Biclique Cryptanalysis of TWINE

Mustafa Çoban[1,2], Ferhat Karakoç[1,3], and Özkan Boztaş[1,4]

[1] TÜBİTAK BİLGEM UEKAE, 41470, Gebze, Kocaeli, Turkey
{mustafacoban, ferhatk, ozkan}@uekae.tubitak.gov.tr
[2] Sakarya University, Mathematics Department, Sakarya, Turkey
[3] Istanbul Technical University, Computer Engineering Department, Istanbul, Turkey
[4] Middle East Technical University, Institute of Applied Mathematics, Cryptography Department, Ankara, Turkey

Abstract. TWINE is a lightweight block cipher proposed at ECRYPT Workshop on Lightweight Cryptography 2011, Belgium. The cipher consists of 36 rounds and has two versions TWINE-80 and TWINE-128 supporting key lengths of 80 and 128 bits, respectively. The block length of the two versions is 64-bit. In this paper, we present the first single-key attacks on the both versions of the cipher. In these attacks, we use the recently developed biclique technique. The complexities of the attacks on TWINE-80 and TWINE-128 are $2^{79.10}$ and $2^{126.82}$ respectively and the data requirement for the two attacks is 2^{60}.

Key words: TWINE, lightweight block cipher, biclique cryptanalysis, meet-in-the-middle attack.

1 Introduction

The needs for security and privacy issues in resource constraint platforms such as RFID tags and sensor nodes give rise to design lightweight cryptographic algorithms. Some of the lightweight algorithms recently proposed are HIGHT [6], PRESENT [2], KATAN/KTANTAN [3], PRINTcipher [7], KLEIN [4], LED [5], Piccolo [9], and TWINE [10].

In this paper, we give our cryptanalytic results on TWINE. TWINE supports two key lengths, 80 and 128 bits. For each key length, encryption functions are the same but the key schedules are different. Corresponding to the key lengths, we denote TWINE-80 and TWINE-128. To the best of our knowledge, the most powerful attack is the impossible differential attacks against 23-round TWINE-80 and 24-round TWINE-128 with time complexities of $2^{76.88}$ and $2^{115.10}$ encryptions, respectively [10]. In this paper, we present attacks on the full TWINE-80 and TWINE-128. In these attacks, we use the biclique technique [1]. The complexities of the attacks on TWINE-80 and TWINE-128 are $2^{79.10}$ and $2^{126.82}$, respectively.

The organization of the paper is as follows. In Section 2 we give the notation which we use throughout the paper and a short description of TWINE algorithm. We overview the biclique technique in Section 3. In Section 4 and 5 we present the attacks on TWINE-80 and TWINE-128, respectively. We conclude the paper in Section 6.

2 Notation and a Short Description of TWINE

2.1 Notation

Throughout the paper, we use the following notations:

A	: a bit string
$A(i)$	: i-th nibble of A. The left most nibble is $A(0)$.
$A(i, j, ..., k)$	: concatenation of i, j, ... , k-th nibbles of A.
$A(i - j)$	: concatenation of i, $(i + 1)$, ..., j-th nibbles of A where $i \leq j$.
$A[i]$	: i-th bit of A. The left most bit of A is $A[0]$.
$A[i, j, ..., k]$	: concatenation of i, j, ..., k-th bits of A.
$A[i - j]$	: concatenation of i, $(i + 1)$, ..., j-th bits of A where $i \leq j$.
$A <<< i$	: i-bit cyclic left shift of A.
$A\|\|B$	: concatenation of A and B.
RK_i	: 32-bit round key used in the i-th round where $1 \leq i \leq 36$.
K^i	: k-bit value calculated in the key schedule where k is the key length of the cipher.
X_i	: the output of the i-th round where X_0 is the plaintext and X_{36} is the ciphertext.

2.2 TWINE

TWINE is a block cipher supporting two key lengths, 80 and 128 bits. The global structure of TWINE algorithm is a variant of Type 2 generalized Feistel structure [12] with 16 4-bit sub-blocks. Each version of the algorithm has the same round function depicted in Figure 1 and consists of 36 rounds.

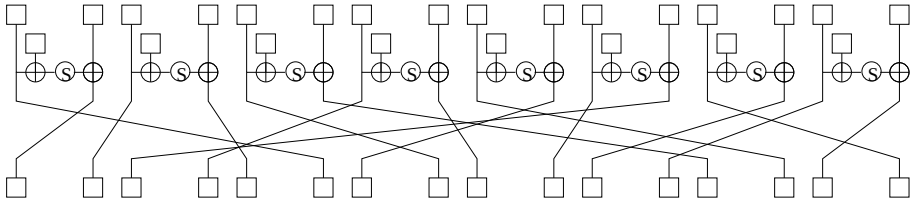

Fig. 1. One round of TWINE

In the round function, the key addition is applied before the S-box operation as seen in the figure and then the permutation is performed. In the last round the permutation does not exist.

The two versions of TWINE have key schedules which consist of S-box, round constant addition, $CON^i = CON^i_H||CON^i_L$, and permutation operations.

The key schedule of TWINE-80 generates 36 32-bit round keys from the 80-bit master key as follows:

1. $K^0 = K$

2. $RK_1 = K^0(1, 3, 4, 6, 13, 14, 15, 16)$
3. for i=1,2,...,35 do the followings
4. – $K^i = K^{i-1}$
   – $K^i(1) = K^i(1) \oplus S[K^i(0)]$
   – $K^i(4) = K^i(4) \oplus S[K^i(16)]$
   – $K^i(7) = K^i(7) \oplus (0||CON_H^i)$
   – $K^i(19) = K^i(19) \oplus (0||CON_L^i)$
   – $K^i(0, 1, 2, 3) = K^i(0, 1, 2, 3) <<< 4$
   – $K^i = K^i <<< 16$
   – $RK_{i+1} = K^i(1, 3, 4, 6, 13, 14, 15, 16)$

TWINE-128 has the following key schedule which generates 36 32-bit round keys from the 128-bit master key.

1. $K^0 = K$
2. $RK_1 = K^0(2, 3, 12, 15, 17, 18, 28, 31)$
3. for i=1,2,...,35 do the followings
4. – $K^i = K^{i-1}$
   – $K^i(1) = K^i(1) \oplus S[K^i(0)]$
   – $K^i(4) = K^i(4) \oplus S[K^i(16)]$
   – $K^i(23) = K^i(23) \oplus S[K^i(30)]$
   – $K^i(7) = K^i(7) \oplus (0||CON_H^i)$
   – $K^i(19) = K^i(19) \oplus (0||CON_L^i)$
   – $K^i(0, 1, 2, 3) = K^i(0, 1, 2, 3) <<< 4$
   – $K^i = K^i <<< 16$
   – $RK_{i+1} = K^i(2, 3, 12, 15, 17, 18, 28, 31)$

Our attacks are independent from the S-box and constants so we skip the details. For a complete description of the algorithm one can refer to [10].

3 An Overview of the Biclique Cryptanalysis Technique

In this section, we give an overview of the biclique technique on block ciphers proposed in [1]. In the biclique attack, firstly the key space is divided into 2^{k-2d} subspaces in which there exists 2^{2d} keys where k and d is the key length and the dimension of the biclique, respectively. Then, for all the key subspaces the two steps, biclique construction and meet-in-the-middle attack, are applied. To perform the two steps, firstly the cipher E is considered as a composition of three parts f, g, and h where $E = h \circ g \circ f$. Then, a biclique is constructed on the first or last part (in the attack on TWINE-80 and TWINE-128 we construct the bicliques on the first and last part as done in [11] and [1], respectively). After that, the meet-in-the-middle attack is applied on the remaining parts. In this section we give the attack idea in the case that the biclique is constructed in the first part. The attack idea is similar for the other case.

A d dimensional biclique is a 3-tuple $(\{P_i\}, \{S_j\}, \{K_{i,j}\})$ such that $f_{K_{i,j}}(P_i) = S_j$, $\forall i, j \in \{0, 1, ..., 2^d - 1\}$. Two methods to construct a biclique are given in [1].

In this work, we use one of the methods called independent related-key differentials. In this method, first S_0 is calculated from a chosen P_0 under the key $K_{0,0}$. Then, S_j and P_i values are calculated from P_0 and S_0 using $K_{0,j} = K_{0,0} \oplus \Delta_j^K$ and $K_{i,0} = K_{0,0} \oplus \nabla_i^K$, respectively. Lets the differential trails in forward and backward directions called as Δ_j and ∇_i, respectively. If the trails does not have common active non-linear operations such as S-boxes then the probability of the equation $f_{K_{i,j}}(P_i) = S_j$ where $K_{i,j} = K_{0,0} \oplus \nabla_i^K \oplus \Delta_j^K$ is 1 as proved in [1].

The second step is the meet-in-the-middle attack on $h \circ g$. In this step first $C_i = E_K(P_i)$ values are obtained from the encryption oracle. After that a partial matching is searched at some portion v between g and h. This matching step has two sub-steps also. Firstly, in the forward and backward direction the internal values which affect the value of v and does not depend on the value of i and j respectively calculated using $K_{0,j}$ and $K_{i,0}$. Then the remaining internal values which affect the values of v are calculated using $K_{i,j}$ for all i and j. The keys $K_{i,j}$ are selected as candidate keys which lead to a matching on v. The number of candidates will be approximately 2^{2d-m} in a subspace where m is the bit length of v. The total number of key candidates will be $2^{k-2d} \times 2^{2d-m} = 2^{k-m}$. At the end approximately 2^{k-m} encryptions using $\lceil \frac{k-m}{n} \rceil$ plaintext-ciphertext pairs are performed to eliminate all the wrong candidates.

4 Biclique Cryptanalysis of TWINE-80

In this section, we present a biclique attack on the full TWINE-80. Firstly, we divide the key space into 2^{72} subspaces of 2^8 keys each. Then for each key subspace we construct a biclique on the first 8 rounds and by using this biclique we apply the meet-in-the-middle attack on the last 28 rounds of the cipher.

4.1 Key Partitioning

The base keys of the key subspaces are of the form $K_{0,0} = (* * * * \,|\, * * * 0|\, * * * *|0 * * * \,|\, * * * *)$, where two nibbles are fixed to zero and the remaining 18 nibbles determine the subspace. The 2^8 keys $\{K_{i,j}\}$ in the subspaces are taken as follows

$$K_{i,j} = K_{0,0} \bigoplus (0000|000i|0000|j000|0000), i, j \in \{0, 1, \ldots, 2^4 - 1\}.$$

4.2 Biclique Construction on 8 Rounds

First of all, S_0 is calculated from a randomly chosen P_0 as $S_0 = f_{K_{0,0}}(P_0)$ where f is the first 8 rounds of the cipher. Then a biclique is constructed using the following two sets of 2^4 related-key differentials with respect to the base computation $P_0 \xrightarrow{K_{0,0}} S_0$.

1. Δ_j-differentials over f. Each related-key differential in the first set maps input difference $\Delta P = 0$ to an output difference $\Delta_j = \Delta S = S_0 \bigoplus S_j$ under the key difference $\Delta_j^K = (0000|0000|0000|j000|0000)$.

$$0 \xrightarrow[f]{\Delta_j^K} \Delta_j.$$

2. ∇_i-differentials over f^{-1}. Each related-key differentials in the second set maps input difference $\Delta S = 0$ to an output difference $\nabla_i = \nabla P = P_0 \bigoplus P_i$ under the key difference $\nabla_i^K = (0000|000i|0000|0000|0000)$.

$$0 \xrightarrow[f^{-1}]{\nabla_i^K} \nabla_i.$$

Δ_j and ∇_i differentials are given in Figure 2.

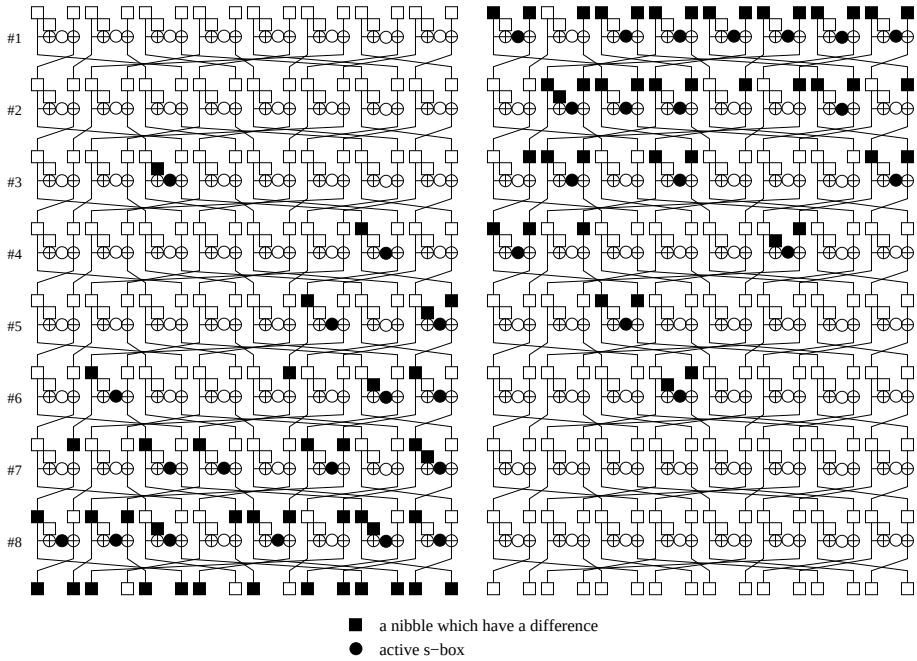

Fig. 2. Δ_j and ∇_i differential trails for TWINE-80 on the left and right respectively.

As seen in the figure the differential trails do not share any active S-box. Thus

$$\nabla_i \xrightarrow[f]{\Delta_j^K \oplus \nabla_i^K} \Delta_j, \forall i, j \in \{0, 1, \ldots, 2^4 - 1\}.$$

As a result, the triple $(\{P_i\}, \{S_j\}, \{K_{i,j}\})$ with the definition

$$\begin{gathered} P_i = P_0 \bigoplus \nabla_i, \\ S_j = S_0 \bigoplus \Delta_j, \\ K_{i,j} = K_{0,0} \bigoplus \Delta_j^K \bigoplus \nabla_i^K \end{gathered}$$

is a 8-round biclique of dimension 4.

Note that there is no difference in the 2-nd nibble of the plaintext in the biclique. Thus we can use the plaintexts whose 2-nd nibble is a fixed value. This reduces the data requirement of the attack to 2^{60}.

4.3 The Meet-in-the-Middle Step

By the biclique construction, 2^4 plaintexts P_i and 2^4 intermediate states S_j are available with corresponding $K_{i,j}$'s. First of all, we obtain C_i in the chosen plaintext scenario. Then, we check if there is some i, j such that

$$C_i \xrightarrow[g^{-1} \circ h^{-1}]{K_{i,j}} S_j. \tag{1}$$

where g and h as the composition of the rounds from the beginning of the 9-th round to the end of the 19-th round and from the beginning of the 20-th round to the end of the 36-th round respectively. For each one of the 2^{72} key subspaces, the complexity of this stage is 2^8. Hence, the overall complexity of the attack will be near exhaustive search, but we can reduce this complexity applying the attack given in Algorithm 1. In the algorithm the nibble $X_{19}(3)$ is taken as the matching variable v. To meet on v, we do partial calculations in forward direction starting from S_j and in backward direction starting from C_i.

Algorithm 1

1: S_j and C_i's are given.
2: **for** j in 0,1,...15 **do**
3: Calculate the nibbles colored in gray in the forward direction in Figure 3 and $S[X_{10}(2) \oplus RK_{10}(1)]$, $S[X_{11}(12) \oplus RK_{11}(6)]$, $S[X_{12}(2) \oplus RK_{12}(1)]$, $S[X_{12}(6) \oplus RK_{12}(3)]$, $S[X_{12}(2) \oplus RK_{12}(1)]$, $S[X_{13}(12) \oplus RK_{13}(6)]$ using $K_{0,j}$ and S_j.
4: **for** i in 0,1,...15 **do**
5: Calculate the nibbles colored in black in the forward direction in Figure 3 using $K_{i,j}$ and the values calculated in step 3.
6: Store $X_{19}(3)$ in the $(16 \times i + j)$-th cell of a table called A.
7: **end for**
8: **end for**
9: **for** i, j in 0,1,...15 **do**
10: Calculate the nibbles colored in black in the backward direction in Figure 3 using C_i and $K_{i,j}$.
11: **if** The calculated value of $X_{19}(3)$ is equal to the value in the $(16 \times i + j)$-th cell of A **then**
12: **if** $K_{i,j}$ satisfies another plaintext-ciphertext pair **then**
13: Output $K_{i,j}$ as the right key
14: **end if**
15: **end if**
16: **end for**

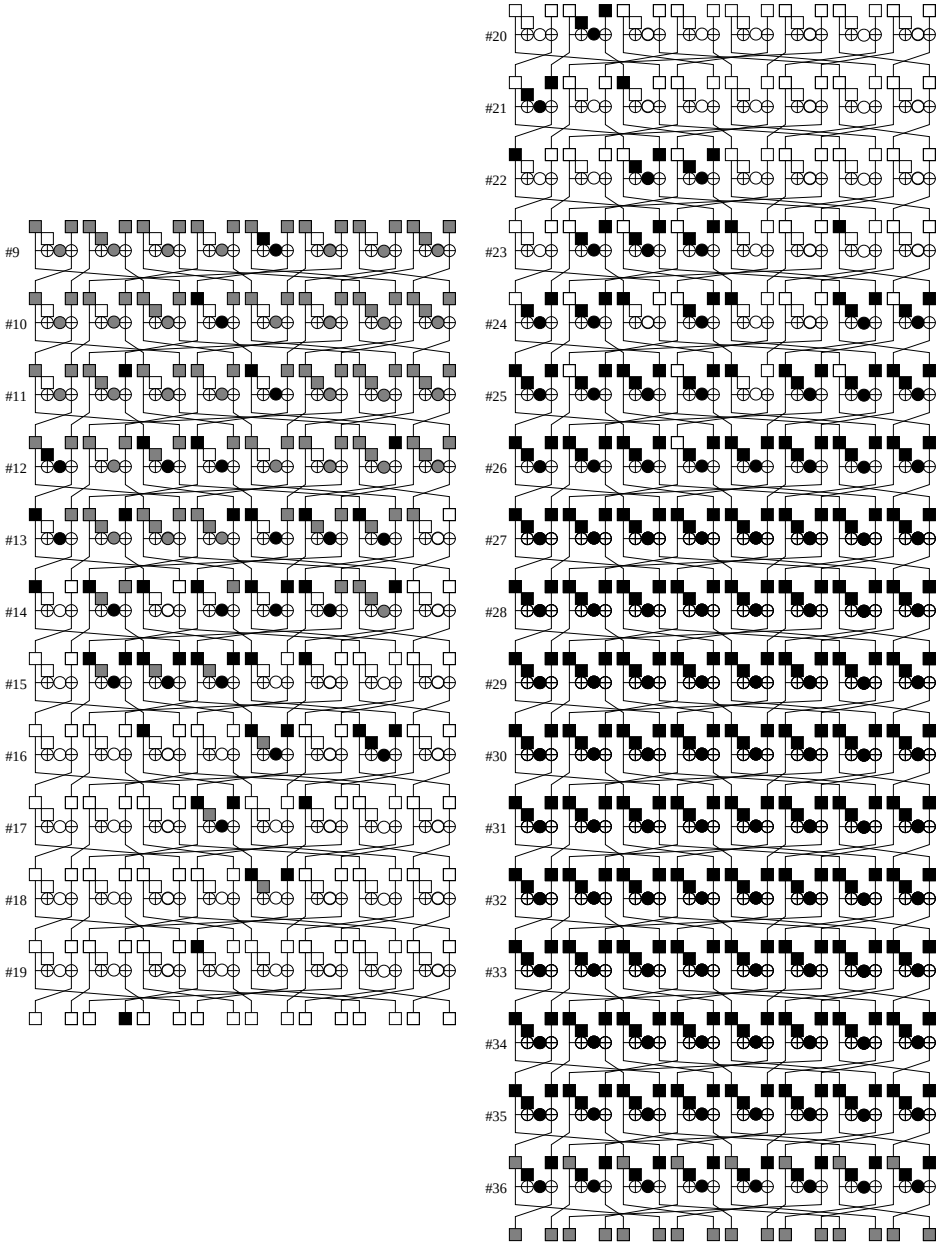

Fig. 3. Meet-in-the-middle step for TWINE-80. Note that the permutation in the last round actually does not exist.

4.4 The Attack Complexity

The computational complexity of the attack is composed of several parts. In the biclique construction step, for each of the 2^{72} key subspaces we perform 2^4

8-round encryptions to compute 2^4 intermediate states S_j and $2^4 - 1$ 8-round decryptions to compute 2^4 P_i's. Therefore, $\frac{2^4 \times 8 + (2^4-1) \times 8}{36} \approx 2^{2.78}$ encryptions are performed to construct a biclique. In the meet-in-the-middle attack given in Algorithm 1, the total number of S-box calculated for 2^4 and 2^8 different keys are 30 and 127 as seen in Figure 3 as gray and black S-boxes, respectively. Since the average number of remaining keys after the condition in Step 11 is $2^8 \times 2^{-4} = 2^4$, we perform 2^4 encryption operations in Step 12. Thus the total number of operations in the algorithm is $\frac{2^4 \times 30 + 2^8 \times 127}{36 \times 8} + 2^4 \approx 2^{7.03}$ encryptions. Therefore, $2^{2.78} + 2^{7.03} \approx 2^{7.10}$ encryptions are performed for each key subspace. As a result, the overall complexity of the attack on the full TWINE-80 is approximately $2^{79.10}$ encryptions with 2^8 memory.

5 Biclique Cryptanalysis of TWINE-128

In this section, we introduce a biclique attack on the full TWINE-128. In this attack we divide the key space considering K^{32}. The attack steps are similar to the attack steps given in the previous section. Firstly, we divide the key space into 2^{120} subspaces of 2^8 keys each and then for each key subspace we construct a biclique on the last 11 rounds and by using this biclique we perform the meet-in-the-middle attack on the first 25 rounds.

5.1 Key Partitioning

The key subspaces are enumerated by 2^{120} base keys of the form $K^{32}_{0,0} = (****|****|****|****|0***|**0*|****|****)$. The 2^8 keys $\{K^{32}_{i,j}\}$ in a subspace are taken as follows $K^{32}_{i,j} = K^{32}_{0,0} \bigoplus (0000|0000|0000|0000|i000|00j0|0000|0000), i, j \in \{0, 1, \ldots, 2^4 - 1\}$.

5.2 Contructing a Biclique

First of all, S_0 is calculated from a randomly chosen C_0 as $S_0 = h^{-1}_{K^{32}_{0,0}}(C_0)$ where h is the last 11 rounds of the cipher. Then a biclique is constructed using the following two sets of 2^4 related-key differentials with respect to the base computation $C_0 \xrightarrow[h^{-1}]{K^{32}_{0,0}} S_0$.

1. Δ_j-differentials over h^{-1}. Each related-key differential in the first set maps input difference $\Delta C = 0$ to an output difference $\Delta_j = \Delta S = S_0 \bigoplus S_j$ under the key difference $\Delta^K_j = (0000|0000|0000|0000|0000|00j0|0000|0000)$.

$$0 \xrightarrow[h^{-1}]{\Delta^K_j} \Delta_j.$$

2. ∇_i-differentials over h. Each related-key differentials in the second set maps input difference $\Delta S = 0$ to an output difference $\nabla_i = \nabla C = C_0 \bigoplus C_i$ under the key difference $\nabla^K_i = (0000|0000|0000|0000|i000|0000|0000|0000)$.

$$0 \xrightarrow[h]{\nabla_i^K} \nabla_i.$$

Δ_j and ∇_i differentials are given in Figure 4.

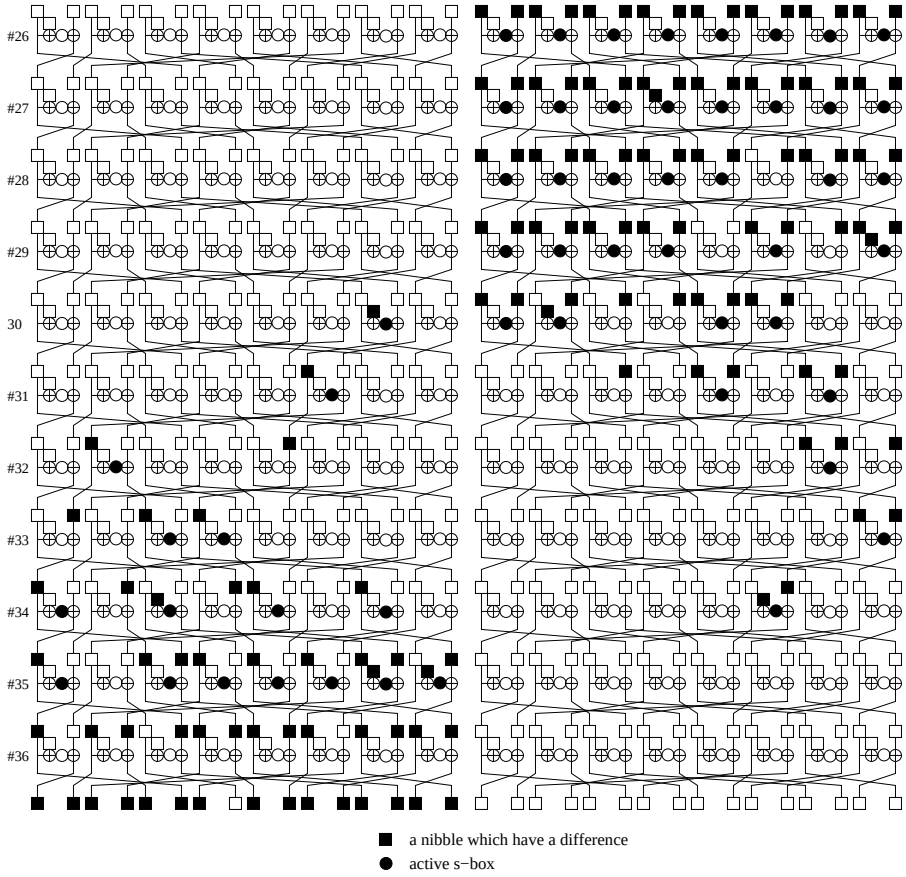

Fig. 4. ∇_i and Δ_j differential trails for TWINE-128 on the left and right respectively. Note that the permutation in the last round actually does not exist.

As seen in the figure the differential trails do not share any active S-box. Thus

$$\nabla_j \xrightarrow[h]{\Delta_j^K \oplus \nabla_i^K} \Delta_i, \forall i, j \in \{0, 1, \ldots, 2^4 - 1\}.$$

As a result, the triple $(\{S_j\}, \{C_i\}, \{K_{i,j}\})$ with the definition

$$\begin{aligned} C_i &= C_0 \bigoplus \nabla_i, \\ S_j &= S_0 \bigoplus \Delta_j, \\ K_{i,j} &= K_{0,0} \bigoplus \Delta_j^K \bigoplus \nabla_i^K \end{aligned}$$

is a 11-round biclique of dimension 4.

Note that there is no difference in the 7-th nibble of the ciphertext in the biclique. Thus we can use the ciphertexts whose 7-th nibble is a fixed value. This reduces the data requirement of the attack to 2^{60}.

5.3 The Meet-in-the-Middle Step

The attack is very similar to that in TWINE-80. In this attack we choose the subcipher f from the beginning of the 1-st round to the end of the 6-th round, and the subcipher g from the beginning of the 7-th round to the end of the 25-th round. The nibble $X_6(11)$ is taken as the matching variable v. The meet-in-the-middle step is given in Algorithm 2.

Algorithm 2

1: P_i and S_j's are given.
2: **for** i in 0,1,...15 **do**
3: Calculate the nibbles colored in gray in the forward direction in Figure 5 using $K^{32}_{i,0}$ and P_i.
4: **for** j in 0,1,...15 **do**
5: Calculate the nibbles colored in black in the forward direction in Figure 5 using $K^{32}_{i,j}$ and the values calculated in step 3.
6: Store $X_6(11)$ in the $(16 \times i + j)$-th cell of a table called A.
7: **end for**
8: **end for**
9: **for** j in 0,1,...15 **do**
10: Calculate the nibbles colored in gray in the backward direction in Figure 5 and $S[X_{23}(15) \oplus RK_{23}(6)]$, $S[X_{22}(11) \oplus RK_{22}(7)]$, $S[X_{21}(7) \oplus RK_{21}(2)]$, $S[X_{21}(3) \oplus RK_{21}(3)]$, $S[X_{20}(3) \oplus RK_{20}(3)]$, $S[X_{20}(15) \oplus RK_{20}(6)]$ using $K^{32}_{0,j}$ and S_j.
11: **for** i in 0,1,...15 **do**
12: Calculate the nibbles colored in black in the backward direction in Figure 5 using S_j, $K^{32}_{i,j}$ and the values calculated in step 10.
13: **if** The calculated value of $X_6(11)$ is equal to the value in the $(16 \times i + j)$-th cell of A **then**
14: **if** $K^{32}_{i,j}$ satisfies another plaintext-ciphertext pair **then**
15: Output $K^{32}_{i,j}$ as the right key
16: **end if**
17: **end if**
18: **end for**
19: **end for**

5.4 The Attack Complexity

For each key subspace, to construct a biclique $\frac{2^4 \times 11 + (2^4 - 1) \times 11}{36} \approx 2^{3.24}$ encryptions are performed. Also, in the meet-in-the-middle step $\frac{2^8 \times 96 + 2^4 \times 39}{36 \times 8} + 2^4 \approx$

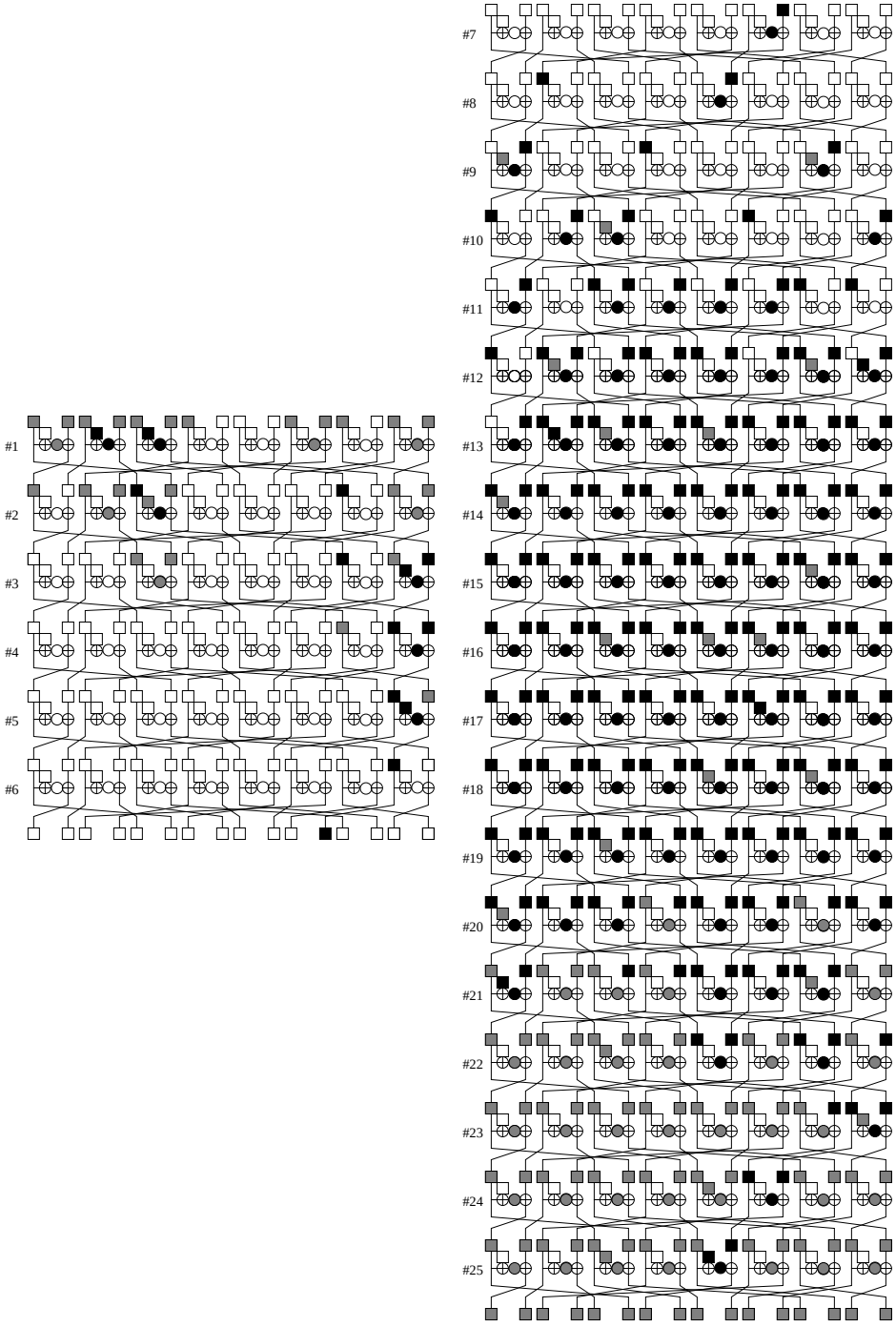

Fig. 5. Meet-in-the-middle step for TWINE-128.

$2^{6.69}$ encryptions are needed. Thus, $2^{3.24} + 2^{6.69} \approx 2^{6.82}$ encryptions are performed for each key subspace. As a result, the overall complexity of the biclique attack is approximately $2^{126.82}$ encryptions with 2^8 memory.

6 Conclusion

In this paper, we present the first single-key attacks on the full TWINE-80 and TWINE-128 by using recently developed biclique attack technique. In the both attacks 2^{60} data and 2^8 memory are required and the time complexities of the attacks on TWINE-80 and TWINE-128 are $2^{79.10}$ and $2^{126.82}$ encryption operations, respectively.

References

1. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full aes. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.
2. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
3. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. Katan and ktantan - a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.
4. Zheng Gong, Svetla Nikova, and Yee Wei Law. Klein: A new family of lightweight block ciphers. In Ari Juels and Christof Paar, editors, *RFIDSec*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2011.
5. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The led block cipher. In Preneel and Takagi [8], pages 326–341.
6. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. Hight: A new block cipher suitable for low-resource device. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer, 2006.
7. Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. Printcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.
8. Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.
9. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In Preneel and Takagi [8], pages 342–357.

10. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. Twine: A lightweight, versatile block cipher. In Proceedings of ECRYPT Workshop on Lightweight Cryptography, 2011. http://www.uclouvain.be/.
11. Shao zhen Chen and Tian min Xu. Biclique attack of the full aria-256. *IACR Cryptology ePrint Archive*, 2012:11, 2012.
12. Yuliang Zheng, Tsutomu Matsumoto, and Hideki Imai. On the construction of block ciphers provably secure and not relying on any unproved hypotheses. In Gilles Brassard, editor, *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 461–480. Springer, 1989.