MILP-based Differential Attack on Round-reduced GIFT*

Baoyu Zhu[1], Xiaoyang Dong[2], and Hongbo Yu[1,3]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, P. R. China
{xiaoyangdong,yuhongbo}@tsinghua.edu.cn
[2] Institute for Advanced Study, Tsinghua University, P. R. China
[3] Science and Technology on Communication Security Laboratory, Chengdu, P.R. China

Abstract. At Asiacrypt 2014, Sun et al. proposed a MILP model [20] to search for differential characteristics of bit-oriented block ciphers. In this paper, we improve this model to search for differential characteristics of GIFT [2], 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-round differential characteristic and a number of iterative 4-round differential characteristics with our MILP-based model. We give a key-recovery attack on 19-round GIFT-64. For GIFT-128, we find a 18-round differential characteristic and give the first attack on 22-round 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 [5], CLEFIA [17], LED [10], PRINCE [6], SIMON and SPECK [3] have been published in last decades. GIFT [2] 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.

Many classical cryptanalysis methods could be converted to mathematical optimization problems which aims to achieve the minimal or maximal value of

* This paper is a corrected version of CT-RSA 2019. We would like to thank Siang Meng Sim for pointing out the error in the attack on 23-round GIFT-128 in the original paper at CT-RSA 2019 [22].

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 for 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 for differential and linear trails [20], whose main idea is to derive some linear inequalities through the H-Representation of the convex hull of all differential patterns and linear bias of S-box. Xiang et al. [21] introduced a MILP model to search for integral distinguisher, Sasaki et al. [16] and Cui et al. [7] 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 [8], MILP-based conditional cube attacks [11] on Keccak [4], etc.

Our Contributions

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

In this paper, we generalize an efficient two-stage MILP-based model inspired by Sun et al.'s two-stage model [18]. 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. Then the inner-MILP part produces the differential characteristic with maximal probability, the differential characteristic should match the truncated differential. With our two-stage model, we find some 12-round differential characteristics of GIFT-64, some of the differential characteristics are iterative. Moreover, using a 12-round differential characteristic with probability of 2^{-60}, we give an attack on 19-round 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. The output difference of a sub-MILP-model should be served 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 the given bound. Using our algorithm, we find some new differential characteristics, including a new 18-round differential characteristic with probability 2^{-109}. We give the first

attack on 22-round[4] GIFT-128 (out of 40 full rounds) with the 18-round differential characteristic. All of the source code is uploaded to GitHub (`https://github.com/zhuby12/MILP-basedModel`).

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	2^{96}	2^{63}	2^{63}	[2]
GIFT-64	MitM	15	2^{120}	2^{8}	2^{64}	[2]
GIFT-64	MitM	15	2^{112}	2^{16}	2^{64}	[14]
GIFT-64	Differential	19	2^{112}	2^{80}	2^{63}	Ours
GIFT-128	Differential	22	2^{114}	2^{53}	2^{114}	Ours

2 Preliminaries

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 thirty-two 4-bit S-boxes for each round.

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.

[4] The attack on 23-round GIFT-128 in our original paper at CT-RSA 2019 [22] is wrong. With the 18-round differential characteristic, we could only get a 22-round key-recovery attack.

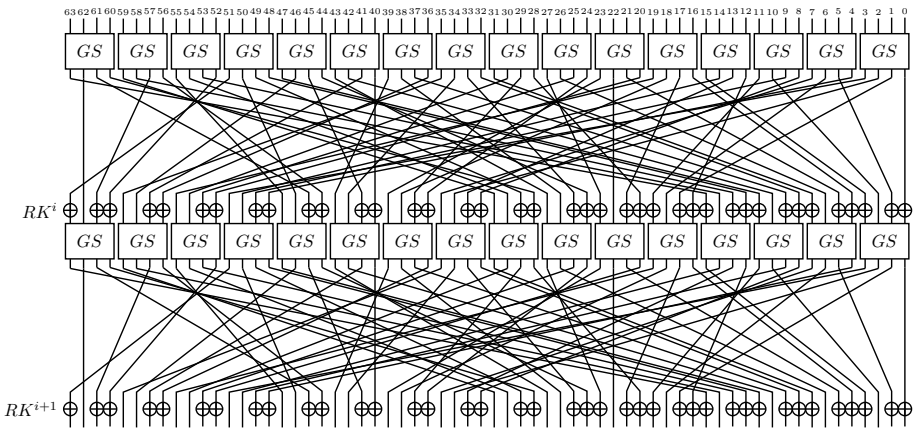

Fig. 1. Two rounds of GIFT-64

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 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 in each round. 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. [12] introduced MILP model to count the number of differentially active S-boxes for word-oriented block ciphers.

Definition 1. *Consider a differential characteristic state 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. [20] extended Mouha et al.'s framework [12] 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 differential characteristic state 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.

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

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

This is called the H-Representation of a $w \times v$ S-box, in which α and β are constant. With the help of SageMath [1], hundreds of linear inequalities can be derived by the differential distribution table of a S-box. But the inequalities is redundant in general, for example, the number of inequalities of GIFT S-box given by SageMath is 237. Because the efficiency of the MILP optimizer is reduced radically when the amount of linear inequalities increase, adding all of the inequalities to the MILP model will make the model insolvable in practical time.

In order to minimize the number of the set of inequalities, Sasaki et al. raised a MILP-based reduction algorithm in [15] to find the optimal combination with minimal number of linear inequalities from hundreds of inequalities in the H-representation of the convex hull. 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 optimizer.

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

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

Two-stage search strategy to find differential characteristics of block ciphers is used in [9,13,18]. In the first step, truncated differential characteristics with minimal active S-box will be found. Then, concrete differential characteristics matching the truncated differential characteristic can be found in a subroutine algorithm. In previous works, one first chose 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. In this section, we propose Algorithm 1 to search for the best or better differential characteristic.

Algorithm 1 MILP-based differential characteristic searching algorithm

Require: r-round block cipher; valid cutting-off inequalities from the convex hull of the S-box; Mr is the minimal number of active S-boxes in all of the r-round differential characteristics.

Ensure: The highest probability; differential characteristics with high probability.

1: Define $MPr = 2^{-64}$ as the initial differential probability of GIFT-64.
2: In the Outer-MILP part, construct a model $\mathcal{M}_1$ describing the differential behavior of the cipher. The target value of $\mathcal{M}_1$ is a truncated differential characteristic, which active S-boxes number is minimum in current solution space. Define $Mr_{bound} = Mr$ as the lower bound of the number of active S-box in $\mathcal{M}_1$.
3: Solve the model $\mathcal{M}_1$ using an MILP optimizer.
4: **if** A feasible solution $\mathcal{TD}$ is found in $\mathcal{M}_1$, save it to a file. **then**
5: $\diamondsuit$ *begin of Inner-MILP part*
6: Construct a MILP model $\mathcal{M}_2$ describing the differential behavior of the cipher and add the truncated differential characteristic $\mathcal{TD}$ as a constraint to $\mathcal{M}_2$. The objective function of $\mathcal{M}_2$ is the differential characteristic with maximal probability.
7: 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.
8: $\diamondsuit$ *end of Inner-MILP part*
9: **end if**
10: Remove the truncated differential $\mathcal{TD}$ from the feasible region of $\mathcal{M}_1$.
11: Solve $\mathcal{M}_1$ again. If a new solution $\mathcal{TD}$ is found and its active S-boxes number is equal to Mr, save it and go to step 5. Else go to step 12.
12: If the number of active S-boxes of is more than Mr and less than $Mr + 3$, set Mr_{bound} equal to $Mr_{bound} + 1$, go to step 5. If a new solution $\mathcal{TD}$ is not found or the number of active S-boxes of $\mathcal{TD}$ is greater than or equal to $Mr + 3$, **return** MPr and the collection of solution x.

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 for Differentials of GIFT-64

Algorithm 1 needs two convex hulls about the S-box in the outer-MILP part and the inner-MILP part respectively. First, we compute the H-representation 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 [15], 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 [19]. 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 tight bound of number of active S-boxes for 11-round and 12-round reduced GIFT-64, which are 22 and 24 respectively. Using the Algorithm 1, we find many 12-round differential characteristics. The highest probability of 12-round differential characteristic is 2^{-59}, the 12-round

differential characteristic with highest probability is shown in Table 4. Meanwhile we get dozens of differential characteristics with probability 2^{-60}.

Table 4. 12-round Differential Characteristic with Probability 2^{-59}

Round	Differential-1	Probability
Input	0c00 0000 0060 0000	1
1st round	0000 0000 0000 4020	2^{-4}
2nd round	0005 0000 0005 0000	2^{-8}
3rd round	0000 0000 2020 0000	2^{-14}
4th round	0050 0000 0050 0000	2^{-18}
5th round	0000 0000 0000 2020	2^{-24}
6th round	0005 0000 0005 0000	2^{-28}
7th round	0000 0000 2020 0000	2^{-34}
8th round	0050 0000 0050 0000	2^{-38}
9th round	0000 0000 0000 2020	2^{-44}
10th round	0000 0000 0005 000a	2^{-49}
11th round	0080 0000 0000 0001	2^{-54}
12th round	1008 0000 0002 2000	2^{-59}

We observe that some of 12-round characteristics are iterative. As a result, we get eight 4-round differential characteristics with probability 2^{-20} totally. These 4-round characteristics are iterative, namely, their input states are identical to their output states. One of them is shown in Table 5, and these characteristics can be extended to more rounds. So we get one of 12-round differential characteristics cycled by three 4-round differential characteristics with probability 2^{-60} in Table 6. A 13-round characteristic with probability 2^{-64} can also be generated by adding another round at the beginning of 12-round differential characteristic. Note that the designers of GIFT claimed that the differential probability of 13-round GIFT-64 will be lower than 2^{-63}. Our result does not violate the claim, however the gap is very small.

Table 5. 4-round 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}

4.3 Attack on 19-round GIFT-64

Using the 12-round differential characteristic with probability 2^{-60} in Table 6, we could launch a key-recovery attack against 19-round GIFT-64. We choose this differential characteristic because its active bits in the head and tail is less than others. As shown in Table 7, we add three rounds at its beginning and

Table 6. 12-round 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}

four rounds at the end of the differential characteristic. Therefore, we can attack 19-round 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.

Table 7. 19-round Differential Attack on GIFT-64

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
⋮	⋮
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}	???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ???? ????

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 sixteen bits undetermined in ΔX_P^1, i.e. the bit labeled by "?" in ΔX_P^1 of Table 7, 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 7. 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 $2^{47} \times 2^{16} = 2^{63}$.

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 8. Round Keys of GIFT-64

Key recovery

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^{15}, k_1^{14}, k_1^{13}, k_1^{12}, 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.

For each of the $2^{47+31} = 2^{78}$ pairs, we repeat the following key guessing phase. 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 bits state and has thirty-two 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-round differential characteristics of GIFT-128. We test Algorithm 1 on 9-round 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-round differential characteristic for a block cipher. We first divide it as r_i-round $(i = 1, 2, ..., t)$ sub-ciphers and $\sum_1^t r_i = r$. We choose probability thresholds for r_1-round,r_2-round,...,r_t-round ciphers as $P_{r_1}, P_{r_2}, ..., P_{r_t}$, so that the probability p_{r_i} for r_i-round sub-cipher should be larger than P_{r_i}. Choose a threshold value P_{target} for r-round. If $p_{r_1} p_{r_2} \ldots p_{r_t}$ is larger than P_{target}, an acceptable solution is found.

As shown in Figure 2, for r_i-round sub-cipher, the input state are fixed as the output state of the differential characteristic $\mathcal{D}_{i-1}$ of r_{i-1}-round 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}-round 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-round,r_2-round,...,r_t-round sub-ciphers that could be connected to produce a r-round differential characteristic.

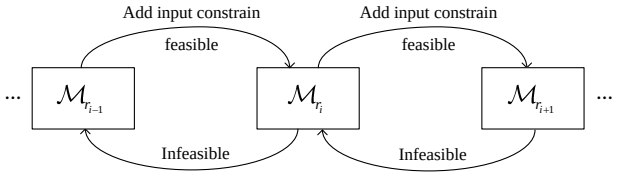

Fig. 2. The framework of our search algorithm

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-round 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-round 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-round 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-round 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 9. The 12-round and 14-round differential characteristics are shown in Appendix C.

Table 9. 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

The 18-round characteristic, shown in Table 10 is constructed by the connection of the following three 4-round differential characteristics and a 6-round differential characteristic:

Table 10. 18-round Differential Characteristic of GIFT-128

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}

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

With the 18-round differential characteristic, we can add three rounds at its beginning and one round at the end to attack 22-round reduced GIFT-128. The attack procedure is similar to subsection 4.3.

Data collection

As shown in Table 11, there are 48 active bits in ΔX_P^1, so we construct data structure by traversing these active bits. Therefore 2^n structures can generate $2^n \times 2^{48\times 2-1} = 2^{n+95}$ pairs. After filtered by 88 zero bits in ΔX_P^{22}, only $2^{n+95-88} = 2^{n+7}$ pairs left.

Key Recovery

When processing the key recovery, the guessing key bits include: in the first round, 24 key bits, i.e. $k_5^{15\sim 8}$, $k_5^{3\sim 0}$, $k_1^{15\sim 8}$, $k_1^{3\sim 0}$; in the 2nd round, 10 key bits, i.e. k_7^{15}, k_7^{14}, k_3^{15}, k_3^{14}, k_7^{12}, k_3^{12}, k_7^{7}, k_3^{7}, k_7^{4}, k_3^{4}; in the 22nd round, 20 key bits, i.e., k_7^{13}, k_7^{12}, k_7^{11}, k_7^{10}, k_6^{9}, k_6^{7}, k_6^{2}, k_6^{0}, k_6^{14}, k_6^{12}, k_3^{8}, k_3^{6}, k_3^{4}, k_3^{2}, k_2^{7}, k_2^{6}, k_2^{5}, k_2^{4}, k_2^{1}, k_2^{15}. Since k_3^4 are involved both in the 2nd round and 22nd round, there are 24+10+20-1=53 key bits are involved in the key recovery phase. Hence, we construct 2^{53} independent counters for the possible values of the 53 key bits above.

The whole attack procedure is a guess and filter approach for the 2^{n+7} pairs:

1. In the first round, guess two key bits k_5^{15}, k_1^{15}, then we can partially encrypt the plaintexts. As the middle values of right pairs should obey $\Delta X_S^2\{127\} = 0$, $\Delta X_S^2\{126\} = 1$, $\Delta X_S^2\{124\} = 0$, the pairs can be filtered with a probability of 2^{-3}. There are 2^{n+4} pairs left.
2. Similarly, guess two key bits for each active S-box, we get filters $2^{-2}, 2^{-2}, 2^{-3}$, $2^{-3}, 2^{-3}, 2^{-3}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-3}, 2^{-4}$. There are $2^{n+4-32} = 2^{n-28}$ pairs left.
3. In the second round, guess two key bits for each active S-boxes, we get filters $2^{-3}, 2^{-3}, 2^{-2}, 2^{-3}, 2^{-2}$. There are $2^{n-28} = 2^{n-41}$ pairs left.
4. In the 22nd round, there are 10 active S-boxes. In each active S-box, we guess 2 key bits and filtered pairs by a factor 2^{-4}. There are $2^{n-41-40} = 2^{n-81}$ pairs left.

Complexity

In the data collection phase, we choose $n = 66$, so the data complexity is $2^{66+48} = 2^{114}$ chosen plaintexts. Under the correct key guessing, number of the right pairs are $2^{66+95-109-48} = 16$. For the wrong key guessing, the expected counter is $2^{66-81} = 2^{-15}$.

The time complexity of the first step is about $2^{66+7+2} = 2^{75}$. Similar time complexities are cost in other steps. The whole time complexity is bounded by the chosen plaintexts, which is 2^{114}.

Table 11. 22-round Differential Attack on GIFT-128

ΔP	???? ????
ΔX_S^1	0?00 00?0 000? ?000 0?00 00?0 000? ?000 0?00 00?0 000? ?000 0?00 00?0 000? ?000 0??0 00?? ?00? ??00 0??0 00?? ?00? ??00 0??0 00?? ?00? ??00 0??0 00?? ?00? ??00
ΔX_P^1	???? ???? ???? ???? ???? ???? ???? ???? 0000 0000 0000 0000 ???? ???? ???? ???? 0000 0000 0000 0000 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 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_S^2	01?0 00?? ?00? ?100 0?00 00?0 000? 1000 0000 0000 0000 0000 0000 01?0 00?? 100? 1100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_P^2	?1?? 1??? 0000 11?? 0000 0000 0000 0000 ?1?? 0000 0000 11?? 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_K^2	?1?? 1??? 0000 11?? 0000 0000 0000 0000 ?1?? 0000 0000 11?? 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_S^3	0010 0001 0000 0100 0000 0000 0000 0000 0010 0000 0000 0100 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_P^3	0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0110 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
ΔX_K^3	0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0110 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
4th round input	0000 0000 0000 0000 0000 0000 0000 0000 0111 0000 0110 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
⋮	⋮
21th round input	0000 0000 0000 0000 0000 0001 0000 0000 0000 0000 0010 0000 0000 1000 0000 0000 0000 0000 0001 0100 0000 0100 0000 0100 0000 0000 0000 0010 0000 0010 0000 0010
ΔX_S^{22}	0000 0000 0000 0000 0000 ???? 0000 0000 0000 0000 ???? 0000 0000 ???? 0000 0000 0000 0000 ???? ???? 0000 ???? 0000 ???? 0000 0000 0000 ???? 0000 ???? 0000 ????
ΔX_P^{22}	0000 00?0 000? 00?0 ?00? ?0?0 ?000 ?0?0 0000 000? ?000 000? ??00 0?0? 0?00 0?0? 0000 ?000 0?00 ?000 0??0 ?0?0 00?0 ?0?0 0000 0?00 00?0 0?00 00?? 0?0? 000? 0?0?
ΔX_K^{22}	0000 00?0 000? 00?0 ?00? ?0?0 ?000 ?0?0 0000 000? ?000 000? ??00 0?0? 0?00 0?0? 0000 ?000 0?00 ?000 0??0 ?0?0 00?0 ?0?0 0000 0?00 00?0 0?00 00?? 0?0? 000? 0?0?

Round	Key bit
1st round	$k_5^{15}, k_5^{14}, k_5^{13}, k_5^{12}, k_5^{11}, k_5^{10}, k_5^9, k_5^8, k_5^7, k_5^6, k_5^5, k_5^4, k_5^3, k_5^2, k_5^1, k_5^0$
	$k_4^{15}, k_4^{14}, k_4^{13}, k_4^{12}, k_4^{11}, k_4^{10}, k_4^9, k_4^8, k_4^7, k_4^6, k_4^5, k_4^4, k_4^3, k_4^2, k_4^1, k_4^0$
	$k_1^{15}, k_1^{14}, k_1^{13}, k_1^{12}, k_1^{11}, k_1^{10}, k_1^9, k_1^8, k_1^7, k_1^6, k_1^5, k_1^4, k_1^3, k_1^2, k_1^1, k_1^0$
	$k_0^{15}, k_0^{14}, k_0^{13}, k_0^{12}, k_0^{11}, k_0^{10}, k_0^9, k_0^8, k_0^7, k_0^6, k_0^5, k_0^4, k_0^3, k_0^2, k_0^1, k_0^0$
2nd round	$k_7^{15}, k_7^{14}, k_7^{13}, k_7^{12}, k_7^{11}, k_7^{10}, k_7^9, k_7^8, k_7^7, k_7^6, k_7^5, k_7^4, k_7^3, k_7^2, k_7^1, k_7^0$
	$k_6^{15}, k_6^{14}, k_6^{13}, k_6^{12}, k_6^{11}, k_6^{10}, k_6^9, k_6^8, k_6^7, k_6^6, k_6^5, k_6^4, k_6^3, k_6^2, k_6^1, k_6^0$
	$k_3^{15}, k_3^{14}, k_3^{13}, k_3^{12}, k_3^{11}, k_3^{10}, k_3^9, k_3^8, k_3^7, k_3^6, k_3^5, k_3^4, k_3^3, k_3^2, k_3^1, k_3^0$
	$k_2^{15}, k_2^{14}, k_2^{13}, k_2^{12}, k_2^{11}, k_2^{10}, k_2^9, k_2^8, k_2^7, k_2^6, k_2^5, k_2^4, k_2^3, k_2^2, k_2^1, k_2^0$
22nd round	$k_7^9, k_7^8, k_7^7, k_7^6, k_7^5, k_7^4, k_7^3, k_7^2, k_7^1, k_7^0, k_7^{15}, k_7^{14}, k_7^{13}, k_7^{12}, k_7^{11}, k_7^{10}$
	$k_6^{11}, k_6^{10}, k_6^9, k_6^8, k_6^7, k_6^6, k_6^5, k_6^4, k_6^3, k_6^2, k_6^1, k_6^0, k_6^{15}, k_6^{14}, k_6^{13}, k_6^{12}$
	$k_3^9, k_3^8, k_3^7, k_3^6, k_3^5, k_3^4, k_3^3, k_3^2, k_3^1, k_3^0, k_3^{15}, k_3^{14}, k_3^{13}, k_3^{12}, k_3^{11}, k_3^{10}$
	$k_2^{11}, k_2^{10}, k_2^9, k_2^8, k_2^7, k_2^6, k_2^5, k_2^4, k_2^3, k_2^2, k_2^1, k_2^0, k_2^{15}, k_2^{14}, k_2^{13}, k_2^{12}$

Table 12. Round Keys of GIFT-128

6 Conclusion

In this paper, first, we design a more efficient MILP-based differential search model. Using this model, we give a 12-round differential characteristic with probability 2^{-60} and get the first 19-round 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-round differential characteristic with probability 2^{-109} and obtain the first 22-round 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. 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`
4. Berton, G., Daemen, J., Peeters, M., Assche, G.V.: The Keccak sponge function family, `http://keccak.noekeon.org/`
5. 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
6. 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)
7. 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`
8. 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
9. Gerault, D., Minier, M., Solnon, C.: Constraint Programming Models for Chosen Key Differential Cryptanalysis. In: Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings. pp. 584-601 (2016)

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), https://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: 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
13. Pierre-Alain, F., Jérémy, J., Thomas, P.: Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In: Advances in Cryptology – CRYPTO 2013. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 183-203.
14. Sasaki, Y.: Integer Linear Programming for Three-Subset Meet-in-the-Middle Attacks: Application to GIFT. In: Advances in Information and Computer Security - 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, September 3-5, 2018, Proceedings. pp. 227-243 (2018)
15. 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)
16. 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), https://doi.org/10.1007/978-3-319-56617-7_7
17. 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)
18. 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`
19. 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`
20. 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)

21. 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), https://doi.org/10.1007/978-3-662-53887-6_24
22. Zhu, B., Dong, X., Yu, H.: MILP-Based Differential Attack on Round-Reduced GIFT. In: Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4-8, 2019, Proceedings. pp. 372-390, (2019). https://doi.org/10.1007/978-3-030-12612-4_19

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

Table 13. 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 Some 4-round Iterative Differential Characteristics of GIFT-64

Table 14. 4-round Iterative Differential Characteristics of GIFT-64

Round	Input Difference	Probability
Input	0005 0000 0005 0000	1
1st	0000 0000 2020 0000	2^{-6}
2nd	0050 0000 0050 0000	2^{-10}
3rd	0000 0000 0000 2020	2^{-16}
4th	0005 0000 0005 0000	2^{-20}
Input	0000 000a 0000 000a	1
1st	0000 0000 0000 0101	2^{-4}
2nd	000a 0000 000a 0000	2^{-10}
3rd	0000 0000 0000 1010	2^{-14}
4th	0000 000a 0000 000a	2^{-20}
Input	0000 00a0 0000 00a0	1
1st	0101 0000 0000 0000	2^{-4}
2nd	a000 0000 a000 0000	2^{-10}
3rd	0000 0000 1010 0000	2^{-14}
4th	0000 00a0 0000 00a0	2^{-20}
Input	0000 0000 0101 0000	1
1st	00a0 0000 00a0 0000	2^{-6}
2nd	1010 0000 0000 0000	2^{-10}
3rd	0000 a000 0000 a000	2^{-16}
4th	0000 0000 0101 0000	2^{-20}
Input	0000 0202 0000 0000	1
1st	0000 0500 0000 0500	2^{-4}
2nd	0202 0000 0000 0000	2^{-10}
3rd	0000 5000 0000 5000	2^{-14}
4th	0000 0202 0000 0000	2^{-20}
Input	0000 1010 0000 0000	1
1st	0000 0a00 0000 0a00	2^{-6}
2nd	0000 0101 0000 0000	2^{-10}
3rd	0a00 0000 0a00 0000	2^{-16}
4th	0000 1010 0000 0000	2^{-20}
Input	0000 0050 0000 0050	1
1st	0000 0000 0000 0202	2^{-6}
2nd	0000 0005 0000 0005	2^{-10}
3rd	0000 0000 0202 0000	2^{-16}
4th	0000 0050 0000 0050	2^{-20}
Input	0500 0000 0500 0000	1
1st	2020 0000 0000 0000	2^{-6}
2nd	5000 0000 5000 0000	2^{-10}
3rd	0000 2020 0000 0000	2^{-16}
4th	0500 0000 0500 0000	2^{-20}

C 12-round and 14-round Differential Characteristics of GIFT-128

Table 15. 12-round Differential Characteristic of GIFT-128

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 0a00 0a00 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 16. 14-round Differential Characteristic of GIFT-128

Round	Input Difference	Probability
Input	0000 0000 0000 0000 0000 0706 0000 0000	1
1st	0000 0000 0000 0000 0000 0a00 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	00a0 00a0 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 c000 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 c000 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}