Unbalanced Biclique Cryptanalysis of Full-Round GIFT

GUOYONG HAN¹,², HONGLUAN ZHAO³, CHUNQUAN ZHAO¹
¹School of Management Engineering, Shandong Jianzhu University, Jinan, China
²School of Information Science and Engineering, Shandong Normal University, Jinan, China (e-mail: hunguoyong2009@163.com)
³School of Computer Science and Technology, Shandong Jianzhu University, Jinan, China (e-mail: hongluanzhao@163.com)

Corresponding author: Hongluan Zhao (e-mail: hongluanzhao@163.com).

This work was supported in part by National Natural Science Foundation of China (Nos. 61672330, 61602287, 61802235 and 11771256), and the State Scholarship Fund No. 201808370069 from China Scholarship Council, and the Key Research Development Project of Shandong Province (Nos. 2015GGX101047, 2016GGX101024 and 2018GGX106006).

Abstract

GIFT is a family of lightweight block ciphers presented at CHES 2017. Biclique cryptanalysis is proposed to attack the full AES by Bogdanov et al. in ASIACRYPT 2011. The attack can decrease computation complexity using the technology of meet-in-the-middle and reduce data complexity utilising the biclique structure. In this paper, we first provide an unbalanced biclique attack on full round GIFT. The master key has been recovered for the full round GIFT-64 by a 5-round $4 \times 16$ unbalanced biclique with data complexity of $2^{16}$ and time complexity of $2^{122.95}$. Furthermore, a 4-round $8 \times 24$ unbalanced biclique is constructed on GIFT-128 to recover the master key with data complexity of $2^{60}$ and computational complexity of $2^{118.38}$, respectively. The research results show GIFT algorithm has weak immunity to biclique cryptanalysis.

Index Terms

GIFT, Lightweight Block Cipher, Unbalanced Biclique, MITM

I. INTRODUCTION

The widespread analysis methods include differential cryptanalysis [1] [2], linear cryptanalysis, meet-in-the-middle (MITM) [3], division cryptanalysis [4] and biclique attack [5] [6]. Biclique cryptanalysis is a typical key-recovery attack that is proposed to attack the full AES by Bogdanov et al. in ASIACRYPT 2011 [7]. The attack can decrease computation and data complexity by using the main idea of MITM attack and the basic principle of the biclique structure, where the MITM attack is a typical method in the cryptanalysis of block ciphers and has been improved by many techniques [8], including splice-and-cut and so on. The researchers provided two biclique methods for AES, that is, the long and the independent related-key differentials bicliques.

The biclique attack is a variant of the MITM of cryptanalysis and achieves good results in the analysis of SHA hash function family [9]. The biclique attack can decrease computation complexity using the technology of meet-in-the-middle and reduce data complexity utilising the biclique structure. The biclique attack will be used widely in the security analyses of many block ciphers such as Midori [10], Skinny [11], TWINE [12], PRESENT [13] [14], Piccolo [15] [16], HIGHT [17] [19], IDEA [18] and KLEIN [20].

A favorable hardware efficiency has become a major design trend in cryptography given the increasing importance of ubiquitous computing. Many lightweight algorithms have been proposed recently, especially the block cipher GIFT [21] [22]. GIFT is a family of lightweight block ciphers presented by Banik et al. at CHES 2017. The designers adopted an substitution permutation network (SPN) structure which is similar to PRESENT [23]. Two versions of GIFT are used with 64 and 128-bit state sizes, and the round numbers are 28 and 40 respectively. However, GIFT has been attacked by several cryptographers with different cryptanalysis methods. In 2018, Zhao et al. provided a differential cryptanalysis over a 16-round GIFT-64 [1], with $2^{62}$ chosen plaintexts and $2^{83}$ computational complexity. Zhu et al. showed another differential cryptanalysis over a 19-round GIFT-64 [2], with data complexity of $2^{62.4}$ and time complexity of $2^{111.4}$, and a 25-round GIFT-128 with data complexity of $2^{125}$ and time complexity of $2^{125}$.

In this paper, we focus on the biclique cryptanalysis of GIFT block cipher. The crucial point of the biclique attack is building a biclique structure at the ciphertext (or plaintext), thereby connecting $2^{d_1}$ ciphertexts (or plaintexts) and $2^{d_2}$ intermediate states. $d_1 = d_2 = d$ is a d-dimension biclique
structure.

**TABLE 1. Summary of the Attacks on GIFT**

<table>
<thead>
<tr>
<th>Target strategy</th>
<th>Round (full round)</th>
<th>Data Computation</th>
<th>Attack Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIFT-64</td>
<td>16(26)</td>
<td>$2^{52}$</td>
<td>Differential cryptanalysis</td>
<td>[2]</td>
</tr>
<tr>
<td>GIFT-64</td>
<td>19(28)</td>
<td>$2^{52} \cdot 4^2$</td>
<td>Differential cryptanalysis</td>
<td>[1]</td>
</tr>
<tr>
<td>GIFT-64</td>
<td>28(28)</td>
<td>$2^{52}$, $2^{116}$</td>
<td>Differential cryptanalysis</td>
<td>[2]</td>
</tr>
<tr>
<td>GIFT-128</td>
<td>17(40)</td>
<td>$2^{12}$</td>
<td>Bi-cyclic cryptanalysis</td>
<td>III.C</td>
</tr>
<tr>
<td>GIFT-128</td>
<td>25(40)</td>
<td>$2^{12}$</td>
<td>Differential cryptanalysis</td>
<td>[1]</td>
</tr>
<tr>
<td>GIFT-128</td>
<td>40(40)</td>
<td>$2^{80}$, $2^{18} \cdot 88$</td>
<td>Bi-cyclic cryptanalysis</td>
<td>IV.C</td>
</tr>
</tbody>
</table>

A. OUR CONTRIBUTIONS

We study the characteristics of the algorithm structure deeply and the diffusion properties of key schedule. We provide an unbalanced biclique attack on full GIFT for the first several rounds. The research results show that GIFT algorithm has weak immunity to biclique cryptanalysis.

1. We present a 5-round $4 \times 16$ unbalanced biclique to key recovery of full round GIFT-64, with data and computational complexities of $2^{16}$ and $2^{122.88}$, respectively.

2. A 4-round $8 \times 24$ unbalanced biclique structure on GIFT-128 is provided, with data and computational complexities of $2^{30}$ and $2^{118.88}$, correspondingly.

The comparisons between our scheme and other methods are summarized in Table 1.

B. ORGANIZATION

This paper is organized as follows. Research and development on algorithm GIFT is summarized in Section I. The notations used throughout this paper and a brief description of GIFT-64/128 are introduced in Section II. The principle of biclique attack is briefly discussed and the key recovery attacks on full round GIFT-64 by unbalanced bicliques is also provided in Section III. Then, the biclique attack on full round GIFT-128 and the data and computational complexities are presented in Section IV. Finally, we draw our conclusions and summarize this paper in Section V.

II. DESCRIPTION OF GIFT

A. NOTATIONS

\( P \): plaintext.

\( C \): ciphertext.

\( M \): the intermediate state.

\( M_i \): the \( i \)-th cell of the intermediate state \( M \).

\( K_i \): the \( i \)-th group of the key \( K \) \((0 \leq i \leq 15)\).

\( K_i^{[m, n]} \): the \( m \)-th and \( n \)-th bits of the \( K_i \).

\( A_{(i)} \): \{0, 1\}.

\( x \): \{0, 1\} .

\( \parallel \): concatenation.

\( F^r_i \): the \( i \)-th cell (4 bits) of the state after AddRoundKey of the \( r \)-th round function.

\( F^r_{ij} \): the \( i \)-th and \( j \)-th cells of \( F^r \).

\( a_{(i)} \): \( b \) denoting the bit length of \( a \).

B. GENERAL DESCRIPTION OF GIFT

GIFT is a lightweight block cipher of the SPN structure. There are two versions, GIFT-64 and GIFT-128, where the sizes of state are 64 and 128 bits, and the numbers of round are 28 and 40, respectively. The key sizes of both versions are the same 128 bits. The overall structure of GIFT-64 is illustrated in Figure 1, and the number of S-boxes for GIFT-128 is 32.

Round Function. The round function of GIFT is composed of the following 3 steps.

1. SubCell: the same S-box \((4 \times 4)\) is applied parallelly to each nibble \( S_i \), \(0 \leq i \leq 15\) for GIFT-64, \(0 \leq i \leq 31\) for GIFT-128, respectively.(See in Table 2)

2. PermBits: The bit permutation of GIFT-64 and GIFT-128 are shown in Table 3 and Table 4, respectively.

3. AddRoundKey: The round key RK is extracted from the master key \( K \). A round key is first extracted from the master key \( K \) before the master key state updates. The 128-bit master key of GIFT is represented as follows. 

\( K = K_7 \parallel K_6 \parallel K_5 \parallel K_4 \parallel K_3 \parallel K_2 \parallel K_1 \parallel K_0 \), where \( K_i \) is a 16-bit subkey.

For GIFT-64, 32-bit of the key state are extracted from the master key \( K \) as the round key: RK=U\parallel V, where \( K_1 \rightarrow U \) and \( K_0 \rightarrow V \). The rule that RK is XORed to \( b_{4i+1} \) and \( b_{4i} \) of the intermediate state, respectively, i.e., \( b_{4i+1} \leftarrow b_{4i+1} \oplus v_i \) and \( b_{4i} \leftarrow b_{4i} \oplus v_i \), where \( i \in \{0, 1, 2, \cdots , 15\} \).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Citation information: DOI 10.1109/ACCESS.2019.2945006, IEEE Access

VOLUME 4, 2016

Author et al.: Preparation of Papers for IEEE TRANSACTIONS and JOURNALS
128 are updated as follows: $u\oplus K_0 \rightarrow U$ and $b_1 \oplus b_4+1 \rightarrow V$. RK is XORed to $b_{4i+2}$ and $b_{4i+1}$ of the intermediate state, respectively, i.e., $b_{4i+2} \leftarrow b_{4i+2} \oplus u_i$ and $b_{4i+1} \leftarrow b_{4i+1} \oplus v_i$, where $i \in \{0, 1, 2, \ldots, 31\}$.

The $b_i$ represents the $i$-th bit of the intermediate state. The $u_i$ and $v_i$ represent the $i$-th bit of $U$ and $V$.

The key schedule. The key state for GIFT-64 and GIFT-128 are updated as follows: $K_7 || K_6 || K_5 || K_4 || K_3 || K_2 || K_1 || K_0 \leftarrow K_1 \gg 2 || K_0 \gg 12 || K_7 || K_6 || K_5 || K_4 || K_3 || K_2$.

III. BICLIQUE ATTACK ON GIFT-64

A. DEFINITION OF BICLIQUE

Biclique cryptanalysis is divided into 2 steps: constructing biclique structure and the MITM attack. The biclique structure determines the data complexity of the whole attack and the MITM attack reduces the computation complexity. The detailed steps of the attack and the basic principle of the biclique attack are presented in [7].

The biclique structure links $2^{d_1}$ plaintexts $\{P_i\}$ to $2^{d_2}$ intermediate states $\{S_j\}$. The core idea is to search two as possible long differential paths which share no active state cells. The biclique structure can be considered as a subcipher, namely $f$, i.e., $f_K(P) = S$, where $K$ is a set of $2^{d_1+d_2}$ keys $\{K_{[i,j]}\}$:

$$
\{K_{[i,j]}\} = 
\begin{bmatrix}
K_{[0,0]} & K_{[0,1]} & \cdots & K_{[0,2^{d_1}-1,1]} \\
K_{[1,0]} & K_{[1,1]} & \cdots & K_{[1,2^{d_1}-1,1]} \\
\vdots & \vdots & \ddots & \vdots \\
K_{[2^{d_2}-1,0]} & K_{[2^{d_2}-1,1]} & \cdots & K_{[2^{d_2}-1,2^{d_1}-1,1]}
\end{bmatrix}
$$

(1)

The 3-tuple $\{\{P_i\}, \{S_j\}, \{K_{[i,j]}\}\}$ is called a biclique structure.

B. FIVE-ROUND $4 \times 16$ UNBALANCED BICLIQUE ON GIFT-64

Phase 1. Key Partitioning. The 128 bites $K$ is divided into $2^{108}$ groups, and each group key consists of a $2^4 \times 2^{16}$ matrix: $\{K_{[i,j]}\}$. Let 20 bits ($K_{[0,0]}$) be $0_{20}$ and enumerate the rest of 108 bits ($K_{[i,j]}$). The round key (RK) schedule is seen in Section 2. We construct a $4 \times 16$ unbalanced biclique structure on GIFT-64 by $K_{[5, 4]} || K_{[2, 0]}$ and $K_{[15, 14, 13, 12, 11, 10, 9, 8]} || K_{[0, 15, 14, 13, 12, 11, 10, 9, 8]}$.

The $K_{[0,0]}, K_{[0, j]}, K_{[i, 0]}$ and $K_{[i, j]}$ are as follows:

$$
\begin{align*}
K_{[0,0]} &= [A_{(16)} \oplus A_{(16)}, A_{(16)} \oplus A_{(16)}, A_{(16)}, A_{(16)}, 0_{(20)}] \\
K_{[i,0]} &= K_{[0,0]} \oplus [0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}] \\
K_{[0,j]} &= K_{[0,0]} \oplus [0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}] \\
K_{[i,j]} &= K_{[0,0]} \oplus K_{[0, j]} \oplus K_{[i, 0]} \oplus K_{[i, j]}
\end{align*}
$$

where $A \in \{0, 1\}$.

Phase 2. Five-Round $4 \times 16$ Unbalanced Biclique. We can construct a five-round $4 \times 16$ unbalanced biclique structure on GIFT-64 (Figure 2) utilizing the above key grouping scheme. The biclique structure links $2^4$ plaintexts to $2^{16}$ intermediate states in each group key. The steps of constructing the biclique structure are as follows:

Step 1. In Figure 2(a), let $P_0 = 0_{64}$ and encrypts $P_0$ for five rounds to obtain $S_0$, i.e., $S_0 = f_{K_{[i,0]}}(P_0)$. This process is called basic operations.

Step 2. The attacker encrypts $P_0$ with different keys $K_{[i,0]}$ for $i \in \{0, 1\}^{16}$ to obtain the corresponding intermediate states $S_i$ (Figure 2(b)). The differences between $K_{[0,0]}$ and $K_{[i,0]}$ lead to the computation complexity. The diagonal
stripes cells should be computed $2^{2}\cdot1$ times and the blue cells should be computed $2^{4}\cdot1$ times. The white cells must not be computed because this process shares the basic operations in Step 1. In this step, the attacker obtains $f(P_0)^{K_{i,j}}S_i$.

**Step 3.** The attacker decrypts $S_0$ with different keys $K_{[0,j]}$ for $j \in \{0,1\}^4$ (Figure 2(c)) to obtain the corresponding plaintexts $P_j$. The differences between $K_{[0,0]}$ and $K_{[0,j]}$ bring the differences in certain cells. The diagonal stripes cells should be computed $2^2\cdot1$ times and the white cells have been computed in Step 1. Thus, the attacker obtains $f^{-1}(S_0)^{K_{[0,j]}}P_j$.

These two differential paths have no intersection in the first five rounds. Then, it is easy to verify that $f(P_j)^{K_{[i,j]}}S_i$ is always holds for all $i \in \{0,1\}^8$ and $j \in \{0,1\}^4$ as shown in Figure 2. So, we can obtain a five-round $4 \times 16$ unbalanced biclique structure for each key group.

**Phase 3. Matching Over 23 Rounds.** In order to decrease computation complexity, $V = F_{12,8}^{3,4,0}$ an 16-bit output of 9-th round, is selected as the internal matching variable (Figure 3) in two directions to attain the correct key.

**Forward Direction.** We encrypt $S_i$ under the key $K_{[i,0]}$ to attain $S_i^{K_{[i,0]}} \overrightarrow{V_{1,0}}$. Then, we encrypt $S_i$ by using all the possible $2^4 - 1$ keys $K_{[i,j]}$ to attain $S_i^{K_{[i,j]}} \overrightarrow{V_{i,j}}$. The differences between $K_{[i,0]}$ and $K_{[i,j]}$ lead to computation complexities. In Figure 3 (left part), the vertical stripes cells should be computed $2^4$ times and the white cells are computed once. The diagonal stripes cells should be computed $2^2$ times and the yellow cells are computed once.

**Search Candidates.** In the last process, we verify $2^{108}$ keys utilizing the 16-bit matching variable of $V_{i,j}$ and $\overrightarrow{V_{i,j}}$ for all $i \in \{0,1\}^8$ and $j \in \{0,1\}^4$. Then, the number of the remaining candidate key is $2^4$ on average in each key group. We exhaustively check the remaining $2^{108}$ candidate key until the correct key is found.

**C. COMPLEXITIES OF FIVE-ROUND UNBALANCED BICLIQUE CRYPTANALYSIS ON GIFT-64**

**Data Complexity.** In Figure 2(c), for each unbalanced biclique structure, we decrypt $S_0$ with the keys $K_{[0,j]}$ to obtain $P_j$. All the plaintexts have differences only in four cells($S_3$, $S_2$, $S_1$ and $S_0$). Thus, the data complexity does not exceed $2^{16}$.

**Computational Complexity.** The computation complexity of the attack depends mainly on the number of the SubCell. Each round of GIFT-64 is composed of 16 SubCells and single encryption includes $28 \times 16 = 448$ SubCells. For each key of the $2^{108}$ groups, the specific computation is as follows.

**Biclique Complexity.** The 24 SubCells (Figure 2(b), noted with diagonal stripes) need to compute $2^2$ times and 24 SubCells (Figure 2(b), noted with blue) need to compute $2^4$ times. In Figure 2(c), 6 SubCells (noted with diagonal stripes) need to compute $2^2$ times. The rest of 26 SubCells are computed once. Then the summation is $2^4 \times 24 + 2^2 \times 30 + 26$ SubCells computations. Thus, the computation complexity of a biclique structure is approximately $2^9\cdot24$ full round GIFT-64 encryptions.

**Matching Complexity.** In the forward direction (Figure 3, left) of a single $S_i$, 4 SubCells (noted with red) need to compute $2^4$ times, and 24 SubCells (noted with diagonal stripes)
need to compute $2^2$ times. 4 SubCells (noted with vertical stripes) need to compute $2^1$ times, and 36 SubCells (noted with yellow) are computed only once. 12 SubCells (noted with white) must not be computed. Thus, the complexity of this process is $2^{16} \times (2^4 \times 4 + 2^2 \times 24 + 2^1 \times 4 + 48)$ SubCells, which is approximately $2^{14.94}$ full round GIFT-64 encryptions.

In the backward direction (Figure 3, right) of a single C_j, 212 SubCells (noted with blue) need to compute $2^4$ times, and 16 SubCells (noted with vertical stripes) need to compute $2^1$ times. 48 SubCells (noted with yellow) are computed only once and 12 SubCells (noted with white) must not be computed. Thus, the complexity of this process is $2^4 \times (2^4 \times 212 + 2^1 \times 16 + 36)$ SubCells, which is approximately $2^{6.95}$ full round GIFT-64 encryptions.

Finally, $2^{20}$ key candidates are verified by a matching variable (16-bit) in each group, and the average of $2^{20-16} = 2^4$ candidate key should be rechecked.

Thus, the total computational complexity of the unbalanced biclique attack on GIFT-64 is:

$$C = 2^{108} \times (2^{0.24} + 2^{14.94} + 2^{6.95} + 2^4) \approx 2^{122.95} \quad (3)$$

Memory Complexity. We need to store $2^4 \times 16$ bits (the backward direction) for the attack.

IV. BICLIQUE ATTACK ON GIFT-128

A. FOUR-ROUND 8 $\times$ 24 UNBALANCED BICLIQUE ON GIFT-128

Phase 1. Key Partitioning.

The 128 bits $K$ is divided into $2^{96}$ groups and each group key consists of a $2^8 \times 2^{24}$ matrix: $\{K_{i,j}\}$. Similar to Section 3.2, we construct an $8 \times 24$ unbalanced biclique structure utilizing $K_4[14, 12, 10, 8]|| K_1[8, 6, 4, 2]$ and $K_7[15, 14, 13, 12, 7, 6, 5, 4]|| K_6[7, 6, 5, 4]|| K_5[15, 14, 13, 12, 7, 6, 5, 4]|| K_2[15, 14, 13, 12]$. The master key $K$ is grouped as follows:

$$K_{[0,0]} = [A_4(0)\mid A_4(4)\mid A_1(0)\mid A_1(4)\mid A_1(8)\mid A_4(0)]$$
$$K_{[1,1]} = [A_4(0)\mid A_4(4)\mid A_1(0)\mid A_1(4)\mid A_1(8)\mid A_4(0)]$$
$$K_{[0,1]} = [A_4(0)\mid A_4(4)\mid A_1(0)\mid A_1(4)\mid A_1(8)\mid A_4(0)]$$
$$K_{[1,0]} = [A_4(0)\mid A_4(4)\mid A_1(0)\mid A_1(4)\mid A_1(8)\mid A_4(0)]$$

where $A_1(1) \in \{0, 1\}$. 

FIGURE 3. Partial matching over 23 rounds for GIFT-64.
Phase 2. Four-Round 8 × 24 Unbalanced Biclique. We create a four-round 8 × 24 unbalanced biclique structure on GIFT-128 (Figure 4) utilizing the above key grouping scheme. The biclique structure links 28 plaintexts to 224 intermediate states in each group key. The steps of constructing the biclique structure are as follows:

Step 1. The basic operation is similar to that in Section III.B.

Step 2. The attacker encrypts \( P_0 \) under different keys \( K_{[i,0]} \) for \( i \in \{0,1\}^{24} \) to obtain the corresponding intermediate states \( S_i \) (Figure 4, left). The differences between \( K_{[0,0]} \) and \( K_{[i,0]} \) lead to the computation complexity. The vertical stripes cells need to compute 21-1 times, and the diagonal stripes cells need to compute 22-1 times. The blue cells need to compute 2\(^{-1}\) times, and the white cells must not be computed because this process shares the basic operations in Step 1. In this step, the attacker obtains \( f(P_0)^{K_{[i,0]}}S_i \).

Step 3. The attacker decrypts \( S_0 \) under different keys \( K_{[0,j]} \) for \( j \in \{0,1\}^8 \) (Figure 4, right) to obtain the corresponding plaintexts \( P_j \). The differences between \( K_{[0,0]} \) and \( K_{[0,j]} \) bring the differences in certain cells. The vertical stripes cells need to compute 2\(^{-1}\) times, and the diagonal stripes cells need to compute 2\(^{-2}\) times. The red cells need to compute 2\(^{-1}\) times, and the white cells must not be computed because this process shares the basic operations in Step 1. Thus, the attacker obtains \( f^{-1}(S_0)^{K_{[0,j]}}P_j \).

These two differential paths have no intersection in the first four rounds. Then, it is easy to verify that \( f(P_j)^{K_{[i,0]}}S_i \) always holds for all \( i \in \{0,1\}^{24} \) and \( j \in \{0,1\}^8 \) as shown in Figure 4. So, we can obtain a four-round 8 × 24 unbalanced biclique structure for each key group.

Phase 3. Matching Over 36 Rounds. In order to decrease computation complexity, \( V = F_{25,24,17,16,9,1,0,1} \), an 32-bit output of 8-th round, are selected as the internal matching variable (Figure 5) in two directions to attain the correct key.

Forward Direction. We encrypt \( S_i \) under the key \( K_{[i,0]} \) to attain \( S_i^{K_{[i,0]}}V_0^1 \). Then, we encrypt \( S_i \) by using all the possible 2\(^8\) - 1 keys \( K_{[i,j]} \) to attain \( S_i^{K_{[i,j]}}V_0^1 \). The differences between \( K_{[i,0]} \) and \( K_{[i,j]} \) lead to computation complexities. In Figure 5 (left), the white cells are not active and must not be calculated. The vertical stripes cells should be computed 2\(^{-1}\) times, and the diagonal stripes cells should be computed 2\(^{-2}\) times. The diagonal crosshatch cells should be computed 2\(^{-2}\) times, and the red cells should be computed 2\(^{-1}\) times. The yellow cells are computed once.

Backward Direction. Firstly, we encrypt the plaintexts \( P_j \) for \( j \in \{0,1\}^8 \) to attain 2\(^8\) ciphertexts \( C_{i,j} \) and decrypt \( C_j \) under the key \( K_{[0,j]} \) to attain \( C_j^{K_{[0,j]}}V_0^1 \). Then, we decrypt \( C_j \) with all the possible 2\(^{24}\) - 1 keys \( K_{[i,j]} \) to obtain \( C_j^{K_{[i,j]}}V_0^1 \). The differences between \( K_{[i,j]} \) and \( K_{[0,0]} \) lead to computation complexities. In Figure 5 (right), the white cells are not active and must not be calculated. The vertical stripes cells should be computed 2\(^{-1}\) times, and the diagonal stripes cells should be computed 2\(^{-2}\) times. The diagonal crosshatch cells should be computed 2\(^{-2}\) times, and the red cells should be computed 2\(^{-1}\) times. The yellow cells are computed once.

Search Candidates. In the last process, we verify 2\(^{32}\) keys utilizing the 32-bit matching variable of \( V_0 \) and blue \( V_0^1 \) for all \( i \in \{0,1\}^{24} \) and \( j \in \{0,1\}^8 \). Then, the number of the remaining candidate key is 2\(^8\) on average in each key group. We exhaustively check the remaining 2\(^{36}\) candidate keys until the correct key is found.

B. Complexities of Four-Round Unbalanced Biclique Cryptanalysis on GIFT-128

Data Complexity. In Figure 4(right), for each unbalanced biclique structure, we decrypt \( S_0 \) with the keys \( K_{[0,j]} \) to obtain \( P_j \). All the plaintexts do not have differences only in 12 cells (S31, S30, S29, S28, S26, S24, S23, S22, S21, S20, S18 and S16). However, there are differences in the remaining 20 cells. Thus, the data complexity does not exceed 2\(^{80}\).

Computational Complexity. The computation complexity of the attack depends mainly on the number of the SubCell. Each round of GIFT-128 is composed of 16 SubCells and single encryption includes 40 × 32 = 1280 SubCells. For each of the 2\(^{36}\) groups, the specific computation is as follows.

Biclique Complexity. In Figure 4(left), 16 SubCells (noted with blue) are calculated 2\(^{4}\) times, 8 SubCells (noted with diagonal stripes) are calculated 2\(^{2}\) times, and 8 SubCells (noted with vertical stripes) are calculated 2\(^{1}\) times. In Figure 4(right), 4 SubCells (noted with red) are calculated 2\(^{4}\) times, 8 SubCells (noted with diagonal stripes) are calculated 2\(^{2}\) times, and 16 SubCells (noted with vertical stripes) are calculated 2\(^{1}\) times. The remaining 68 SubCells are calculated only once. Thus, the total is 2\(^{4}\) × 20 + 2\(^{4}\) × 16 + 2\(^{4}\) × 24 + 68 SubCells calculations. Thus, the computation complexity of a biclique structure is 500 SubCells, which is approximately 2\(^{-1.36}\) full round GIFT-128 encryptions.

Matching Complexity. In the forward direction (Figure 5, left) of a single \( S_i \), 8 SubCells (noted with red) need to compute 2\(^{4}\) times, and 2 SubCells (noted with diagonal crosshatch) need to compute 2\(^{4}\) times. 46 SubCells (noted with diagonal stripes) need to compute 2\(^{2}\) times, and 8 SubCells (noted with vertical stripes) need to compute 2\(^{1}\) times. 72 SubCells (noted with yellow) are computed only once, and 24 SubCells (noted with white) must not be computed. Thus, the complexity of this process is 2\(^{24}\) × (2\(^{4}\) × 8 + 2\(^{4}\) × 2 + 2\(^{4}\) × 24 + 68) SubCells, which is approximately 2\(^{22.38}\) full round GIFT-128 encryptions.

In the backward direction (Figure 5, right) of a single \( C_j \), 872 SubCells (noted with blue) need to compute 2\(^{4}\) times, and 20 SubCells (noted with diagonal crosshatch) need to compute 2\(^{4}\) times. 28 SubCells (noted with diagonal stripes) need to compute 2\(^{2}\) times, and 8 SubCells (noted with vertical stripes) need to compute 2\(^{1}\) times. 24 SubCells (noted with yellow) are computed only once, and 40 SubCells (noted with white) must not be computed. Thus, the complexity of this process is 2\(^{8}\) × (2\(^{4}\) × 872 + 2\(^{3}\) × 20 + 2\(^{2}\) × 28 + 2\(^{1}\) × 24) SubCells.
8 + 24) SubCells, which is approximately $2^{11.48}$ full round GIFT-128 encryptions.

Finally, $2^{32}$ key candidates are verified by a matching variable(32-bit) in each group, and the average of $2^{32} - 32 = 1$ candidate key should be rechecked.

Thus, the total computational complexity of the four-round unbalanced biclique attack on GIFT-128 is:

$$C \approx 2^{96} \times (2^{-1.36} + 2^{22.38} + 2^{11.48} + 2^0) \approx 2^{11.38}$$

**Memory Complexity.** We need to store $2^4 \times 16$ bits (the backward direction) for the attack.

**V. CONCLUSION**

In this paper, we propose a novel method for the attack of a full-round GIFT block cipher. Additionally, we describe...
the construction of a biclique structure and the analysis of the cipher. We present a full-round biclique cryptanalysis of GIFT by investigating the simple key schedule and encryption structure.

Then, we construct a five-round $4 \times 16$ unbalanced biclique on GIFT-64, with data complexities of $2^{16}$ and computational complexities of $2^{122.95}$, respectively. Moreover, we use a four-round $8 \times 24$ unbalanced biclique on GIFT-128 with data complexities of $2^{80}$ and computational complexities of $2^{118.38}$, respectively.

These results are superior to the currently known results, thereby indicating that the biclique attack can easily attack certain ciphers with slow diffusion and simple key schedule. Thus, the designers of lightweight ciphers must improve the implementation efficiency, key schedule complexity and diffusion speed thereof.

**REFERENCES**


