Postprint of Biclique Analysis of the GIFT-64 Algorithm

Authors: Guo Weibo, Liu Bin, Wang Yang

Date: 2019-04-01T00:00:00+00:00

Abstract

GIFT is a lightweight block cipher featuring high implementation efficiency and low power consumption, for which security evaluation research remains scarce. By utilizing the Biclique attack method and combining it with the algorithm' s key schedule and the information leakage patterns of the round function structure, we present results for both balanced Biclique attacks and Star attacks on GIFT-64. The balanced Biclique attack on GIFT-64 requires a data complexity of 2^32 and a computational complexity of 2^127.36; the Star attack on GIFT-64 requires a data complexity of 2 and a computational complexity of 2^127.48. This represents the first security analysis result for the full-round GIFT-64 algorithm.

Full Text

Biclique Analysis of GIFT-64

Guo Weibo[1], Liu Bin[1], Wang Yang[2] [1]Information Engineering University, Zhengzhou 450001, China [2]Xi' an Division of Surveying & Mapping, Xi' an 710054, China

Abstract: GIFT is a lightweight block cipher with high implementation efficiency and low power consumption, yet there are few research results evaluating its security. Using the Biclique attack method and combining it with the information leakage patterns of the algorithm' s key scheduling and round function structure, we present balanced Biclique attack and Star attack results for GIFT-64. The balanced Biclique attack on GIFT-64 requires a data complexity of 2^{32} and a computational complexity of $2^{127.36}$; the Star attack requires a data complexity of 2 and a computational complexity of $2^{127.48}$. These are the first security analysis results for the full-round GIFT-64 algorithm.

Key words: lightweight block cipher; GIFT algorithm; cryptanalysis; Biclique analysis; Star attack

0 Introduction

GIFT is a lightweight block cipher with an SPN structure proposed by Banik et al. at CHES 2017, designed for resource-constrained environments such as IoT and wireless sensor networks. Based on the design philosophy of the PRESENT algorithm, GIFT achieves even lower power consumption and superior implementation efficiency compared to PRESENT. The algorithm employs a 128-bit key and is available in two variants: GIFT-64 and GIFT-128, differing in block size.

Biclique analysis, introduced by Bogdanov et al. at Asiacrypt 2011, is a novel attack method against block ciphers and hash functions. Essentially a meet-in-the-middle attack that leverages differential cryptanalysis techniques, it exploits information leakage from the cipher's structure and key scheduling to achieve key recovery. This method generally enables security analysis of full-round block ciphers. Initially applied to the full AES with complexity below exhaustive search, Biclique analysis has subsequently been used to evaluate other lightweight ciphers including LBlock, PRESENT, and Piccolo.

Existing security analyses of GIFT are limited. However, due to its structural similarity to PRESENT, many analysis techniques applicable to PRESENT can be adapted to GIFT. The designers briefly analyzed GIFT' s resistance against differential cryptanalysis, linear cryptanalysis, invariant subspace attacks, and algebraic attacks. Zhao et al. found a 10-round differential distinguisher for GIFT-64 and presented differential cryptanalysis results for 16-round and 17-round versions. This paper employs Biclique analysis to provide the first security evaluation of full-round GIFT-64 under both balanced Biclique and Star attacks.

1.1 GIFT Algorithm Description

GIFT adopts an SPN structure with a design philosophy similar to PRESENT, commemorating the tenth anniversary of PRESENT' s introduction. Unlike PRESENT, GIFT' s S-boxes are not constrained by a branch number of 3, offering better implementation efficiency.

The algorithm is available in two variants: GIFT-64 with 28 rounds and GIFT-128 with 40 rounds, both using a 128-bit key. This analysis focuses on GIFT-64' s security against Biclique attacks, so we detail only GIFT-64' s structure.

Each round of GIFT-64 consists of three operations in sequence: S-box transformation, bit permutation, and key addition. The round function structure is illustrated in [Figure 1: see original paper].

a) S-box Transformation. The S-box is the sole nonlinear component of GIFT, composed of 16 parallel 4-bit invertible S-boxes that operate on each nibble of the state to provide confusion. The specific 4-bit S-box used in GIFT-64 is shown in .

b) Bit Permutation. The permutation layer operates at the bit level, moving

the state value at position i to position $P(i)$. The algorithm' s input state is numbered from right to left as bits 0 through 63. The specific permutation table for GIFT-64 is given in .

c) Key Addition. This step consists of round key addition and round constant addition. For the key addition, the key schedule generates a 32-bit round key $RK_r = U||V$, where U and V are each 16 bits. These are XORed with state values: $b_{4i} \leftarrow b_{4i} \oplus u_i$ for $i \in \{0, \dots, 15\}$ and $b_{4i+1} \leftarrow b_{4i+1} \oplus v_i$ for $i \in \{0, \dots, 15\}$. For the constant addition, a single bit "1" and a 6-bit constant $C = c_5c_4c_3c_2c_1c_0$ are XORed with state bits at positions 63, 23, 19, 15, 11, 7, and 3: $b_{63} \leftarrow b_{63} \oplus 1$, $b_{23} \leftarrow b_{23} \oplus c_5$, $b_{19} \leftarrow b_{19} \oplus c_4$, $b_{15} \leftarrow b_{15} \oplus c_3$, $b_{11} \leftarrow b_{11} \oplus c_2$, $b_7 \leftarrow b_7 \oplus c_1$, $b_3 \leftarrow b_3 \oplus c_0$.

Key Schedule. For GIFT-64, the round key RK_r consists of two 16-bit words extracted from the key state K before state update: $RK_r = U||V$. The key state is updated as follows: $K \leftarrow (k_{76}||\dots||k_0) \gg 2$, where $\gg$ denotes right rotation. The round constants are defined via an LFSR updating a 6-bit state initialized to 0: $(c_5, c_4, c_3, c_2, c_1, c_0) \leftarrow (c_4, c_3, c_2, c_1, c_0, c_5 \oplus c_4)$. lists the round constants for each round. Further design details are available in [1].

1.2 Basic Principles of Biclique Analysis

Biclique analysis requires constructing a Biclique structure. Bogdanov et al. [3] proposed two construction methods for AES: Independent Biclique and Long Biclique. Since Independent Biclique is simpler to construct and enables longer attack rounds, it is commonly adopted in subsequent research, including this work. Based on the dimension of the constructed structure, attacks can be categorized as balanced Biclique, unbalanced Biclique, or Star attacks. We now describe the Biclique structure and attack procedure.

1) Biclique Structure. A Biclique structure is essentially a bipartite graph, typically represented as a triple. Let f be an r-round sub-cipher that maps 2^{d_1} ciphertext elements C_i to 2^{d_2} intermediate state elements S_j under key $K[i, j]$, i.e., $S_j \leftarrow f^{-1}_{K[i,j]}(C_i)$ for all $i \in \{0, 1, \dots, 2^{d_1} - 1\}$ and $j \in \{0, 1, \dots, 2^{d_2} - 1\}$. The structure $(\{C_i\}, \{S_j\}, K[i, j])$ is called a d-dimensional Biclique structure where $d = d_1 = d_2$. When $d_1 = d_2 \neq 0$, it is a balanced Biclique; when $d_1 \neq d_2$ and both are non-zero, it is an unbalanced Biclique; when $d_1 = 0$ or $d_2 = 0$, it is called a Star structure. [Figure 2: see original paper] shows a general ciphertext-direction Biclique; plaintext-direction Bicliques can also be constructed depending on the algorithm.

2) Biclique Attack Procedure. A block cipher E can be viewed as a composition of sub-ciphers: $P \xleftarrow{e_2} V \xrightarrow{e_1} S \xleftarrow{f} C$, where f is the sub-cipher used to construct the Biclique and e_1, e_2 are matching sub-ciphers. The attack consists of four steps: key partitioning, Biclique construction, state matching, and key filtering.

2 Balanced Biclique Analysis of GIFT-64

Using the Biclique construction method from [3], we construct a plaintext-direction 5-round (4,4) balanced Biclique structure through appropriate key partitioning, then present a full-round GIFT-64 security analysis.

a) Key Partitioning. Based on the key schedule, we select bits $[0, 12]$, $[0, 4]$, $[0, 1]$, $[0, 1]$ of the key as active bits. Let $K[0, 0]$ denote the master key with these 8 bit positions set to 0 while remaining bits are traversed. The 128-bit master key is partitioned into 2^{120} sets, each containing 2^8 keys $K[i, j]$.

b) Balanced Biclique Construction. Using the partitioned key space, we construct related-key differential trails Δ_i and ∇_j through the corresponding round keys. Analysis shows that Δ_i and ∇_j affect no overlapping S-boxes, making them independent and yielding a 5-round (4,4) balanced Biclique structure, as shown in [Figure 3: see original paper].

c) State Matching. From the 5-round (4,4) balanced Biclique, we obtain 2^8 plaintext states P_i and their corresponding ciphertext states C_j under the correct key. We select bits 44-47 of round 17 output as the matching vector. Decrypting upward 11 rounds forms the backward matching phase, while encrypting downward 12 rounds forms the forward matching phase. [Figure 4: see original paper] illustrates the matching process. In the forward phase, we precompute 32 S-boxes, recompute 12 S-boxes twice, 4 S-boxes 2^4 times, and 148 S-boxes 2^8 times. In the backward phase, we precompute 36 S-boxes, recompute 4 S-boxes twice, 8 S-boxes 2^4 times, and 101 S-boxes 2^8 times. In the figures, white indicates no computation, light gray indicates precomputation, and dark gray indicates recomputation.

d) Key Filtering. With 4 matching bits (44-47), there are 2^4 possible values, giving an average false key pass probability of 2^{-4}. Each key set contains 2^8 keys, so on average 2^4 keys pass filtering, yielding 2^4 candidate keys per set. Finally, each candidate key is tested with full-round encryption to recover the correct key.

Theorem 1. Using a 5-round (4,4) balanced Biclique structure, we can recover the full-round GIFT-64 master key with data complexity 2^{32} and computational complexity $2^{127.36}$.

Proof. The data complexity is determined by the chosen plaintexts needed for Biclique construction. As shown in [Figure 3: see original paper], we traverse bits 0-31 of the plaintext while fixing bits 32-63, requiring 2^{32} chosen plaintexts.

The computational complexity comprises three parts. First, Biclique construction requires precomputing 64 S-boxes, recomputing 12 S-boxes 2^4 times, and recomputing 4 S-boxes 2^8 times, totaling 176 S-box computations. Second, state matching includes forward and backward phases. Forward matching precomputes 32 S-boxes, recomputes 12 S-boxes twice, 4 S-boxes 2^4 times, and 148 S-boxes 2^8 times, totaling $32+12\times2+4\times2^4+148\times2^8 = 39,040$ S-boxes. Back-

ward matching precomputes 36 S-boxes, recomputes 4 S-boxes twice, 8 S-boxes 2^4 times, and 101 S-boxes 2^8 times, totaling $36+4\times2+8\times2^4+101\times2^8 = 27,072$ S-boxes. The matching phase thus requires $39,040 + 27,072 = 66,112$ S-box computations. Third, key filtering tests 2^4 candidate keys per set, requiring 2^4 full-round encryptions.

The total complexity is therefore $176 + 66,112 + 2^4 \approx 2^{16}$ S-box computations. Since each GIFT-64 round contains 16 S-boxes and we attack 28 rounds, the complexity is $2^{16}/(16 \times 28) \approx 2^{127.36}$ full-round GIFT-64 encryptions.

3 Star Attack on GIFT-64

While the previous section used balanced Biclique analysis, this section presents a Star attack yielding the lowest data complexity for full-round GIFT-64. The Star attack, proposed by Bogdanov et al. [11] in 2014 as an unbalanced Biclique variant, was first applied to AES. It follows a similar procedure but requires only 2-3 known plaintexts at the cost of increased computational complexity.

a) Key Partitioning. We select key bits $[0, 1, 8, 11]$ in both U and V as active bits. Let $K[0, 0]$ be the master key with these 8 bits set to 0 while remaining bits are traversed. The 128-bit key is partitioned into 2^{120} sets, each containing 2^8 keys $K[i, j]$.

b) Star Structure Construction. Using the partitioned key space, we construct related-key differential trails Δ_i and ∇_j where Δ_i affects bits $[0, 1]$ and ∇_j affects bits $[8, 11]$. As with the balanced Biclique, Δ_i and ∇_j affect no overlapping S-boxes, yielding a 4-round 8-dimensional Star structure shown in [Figure 5: see original paper].

c) State Matching. Using the 4-round 8-dimensional Star structure, we select bits 44-47 of round 16 output as the matching vector. Forward matching precomputes 8 S-boxes, recomputes 8 S-boxes 2^4 times, and 164 S-boxes 2^8 times. Backward matching precomputes 34 S-boxes, recomputes 4 S-boxes twice, 2 S-boxes 2^4 times, 8 S-boxes 2^8 times, and 117 S-boxes 2^8 times. [Figure 6: see original paper] shows the matching paths with the same color coding as [Figure 4: see original paper].

d) Key Filtering. With 4 matching bits, the false key pass probability is 2^{-4}. Each key set contains 2^8 keys, so on average 2^4 keys pass filtering, leaving 2^4 candidates per set. Each candidate is verified with full-round encryption.

Theorem 2. Using a 4-round 8-dimensional Star structure, we can recover the full-round GIFT-64 master key with data complexity 2 and computational complexity $2^{127.48}$.

Proof. The computational complexity has three components. First, Star construction precomputes 60 S-boxes and recomputes 4 S-boxes 2^4 times, totaling 76 S-box computations. Second, forward matching requires $8+8\times2^4+164\times2^8 = 42,120$ S-box computations. Backward matching requires $34 + 4 \times 2 + 2 \times 2^4 +$

$8 \times 2^8 + 117 \times 2^8 = 30,130$ S-box computations. The matching phase totals $42,120 + 30,130 = 72,250$ S-box computations. Third, key filtering tests 2^4 candidates per set.

The total is $76 + 72,250 + 2^4 \approx 2^{16.14}$ S-box computations, equivalent to $2^{127.48}$ full-round encryptions. The data complexity is 2 chosen plaintexts, giving a success probability of 1.

4 Conclusion

This paper analyzed GIFT-64' s security against balanced Biclique and Star attacks. By exploiting information leakage from GIFT-64' s linear key schedule, we presented the first security analysis results for full-round GIFT-64 with optimal computational complexity and minimal data complexity. Future work will extend this Biclique analysis to GIFT-128, providing a more comprehensive security evaluation of the GIFT family.

References

[1] Banik S, Pandey S K, Peyrin T, et al. GIFT: a small PRESENT [C]//Proc of International Conference on Cryptographic Hardware and Embedded Systems. Cham: Springer, 2017: 321-345.

[2] Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher [C]//Proc of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2007: 450-466.

[3] Bogdanov A, Khovratovich D, Rechberger C. Biclique cryptanalysis of the full AES [C]//Advances in Cryptology. 2011: 344-371.

[4] National Institute of Standard and Technology (NIST). Federal information processing standards publication 197 (FIPS Pub 197): specification for the advanced encryption standard (AES)[S]. 2001.

[5] Wang Yanfeng, Wu Wenling, Yu Xiaoli, et al. Security on LBlock against biclique cryptanalysis[C]//Proc of International Workshop on Information Security Applications. Berlin: Springer, 2012: 1-14.

[6] Jeong K, Kang H C, Lee C, et al. Biclique cryptanalysis of lightweight block ciphers PRESENT, piccolo and LED [J]. IACR Cryptology ePrint Archive, 2012, 2012: 621.

[7] Wang Yanfeng, Wu Wenling, Yu Xiaoli, et al. Biclique Cryptanalysis of Reduced-Round Piccolo Block Cipher [C]//Proc of International Conference on Information Security Practice and Experience. Berlin: Springer, 2012: 337-352.

[8] Ahmadi S, Ahmadian Z, Mohajeri J, et al. Low-data complexity biclique cryptanalysis of block ciphers with application to Piccolo and Hight [J]. IEEE Trans on Information Forensics and Security, 2014, 9(10): 1641-1652.

[9] 赵静远, 徐松艳, 张子剑, 等. 轻量级分组密码算法 GIFT 的差分分析 [J]. 密码学报, 2018, 5(4): 335-343. (Zhao Jingyuan, Xu Songyan, Zhang Zijian, et al. Differential analysis of lightweight block cipher GIFT[J]. Journal of Cryptologic Research, 2018, 5(4): 335-343.)

[10] Bogdanov A, Chang D, Ghosh M, et al. Biclique with minimal data and time complexity for AES [C]//Information Security and Cryptology. 2014: 160-174.

[11] 崔竞一, 郭建胜, 刘翼鹏. 广义 Independent Biclique 攻击框架及其应用 [J]. 计算机学报, 2018, 41(2): 349-367. (Cui Jingyi, Guo Jiansheng, Liu Yipeng. Generalized Independent Biclique automated attack framework and its applications[J]. Chinese Journal of Computers, 2018, 41(2): 349-367.)

Note: Figure translations are in progress. See original paper for figures.