MILP-based Differential Attack on Round-reduced GIFT

Baoyu Zhu[1], Xiaoyang Dong[2], and Hongbo Yu[1]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China,
[2] Institute for Advanced Study, Tsinghua University, P. R. China
{yuhongbo}@mail.tsinghua.edu.cn

Abstract. At Asiacrypt 2014, Sun *et al.* proposed a MILP model[19] to search differential characteristics for bit-oriented block ciphers. In this paper, we improve this model to search differential characteristics of GIFT[3], a new lightweight block cipher proposed at CHES 2017. GIFT has two versions, namely GIFT-64 and GIFT-128. For GIFT-64, we find the best 12 rounds differential characteristic with our MILP-based model and give a key-recovery attack on 19 rounds GIFT-64. For GIFT-128, we find a 18 rounds differential characteristic and give the first attack on 23 rounds GIFT-128.

Keywords: GIFT, Differential Cryptanalysis, Lightweight Block Cipher, MILP

1 Introduction

In recent years, research on lightweight block ciphers has received a lot of attentions. Lightweight block ciphers are widely used in Internet of things and wireless communication because their structures are simple and they can be run in low-power environment. Many lightweight block ciphers such as PRESENT[6], CLEFIA[16], LED[10], PRINCE[7], SIMON and SPECK[4] have been published in last decades. GIFT[3] is a new lightweight block cipher proposed by Banik *et al.* at CHES 2017, which is designed to celebrate 10 years of PRESENT. GIFT has an SPN structure which is similar to PRESENT. It has two versions, namely GIFT-64 and GIFT-128, whose block sizes are 64 and 128, and the round numbers are 28 and 40 respectively.

Recently, many classical cryptanalysis methods could be converted to mathematical optimization problems which aims to achieve the minimal or maximal value of an objective function under certain constraints. Mixed-integer Linear Programming (MILP) is the most widely studied technique to solve these optimization problems. One of the most successful applications of MILP is to search differential and linear trails. Mouha *et al.* first applied MILP method to count active S-boxes of word-based block ciphers[12]. Then, at Asiacrypt 2014, Sun *et al.* extended this technique to search differential and linear trails[19], whose main idea is to derive some linear inequalities through the H-Representation of the convex hull of all differential patterns of S-box. Xiang *et al.*[20] introduced a MILP model to search integral distinguisher, Sasaki *et al.*[15] and Cui *et al.*[8] gave the MILP-based impossible differential search model independently. There are many MILP-based tools proposed already, such as MILP-based differential/linear search model for ARX ciphers[9], MILP-based conditional cube attacks[11] on Keccak[5], etc.

Our Contributions

The designers of GIFT provided the various cryptanalysis[3] on GIFT. They use MILP to compute the lower bounds for the number of active S-boxes in both differential cryptanalysis firstly. And then round-reduced differential differential probability of GIFT is presented. For GIFT-64, they provided a 9 rounds differential characteristic with probability of $2^{-44.415}$ and they expected that the differential probability of 13 rounds GIFT-64 will be lower than 2^{-63}. For GIFT-128, they provided a 9 rounds differential probability of 2^{-47} and they expected that the differential probability of 26 rounds GIFT-128 will be lower than 2^{-127}. The designers did not present actual attack on GIFT in [3].

In this paper, we generalize an efficient two-stage MILP-based model inspired by Sun *et al.*'s two-stage model[17]. Our model includes two interactive submodels, denoted as outer-MILP and inner-MILP part. The outer-MILP part obtains the minimal active S-boxes, namely, the truncated differential. And then the inner-MILP part produce the differential characteristic that matches the truncated differential with maximal probability. With our two-stage model, we find some differential characteristics of GIFT-64. Moreover, using the 12 rounds differential characteristic with probability of 2^{-60}, we give an attack on 19 rounds reduced GIFT-64 (out of 28 full rounds) with time complexity 2^{112}, memory complexity 2^{80} and data complexity 2^{63}.

In addition, we also improved our search model to find differential characteristics of GIFT-128. Firstly, the algorithm solves a sub-MILP-model to obtain an acceptable differential characteristic with small number of rounds. Then the produced output difference serves as input difference of the following sub-MILP-model. The sub-MILP-model is iterated until the probability of the whole differential characteristic is higher than our given bound. Using our algorithm, we find some new differential characteristics, including a new 18 rounds differential characteristic with probability 2^{-109}. Using the 18 rounds differential characteristic we give the first attack on 23 rounds GIFT-128 (out of 40 full rounds).

The summary of differential analysis of GIFT is shown in Table 1.

Table 1. Summary of cryptography analysis on GIFT

	Type	Rounds	Time	Memory	Data	Source
GIFT-64	Integral	14	-	-	-	[3]
GIFT-64	Differential	19	2^{112}	2^{80}	2^{63}	Ours
GIFT-128	Differential	23	2^{120}	2^{86}	2^{120}	Ours

2 Parliminaries

2.1 Description of GIFT

GIFT has an SPN structure which is similar to PRESENT. It has two versions, namely GIFT-64 and GIFT-128, whose block sizes are 64 and 128 and round numbers are 28 and 40 respectively. Both versions have a key length of 128 bits.

Each round of GIFT consists of three steps: SubCells, PermBits and AddRoundKey. The round function of GIFT-64 is shown in Figure 1. Similarly, GIFT-128 adopts 32 4-bit S-boxes for each round.

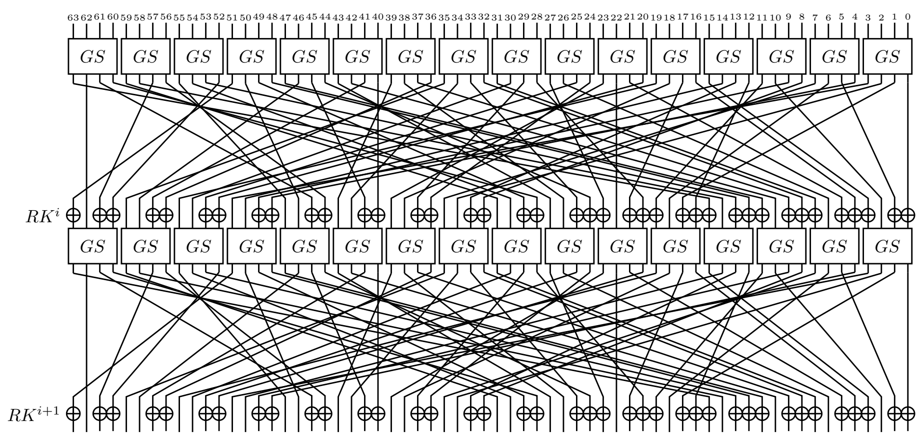

Fig. 1. 2 Rounds of GIFT-64

SubCells Both versions of GIFT use the same invertible 4-bit S-box, which is the only nonlinear component of the algorithm. The action of this S-box in hexadecimal notation is given in Table 2.

Table 2. Sbox of GIFT

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
GS(x)	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

PermBits The bit permutation used in GIFT-64 and GIFT-128 are given in Table 3.

Table 3. Specifications of GIFT Bit Permutation

GIFT-64	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$P_{64}(i)$	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
	$P_{64}(i)$	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
	$P_{64}(i)$	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
	$P_{64}(i)$	12	29	46	63	60	13	30	47	44	61	14	31	28	45	62	15
GIFT-128	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	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
	$P_{128}(i)$	28	61	94	127	124	29	62	95	92	125	30	63	60	93	126	31

AddRoundKey The round key RK is extracted from the key state. A round key is *first* extracted from the key state before the key state update.

For GIFT-64, two 16-bit words of the key state are extracted as the round key $RK = U||V$. U and V are XORed to b_{4i+1} and b_{4i} of the cipher state respectively. b_i represents the *i-th* bit of the cipher state. u_i and v_i represent the *i-th* bit of

U and V.

$$U \leftarrow k_1, V \leftarrow k_0$$

$$b_{4i+1} \leftarrow b_{4i+1} \oplus u_i, b_{4i} \leftarrow b_{4i} \oplus v_i, \forall i \in \{0, \cdots, 15\}$$

For GIFT-128, four 16-bit words of the key state are extracted as the round key $RK = U||V$. U and V are XORed to b_{4i+2} and b_{4i+1} of the cipher state respectively.

$$U \leftarrow k_5||k_4, V \leftarrow k_1||k_0$$

$$b_{4i+2} \leftarrow b_{4i+2} \oplus u_i, b_{4i+1} \leftarrow b_{4i+1} \oplus v_i, \forall i \in \{0, \cdots, 31\}$$

The key state for two versions are updated as follows,

$$k_7||k_6|| \cdots ||k_1||k_0 \leftarrow k_1 \ggg 2||k_0 \ggg 12|| \cdots ||k_3||k_2$$

Round Constants For both versions of GIFT, a single bit "1" and a 6-bit round constant $C = \{c_5, c_4, c_3, c_2, c_1, c_0\}$ are XORed into the cipher state at bit position n-1,23,19,15,11,7,3 respectively. For GIFT-64, n-1 is 63 and for GIFT-128, n-1 is 127. $\{c_5, c_4, c_3, c_2, c_1, c_0\}$ are initialized to "0", and they are updated as follow:

$$(c_5, c_4, c_3, c_2, c_1, c_0) \leftarrow (c_4, c_3, c_2, c_1, c_0, c_5 \oplus c_4 \oplus 1)$$

2.2 Notations

K_i^j	The j-th bit of the i-th round key
ΔP	The differential in the plaintext
ΔX_S^i	The differential in the output of the i-th round's Sbox
ΔX_P^i	The differential in the output of the i-th round's Permutation
ΔX_K^i	The differential in the output of the i-th round's AddKey
$\Delta X_{S,P,K}^i$	ΔX_S^i or ΔX_P^i or ΔX_K^i
$\Delta X_{S,P,K}^i\{m\}$	The m-th bit of $\Delta X_{S,P,K}^i$
$\Delta X_{S,P,K}^i\{m_l\text{-}m_t\}$	The (m_t-m_l+1) bits totally from the m_l-th bit to the m_t-th bit of $\Delta X_{S,P,K}^i$

3 Related Works

3.1 Mouha *et al.*'s Framework for Word-Oriented Block Ciphers

Mouha *et al.*[13] introduced MILP model to count the number of differentially active S-boxes for word-oriented block ciphers.

Definition 1. *Consider a string Δ consisting of n bytes $\Delta = (\Delta_0, \Delta_1, \ldots, \Delta_{n-1})$. Then, the difference vector $x = (x_0, x_1, \ldots, x_{n-1})$ corresponding to Δ is defined as*

$$x_i = \begin{cases} 0 & if\, \Delta_i = 0, \\ 1 & otherwise. \end{cases} \tag{1}$$

Based on Definition 1, Mouha *et al.* translated the XOR operation and the linear transformation to linear inequalities as follows:

- **Equations describing the XOR operation:** Suppose the input difference vector for the XOR operation be $(x^{\oplus}_{in1}, x^{\oplus}_{in2})$ and the corresponding output difference vector be $x^{\oplus}_{out}$. The following constraints will make sure that when $x^{\oplus}_{in1}$, $x^{\oplus}_{in2}$ and $x^{\oplus}_{out}$ are not all zero, then there are at least two of them are nonzero:

$$\begin{cases} x^{\oplus}_{in1} + x^{\oplus}_{in2} + x^{\oplus}_{out} \geq 2d_{\oplus} \\ d_{\oplus} \geq x^{\oplus}_{in1}, d_{\oplus} \geq x^{\oplus}_{in2}, d_{\oplus} \geq x^{\oplus}_{out} \end{cases} \tag{2}$$

  where $d_{\oplus}$ is a dummy variable taking values in {0,1}.
- **Equations describing the linear transformation:** Assume linear transformation L transforms the input difference vector $(x^L_1, x^L_2, \ldots, x^L_{m-1})$ to the output difference vector $(y^L_1, y^L_2, \ldots, y^L_{m-1})$. Given the differential branch number $\mathcal{B}_{\mathcal{D}}$. The following constraints can describe the relation between the input and output difference vectors, they should be subject to:

$$\begin{cases} \sum_i^{m-1} x^L_i + \sum_i^{m-1} y^L_i \geq \mathcal{B}_{\mathcal{D}} d^L \\ d^L \geq x^L_i, d^L \geq y^L_i, i \in \{0, ..., m-1\} \end{cases} \tag{3}$$

  where d^L is a dummy variable taking values in {0,1}.

3.2 Sun *et al.*'s Framework for Bit-Oriented Block Ciphers

At Asiacrypt 2014, Sun *et al.*[19] extended Mouha *et al.*'s framework[13] to bit-oriented ciphers. For bit-oriented ciphers, Mouha *et al.*'s descriptions of XOR operation and linear transformation are also suitable.

Definition 2. *Consider a string Δ consisting of n bits $\Delta = (\Delta_0, \Delta_1, \ldots, \Delta_{n-1})$. Then, the difference vector $x = (x_0, x_1, \ldots, x_{n-1})$ corresponding to Δ is defined as*

$$x_i = \begin{cases} 0 & if\, \Delta_i = 0, \\ 1 & if\, \Delta_i = 1. \end{cases} \tag{4}$$

Based on Definition 2, Sun *et al.* translated the S-box operation to linear inequalities as follow:

– **Equations describing the S-box operation** Suppose $(x_0, \ldots, x_{w-1})$ and $(y_0, \ldots, y_{v-1})$ are the input and output bit-level differences of an $w \times v$ S-box. A is a dummy variable taking values in $\{0,1\}$ to describe whether the S-box is active or not. $A = 1$ holds if and only if $x_0, x_1, \ldots, x_{w-1}$ are not all zero. The following constraints should be obeyed:

$$\begin{cases} A - x_i \geq 0, i \in \{0, \ldots, w-1\} \\ \sum_i^{w-1} x_i - A \geq 0 \end{cases} \tag{5}$$

3.3 Valid Cutting-off Inequalities from the Convex Hull of S-box

The convex hull of a set Q of discrete points in $\mathbb{R}^n$ is the smallest convex that contains Q. A convex hull in $\mathbb{R}^n$ can be described as the common solutions of a set of finitely many linear equalities and inequalities.

If we treat a differential of an $w \times v$ S-box as a discrete point in $\mathbb{R}^{w+v}$, then we can get a set of finitely many discrete points which includes all possible differential patterns of this S-box. Suppose $p = (x, y) = (x_0, \ldots, x_{w-1}, y_0, \ldots, y_{v-1})$ is a differential pattern of an $w \times v$ S-box, in which x is the input difference vector and y is the output difference vector. If a differential pattern p is possible, it belongs to the set of the possible differential patterns of S-box. As a result, we can describe this finitely set with the following inequalities:

$$\begin{cases} \alpha_{0,0}x_0 + \ldots + \alpha_{0,w-1}x_{w-1} + \beta_{0,0}y_0 + \ldots + \beta_{0,v-1}y_{v-1} + \gamma_0 \geq 0 \\ \cdots \\ \alpha_{n,0}x_0 + \ldots + \alpha_{n,w-1}x_{w-1} + \beta_{n,0}y_0 + \ldots + \beta_{n,v-1}y_{v-1} + \gamma_n \geq 0 \end{cases} \tag{6}$$

This is called the H-Representation of a $w \times v$ S-box. With the help of SageMath[1], hundreds of linear inequalities can be derived by differential patterns of S-box. The number of inequalities is very large in general, for example, the number of inequalities of GIFT S-box given by SageMath is 237. Adding all of them to the MILP model will make it insolvable in practical time because the efficiency of a MILP model is reduced radically when the amount of linear inequalities increase. To overcome it, Sun *et al.* invented a greedy algorithm in [19] for selecting inequalities from the convex hull.

In order to minimize the number of the set of inequalities, Sasaki *et al.* raised a MILP-based reduction algorithm in [14] to find the optimal combination with minimal number of linear inequalities from hundreds of inequalities in the H-representation of the convex hull, which remove all the impossible differential patterns of S-box. The algorithm considers each impossible pattern in the DDT of S-box. An impossible pattern should be excluded from the solution space by at least one inequality. Under these constraints, we can minimize the number of inequalities by using MILP model.

4 MILP-based Model to Search Differential Characteristic For GIFT-64

4.1 MILP-based two-stage algorithm to search differential characteristic

In [17], Sun *et al.* raised a two-stage search algorithm to find differential characteristics of block ciphers. In Sun *et al.*'s model, truncated differential characteristics with minimal active S-box will be found firstly, and then differential characteristics matching the truncated differential characteristic can be found in another model. Sun *et al.*'s model choose a prespecified threshold of the number of active S-box. However, it is possible that the characteristic with the highest probability do not have the minimal number of active S-box. Inspired by Sun *et al.*'s model, we propose Algorithm 1 to search the best or better differential characteristic.

Algorithm 1 New differential characteristic searching algorithm based on Inner and Outer-MILP Loop

Require: r round block ciphers; valid cutting-off inequalities from the convex hull of the S-box; m—number of S-boxes in one round.

Ensure: Minimal number of active S-boxes $MinSb$; differential characteristic with maximal probability.

1: Define MPr as the current minimal differential probability.
2: In the Outer-MILP part, construct an MILP model $\mathcal{M}_1$ describing the differential behavior of the cipher whose objective function is the minimal active S-boxes.
3: Initial $MPr = 2^{-200}$. Initial $MinSb$ as $r \times m$.
4: Solve the model $\mathcal{M}_1$ using an MILP optimizer.
5: **if** A feasible solution $\mathcal{TD}$ is found in $\mathcal{M}_1$, save it to a file. **then**
6:     $\diamondsuit$ *begin of Inner-MILP part*
7:     Construct an MILP model $\mathcal{M}_2$ describing the differential behavior of the cipher and add $\mathcal{TD}$ as a constraint to $\mathcal{M}_2$. The objective function is the characteristic with maximal probability.
8:     Solve the model using an MILP optimizer. If a feasible solution x is found, save x and its probability Pr to the file. If $Pr > MPr$, set MPr equal to Pr. (If only the minimal number of active S-boxes is required, it returns $MinSb = \sum A_{i,j}$.)
9:     $\diamondsuit$ *end of Inner-MILP part*
10: **end if**
11: Add the linear inequality $l^{(\mathcal{TD})}$ to remove the truncated differential $\mathcal{TD}$ from the feasible region of $\mathcal{M}_1$.
12: Solve $\mathcal{M}_1$ again, if a new solution $\mathcal{TD}$ is found, save it and go to step 5 (process the *Inner-MILP part*). Else go to step 12.
13: Terminate the procedure and extract all the best differential characteristics and their corresponding truncated differentials $\mathcal{TD}$. Extract the best characteristic with probability MPr.

Algorithm 1 does not need the predefined threshold and could get the characteristic with highest probability definitely. Algorithm 1 includes two interactive sub-models, denoted as outer-MILP part and inner-MILP part. The two stages are interactive. In the outer-MILP part, the objective function is the minimal active S-boxes. When a solution is found in the outer-MILP part, the truncated differential that contains the information of the positions of active S-boxes will input the inner-MILP part as constraints. In the inner-MILP part, it produces the differential characteristic with maximal probability that matches the truncated differential. Then the algorithm goes to the outer-MILP part with the truncated differential removed from its feasible region.

In addition, the maximal probability of the derived differential characteristic is also used to reduce the feasible region of the outer-MILP part dynamically. In details, if a differential characteristic with larger probability could be found in the next loops, the number of active S-boxes produced in outer-MILP part must be lower than a certain bound. The bound is dynamically computed by the current maximal probability. When the outer-MILP part is infeasible, the algorithm returned.

We apply Algorithm 1 to search for differential characteristics for GIFT-64, and get some interesting results.

4.2 Search Differentials of GIFT-64

Algorithm 1 needs two kinds convex hulls about the S-box in the outer-MILP part and the inner-MILP part respectively. First, we compute the H-presentation of convex hull of differential patterns of S-box in Appendix A. Using SageMath, 237 inequalities are produced in the H-Representation of the convex hull of GIFT S-box, then after selecting inequalities by the method introduced in [14], we get 21 inequalities. Second, we study the convex hull of differential patterns with probabilities of the S-box. Sun *et al.* introduced the differential distribution probability of S-box to MILP-model in [18]. Since, for GIFT S-box, there are 4 possible probabilities, i.e. 1, $2^{-1.415}$, 2^{-2}, 2^{-3}, we need three extra bits (p_0, p_1, p_2) to encode the differential patterns with probability. The new differential pattern is $(x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3; p_0, p_1, p_2) \in \mathbb{F}_2^{8+3}$ which satisfies Equation 7.

$$\begin{cases} (p_0, p_1, p_2) = (0, 0, 0), \text{if } \Pr_s[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)] = 1 = 2^{-0} \\ (p_0, p_1, p_2) = (0, 0, 1), \text{if } \Pr_s[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)] = 6/16 = 2^{-1.415} \\ (p_0, p_1, p_2) = (0, 1, 0), \text{if } \Pr_s[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)] = 4/16 = 2^{-2} \\ (p_0, p_1, p_2) = (1, 0, 0), \text{if } \Pr_s[(x_0, x_1, x_2, x_3) \to (y_0, y_1, y_2, y_3)] = 2/16 = 2^{-3} \end{cases} \tag{7}$$

Then the objective function is changed to minimize $\sum(3 \times p_0 + 2 \times p_1 + 1.415 \times p_2)$.

We implement the Algorithm 1 to search for differential characteristics for GIFT-64. In the Outer-MILP part of the Algorithm 1, the objective function is to minimize active S-boxes. We get the tighter bound of number of active S-boxes for 11 and 12 rounds reduced GIFT-64, which are 22 and 24 respectively. In addition, we get a 4 rounds differential characteristic with probability 2^{-20} as shown in Table 4, and this differential characteristic can be extended to more rounds because its input differential and output differential are same. So we get a 12 rounds differential characteristics cycled by three 4 rounds differential characteristics with probability 2^{-60} in Table 5. A 13 rounds characteristic with probability 2^{-64} can also be generated by adding another round at the beginning of 12 rounds differential characteristic. Note that the designers of GIFT claimed that the differential probability of 13 rounds GIFT-64 will be lower than 2^{-63}. Our result does not violate the claim, however the gap is very small.

Table 4. 4 rounds Differential Characteristic with Probability 2^{-20}

Round	Differential-1	Probability
Input	0000 0000 0000 1010	1
1st round	0000 000a 0000 000a	2^{-6}
2nd round	0000 0000 0000 0101	2^{-10}
3rd round	000a 0000 000a 0000	2^{-16}
4th round	0000 0000 0000 1010	2^{-20}

Table 5. 12 rounds Differential Characteristic with Probability 2^{-60}

Round	Differential	Probability
Input	0000 0000 0000 1010	1
1st round	0000 000a 0000 000a	2^{-6}
2nd round	0000 0000 0000 0101	2^{-10}
3rd round	000a 0000 000a 0000	2^{-16}
4th round	0000 0000 0000 1010	2^{-20}
5th round	0000 000a 0000 000a	2^{-26}
6th round	0000 0000 0000 0101	2^{-30}
7th round	000a 0000 000a 0000	2^{-36}
8th round	0000 0000 0000 1010	2^{-40}
9th round	0000 000a 0000 000a	2^{-46}
10th round	0000 0000 0000 0101	2^{-50}
11th round	000a 0000 000a 0000	2^{-56}
12th round	0000 0000 0000 1010	2^{-60}

4.3 Attack on 19 rounds GIFT-64

Using the 12 rounds differential characteristic with probability 2^{-60} in Table 5, we could launch a key-recovery attack against 19 rounds GIFT-64. As shown in

Table 6, we add 3 rounds at its beginning and 4 rounds at the end of the differential characteristic. Therefore, we can attack 19 rounds GIFT-64. According to the key schedule, the round key used in 1-*st*, 2-*nd*, 16-*th*, 17-*th*, 18-*th* and 19-*th* round corresponds to (k_1, k_0), (k_3, k_2), $(k_7 \ggg 6, k_6 \ggg 4)$, $(k_1 \ggg 8, k_0)$, $(k_3 \ggg 8, k_2)$ and $(k_5 \ggg 8, k_4)$ in initial key state $(k_7, k_6, k_5, k_4, k_3, k_2, k_1, k_0)$, respectively.

ΔP	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_S^1	?000 0?00 00?0 000? ?000 0?00 00?0 000? ?000 0?00 00?0 000? ?000 0?00 00?0 000?
ΔX_P^1	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ???? ???? ???? ????
ΔX_K^1	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ???? ???? ???? ????
ΔX_S^2	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0?0? 10?0 0?0? 10?0
ΔX_P^2	0000 0000 0000 1??? 0000 0000 0000 0000 0000 0000 0000 1??? 0000 0000 0000 0000
ΔX_K^2	0000 0000 0000 1??? 0000 0000 0000 0000 0000 0000 0000 1??? 0000 0000 0000 0000
ΔX_S^3	0000 0000 0000 0001 0000 0000 0000 0000 0000 0000 0000 0001 0000 0000 0000 0000
ΔX_P^3	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0001 0000
ΔX_K^3	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0001 0000
4th round input	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0001 0000
$\vdots$	$\vdots$
15th round output	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001 0000 0001 0000
ΔX_S^{16}	0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ???? 0000 ???? 0000
ΔX_P^{16}	0000 0000 0000 0?0? 0000 0000 0000 ?0?0 0000 0000 0000 0?0? 0000 0000 0000 ?0?0
ΔX_K^{16}	0000 0000 0000 0?0? 0000 0000 0000 ?0?0 0000 0000 0000 0?0? 0000 0000 0000 ?0?0
ΔX_S^{17}	0000 0000 0000 ???? 0000 0000 0000 ???? 0000 0000 0000 ???? 0000 0000 0000 ????
ΔX_P^{17}	?000 ?000 ?000 ?000 0?00 0?00 0?00 0?00 00?0 00?0 00?0 00?0 000? 000? 000? 000?
ΔX_K^{17}	?000 ?000 ?000 ?000 0?00 0?00 0?00 0?00 00?0 00?0 00?0 00?0 000? 000? 000? 000?
ΔX_S^{18}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_P^{18}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_K^{18}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_S^{19}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_P^{19}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????
ΔX_K^{19}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????

Table 6. 19 rounds Differential Attack on GIFT-64

Data collection

Since GIFT-64 does not have whitening key layer at the beginning, after the P permutation of the first round, we could build 2^n structures. Each structure traverses the 16 bits undetermined in ΔX_P^1, i.e. the bit labeled by "?" in ΔX_P^1 of Table 6, thus it can generate $2^{16\times 2-1} = 2^{31}$ pairs obeying the differential. Therefore, 2^n structures can generate $2^n \times 2^{31} = 2^{n+31}$ pairs.

For such a pair, it has an average probability of 2^{-16} to meet the differential in 4-*th* round in Table 6. Then, the pair encrypted with the right key will obey the differential after 15th round with probability of 2^{-60}. While the pair with a wrong key will obey it with a random probability of 2^{-64}. Therefore, with the right key guess, $2^{n+31} \times 2^{-16} \times 2^{-60} = 2^{n-45}$ pairs will obey the differential after 15th round. Here we choose $n = 47$. So the data complexity is $2^{47} \times 2^{16} = 2^{63}$.

Key recovery

Round	Key bit
1st round	$k_1^{15}, k_1^{14}, k_1^{13}, k_1^{12}, k_1^{11}, k_1^{10}, k_1^{9}, k_1^{8}, k_1^{7}, k_1^{6}, k_1^{5}, k_1^{4}, k_1^{3}, k_1^{2}, k_1^{1}, k_1^{0}$
	$k_0^{15}, k_0^{14}, k_0^{13}, k_0^{12}, k_0^{11}, k_0^{10}, k_0^{9}, k_0^{8}, k_0^{7}, k_0^{6}, k_0^{5}, k_0^{4}, k_0^{3}, k_0^{2}, k_0^{1}, k_0^{0}$
2nd round	$k_3^{15}, k_3^{14}, k_3^{13}, k_3^{12}, k_3^{11}, k_3^{10}, k_3^{9}, k_3^{8}, k_3^{7}, k_3^{6}, k_3^{5}, k_3^{4}, k_3^{3}, k_3^{2}, k_3^{1}, k_3^{0}$
	$k_2^{15}, k_2^{14}, k_2^{13}, k_2^{12}, k_2^{11}, k_2^{10}, k_2^{9}, k_2^{8}, k_2^{7}, k_2^{6}, k_2^{5}, k_2^{4}, k_2^{3}, k_2^{2}, k_2^{1}, k_2^{0}$
16th round	$k_7^{5}, k_7^{4}, k_7^{3}, k_7^{2}, k_7^{1}, k_7^{0}, k_7^{15}, k_7^{14}, k_7^{13}, k_7^{12}, k_7^{11}, k_7^{10}, k_7^{9}, k_7^{8}, k_7^{7}, k_7^{6}$
	$k_6^{3}, k_6^{2}, k_6^{1}, k_6^{0}, k_6^{15}, k_6^{14}, k_6^{13}, k_6^{12}, k_6^{11}, k_6^{10}, k_6^{9}, k_6^{8}, k_6^{7}, k_6^{6}, k_6^{5}, k_6^{4}$
17th round	$k_1^{7}, k_1^{6}, k_1^{5}, k_1^{4}, k_1^{3}, k_1^{2}, k_1^{1}, k_1^{0}, k_1^{15}, k_1^{14}, k_1^{13}, k_1^{12}, k_1^{11}, k_1^{10}, k_1^{9}, k_1^{8}$
	$k_0^{15}, k_0^{14}, k_0^{13}, k_0^{12}, k_0^{11}, k_0^{10}, k_0^{9}, k_0^{8}, k_0^{7}, k_0^{6}, k_0^{5}, k_0^{4}, k_0^{3}, k_0^{2}, k_0^{1}, k_0^{0}$
18th round	$k_3^{7}, k_3^{6}, k_3^{5}, k_3^{4}, k_3^{3}, k_3^{2}, k_3^{1}, k_3^{0}, k_3^{15}, k_3^{14}, k_3^{13}, k_3^{12}, k_3^{11}, k_3^{10}, k_3^{9}, k_3^{8}$
	$k_2^{15}, k_2^{14}, k_2^{13}, k_2^{12}, k_2^{11}, k_2^{10}, k_2^{9}, k_2^{8}, k_2^{7}, k_2^{6}, k_2^{5}, k_2^{4}, k_2^{3}, k_2^{2}, k_2^{1}, k_2^{0}$
19th round	$k_5^{7}, k_5^{6}, k_5^{5}, k_5^{4}, k_5^{3}, k_5^{2}, k_5^{1}, k_5^{0}, k_5^{15}, k_5^{14}, k_5^{13}, k_5^{12}, k_5^{11}, k_5^{10}, k_5^{9}, k_5^{8}$
	$k_4^{15}, k_4^{14}, k_4^{13}, k_4^{12}, k_4^{11}, k_4^{10}, k_4^{9}, k_4^{8}, k_4^{7}, k_4^{6}, k_4^{5}, k_4^{4}, k_4^{3}, k_4^{2}, k_4^{1}, k_4^{0}$

Table 7. Round Keys of GIFT-64

When processing the key recovery, the guessing key bits include: k_1^3, k_1^2, k_1^1, k_1^0, k_0^3, k_0^2, k_0^1, k_0^0 in 1st round, k_3^{12}, k_2^{12}, k_3^4, k_2^4 in 2nd round; k_7^6, k_6^8, k_7^{14}, k_6^0 in 16th round, k_1^7, k_1^6, k_1^5, k_1^4, k_0^3, k_0^2, k_0^1, k_0^0 in 17th round, as well as all 64 key bits in 18th, 19th round. Totally, we construct 2^{80} counters for the possible values of the 80 key bits above. The whole attack procedure is a guess and filter approach. Guess two key bits k_1^0, k_0^0, then we can partially encrypt the plaintexts.

As the middle values of right pairs should obey $\Delta X_S^2\{0\} = 0$, $\Delta X_S^2\{2\} = 0$, $\Delta X_S^2\{3\} = 1$, the $(plaintext, ciphertext)$ pairs can be filtered with a probability of 2^{-3}. Similarly, guessing k_1^i, k_0^i, $i = 1, 2, 3$ and partially encrypt, corresponding conditions in $\Delta X_S^2\{5, 7\}$, $\Delta X_S^2\{8, 10, 11\}$, $\Delta X_S^2\{13, 15\}$ can filter the pairs with 2^{-2}, 2^{-3} and 2^{-2}. Totally 1st round provide a filtering probability of 2^{-10}.

Similarly, the encryption at 2-*nd*, 16-*th*, 17-*th*, 18-*th* round can filter the pairs with probability 2^{-6}, 2^{-8}, 2^{-8}, 2^{-48} while all 32 key bits in 19th round need to be guessed. Thus, 2^{-2} pairs will be left for a random key, while 4 pairs should be left for a right key.

The time complexity is $2^2 \times 2^{31+47} \times 2^{32} = 2^{112}$, the data complexity is 2^{63} and the memory complexity is 2^{80}.

5 Improved MILP-based Method to Find Differential for GIFT-128

GIFT-128 adopts 128-bit state and has 32 4-bit S-boxes in each round. The variables and constrains are twice as many as GIFT-64. The designers of GIFT[2] gives 9 rounds differential characteristics of GIFT-128. We test Algorithm 1 on 9 rounds GIFT-128 and obtain the designers' conclusion. But it costs days to solve. In this section, we devise a segmented MILP-based method to search for longer differential characteristics for GIFT-128.

Suppose we aim to find a r rounds differential characteristic for a block cipher. We first divide it as r_i rounds $(i = 1, 2, ..., t)$ sub-ciphers and $\sum_1^t r_i = r$. We choose probability thresholds for $r_1, r_2, ..., r_t$ rounds ciphers as $P_{r_1}, P_{r_2}, ..., P_{r_t}$, so that the probability p_{r_i} for r_i rounds sub-cipher should be larger than P_{r_i}. Choose a threshold value P_{target} for r rounds. If $p_{r_1} p_{r_2} \dots p_{r_t}$ is larger than P_{target}, an acceptable solution is found.

As shown in Figure 2, for r_i rounds sub-cipher, the input difference are fixed as the output difference of the differential characteristic $\mathcal{D}_{i-1}$ of r_{i-1} rounds sub-cipher, and construct the MILP model $\mathcal{M}_{r_i}$. If $\mathcal{M}_{r_i}$ is feasible, we continue to construct $\mathcal{M}_{r_{i+1}}$ for r_{i+1} rounds sub-cipher; else, we remove $\mathcal{D}_{i-1}$ from $\mathcal{M}_{r_{i-1}}$, and solve it again. The search terminates until we find the differential characteristics of $r_1, r_2, ..., r_t$ rounds sub-ciphers that could be connected to produce a r rounds differential characteristic.

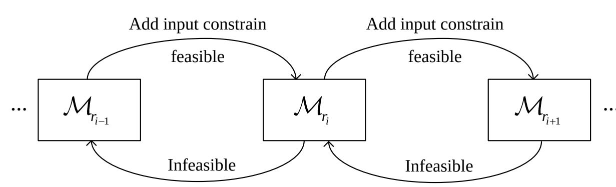

Fig. 2. The framework of our search algorithm

We apply this model to search for differential characteristics for GIFT-128. It is indeed a heuristic and empirical process. For GIFT-128, it is time consuming to solve a more than 6 rounds MILP model. In order to keep the efficiency, we choose $r_i < 6$. P_{r_i} is chosen more flexible. According to the designers' analysis in [2], for 3/4/5 rounds GIFT-128, the numbers of minimum active S-boxes are 3, 5, and 7, respectively. The length of the sub-cipher can neither be too short nor be too long. If the number of rounds is smaller than 2, this sub-MILP-model is unnecessary to solve. On the other hand, if the number of rounds is bigger than 6 or 7, it costs too much time to solve the sub-model that we cannot bear. We do not want the probability of r_i rounds differential characteristic of GIFT-128 to be much smaller than the highest one. So P_{r_i} are chosen according to the minimum active S-boxes of r_i rounds GIFT-128. In this section, we choose $P_{r_i=3} = 2^{-30}$, $P_{r_i=4} = 2^{-40}$ and $P_{r_i=5} = 2^{-50}$ to act as the exact lower bound of differential probability of each sub-model.

We use this model and the strategies above choosing parameters to search for differential characteristics for GIFT-128. We list some results in Table 8. The 12 and 14 rounds differential characteristics are shown in Appendix B.

Table 8. Probabilities of Some Differential Characteristics of GIFT-128

Round	Parameters for r_i	Probability	Source
9	–	2^{-47}	[2]
12	$r_1 = r_2 = r_3 = r_4 = 3$	$2^{-62.415}$	Ours
14	$r_1 = r_2 = 4$ and $r_3 = 6$	2^{-85}	Ours
18	$r_1 = r_2 = r_3 = 4$ and $r_4 = 6$	2^{-109}	Ours

Table 9. 18 rounds Differential Characteristic

Round	Input Difference	Probability
Input	0000 0000 7060 0000 0000 0000 0000 0000	1
1st	0000 0000 0000 0000 0000 0000 00a0 0000	2^{-5}
2nd	0000 0010 0000 0000 0000 0000 0000 0000	2^{-7}
3rd	0000 0000 0800 0000 0000 0000 0000 0000	2^{-10}
4th	0020 0000 0010 0000 0000 0000 0000 0000	2^{-12}
5th	0000 0000 0000 0000 4040 0000 2020 0000	2^{-17}
6th	0000 5050 0000 0000 0000 5050 0000 0000	2^{-25}
7th	0000 0000 0000 0000 0000 0000 0a00 0a00	2^{-37}
8th	0000 0000 0000 0011 0000 0000 0000 0000	2^{-41}
9th	0008 0000 0008 0000 0000 0000 0000 0000	2^{-47}
10th	0000 0000 0000 0000 2020 0000 1010 0000	2^{-51}
11th	0000 5050 0000 0000 0000 5050 0000 0000	2^{-61}
12th	0000 0000 0a00 0a00 0000 0000 0000 0000	2^{-73}
13th	0000 0000 0011 0000 0000 0000 0000 0000	2^{-77}
14th	0090 0000 00c0 0000 0000 0000 0000 0000	2^{-83}
15th	1000 0000 0080 0000 0000 0000 0000 0000	2^{-89}
16th	0010 0000 0000 0000 0000 0000 8020 0000	2^{-94}
17th	0000 0000 8000 0020 0000 0050 0000 0020	2^{-101}
18th	0000 0100 0020 0800 0014 0404 0002 0202	2^{-109}

The 18 rounds characteristic, shown in Table 9, is constructed by the connection of the following three 4 rounds differential characteristics and a 6 rounds differential characteristic:

$$(0000\,0000\,7060\,0000\,0000\,0000\,0000\,0000) \xrightarrow{4-round,\ 2^{-12}} (0020\,0000\,0010\,0000\,0000\,0000\,0000\,0000)$$
$$(0020\,0000\,0010\,0000\,0000\,0000\,0000\,0000) \xrightarrow{4-round,\ 2^{-29}} (0000\,0000\,0000\,0011\,0000\,0000\,0000\,0000)$$
$$(0000\,0000\,0000\,0011\,0000\,0000\,0000\,0000) \xrightarrow{4-round,\ 2^{-32}} (0000\,0000\,0a00\,0a00\,0000\,0000\,0000\,0000)$$
$$(0000\,0000\,0a00\,0a00\,0000\,0000\,0000\,0000) \xrightarrow{6-round,\ 2^{-36}} (0000\,0100\,0020\,0800\,0014\,0404\,0002\,0202)$$

With the 18 rounds differential characteristic, we can add 3 rounds at its beginning and 2 rounds at the end to attack 23 rounds reduced GIFT-128. The attack procedure is similar to subsection 4.3. The time complexity is 2^{120} which is bounded by the data complexity and the memory complexity is 2^{86} bits to store the key counters.

6 Conclusion

In this paper, first, we design a more efficient MILP-based differential search model. Using this model, we give a 12 rounds differential characteristic with probability 2^{-60} and get the first 19 rounds key-recovery attack on GIFT-64. Second, we improve our MILP-based model for block ciphers with large state size. With this model, we give 18 rounds differential characteristic with probability 2^{-109} and obtain the first 23 rounds key-recovery attack on GIFT-128.

MILP can efficiently find high-probabilistic differential characteristics when attacking algorithms whose permutation layer will not cause diffusion. In the future work, we can try to apply heuristic method to constrain global variables, so as to find a higher probability differential characteristics.

References

1. Http://www.sagemath.org/
2. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings. pp. 321–345 (2017), https://doi.org/10.1007/978-3-319-66787-4_16
3. Banik, S., Pandey, S.K., Peyrin, T., Sim, S.M., Todo, Y., Sasaki, Y.: Gift: A small present. Cryptology ePrint Archive, Report 2017/622 (2017), http://eprint.iacr.org/2017/622
4. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The simon and speck families of lightweight block ciphers. Cryptology ePrint Archive, Report 2013/404 (2013), https://eprint.iacr.org/2013/404
5. Berton, G., Daemen, J., Peeters, M., Assche, G.V.: The Keccak sponge function family, http://keccak.noekeon.org/

6. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. pp. 450–466 (2007), https://doi.org/10.1007/978-3-540-74735-2_31
7. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 208–225. Springer (2012)
8. Cui, T., Jia, K., Fu, K., Chen, S., Wang, M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptology ePrint Archive 2016, 689 (2016), http://eprint.iacr.org/2016/689
9. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: MILP-based automatic search algorithms for differential and linear trails for speck. In: Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. pp. 268–288 (2016), https://doi.org/10.1007/978-3-662-52993-5_14
10. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.J.B.: The LED block cipher. In: Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. pp. 326–341 (2011), http://dx.doi.org/10.1007/978-3-642-23951-9_22
11. Li, Z., Bi, W., Dong, X., Wang, X.: Improved conditional cube attacks on keccak keyed modes with MILP method. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. pp. 99–127 (2017), https://doi.org/10.1007/978-3-319-70694-8_4
12. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: International Conference on Information Security and Cryptology. pp. 57–76. Springer (2011)
13. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers. pp. 57–76 (2011), https://doi.org/10.1007/978-3-642-34704-7_5
14. Sasaki, Y., Todo, Y.: New algorithm for modeling s-box in milp based differential and division trail search. In: Farshim, P., Simion, E. (eds.) Innovative Security Solutions for Information Technology and Communications. pp. 150–165. Springer International Publishing, Cham (2017)
15. Sasaki, Y., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III. pp. 185–215 (2017), http://dx.doi.org/10.1007/978-3-319-56617-7_7

16. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-Bit Blockcipher CLEFIA (Extended Abstract). In: Biryukov, A. (ed.) Fast Software Encryption - FSE 2007. Lecture Notes in Computer Science, vol. 4593, pp. 181–195. Springer (2007)
17. Sun, S., Gerault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., Hu, L.: Analysis of aes, skinny, and others with constraint programming. IACR Transactions on Symmetric Cryptology 2017(1), 281–306 (2017), `https://tosc.iacr.org/index.php/ToSC/article/view/595`
18. Sun, S., Hu, L., Wang, M., Wang, P., Qiao, K., Ma, X., Shi, D., Song, L., Fu, K.: Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. Cryptology ePrint Archive, Report 2014/747 (2014), `http://eprint.iacr.org/2014/747`
19. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 158–178. Springer (2014)
20. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. pp. 648–678 (2016), `http://dx.doi.org/10.1007/978-3-662-53887-6_24`

A Difference Distribution Table(DDT) of GIFT S-box

Table 10. DDT of GIFT S-box

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	2	2	0	2	2	2	2	2	0	0	2
2	0	0	0	0	0	4	4	0	0	2	2	0	0	2	2	0
3	0	0	0	0	0	2	2	0	2	0	0	2	2	2	2	2
4	0	0	0	2	0	4	0	6	0	2	0	0	0	2	0	0
5	0	0	2	0	0	2	0	0	2	0	0	0	2	2	2	4
6	0	0	4	6	0	0	0	2	0	0	2	0	0	0	2	0
7	0	0	2	0	0	2	0	0	2	2	2	4	2	0	0	0
8	0	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4
9	0	2	0	2	0	0	2	2	2	0	2	0	2	2	0	0
a	0	4	0	0	0	0	4	0	0	2	2	0	0	2	2	0
b	0	2	0	2	0	0	2	2	2	2	0	0	2	0	2	0
c	0	0	4	0	4	0	0	0	2	0	2	0	2	0	2	0
d	0	2	2	0	4	0	0	0	0	0	2	2	0	2	0	2
e	0	4	0	0	4	0	0	0	2	2	0	0	2	2	0	0
f	0	2	2	0	4	0	0	0	0	2	0	2	0	0	2	2

B 12 and 14 rounds Differential Characteristic of GIFT-128

Table 11. 12 rounds Differential Path

Round	Input Difference	Probability
Input	0000 0000 7060 0000 0000 0000 0000 0000	1
1st	0000 0000 0000 0000 0000 0000 00*a*0 0000	2^{-5}
2nd	0000 0010 0000 0000 0000 0000 0000 0000	2^{-7}
3rd	0000 0000 0800 0000 0000 0000 0000 0000	2^{-10}
4th	0020 0000 0010 0000 0000 0000 0000 0000	2^{-12}
5th	0000 0000 0000 0000 4040 0000 2020 0000	2^{-17}
6th	0000 5050 0000 0000 0000 5050 0000 0000	2^{-25}
7th	0000 0000 0*a*00 0*a*00 0000 0000 0000 0000	2^{-37}
8th	0000 0000 0011 0000 0000 0000 0000 0000	2^{-41}
9th	0090 0000 0000 0000 0060 0000 0000 0000	2^{-47}
10th	1000 0000 0000 0000 0000 0000 0000 2000	2^{-52}
11th	0000 0004 0000 0002 0000 0000 8000 0000	2^{-57}
12th	0000 0000 0404 0020 0200 0010 0101 0000	$2^{-62.415}$

Table 12. 14 rounds Differential Path

Round	Input Difference	Probability
Input	0000 0000 0000 0000 0000 0706 0000 0000	1
1st	0000 0000 0000 0000 0000 0*a*00 0000 0000	2^{-5}
2nd	0000 0000 0000 0100 0000 0000 0000 0000	2^{-7}
3rd	0000 0000 0000 0000 0008 0000 0000 0000	2^{-10}
4th	0000 0000 0000 0000 0000 2000 0000 1000	2^{-12}
5th	0000 0404 0000 0202 0000 0000 0000 0000	2^{-17}
6th	0000 0000 0505 0000 0000 0000 0505 0000	2^{-25}
7th	00*a*0 00*a*0 0000 0000 0000 0000 0000 0000	2^{-37}
8th	1100 0000 0000 0000 0000 0000 0000 0000	2^{-41}
9th	6000 0000 0000 0000 0000 0000 *c*000 0000	2^{-47}
10th	0000 0000 2000 0020 0000 0000 0000 0000	2^{-51}
11th	0041 0000 0000 0000 0014 0000 0000 0000	2^{-55}
12th	9000 0000 0000 *c*000 0000 0000 3000 1000	2^{-66}
13th	0000 0000 0002 0000 0000 0000 8000 0088	2^{-77}
14th	0000 0001 0040 0020 0000 0012 0010 0003	2^{-85}