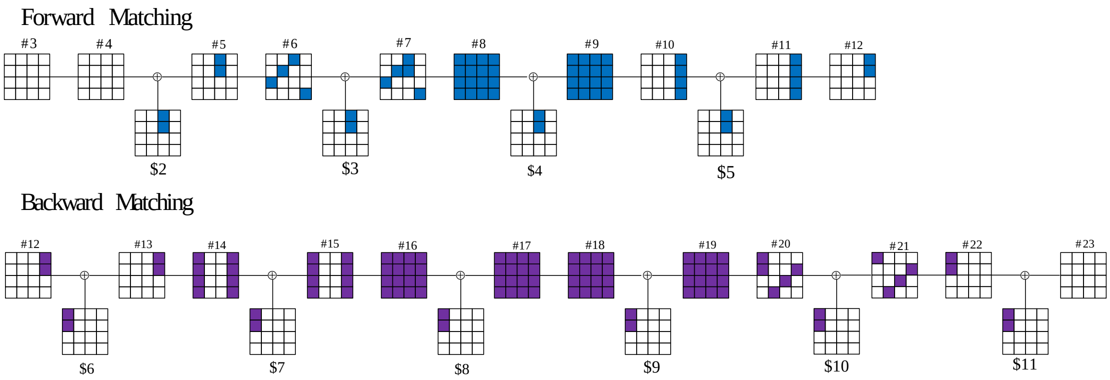

Fig6. Recomputations for PRINCEcoer in forward and backward direction of balanced biclique attack.

3.4 Complexity Analysis

There are 192 S/S^{-1}-boxes in total in all round transformation of PRINCEcore. In the forward matching, we only need to compute 29 S-boxes; Fig6 top illustrates the active nibbles in the S operations in the states directly after the key additions during the matching. In the backward matching, we only need to compute 46 S-boxes. Fig6 bottom illustrates the active nibbles in the S operations in the states directly after the key additions during the matching. During the matching with precomputation, there are 75 S-boxes to be recomputed.

Computation Complexity. For a key group of 2^{16} keys, the recomputation complexity is $C_{recomp} = 2^{16} \cdot \frac{75}{192} \approx 2^{14.64}$. The complexity for constructing a biclique is $C_{biclique} = 2 \cdot 2^{8} \cdot \frac{1}{12} \approx 2^{5.42}$.The complexity of precomputations is $C_{precomp} = 2^{8} \cdot \frac{11}{12} \approx 2^{7.88}$. The complexity to eliminate false positives is $C_{falsepos} = 2^{8}$. Therefore, the total computational complexity is

$$
\begin{aligned}
C_{full} &= 2^{k-2d}(C_{biclique} + C_{recomp} + C_{precomp} + C_{falsepos}) \\
&= 2^{48} \cdot (2^{5.42} + 2^{14.64} + 2^{7.88} + 2^{8}) \\
&= 2^{62.67}
\end{aligned}
$$

Data complexity. The data complexity is determined by the encrypted plaintexts. We fix $P_0 = 0_{(64)}$ for every biclique and all the plaintexts P_i ($i \in \{0,\ldots,2^8-1\}$) share eight nibbles, so the data complexity does not exceed 2^{32} chosen plaintexts.

Memory complexity. During the precomputation, we need to store 2^8 values, so the memory complexity is 2^8.

4 Star-Based Independent-Biclique Attack on PRINCEcore

Inspired by their work [8], we first illustrate a star-based biclique attack on full rounds of PRINCEcore. A star-based biclique is different from balanced biclique with just one state in one vertex set and 2^{2d} states in the other ones. We construct a 1-round star-based independent biclique over the first round of PRINCEcore and apply the matching with precomputations on the remaining rounds.

4.1 Key Partitioning

We divide the 64-bit key space into 2^{48} 16-nibble key groups. The form of partition is as same as section 3.1, so we do not describe key partitioning in this section.

4.2 Constructing Single Round Star-Based Independent-Biclique of Dimension 8

Similar to balanced biclique, stars can be constructed efficiently from independent sets of differentials. Unlike balanced biclique, the necessary differentials form of a star-based biclique is different from the one of balanced biclique. For PRINCEcore, it is easy to construct a star-based independent biclique over the first round. We place the star at the beginning of the cipher, let x be the plaintext and $y_{i,j}$ be the output of round 1 encryption. Fig7 shows the 1-round star-based independent biclique, including the base computation, Δ_i-differentials and ∇_j-differentials. Both Δ_i-differentials and ∇_j-differentials do not share any non-linear components.

Step1. Let x_0 be the plaintext, and obtain $y_{0,0} = f_{K[0,0]}(x_0)$ with key $K[0,0]$. This process is called as base computation (Fig7, left).

Step2. Encrypt x_0 under different keys Δ_i^K ($i \in \{0,...,2^8-1\}$)and obtain $x_0 \xrightarrow[f]{\Delta_i^K} y_i$ (Fig7,middle). This process has the same starting point and ending point as base computation.

Step3. Encrypt x_0 from over the same part of the cipher as step 2 under different keys ∇_j^K ($j \in \{0,...,2^8-1\}$) and obtain $x_0 \xrightarrow[f]{\nabla_j^K} y_j$ (Fig7,right). This process also has the same starting point and ending point as base computation.

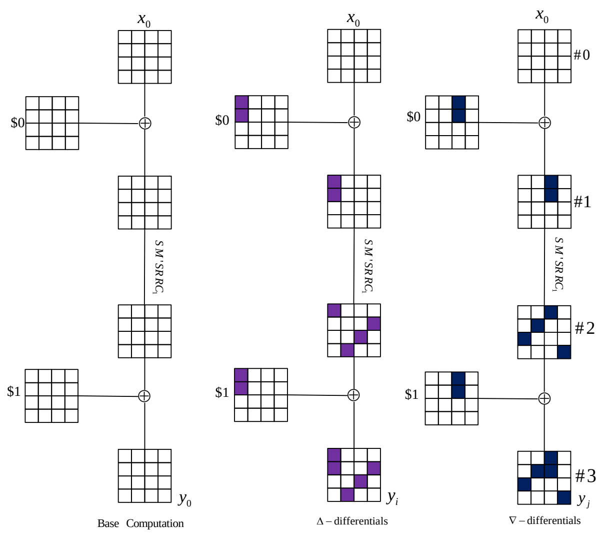

Fig7. Star-based independet biclique over the first round of PRINCEcore

From Fig7, we can observe that both the differentials do not share any non-linear components during the first round. So, it is easy to prove that $x \xrightarrow[f]{K[0,0]\oplus\Delta_i^K\oplus\nabla_j^K} y_{i,j}$ ($i, j \in \{0,\ldots,2^8-1\}$) is true. Therefore, we successfully construct a star-based independent biclique for every key group.

4.3 Matching with Precomputation

In this section, we apply matching with precomputation technique on the remaining rounds. We choose the same nibbles of the state in the middle round, i.e., before 6-th round, as the matching values.

Forward matching. In the forward direction of matching, starting in round 2, a part of the state has to be recomputed. Because difference propagation in these differentials over one round is non-overlapping, no S-boxes has to be recomputed in 2-th round. However, starting in 3-th round and forwards, the propagation affects the whole state (Fig8, top). So 54 S-boxes have to be recomputed in the forward direction of matching.

Backward matching. In the backward direction of matching, 42 S-boxes need to be recomputed (Fig8, bottom).

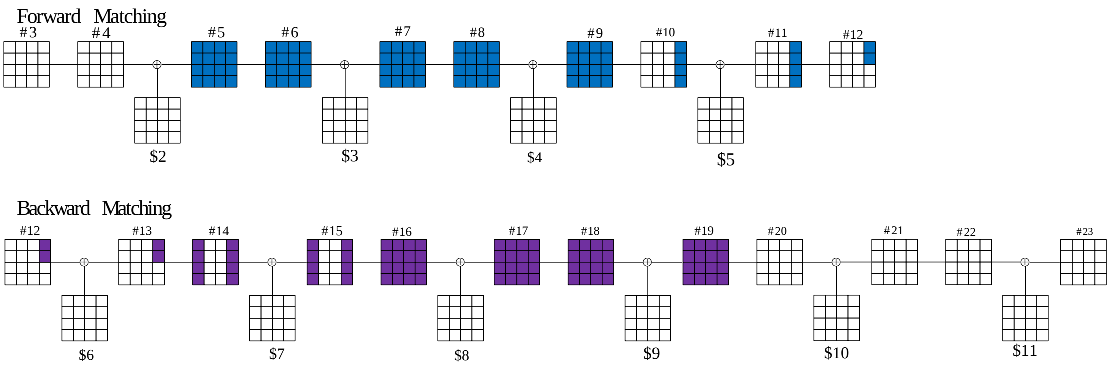

Fig8. Recomputations for PRINCEcore in forward and backward direction of star-based biclique attack.

4.4 Complexity Analysis

During the matching with precomputation, 96 S-boxes have to be recomputed.

Computation Complexity. For a key group of 2^{16} keys, the recomputation complexity is $C_{recomp} = 2^{16} \cdot \frac{96}{192} \approx 2^{15}$. The effort for constructing one biclique is $C_{biclique} = 2 \cdot 2^8 \cdot \frac{1}{12} \approx 2^{5.42}$. The complexity of precomputations is $C_{precomp} = 2^8 \cdot \frac{11}{12} \approx 2^{7.88}$. The complexity to eliminate false positives is $C_{falsepos} = 2^8$. Therefore, the total computational complexity is

$$
\begin{aligned}
C_{full} &= 2^{k-2d}(C_{bilique} + C_{recomp} + C_{precomp} + C_{falsepos}) \\
&= 2^{48} \cdot (2^{5.42} + 2^{15} + 2^{7.88} + 2^{8}) \\
&= 2^{63.02}
\end{aligned}
$$

Data complexity. The data complexity is determined by the encrypted plaintexts. We let x be the plaintext, so the data complexity will be 1. One known plaintext-ciphertext pair can sometimes be enough, and two known plaintext-ciphertext pairs yield a success probability of practically1.

Memory complexity. During the precomputation, we need to store 2^8 values, so the memory complexity is upper bounded by 2^8.

5 Conclusion

In this paper, we concentrate on independent biclique attack on full rounds PRINCEcore. We give two kinds of independent biclique attacks on full rounds PRINCEcore. One is a balanced-biclique attack with the lower computational complexity and data complexity than the previous works. Another is a first star-based biclique attack PRINCEcore with the optimal data complexity. i.e., we first utilize a star-based biclique (unbalanced biclique) to reduce the data complexity to the theoretically attainable minimum. Compared with previous biclique attack, our balanced-biclique attack has an advantage of computational complexity and data complexity. While the data complexity of our star-based independent biclique attack is optimal than the ones of existing kinds of biclique attacks. It's worth mentioning that the structure of biclique is important for the data complexity of the attack, whereas the length of the biclique seems to be correlated with the computational complexity.

References

1. Chae Lim and Tymur Korkishko. mCrypton-A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In Information Security Applications,volume 3786 of LNCS, pages 243-258, Springer, 2006.
2. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai,and Tetsu Iwata. The 128-bit Block Cipher CLEFIA (Extended Abstract).In Fast Software Encryption-FSE 2007, volume 4593 of LNCS, pages 181-195. Springer, 2007.
3. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda,Toru Akishita, and Taizo Shirai. Piccolo: An Ultra-Lightweight Blockcipher. In Cryptographic Hardware and Embedded Systems-CHES 2011, volume 6917 of LNCS, pages 342-57. Springer, 2011.
4. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, AxelPoschmann, Matthew J. B. Robshaw, Yannick Seurin, and Charlotte Vikkelso. PRESENT: An Ultra-Lightweight Block Cipher. In Cryptographic Hardware and Embedded Systems- CHES 2007, volume 4727 of Springer LNCS, pages 450-466,2007.

5. Zheng Gong, Svetla Nikova, and Yee Wei Law. KLEIN: A New Family of Lightweight Block Ciphers. In RFID Security and Privacy-RFIDSec 2011, volume 7055 of LNCS, pages 1-18. Springer, 2011.
6. Julia Borghoff, Anne Canteaut, Tim Guneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalcin. PRINCE-A Low-latency Block Cipher for Pervasive Computing Applications. Cryptology ePrint Archive, Report 2012/591, 2012. http://eprint.iacr.org/.
7. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique Cryptanalysis of the Full AES. Cryptology ePrint Archive, Report 2011/449, 2011. http://eprint.iacr.org/.
8. Andrey Bogdanov,Donghoon Chang, Mohona Ghosh et al. Biclqiues with minimal Data and Time Complexity for AES.ICISC2014.
9. HaHadi Soleimany,Céline Blondeau,Xiaoli Yu,Wenling Wu et.al. Reflection Cryptanalysis Of PRINCE-Like Ciphers. Journal of Cryptology. July 2015, Volume 28, Issue 3, pp 718-744.
10. Farzaneh Abed, Eik List, Stefan Lucks. On the security of the Core of PRINCE Against Biclique and Differential Cryptanalysis. Lacr Cryptology Eprint Archive.2012.
11. Li Lang, Du Guo-quan, ZENG Ting,et.al. Research on the PRINCE Algebraic Attack. MATHEMATICS IN PRACTICE AND THEORY, Vol.45, No.5, pp153-159, 2015.
12. Anne Canteaut, Thomas Tuhr, Henri Gilbert et.al. Multiple differential cryptanalysis of round-reduced PRINCE(Full version),FSE 2014.
13. Zheng Yuan, Zhen Peng, and Ming Mao. A Star-based Independent Biclique Attack on Full RoundsSQUARE.CryptologyePrintArchivec,Report 2015/1201, 2015.http://eprint.iacr.org/.

A Matching with One Nibble of Balanced Biclique Attack on PRINCEcore

If we only choose nibble 12 of the state in the middle round, i.e, before 6-th round, as the matching value (Fig 9), we need to recompute 28 S-boxes In the forward matching and we have to recomputed 41 S-boxes in the backward matching. During the matching with precomputation, 69 S-boxes need to be recomputed.

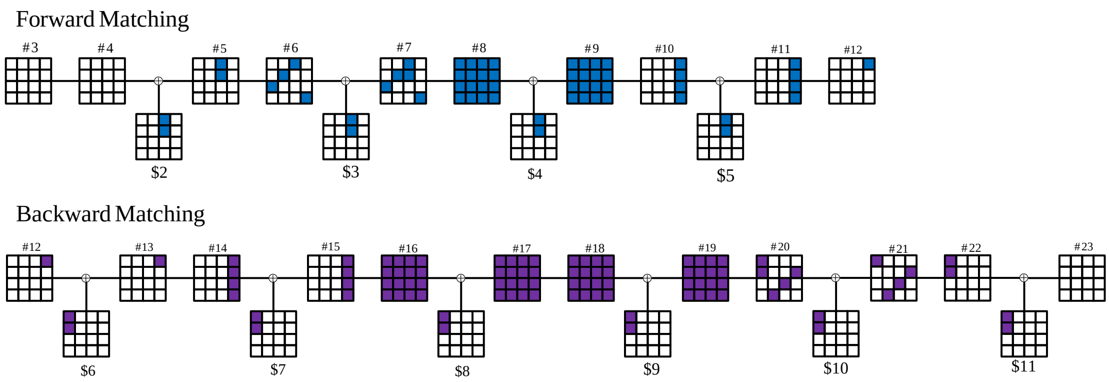

Fig 9. Matching with one nibble in forward and backward direction of balanced biclique attack.

Computation Complexity. For a key group of 2^{16} keys, the recomputation complexity is $C_{recomp} = 2^{16} \cdot \frac{69}{192} \approx 2^{14.52}$. The complexity for constructing a biclique is $C_{biclique} = 2 \cdot 2^{8} \cdot \frac{1}{12} \approx 2^{5.42}$. The complexity of precomputations is $C_{precomp} = 2^{8} \cdot \frac{11}{12} \approx 2^{7.88}$. The complexity to eliminate false positives is $C_{falsepos} = 2^{12}$. Therefore, the total computational complexity is

$$\begin{aligned} C_{full} &= 2^{k-2d}(C_{biclique} + C_{recomp} + C_{precomp} + C_{falsepos}) \\ &= 2^{48} \cdot (2^{5.42} + 2^{14.52} + 2^{7.88} + 2^{12}) \\ &= 2^{62.77} \end{aligned}$$

B Matching with One Nibble of Star-based Biclique Attack on PRINCEcore

If we onlye choose nibble 12 of the state in the middle round, i.e, before 6-th round, as the matching value (Fig 10), we need to recompute 53 S-boxes in the forward matching and we have to recomputed 37 S-boxes in the backward matching. During the matching with precomputation, 90 S-boxes need to be recomputed.

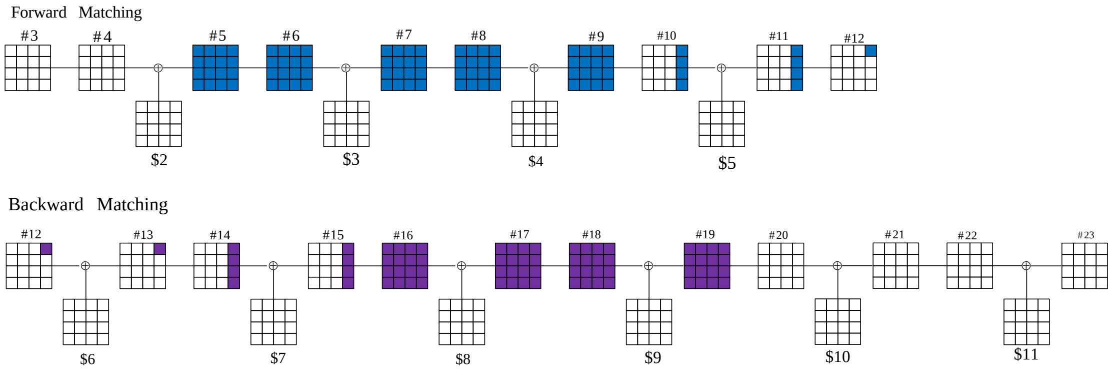

Fig10. Matching with one nibble in forward and backward direction of star-based biclique attack.

Computation Complexity. For a key group of 2^{16} keys, the recomputation complexity is $C_{recomp} = 2^{16} \cdot \frac{90}{192} \approx 2^{14.91}$.The complexity for constructing one biclique is $C_{biclique} = 2 \cdot 2^{8} \cdot \frac{1}{12} \approx 2^{5.42}$. The complexity of precomputations is $C_{precomp} = 2^{8} \cdot \frac{11}{12} \approx 2^{7.88}$. The complexity to eliminate false positives is $C_{falsepos} = 2^{12}$. Therefore, the total computation complexity is

$$
\begin{aligned}
C_{full} &= 2^{k-2d}(C_{bilique} + C_{recomp} + C_{precomp} + C_{falsepos}) \\
&= 2^{48} \cdot (2^{5.42} + 2^{14.91} + 2^{7.88} + 2^{12}) \\
&= 2^{63.1}
\end{aligned}
$$