IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 22, 2019, accepted September 17, 2019, date of publication October 1, 2019, date of current version October 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945006

Unbalanced Biclique Cryptanalysis of Full-Round GIFT

GUOYONG HAN[1,2], HONGLUAN ZHAO[3], AND CHUNQUAN ZHAO[1]

[1]School of Management Engineering, Shandong Jianzhu University, Jinan 250101, China
[2]School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China
[3]School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China

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

This work was supported in part by the National Natural Science Foundation of China under Grant 61672330, Grant 61602287, Grant 61802235, and Grant 11771256, in part by the China Scholarship Council through the State Scholarship Fund under Grant 201808370069, and in part by the Key Research Development Project of Shandong Province under Grant 2015GGX101047, Grant 2016GGX101024, and Grant 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 4×16 unbalanced biclique with data complexity of 2^{16} and time complexity of $2^{122.95}$. Furthermore, a 4-round 8×24 unbalanced biclique is constructed on GIFT-128 to recover the master key with data complexity of 2^{80} 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 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

The associate editor coordinating the review of this manuscript and approving it for publication was Honglong Chen.

TABLE 1. Summary of the attacks on GIFT.

Target algorithm	Round(full round)	Data	Computations	Attack Type	Reference
GIFT-64	16(28)	2^{62}	2^{83}	Differential cryptanalysis	[2]
GIFT-64	19(28)	$2^{62.4}$	$2^{111.4}$	Differential cryptanalysis	[1]
GIFT-64	28(28)	2^{16}	$2^{122.88}$	Biclique cryptanalysis	III.C
GIFT-128	17(40)	2^{62}	2^{115}	Differential cryptanalysis	[2]
GIFT-128	25(40)	2^{125}	2^{125}	Differential cryptanalysis	[1]
GIFT-128	40(40)	2^{80}	$2^{118.38}$	Biclique cryptanalysis	IV.C

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.

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 GIFT algorithm has weak immunity to biclique cryptanalysis.

(1) We present a 5-round 4×16 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 8×24 unbalanced biclique structure on GIFT-128 is provided, with data and computational complexities of 2^{80} and $2^{118.38}$, correspondingly.

The comparisons between our scheme and other methods are summarised 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 $(0 \le i \le 15)$.
$K_i[m, n]$: the m-th and n-th bits of the K_i.
$A_{(i)} : \{0, 1\}^i$.
$x : \{0, 1\}^4$.

TABLE 2. The S-box of GIFT.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

TABLE 3. The bit permutation used in GIFT-64.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	0	17	34	51	48	1	18	35	32	49	2	19	16	33	50	3
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(i)	4	21	38	55	52	5	22	39	36	53	6	23	20	37	54	7
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(i)	8	25	42	59	56	9	26	43	40	57	10	27	24	41	58	11
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(i)	12	29	46	63	60	13	30	47	44	61	14	31	28	45	62	15

TABLE 4. The bit permutation used in GIFT-128.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	0	33	66	99	96	1	34	67	64	97	2	35	32	65	98	3
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(i)	4	37	70	103	100	5	38	71	68	101	6	39	36	69	102	7
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(i)	8	41	74	107	104	9	42	75	72	105	10	43	40	73	106	11
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(i)	12	45	78	111	108	13	46	79	76	109	14	47	44	77	110	15
i	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
P(i)	16	49	82	115	112	17	50	83	80	113	18	51	48	81	114	19
i	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
P(i)	20	53	86	119	116	21	54	87	84	117	22	55	52	85	118	23
i	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
P(i)	24	57	90	123	120	25	58	91	88	121	26	59	56	89	122	27
i	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
P(i)	28	61	94	127	124	29	62	95	92	125	30	63	60	93	126	31

$\|$: concatenation.
F_i^r : the i-th cell(4 bits) of the state after AddRoundKey of the r-th round function.
$F_{i,j}^r$: the i-th and j-th cells of F^r.
$a_{(b)}$: 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.

1) ROUND FUNCTION

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

(1) SubCell: the same S-box(4×4) is applied parallelly to each nibble S_i, $0 \le i \le 15$ for GIFT-64, $0 \le i \le 31$ for GIFT-128, respectively.(Seen 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. $K = K_7\|K_6\|K_5\|K_4\|K_3\|K_2\|K_1\|K_0$, 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: $\mathrm{RK} = U\|V$, where $K_1 \to U$ and $K_0 \to V$. The rule that RK is XORed to b_{4i+1} and b_{4i} of the intermediate state, respectively, i.e., $b_{4i+1} \leftarrow b_{4i+1} \bigoplus u_i$ and $b_{4i} \leftarrow b_{4i} \bigoplus v_i$, where $i \in \{0, 1, 2, \cdots, 15\}$.

For GIFT-128, 64-bit of the key state are extracted from the master key K as the round key: RK = U‖V, where $K_5\|K_4 \to U$ and $K_1\|K_0 \to V$. RK is XORed to b_{4i+2} and b_{4i+1} of the intermediate state, respectively, i.e., $b_{4i+2} \leftarrow b_{4i+2} \bigoplus u_i$ and $b_{4i+1} \leftarrow b_{4i+1} \bigoplus v_i$, where $i \in \{0, 1, 2, \cdots, 31\}$.

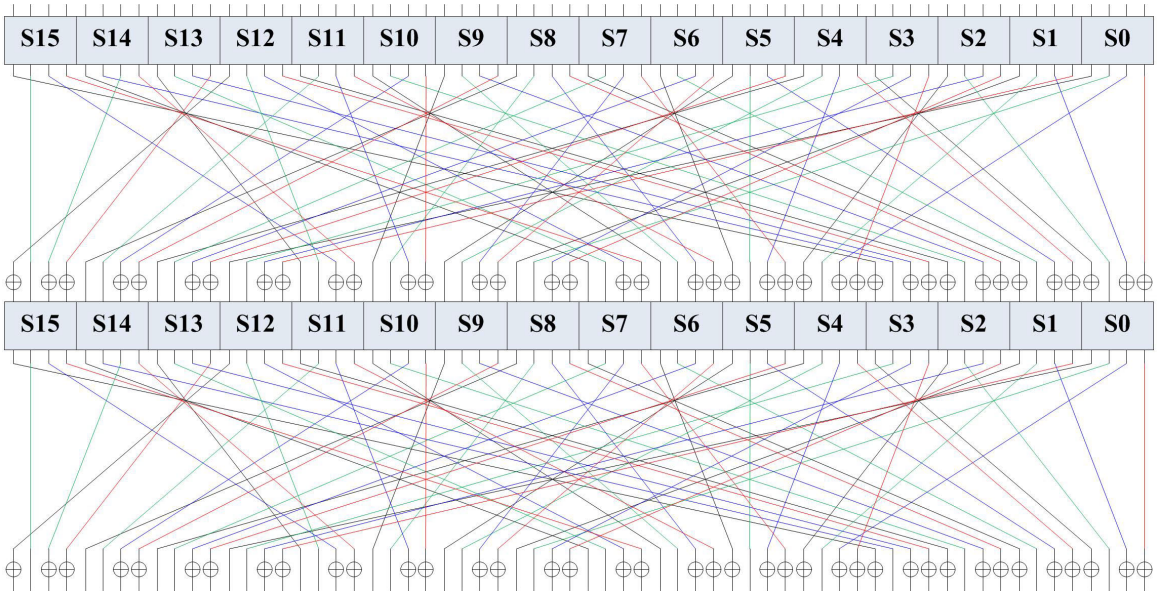

FIGURE 1. Overview of GIFT-64 round function.

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.

2) THE KEY SCHEDULE

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

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 as 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 $\{K_{[i,j]}\}$:

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

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

B. FIVE-ROUND 4 × 16 UNBALANCED BICLIQUE ON GIFT-64

1) 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 $(K_{[i,0]}, K_{[0,j]})$ be $0_{(20)}$ and enumerate the rest of 108 bits $(K_{[0,0]})$. The round key (RK) schedule is seen in Section 2. We construct a 4 × 16 unbalanced biclique structure on GIFT-64 by $K_3[5, 4]\|K_2[1, 0]$ and $K_1[15, 14, 13, 12, 11, 10, 9, 8]\|K_0[15, 14, 13, 12, 11, 10, 9, 8]$. The $K_{[0,0]}$, $K_{[0,j]}$, $K_{[i,0]}$ and $K_{[i,j]}$ are as follows:

$$\begin{cases} K_{[0,0]} = [A_{(16)}, A_{(16)}, A_{(16)}, A_{(16)}, A_{(10)}\|0_{(2)} \\ \qquad \|A_{(4)}, A_{(14)}\|0_{(2)}, 0_{(8)}\|A_{(8)}, 0_{(8)}\|A_{(8)}] \\ K_{[i,0]} = K_{[0,0]} \oplus [0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, \\ \qquad 0_{(16)}, A_{(8)}\|0_{(8)}, A_{(8)}\|0_{(8)}] \\ K_{[0,j]} = K_{[0,0]} \oplus [0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(10)} \\ \qquad \|A_{(2)}\|0_{(4)}, 0_{(14)}\|A_{(2)}, 0_{(16)}, 0_{(16)}] \\ K_{[i,j]} = K_{[0,0]} \oplus K_{[i,0]} \oplus K_{[0,j]} \end{cases} \tag{2}$$

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

2) PHASE 2. FIVE-ROUND 4 × 16 UNBALANCED BICLIQUE

We can construct a five-round 4 × 16 unbalanced biclique stucture on GIFT-64 (Figure 2) utilizing the above key grouping scheme. The biclique stucture links 2^4 plaintexts to 2^{16} intermediate states in each group key. The steps of constructing the biclique stucture 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., $S_0 = f_{K_{[0,0]}}(P_0)$. 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 2^2-1 times and the blue cells should be computed 2^4-1 times. The white cells must not be computed because this process shares the basic operations in Step 1. In this step, the attacker obtains $f(P_0) \xrightarrow{K_{[i,0]}} S_i$.

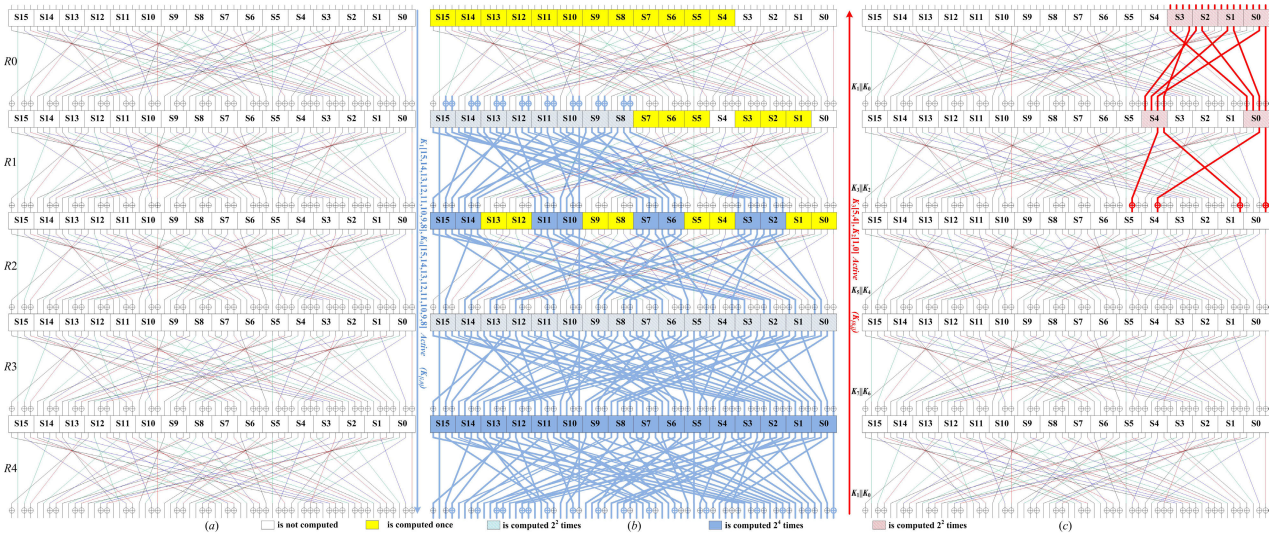

FIGURE 2. Five-round unbalanced biclique on GIFT-64.

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 2^2-1 times and the white cells have been computed in Step 1. Thus, the attacker obtains $f^{-1}(S_0)^{\overrightarrow{K_{[0,j]}}}P_j$.

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

3) PHASE 3. MATCHING OVER 23 ROUNDS

In order to decrease computation complexity, $V = F^9_{12,8,4,0}$, 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.

4) FORWARD DIRECTION

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

5) BACKWARD DIRECTION

Firstly, we encrypt the plaintexts P_j for $j \in \{0,1\}^4$ to attain 2^4 ciphertexts C_j and decrypt C_j under the key $K_{[0,j]}$ to attain $C_j^{\overrightarrow{K_{[0,j]}}}\overleftarrow{V_{0,j}}$. Then, we decrypt C_j with all the possible 2^{16}-1 keys $K_{[i,j]}$ to obtain $C_j^{\overrightarrow{K_{[i,j]}}}\overleftarrow{V_{i,j}}$. The differences between $K_{[i,j]}$ and $K_{[0,j]}$ lead to computation complexities. In Figure 3 (right part), the white cells must not be computed and the vertical stripes cells should be computed 2^1 times. The blue cells should be computed 2^4 times and the yellow cells are computed once.

6) SEARCH CANDIDATES

In the last process, we verify 2^{20} keys utilizing the 16-bit matching variable of $\overrightarrow{V_{i,j}}$ and $\overleftarrow{V_{i,j}}$ for all $i \in \{0,1\}^{16}$ 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

1) 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($S3$, $S2$, $S1$ and $S0$). Thus, the data complexity does not exceed 2^{16}.

2) 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.

3) 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 $2^{0.24}$ full round GIFT-64 encryptions.

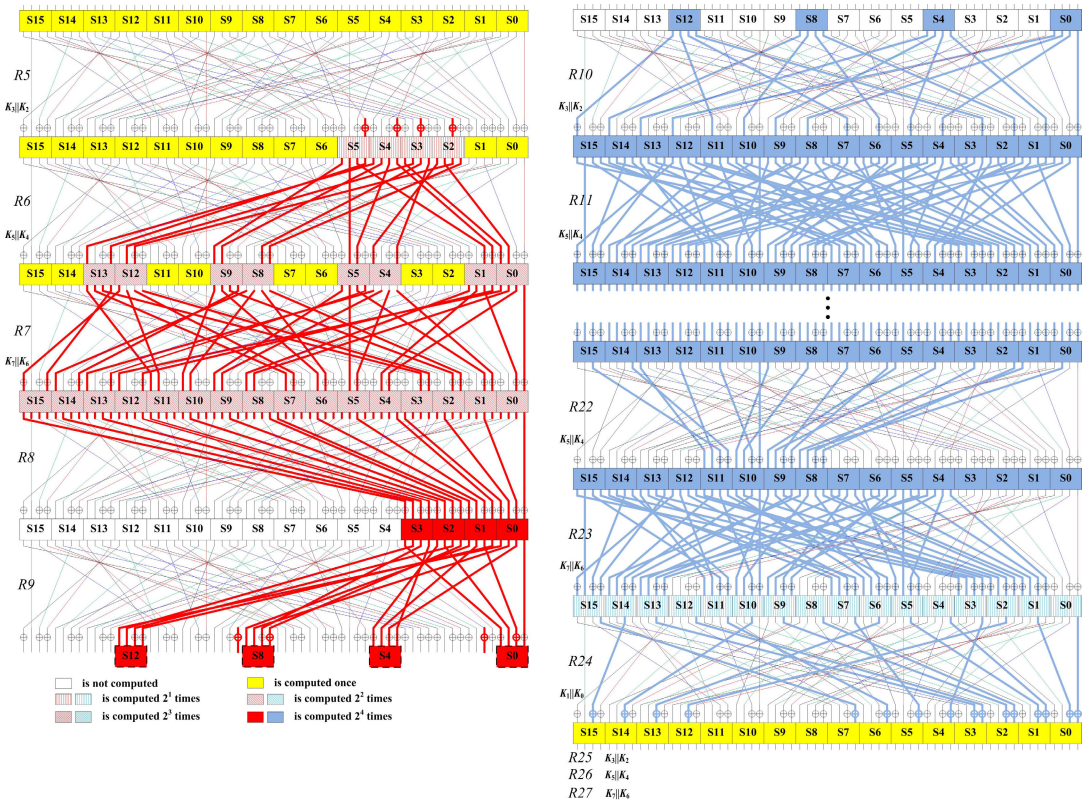

FIGURE 3. Partial matching over 23 rounds for GIFT-64.

4) 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:

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

5) 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 × 24 UNBALANCED BICLIQUE ON GIFT-128

1) 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 8×24 unbalanced biclique structure utilizing $K_4[14, 12, 10, 8] \| K_1[8, 6, 4, 2]$ and $K_7[15, 14, 13, 12, 7, 6, 5, 4] \| K_6[7, 6, 5, 4] \| K_3[15, 14, 13, 12, 7, 6, 5, 4] \| K_2[15, 14, 13, 12]$. The master key K is grouped as follows:

$$\begin{cases} K_{[0,0]} = [0_{(4)} \| A_{(4)} \| 0_{(4)} \| A_{(4)}, A_{(8)} \| 0_{(4)} \| A_{(4)}, \\ \qquad A_{(16)}, A_{(1)} \| 0_{(1)} \| A_{(1)} \| 0_{(1)} \| A_{(1)} \| 0_{(1)} \| A_{(1)} \| \\ \qquad 0_{(1)} \| A_{(8)}, 0_{(4)} \| A_{(4)} \| 0_{(4)} \| A_{(4)}, 0_{(4)} \| A_{(12)}, \\ \qquad A_{(7)} \| 0_{(1)} \| A_{(1)} \| 0_{(1)} \| A_{(1)} \| 0_{(1)} \| A_{(1)} \| 0_{(1)} \| \\ \qquad A_{(1)}, A_{(16)}] \\ K_{[i,0]} = K_{[0,0]} \oplus [A_{(4)} \| 0_{(4)} \| A_{(4)} \| 0_{(4)}, 0_{(8)} \| \\ \qquad A_{(4)} \| 0_{(4)}, 0_{(16)}, 0_{(16)}, A_{(4)} \| 0_{(4)} \| A_{(4)} \| 0_{(4)}, \\ \qquad A_{(4)} \| 0_{(12)}, 0_{(16)}, 0_{(16)}] \\ K_{[0,j]} = K_{[0,0]} \oplus [0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(16)}, 0_{(7)} \\ \qquad \| A_{(1)} \| 0_{(8)}, 0_{(6)} \| A_{(1)} \| 0_{(7)} \| A_{(2)}, 0_{(16)}, 0_{(16)}] \\ K_{[i,j]} = K_{[0,0]} \oplus K_{[i,0]} \oplus K_{[0,j]} \end{cases} \quad (4)$$

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

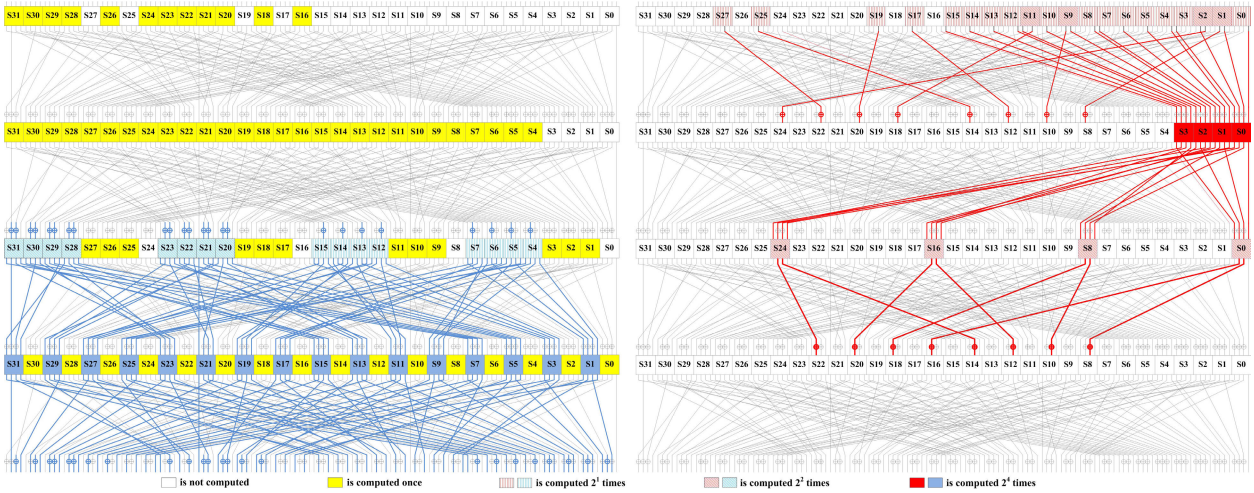

FIGURE 4. Four-round unbalanced biclique on GIFT-128.

2) PHASE 2. FOUR-ROUND 8 × 24 UNBALANCED BICLIQUE

We create a four-round 8×24 unbalanced biclique stucture on GIFT-128 (Figure 4) utilizing the above key grouping scheme. The biclique stucture links 2^8 plaintexts to 2^{24} intermediate states in each group key. The steps of constructing the biclique stucture 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 2^1-1 times, and the diagonal stripes cells need to compute 2^2-1 times. The blue cells need to compute 2^4-1 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 $f(P_0)\xrightarrow{K_{[i,0]}}S_i$.

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 2^1-1 times, and the diagonal stripes cells need to compute 2^2-1 times. The red cells need to compute 2^4-1 times, and the white cells must not be computed because this process shares the basic operations in Step 1. Thus, the attacker obtains $f^{-1}(S_0)\xrightarrow{K_{[0,j]}}P_j$.

These two differential paths have no intersection in the first four rounds. Then, it is easy to verify that $f(P_j)\xrightarrow{K_{[i,j]}}S_i$ 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.

3) PHASE 3. MATCHING OVER 36 ROUNDS

In order to decrease computation complexity, $V = F^8_{25,24,17,16,9,8,1,0}$, 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.

4) FORWARD DIRECTION

We encrypt S_i under the key $K_{[i,0]}$ to attain $S_i\xrightarrow{K_{[i,0]}}\overrightarrow{V_{i,0}}$. Then, we encrypt S_i by using all the possible $2^8 - 1$ keys $K_{[i,j]}$ to attain $S_i\xrightarrow{K_{[i,j]}}\overrightarrow{V_{i,j}}$. 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 2^3 times, and the red cells should be computed 2^4 times. The yellow cells are computed once.

5) BACKWARD DIRECTION

Firstly, we encrypt the plaintexts P_j for $j \in \{0,1\}^8$ to attain 2^8 ciphertexts C_j and decrypt C_j under the key $K_{[0,j]}$ to attain $C_j\xrightarrow{K_{[0,j]}}\overleftarrow{V_{0,j}}$. Then, we decrypt C_j with all the possible 2^{24}-1 keys $K_{[i,j]}$ to obtain $C_j\xrightarrow{K_{[i,j]}}\overleftarrow{V_{i,j}}$. The differences between $K_{[i,j]}$ and $K_{[0,j]}$ 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 2^3 times, and the red cells should be computed 2^4 times. The yellow cells are computed once.

6) SEARCH CANDIDATES

In the last process, we verify 2^{32} keys utilizing the 32-bit matching variable of $\overrightarrow{V_{i,j}}$ and blue $\overleftarrow{V_{i,j}}$ for all $i \in \{0,1\}^{24}$ and $j \in \{0,1\}^8$. Then, the number of the remaining candidate key is 2^0 on average in each key group. We exhaustively check the remaining 2^{96} candidate keys until the correct key is found.

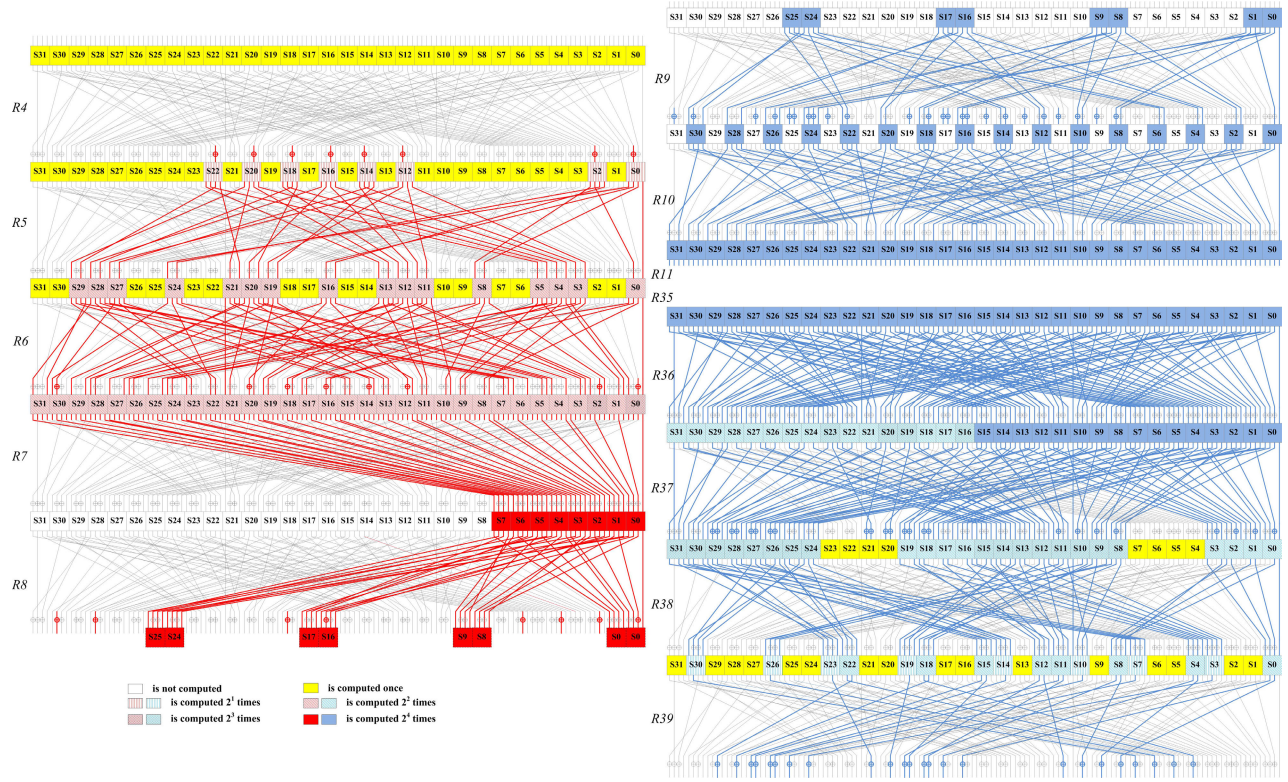

FIGURE 5. Partial matching over 36 rounds for GIFT-128.

B. COMPLEXITIES OF FOUR-ROUND UNBALANCED BICLIQUE CRYPTANALYSIS ON GIFT-128

1) 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}.

2) 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 $40 \times 32 = 1280$ SubCells. For each key of the 2^{96} groups, the specific computation is as follows.

3) 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 $2^4 \times 20 + 2^2 \times 16 + 2^1 \times 24 + 68$ SubCells calculations. Thus, the computation complexity of a biclique structure is 500 SubCells, which is approximately $2^{-1.36}$ full round GIFT-128 encryptions.

4) 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 2^3 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 $2^{24} \times (2^4 \times 8 + 2^3 \times 2 + 2^2 \times 46 + 2^1 \times 8 + 72)$ 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 2^3 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 $2^8 \times (2^4 \times 872 + 2^3 \times 20 + 2^2 \times 28 + 2^1 \times 8 + 24)$ 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:

$$C \approx 2^{96} \times (2^{-1.36} + 2^{22.38} + 2^{11.48} + 2^{0}) \approx 2^{118.38} \quad (5)$$

5) 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 4×16 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 8×24 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

[1] J. Zhao *et al.*, "Differential analysis of lightweight block cipher gift," *J. Cryptologic Res.*, vol. 5, no. 4, pp. 335–343, 2018.
[2] B. Zhu, X. Dong, and H. Yu, "MILP-based differential attack on round-reduced GIFT," in *Proc. Cryptographers Track RSA Conf.*, San Francisco, CA, USA, 2019, pp. 372–390.
[3] Y. Wang and W. Wu, "Meet-in-the-middle attack on TWINE block cipher," (in Chinese), *Ruan Jian Xue Bao/J. Softw.*, vol. 26, pp. 2684–2695, 2015.
[4] W. Zhang and V. Rijmen, "Division cryptanalysis of block ciphers with a binary diffusion layer," Cryptol. ePrint Arch., Tech. Rep. 2017/188, 2017.
[5] O. Özen, K. Varıcı, C. Tezcan, and Ç. Kocair, "Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT," in *Information Security and Privacy* (Lecture Notes in Computer Science), vol. 5594. Berlin, Germany: Springer, 2009, pp. 90–107.
[6] G. Han, W. Zhang, Z. Xing, H. Zhao, and J. Lian, "Unbalanced biclique cryptanalysis of a full round Midori," *IET Commun.*, vol. 13, no. 5, pp. 505–511, Mar. 2019.
[7] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *Advances in Cryptology—ASIACRYPT*, vol. 7073, D. H. Lee and X. Wang, Eds. Berlin, Germany: Springer, 2011, pp. 344–371.
[8] H. Ma and Y. Jia, "Stability analysis for stochastic differential equations with infinite Markovian switchings," *J. Math. Anal. Appl.*, vol. 435, pp. 593–605, Mar. 2016.
[9] D. Khovratovich, C. Rechberger, and A. Savelieva, "Bicliques for preimages: Attacks on Skein-512 and the SHA-2 family," in *Fast Software Encryption*. Berlin, Germany: Springer, 2012, pp. 244–263.
[10] H. Zhao and G. Han, "Biclique cryptanalysis on block cipher Midori," *Int. J. Embedded Syst.*, vol. 11, pp. 229–239, 2019.
[11] Y. Zheng and W. Wu, "Biclique attack of block cipher SKINNY," in *Proc. Int. Conf. Inf. Secur. Cryptol. (Inscrypt)*, 2016, pp. 3–17.
[12] M. Çoban, F. Karakoç, and Ö. Biztaş, "Biclique cryptanalysis of TWINE," Cryptol. ePrint Arch., Tech. Rep. 2012/422, 2012.
[13] M. H. F. Sereshgi, M. Dakhilalian, and M. Shakiba, "Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers," *Secur. Commun. Netw.*, vol. 9, no. 1, pp. 27–33, 2016.
[14] F. Abed, C. Forler, E. List, S. Lucks, and J. Wenzel, "Biclique cryptanalysis of the PRESENT and LED lightweight ciphers," Cryptol. ePrint Arch., Tech. Rep. 2012/591, 2002.
[15] G. Han and W. Zhang, "Improved biclique cryptanalysis of the lightweight block cipher Piccolo," *Secur. Commun. Netw.*, vol. 2017, Mar. 2017, Art. no. 7589306.
[16] Y. Wang, W. Wu, and X. Yu, "Biclique cryptanalysis of reduced-round Piccolo block cipher," in *Information Security Practice and Experience* (Lecture Notes in Computer Science), vol. 7232. Berlin, Germany: Springer, 2012, pp. 337–352.
[17] S. Ahmadi, Z. Ahmadian, J. Mohajeri, and M. R. Aref, "Low-data complexity biclique cryptanalysis of block ciphers with application to Piccolo and HIGHT," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1641–1652, Oct. 2014.
[18] D. Khovratovich, G. Leurent, and C. Rechberger, "Narrow-bicliques: Cryptanalysis of full IDEA," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2012, pp. 392–410.
[19] J. Song, K. Lee, and H. Lee, "Biclique cryptanalysis on lightweight block cipher: HIGHT and Piccolo," *Int. J. Comput. Math.*, vol. 90, pp. 2564–2580, Jan. 2013.
[20] A. Zahra, S. Mahmoud, and R. A. Mohammad, "Biclique cryptanalysis of the full-round KLEIN block cipher," *IET Inf. Secur.*, vol. 9, no. 5, pp. 294–301, 2015.
[21] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A small present: Towards reaching the limit of lightweight encryption," in *Cryptographic Hardware and Embedded Systems—CHES*. Cham, Switzerland: Springer, 2017, pp. 321–345.
[22] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "Gift: A small present," Cryptol. ePrint Arch., Tech. Rep. 2017/622, 2017.
[23] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems—CHES*. Berlin, Germany: Springer, 2007, pp. 450–466.

GUOYONG HAN received the B.E. and M.E. degrees from Shandong University, Jinan, China, in 2002 and 2006, respectively, and the Ph.D. degree from the School of Information Science and Engineering, Shandong Normal University, Jinan, in 2019. He is currently an Associate Professor with the School of Management Engineering, Shandong Jianzhu University. He has published over ten research articles in refereed academic journals and conferences. His research interests include information security, and analysis and design of block ciphers.

HONGLUAN ZHAO received the B.E. and Ph.D. degrees from the School of Mathematics, Shandong University, in 2002 and 2007, respectively. She is currently a Professor with the School of Computer Science and Technology, Shandong Jianzhu University. Her research interests include computer networks and information security.

CHUNQUAN ZHAO received the M.E. degree from Shandong Jianzhu University, Jinan, China, in 2017, where he is currently a Lecturer with the School of Management Engineering. His research interests include cryptography, including block ciphers, and big data analysis.