Biclique Cryptanalysis of the Block Cipher SQUARE

Hamid Mala

Department of Information Technology, University of Isfahan, Isfahan, Iran

Abstract. SQUARE, an 8-round substitution-permutation block cipher, is considered as the predecessor of the AES. In this paper, inspired from the recent biclique attack on the AES [5], we present the first single-key attack on full SQUARE. First, we introduce a biclique for 3 rounds of SQUARE using the independent related-key differentials. Then, we present an attack on the full round of this cipher with a data complexity of about 2^{48} chosen plaintexts and a time complexity of about 2^{126} encryptions.

Key words: Block cipher, cryptanalysis, biclique, differential, SQUARE.

1 Introduction

The structure and mathematical backgrounds used in the design of the SQUARE [6] poses this block cipher as the predecessor of the Advanced Encryption Standard (AES) [7]. This 128-bit block cipher has an 8-round SPN structure and supports the key length of 128 bits. Designed based on the Wide Trail Strategy, SQUARE is secure against differential [1] and linear [11] cryptanalysis. The first cryptanalysis result on this block cipher is a square attack introduced by the designers [6]. This attack can break 6 rounds of the cipher with data, time and memory complexities of 2^{32} chosen plaintexts, 2^{72} encryptions and 2^{72} blocks of memory, respectively. Recently, a related-key boomerang attack on full rounds of this cipher has been introduced in [10] which recovers 16 subkey bits with 2^{36} encryptions and 2^{123} adaptively chosen plaintexts and ciphertexts.

Block cipher cryptanalysis and hash function cryptanalysis share several techniques. Differential Cryptanalysis, a technique originally invented for analysis of block ciphers, now is widely applicable to hash functions [8, 12, 13]. Inversely, two new techniques have been carried over from hash analysis to block cipher analysis. First, local collisions were used in related-key boomerang attacks on AES-192 and AES-256 [2–4], and recently, biclique cryptanalysis which was first introduced for analysis of the hash functions Skein-512 and the SHA-2 family [9], has been exploited to attack the full version of the 3 variants of the AES [5].

In this paper, inspired from the biclique cryptanalysis of the AES [5], we present an attack on SQUARE block cipher. To the best of our knowledge this is the first attack on the full round of this cipher in the single-key scenario. We find a 3-round biclique for the 3 initial rounds of SQUARE using independent related-key differentials. Then we use precomputation and recomputation techniques to recover the whole key. This is considered the second application of biclique attack on a block cipher. Table 1 summarizes our results along with previously known results on SQUARE.

The rest of this paper is organized as follows. Section 2 provides a brief description of the block cipher SQUARE. The concept of biclique attack is reviewed in Section 3. Our proposed biclique attack on SQUARE is presented, and its complexity is evaluated in Section 4. Finally, the paper is concluded in Section 5.

2 A Brief Description of SQUARE

The 128-bit block cipher SQUARE [6] has an 8-round SPN structure that supports 128-bit keys. Let us represent a 128-bit data or key by a 4×4 matrix $A = a_0a_1a_2a_3|a_4a_5a_6a_7|a_8a_9a_{10}a_{11}|a_{12}a_{13}a_{14}a_{15})$

Table 1. Summary of previous attacks and our new attack on SQUARE

Rounds	Data (CP)	Time (Encryptions)	Memory (Blocks)	Attack type	Source
5	2^{11}	2^{40}	small	Square	[6]
5	2^{32}	2^{40}	2^{32}	Square	[6]
6	2^{32}	2^{72}	2^{32}	Square	[6]
8	2^{123}	2^{36}	?	RK Boomerang	[10]
8	2^{48}	2^{126}	2^{16}	Biclique	This work

of bytes, where the byte located in row $i \in \{0,1,2,3\}$ and column $j \in \{0,1,2,3\}$ of A is denoted by a_{4i+j}. Each round of SQUARE applies the following 4 transformations to the state matrix.

- θ is a linear row-wise permutation with differential branch number 5. The state matrix is multiplied to a 4×4 MDS matrix $\mathcal{M}$ in $GF(2^8)$, where

$$\mathcal{M} = \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix}.$$

- γ is a nonlinear substitution layer including 16 invertible 8-bit S-boxes. In our attack, the exact values of the S-box table is not required, so we only consider its invertibility.
- π is a linear transformation transposing the state matrix.
- σ is a bitwise key XOR with the 128-bit round key.

To spot the transformation θ, γ, π and σ in round i, we use the notation $\theta^i, \gamma^i, \pi^i$ and σ^i. The round transformation of SQUARE $\rho_r(A) = \sigma \circ \pi \circ \gamma \circ \theta(A)$ is illustrated in Figure 1. The encryption consists of an initial application of the transformation θ^{-1}, then whitening with the 128-bit subkey rk^0, and finally 8 consecutive round functions.

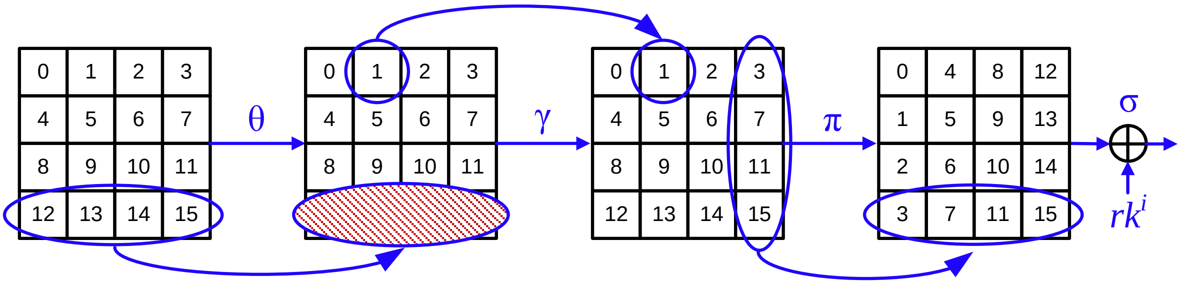

Fig. 1. Round transformation of the block cipher SQUARE

The key schedule of SQUARE generates 9 128-bit round keys $rk^0, rk^1, ..., rk^8$ given the master key K. Let each round key rk^i be regarded as a 4×4 array and $rk^i_{row(j)}$ be its jth row. To generate these round keys, rk^0 is initiated with K, and $rk^{i+1}, i = 0, 1, ..., 7$ is generated based on the following rule.

$$rk^{i+1}_{row(0)} = rk^i_{row(0)} \oplus rotl(rk^i_{row(3)}) \oplus C^i,$$
$$\text{for j=1, 2, 3: } rk^{i+1}_{row(j)} = rk^i_{row(j)} \oplus rk^{i+1}_{row(j-1)},$$

where C^i is a constant and the function $rotl$ rotates its four-byte argument by one byte to the left. Note that the key schedule lacks any diffusion and nonlinear part.

3 Biclique Cryptanalysis of Block Ciphers

Biclique cryptanalysis of block ciphers was originally introduced in [5] to cryptanalyze the 3 variants of the AES. In this section, we customize the concept introduced in [5] for the case where the biclique is constructed in the plaintext side.

Consider the block cipher E as a composition of 3 subciphers: $E = f \circ g \circ h$, where f is located in the plaintext side, g follows f, and h is located in the ciphertext side. Let S be the intermediate state obtained from the application of f on a plaintext P, i.e. $f_K(P) = S$. Suppose f connects 2^d plaintexts $\{P_i\}$ to 2^d intermediate states $\{S_j\}$ with 2^{2d} keys $\{K[i,j]\}$, where

$$\{K[i,j]\} = \begin{bmatrix} K[0,0] & ... & K[0,2^d-1] \\ \vdots & \vdots & \vdots \\ K[2^d-1,0] & ... & K[2^d-1,2^d-1] \end{bmatrix}$$

The 3-tuple $[\{P_i\},\{S_j\},\{K[i,j]\}$ is called a biclique of dimension d if

$$S_j = f_{K[i,j]}(P_i), \quad \forall i,j \in \{0,1,...,2^d-1\}$$

In other words, as illustrated in Figure 2, a biclique is a bipartite graph with $\{P_i\}$ and $\{S_j\}$ as the two parts of vertexes connected via 2^{2d} edges $f_{K[i,j]}$, where each edge has degree 2^d.

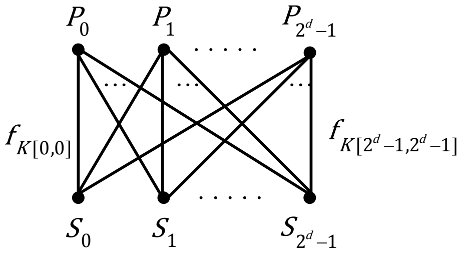

Fig. 2. d-dimensional biclique in the plaintext side

The biclique attack for a block cipher with key length of k bits is performed based on the following four steps.

Key Partitioning. The key space is partitioned into 2^{k-2d} groups of 2^{2d} keys each. A group is considered as a $2^d \times 2^d$ elements $K[i,j]$. For each group of keys:

Biclique Construction. Build a structure of 2^d plaintexts P_i and 2^d intermediate states S_j such that for all $i,j \in \{0,1,...,2^d-1\}$ the relation $S_j = f_{K[i,j]}(P_i)$ is satisfied. To reduce the data complexity, some plaintexts are reused in different bicliques.

Data Collection. Ask for the encryption of plaintexts P_i to obtain the corresponding ciphertexts C_i.

Matching check. Check if there exist i and j values such that $g \circ h_{K[i,j]}(S_j) = C_i$. This step can be performed by a precomputation-recomputation strategy to reduce the time complexity.

4 Biclique Cryptanalysis of SQUARE

In this section we present a biclique attack on full rounds of SQUARE. [5] introduces two methods to construct a biclique: using independent related-key differentials and using interleaving related key differential trails. Here, we follow the first approach to construct a biclique for the first three rounds of SQUARE. Moreover, we use precomputation and recomputation in the final step of the above attack procedure to reduce the time complexity.

4.1 Constructing a 3-Round Biclique of Dimension 8

In this section, we construct a 3-round biclique for the initial three rounds of SQUARE using two independent related-key differentials. We remind that although related-key differentials are used to construct the biclique, the attack is substantially performed in the single-key scenario.

As shown in Figure 3, left, let the key $K[0,0]$ map plaintext $P_0 = 0$ to intermediate state $S_0 = f_{K[0,0]}(P_0)$. Moreover, consider two sets of 2^d related-key differentials with respect to the base computation $P_0 \xrightarrow{K[0,0]} S_0$.

1. Δ_i**-differentials.** Each related-key differential in the first set maps input difference $\Delta S = 0$ to an output difference $\Delta_i = \Delta P = P_0 \oplus P_i$ under the key difference Δ_i^K.

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

According to Figere 3, middle, P_i is of the following form

$$P_i = P_0 \oplus (000*|000*|000*|00**) \oplus \theta(00i0|0000|000i|000i),$$

where '*' denotes any byte difference.

2. ∇_j**-differentials.** Each related-key differential in the second set maps input difference $\Delta S = \nabla_j$ to output difference $\Delta P = 0$ under the key difference ∇_j^K.

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

In fact, given $\Delta P = 0$ and $\Delta rk^2 = (jj00|jj00|jj00|jj00)$, the attacker computes $S_j, j = 0, ..., 2^d - 1$.

Δ_i-differentials and ∇_j-differentials are illustrated in truncated form in Figure 3. Since these two sets of differentials do not share any active S-box, we have

$$\nabla_j \xrightarrow[f^{-1}]{\Delta_i^K \oplus \nabla_j^K} \Delta_i, \quad \forall i, j \in \{0, 1, ..., 2^d - 1\}.$$

Note that all differentials are with respect to the $(P_0, S_0, K[0,0])$, so one can easily deduce

$$S_0 \oplus \nabla_j \xrightarrow[f^{-1}]{K[0,0] \oplus \Delta_i^K \oplus \nabla_j^K} P_0 \oplus \Delta_i, \quad \forall i, j \in \{0, 1, ..., 2^d - 1\}.$$

Hence, the triple$\{P_i, S_j, K[0,0]\}$ with the definition

$$\begin{aligned} &P_i = P_0 \oplus \Delta_i, \\ &S_j = S_0 \oplus \nabla_j, \\ &K[i,j] = K[0,0] \oplus \Delta_i^K \oplus \nabla_j^K \end{aligned}$$

exactly conforms the definition of a biclique of dimension 8.

4.2 Key Partitioning

The 2^{128} possible values in the key space are partitioned into 2^{112} groups of $2^{2d} = 2^{16}$ keys each with respect to the subkey rk^2. The groups are enumerated by 2^{112} base keys of the form $K[0,0] = (0*0*|****|****|****)$, where two bytes are fixed to zero and the remaining 14 bytes take all possible values. Note that the key schedule of SQUARE given each value of rk^2 uniquely determines one value for the master key, so this partitioning is equivalent to a partitioning of the master key space. Let $rk^2[0,0]$ be the subkey of round 2 generated based on the key schedule from the base master key $K[0,0]$. The 2^{16} keys $\{K[i,j]\}$ in a group with a $K[0,0]$ as the base key are constructed from $rk_2[i,j]$, where

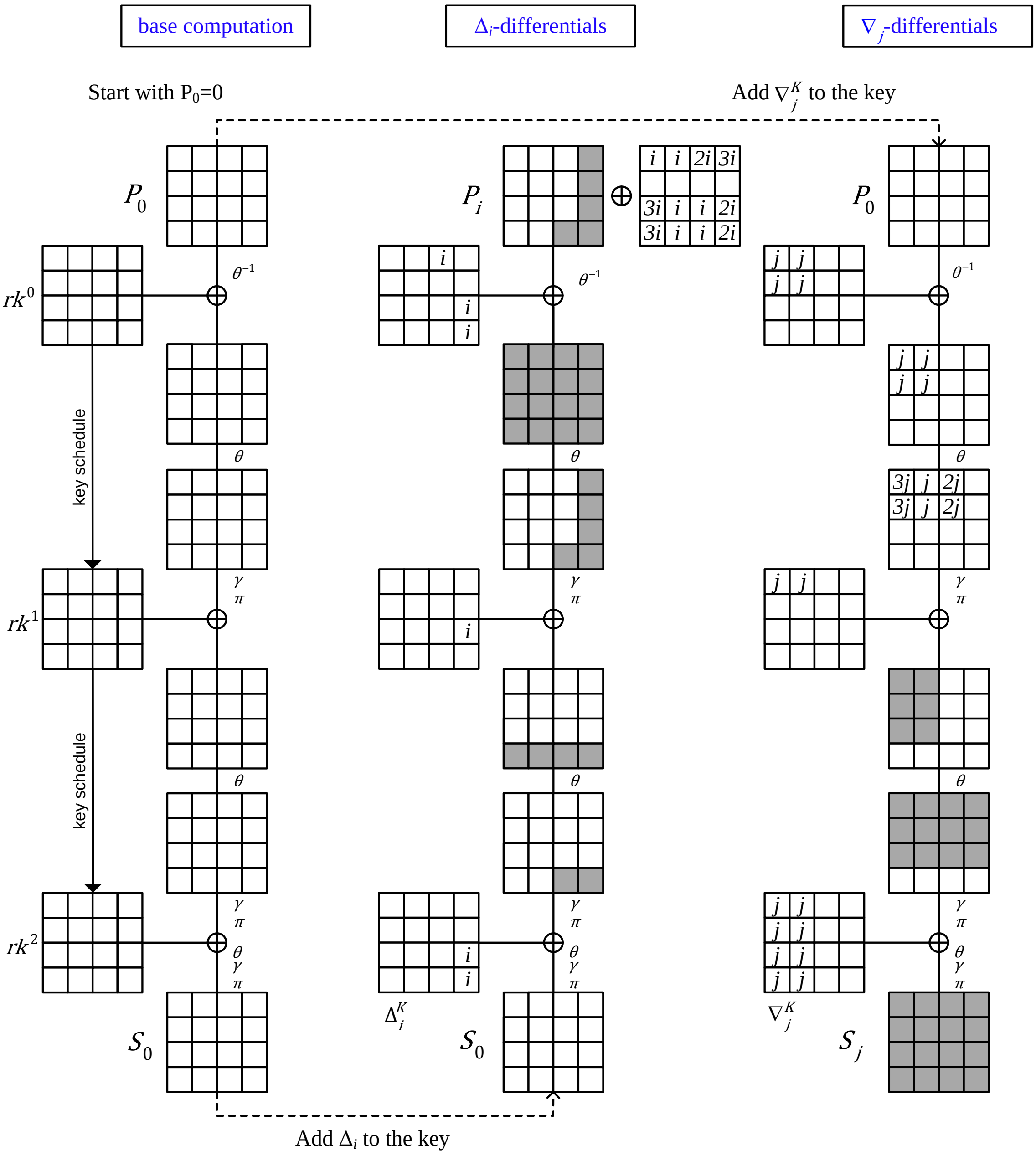

Fig. 3. 3-round biclique of SQUARE from combined independent differentials

$$rk^2[i,j] = rk^2[0,0] \oplus \mathit{\Delta} rk^2 = rk^2[0,0] \oplus (jj00|jj00|jj0i|jj0i), i,j \in \{0,1,...,2^8-1\}.$$

Following this method, the adversary partitions the rk^2 subkey space, and hence the master key space into 2^{112} groups of 2^{16} keys each.

4.3 The Attack Procedure

The attack mainly follows the 4-step procedure mentioned in Section 3. Biclique construction and key partitioning steps were described in Subsections 4.1 and 4.2, respectively. Data collection step is performed based on the chosen plaintext scenario. So, we here elaborate only the matching check stage.

Recall that for each biclique 2^8 plaintexts P_i and 2^8 intermediate states S_j are available, and there is a unique path from each P_i to each S_j through the key $K[i,j]$. Knowing this relation, for $i = 0,1,...,2^d-1$ the adversary obtains C_i in the chosen plaintext scenario. Then she has to check if there is some j such that

$$C_i \xrightarrow[h^{-1}\circ g^{-1}]{K[i,j]} S_j \qquad\qquad (1).$$

The complexity of this stage is 2^{2d} for each of the 2^{k-2d} bicliques. So the overall time complexity will be near exhaustive search, but we can reduce this complexity with precomputation and meet-in-the-middle technique. To do this, first, we perform and store 2^d partial encryptions and 2^d partial decryptions.

$$\text{for } j=0,...,2^d-1: \quad S_j \xrightarrow[g]{K[0,j]} \overrightarrow{v} \text{ and for } i=0,...,2^d-1: \quad \overleftarrow{v} \xleftarrow[h^{-1}]{K[i,0]} C_i$$

up to some matching variable v, which here is a byte of the intermediate state in the junction of g and h subciphers. Thus, to check equation (1) for a particular i, j, we need to recompute only parts of the cipher that differ from the stored values. This approach provides computational advantage of about several bits.

We choose the subcipher g from the σ^3 through π^6 transformation, and h subcipher from the output of g through σ^8. The byte with index zero in the output of π^6 is taken as the matching variable v. Now let us see how the precomputations reduce the computations $S_j \xrightarrow[g]{K[i,j]} \overrightarrow{v}$ and $\overleftarrow{v} \xleftarrow[h^{-1}]{K[i,0]} C_i$.

Encryption direction. The difference between computation $S_j \xrightarrow[g]{K[i,j]} \overrightarrow{v}$ and the precomputation $S_j \xrightarrow[g]{K[0,j]} \overrightarrow{v}$ is influenced by the difference between keys $K[i,j]$ and $K[0,j]$. The difference between different round keys is uniquely determined by the key schedule from $\mathit{\Delta} rk^2 = (0000|0000|000i|000i)$. Hence, as illustrated in Figure 4, to recompute the new value of v we have to recompute 15 S-boxes in round 3, $\frac{1}{4}\theta$ and 4 S-boxes in round 4, $\frac{1}{16}\theta$ and one S-box in round 5.

Decryption direction. The difference between computation $\overleftarrow{v} \xleftarrow[h^{-1}]{K[i,j]} C_i$ and the precomputation $\overleftarrow{v} \xleftarrow[h^{-1}]{K[i,0]} C_i$ is influenced by the difference between keys $K[i,j]$ and $K[i,0]$. The difference between different round keys is uniquely determined by the key schedule from $\mathit{\Delta} rk^2 = (jj00|jj00|jj00|jj00)$. Hence, as illustrated in Figure 5, to recompute the new value of v we have to recompute 6 S-boxes and $\frac{1}{4}\theta$ in round 8, and 4 S-boxes and $\frac{1}{16}\theta$ in round 6.

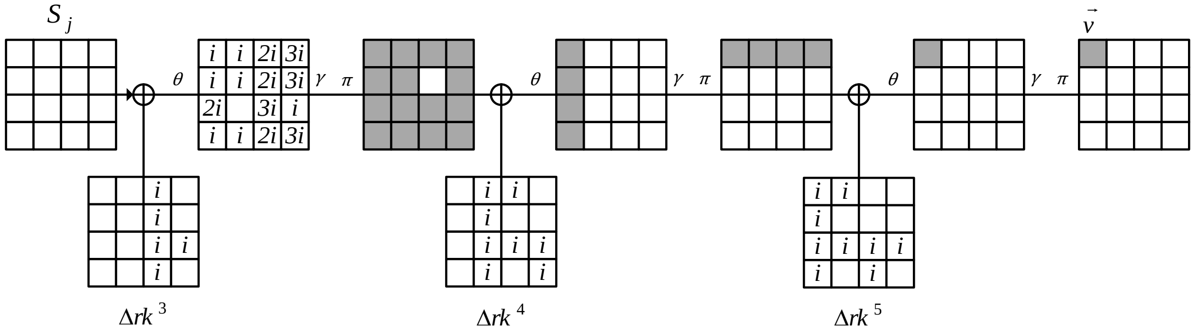

Fig. 4. Recomputation through g subcipher

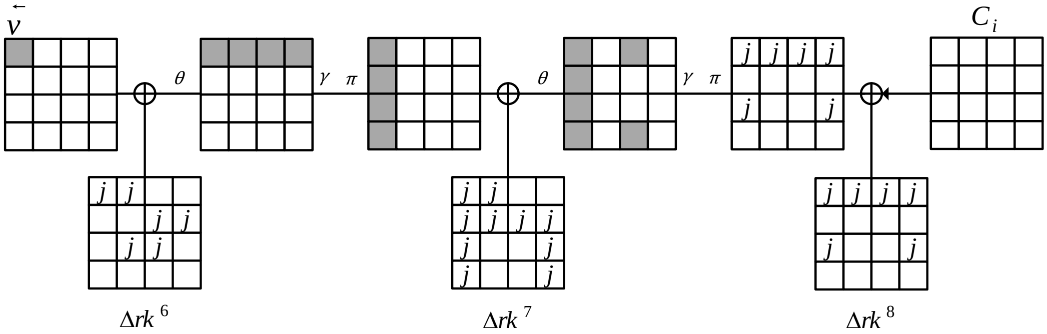

Fig. 5. Recomputation through h^{-1} subcipher

4.4 The Attack Complexity

As discussed in Section 4.1, all the palintext in a biclique are of the form $P_i = P_0 \oplus \Delta_i$. Recall that, as shown in Figure 3, $\Delta_i = (000*|000*|000*|00**) \oplus \theta(00i0|0000|000i|000i)$, where each byte spotted by '*' can take all possible values, and variable i changes from 0 to 255. So, if all the bicliques share $P_0 = 0$ as the base plaintext, then at most 2^{48} different plaitexts P_i will be used in the attack.

The computational complexity of the attack is composed of several parts. In the biclique construction step, for each of the $2^{k-2d} = 2^{112}$ bicliques we perform $2^d = 2^8$ 3-round encryptions to compute 2^d intermediate states S_j. Since the data complexity is 2^{48}, the time complexity of the data collection step is also 2^{48} encryptions. Matching check has a precomputation and a recomputation stages. The precomputation is about 2^d 5-round encryptions. The recomputation complexity for each biclique is about $2^d \times 30$ S-box evaluation plus $2^d \times \frac{10}{16}\theta$ evaluation. Since the S-box evaluation is the dominant part, and the full round encryption has $8 \times 16 = 128$ S-boxes, $2^{2d} \times \frac{30}{128} = 2^{2d-2.1}$ encryptions seems to be a close approximation for the recomputation complexity. Thus the overall time complexity is

$$TC = 2^{k-d} \times \tfrac{3}{8} + 2^{48} + 2^{k-d} \times \tfrac{5}{8} + 2^{n-2.1} \approx 2^{125.9}.$$

The memory complexity is composed of two parts. The memory used to store one biclique is equal to $2^{d+1} = 2^9$ blocks for plaintexts and intermediate states, and $2^{2d} = 2^{16}$ blocks for the corresponding group of keys. The memory for the precomputation of the matching check step is equal to the memory for storing $2^{d+1} = 2^9$ full computation of g and h^{-1}.

5 Conclusion

In this paper, we proposed the first single-key attack on full SQUARE. The attack uses the recently introduced concept of biclique cryptanalysis, which is a technique carried over to the block cipher cryptanalysis from hash function analysis. We introduced a biclique for the 3 initial rounds of SQUARE to present an attack on the full round of this cipher with a data complexity of less than 2^{48} chosen plaintexts and a time complexity of about 2^{126} encryptions. Our attack is the second application of biclique attack to a block cipher.

References

1. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg, 1993.
2. Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 299–319. Springer, Heidelberg, 2010.
3. Biryukov, A., Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg, 2009.
4. Biryukov, A., Khovratovich, D., Nikolic, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg, 2009.
5. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. available at http://eprint.iacr.org/2011/499.pdf, 2011.
6. Daeman, J., Knudsen, L.R., Rijmen, V.: The Block Cipher SQUARE. In: Biham, E. (ed.) FSE'97. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg, 1997.
7. Daemen, J., Rijmen, V.: The design of Rijndael: AES the Advanced Encryption Standard. Springer, Heidelberg, 2002.
8. Khovratovich, D., Naya-Plasencia, M., Röck, A., Schläffer, M.: Cryptanalysis of Luffa v2 Components. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) Selected Areas in Cryptography 2010. LNCS, vol. 6544, pp. 388-409. Springer, Heidelberg, 2010.
9. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for preimages: attacks on Skein–512 and the SHA-2 family. available at http://eprint.iacr.org/2011/286.pdf, 2011.
10. Koo, B., Yeom, Y., Song, J.: Related-Key Boomerang Attack on Block Cipher SQUARE. *IEICE Transaction*, 94-A(1), 3–9, 2011.
11. Matsui, M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT'93, LNCS, vol. 765, pp. 386-397. Springer, 1993.
12. Mendel, F., Peyrin, T., Rechberger, C., Schläffer, M.: Improved Cryptanalysis of the Reduced Grøstl Compression Function, ECHO Permutation and AES Block Cipher. In: Jacobson J.J., Rijmen, V., Safavi-Naini, R. (eds.) Selected Areas in Cryptography 2009. LNCS, vol. 5867, pp. 16-35. Springer, Heidelberg, 2009.
13. Peyrin, T.: Improved Differential Attacks for ECHO and Grøstl. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 370-392. Springer, Heidelberg, 2010.