Related-Key Boomerang Attacks on GIFT with Automated Trail Search Including BCT Effect

Yunwen Liu[1,3] and Yu Sasaki[2]

[1] National University of Defence Technology, Department of Mathematics, China
univerlyw@hotmail.com
[2] NTT Secure Platform Laboratories, Tokyo, Japan, yu.sasaki.sk@hco.ntt.co.jp
[3] imec-COSIC KU Leuven, Belgium

Abstract. In Eurocrypt 2018, Cid et al. proposed a novel notion called the boomerang connectivity table, which formalised the switch property in the middle round of boomerang distinguishers in a unified approach. In this paper, we present a generic model of the boomerang connectivity table with automatic search technique for the first time, and search for (related-key) boomerang distinguishers directly by combining with the search of (related-key) differential characteristics. With the technique, we are able to find 19-round related-key boomerang distinguishers in the lightweight block cipher GIFT-64 and GIFT-128. Interestingly, a transition that is not predictable by the conventional switches is realised in a boomerang distinguisher predicted by the boomerang connectivity table. In addition, we experimentally extend the 19-round distinguisher by one more round. A 23-round key-recovery attack is presented on GIFT-64 based on the distinguisher, which covers more rounds than previous known results in the single-key setting. Although the designers of GIFT do not claim related-key security, bit positions of the key addition and 16-bit rotations were chosen to optimize the related-key differential bound. Indeed, the designers evaluated related-key differential attacks. This is the first work to present better related-key attacks than the simple related-key differential attack. [4]

Keywords: Boomerang connectivity table, GIFT, Automatic search

1 Introduction

Boomerang connectivity table (BCT) [7] is a novel technique proposed by Cid et al. in Eurocrypt 2018 on analysing the middle rounds of boomerang distinguishers. Through the boomerang connectivity table of an S-box, the middle round of a boomerang distinguisher through the S-box layer is described in a unified model similar to differential cryptanalysis with the difference distribution table. As a result, previous methods [3, 4, 8] such as ladder switch and S-box switch

[4] This is a pre-print of an article published in ACISP 2019. The final authenticated version is available online at 10.1007/978-3-030-21548-4.

are special cases of the boomerang transitions predicted by the BCT. Moreover, the boomerang connectivity table reveals new properties in the S-boxes such that new transitions can be derived which are not detectable by any previous methods.

Currently, automatic search has been widely adopted in finding distinguishers in cryptographic primitives, including differential characteristics, impossible differentials and many others [11, 10]. The technique requires an explicit model on the propagation of the differences through a number of rounds, and solves the problem with an MILP (Mixed integer linear programming) or an SMT (Satisfiability module theory) solver. In the scenario of the boomerang attack, due to the lack of unified mathematical model for the middle round of the boomerang distinguishers before the BCT, one searches for differential characteristics in two parts of the encryption function separately, and concatenates them together by analysing the property in the middle round. In ToSC 2017, Cid et al. studied ladder switch for a boomerang attack of Deoxys, searching with an MILP model [6]. Whereas a general technique for the automatic search on boomerang distinguishers is still left unsolved.

In this paper, we propose the first model of the BCT theory with automatic search techniques, and merge it with the search for the related-key differential characteristics. By converting the boomerang connectivity table of an S-box into (vectorial) logical constraints, the propagations of differences through an S-box is completely modeled for the middle round of a boomerang distinguisher. As a result, we are able to search for boomerang distinguishers with a direct evaluation of the middle switches.

As an application, we construct boomerang distinguishers for a recently proposed block cipher GIFT. Proposed by Banik at CHES 2017 [1], GIFT is an improved version of the lightweight block cipher PRESENT [5] with a novel design strategy on the bit-shuffle layer. GIFT-64 and GIFT-128 support 64-bit and 128-bit block sizes, respectively, while both members support the 128-bit key size. With the optimisation on the diffusion of single-bit differences/masks, the number of rounds for GIFT-64 is largely reduced comparing with that of PRESENT. Shortly after the proposal of GIFT, Zhu et al. report a differential attack on 19-round of GIFT-64 based on a 12-round differential distinguisher under the single-key setting [14]. In addition, the security of the cipher against MITM attack and integral cryptanalysis has been studied as well [1, 9]. As far as we know, there is few result on evaluating the cipher in the related-key model. Notice that the key schedule of the GIFT cipher is linear, the attacks under the related-key setting may penetrate more rounds, and reveal a better picture of its security.

Our second contribution is the first third-party security evaluation of the GIFT block cipher in the related-key setting. Based on the automatic search model developed for boomerang distinguishers, we obtain boomerang distinguishers for GIFT-64 (consisting of 28 rounds) and GIFT-128 (consisting of 40 rounds), both cover 19 rounds with two parts of 9-round encryptions and one middle part of 1 round. In addition, with an experimental approach, we extend the 19-round boomerang distinguisher of Gift-64 to several 20-round ones, each with probability $2^{-62.6}$. Afterwards, a key-recovery attack is launched for Gift-64 reduced to 23 rounds, with data complexity $2^{63.3}$ and time complexity 2^{96}. The attack covers about 82% of the entire construction, which well-illustrates the security margin of Gift-64 in the related-key setting. In addition, we give a 21-round attack on Gift-128 based on a 19-round boomerang distinguisher. The attack only reaches (52.5%) of the entire construction. Our analysis implies that the security margin of Gift-128 is better than that of the smaller version. A comparison of our attacks with previous works is summarised in Table 1.

The rest of this paper is organised as follows. In Section 2, an overview of boomerang attacks and the BCT theory is given, as well as an description of the Gift cipher. The mathematical description of the BCT table is converted into an automatic search model in Section 3, with applications to search for boomerang distinguishers in Gift-64 and Gift-128 in Section 4. We extend the boomerang distinguisher into a key-recovery attack for Gift-64 in Section 5. Section 6 concludes the paper.

Table 1. A comparison of attacks on Gift-64 and Gift-128. DC stands for differential cryptanalysis; IC stands for integral cryptanalysis; MITM stands for meet-in-the-middle attack; RK-B stands for related-key boomerang attack.

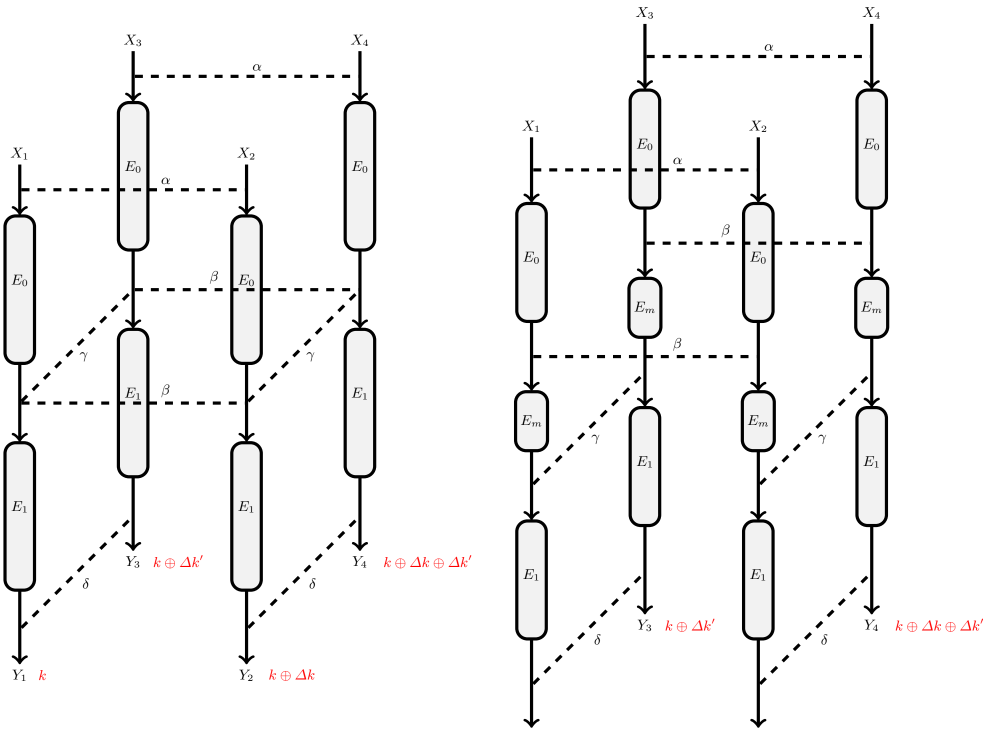
	Type	#rd	Prob.	Attack #rd	Data	Time	cf.
Gift-64	DC	13	2^{-62}	-	-	-	[13]
(28 rounds)	DC	12	2^{-60}	19	2^{63}	2^{112}	[14]
	IC	10	2^{-63}	14	2^{63}	2^{97}	[1]
	MITM			15	2^{64}	2^{120}	[1]
	MITM			15		2^{112}	[9]
	RK-B	20	$2^{-62.6}$	23	$2^{63.3}$	$2^{126.6}$	This paper
Gift-128	DC	18	-	23	2^{120}	2^{120}	[14]
(40 rounds)	RK-B	19	$2^{-121.2}$	21	$2^{126.6}$	$2^{126.6}$	This paper

2 Preliminaries

2.1 Boomerang Attacks

Boomerang attack [12] is an effective cryptanalysis tool, especially for ciphers where the probabilities of the differential characteristics decrease exponentially with respect to the growth of rounds. As a result, the concatenation of two short characteristics may possess a better probability. The diagram of a (related-key) boomerang distinguisher can be illustrated as shown in Figure 1.(1).

The target cipher E is decomposed into two parts E_0 and E_1. Assume that a differential characteristic (α, β) with probability p is found for E_0, and (γ, δ)



(1) Related-key boomerang

(2) Related-key Sandwich

Fig. 1. An illustration of a related-key boomerang (1) and a related-key sandwich (2).

with probability q for E_1. Then the probability of the boomerang distinguisher is

$$\Pr[E^{-1}(E(x) \oplus \delta) \oplus E^{-1}(E(x \oplus \alpha) \oplus \delta) = \alpha] = p^2q^2.$$

The boomerang attack works in a chosen-plaintext and chosen-ciphertext model. In 2001, Biham et al. showed that it is possible to construct a rectangle attack [2] based on a boomerang distinguisher where only the chosen-plaintext setting is required. The technique exploits the fact that a pair of paired values $(x, x \oplus \alpha)$ and $(x', x' \oplus \alpha)$, $x, x' \in \{0,1\}^n$ satisfies the boomerang structure, i.e. $E(x) \oplus E(x') = \delta$ and $E(x \oplus \alpha) \oplus E(x' \oplus \alpha) = \delta$ with probability $p^2q^22^{-n}$, thus may be generated after querying $p^{-1}q^{-1}2^{n/2}$ chosen-plaintext pairs.

2.2 Boomerang Connectivity Table

The partition in the boomerang attack can be extended by decomposing the encryption function into three parts, where the middle round E_m contains many useful transitions. A number of observations and generalisations on boomerang

Table 2. DDT of the GIFT S-box

Δ_i \ Δ_o	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

Table 3. BCT of the GIFT S-box

Δ_i \ ∇_o	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
1	16	0	0	0	0	2	2	0	2	2	2	2	2	0	0	2
2	16	0	4	4	0	8	4	4	0	2	2	0	0	2	2	0
3	16	0	0	0	0	2	2	0	2	0	0	2	2	2	2	2
4	16	4	4	10	4	8	8	6	0	2	0	0	0	2	0	0
5	16	0	2	0	4	2	0	0	2	0	0	4	2	2	2	4
6	16	4	8	6	4	8	4	10	0	0	2	0	0	0	2	0
7	16	0	2	0	4	2	0	0	2	2	2	4	2	0	0	4
8	16	0	0	8	16	0	0	8	0	0	0	8	0	0	0	8
9	16	2	0	2	0	0	2	2	2	0	2	0	2	2	0	0
a	16	8	4	4	0	0	4	4	0	2	2	0	0	2	2	0
b	16	2	0	2	0	0	2	2	2	2	0	0	2	0	2	0
c	16	4	4	8	4	0	0	4	2	0	2	0	2	0	2	0
d	16	2	2	0	4	0	0	0	0	0	2	6	0	2	0	6
e	16	4	0	4	4	0	4	8	2	2	0	0	2	2	0	0
f	16	2	2	0	4	0	0	0	0	2	0	6	0	0	2	6

attack focus on the margin of the decomposition with techniques such as S-box switch, boomerang switch and sandwich attack [4, 8], see Figure 1.(2) for a diagram of a sandwich. Differential behaviours through the S-box are usually summarised in the precomputed table called differential distribution table (DDT). Those research results imply that the transitions of differences in the middle part of a boomerang distinguisher through the S-boxes differ from the prediction from the DDT. In Eurocrypt 2018, Cid *et al.* proposed a novel notion called boomerang connectivity table (BCT), which systematically characterised the propagation of differences and the corresponding probabilities.

Definition 1 (BCT [7]). *Let $S : \{0,1\}^n \rightarrow \{0,1\}^n$ be an invertible function. For input difference Δ_i and output difference ∇_o, the entry (Δ_i, ∇_o) in the boomerang connectivity table $\mathcal{T}(\Delta_i, \nabla_o)$ of S is given by*

$$\mathcal{T}(\Delta_i, \nabla_o) = \#\{x \in \{0,1\}^n \,|\, S^{-1}(S(x) \oplus \nabla_o) \oplus S^{-1}(S(x \oplus \Delta_i) \oplus \nabla_o) = \Delta_i\}.$$

The above definition implies an important feature that the middle round E_m does not require the squared probability p^2 or q^2 because the generation of a right quartet is the probabilistic event over 2^n possibilities. As an example, the DDT and BCT of the GIFT S-box are given in Tables 2 and 3.

The proposal of boomerang connectivity table enables an unified view on the behaviour of the boomerang distinguishers in the middle round(s). Apart from explaining previous results in the literature, the BCT table provides guidance in new improvements on boomerang attacks for certain ciphers.

2.3 The Specification of GIFT

Proposed by Banik *et al.* in CHES 2017, GIFT [1] is a lightweight block cipher which is a descendent of PRESENT [5]. The block size n of GIFT takes 64 bits or 128 bits, and the key size is 128 bits. We denote the corresponding ciphers by GIFT-64 and GIFT-128. One round of GIFT contains only an S-box layer (SubCells), a bit-shuffle (BitPerm) and a round-key injection (AddKey). The round function of GIFT-64 is depicted in Figure 2.

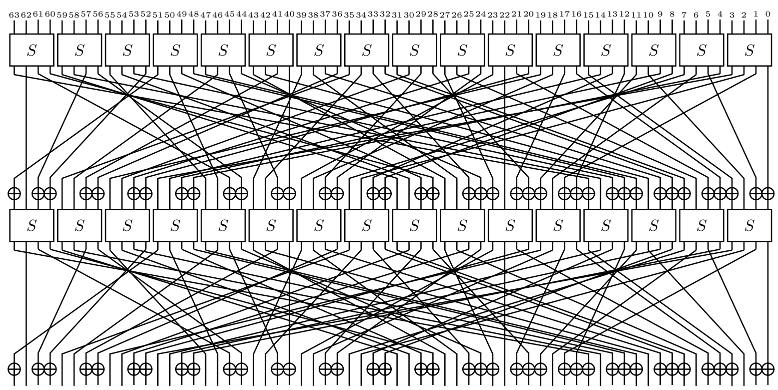


Fig. 2. Two rounds of the block cipher GIFT-64.

Both versions of GIFT adopt the same 4-bit S-box that is different from the S-box in PRESENT.

$$S[16] = \{\mathtt{1}, \mathtt{a}, \mathtt{4}, \mathtt{c}, \mathtt{6}, \mathtt{f}, \mathtt{3}, \mathtt{9}, \mathtt{2}, \mathtt{d}, \mathtt{b}, \mathtt{7}, \mathtt{5}, \mathtt{0}, \mathtt{8}, \mathtt{e}\}.$$

The bit permutation used in GIFT follows a new strategy called BOGI (Bad Output must go to Good Input) to overcome the existence of single active bit path in characteristics. The detail of the permutations can be found in the specification of the cipher [1].

The round keys are XORed to two bits of the 4-bit cells. An $s(= n/2)$-bit round key $RK = U||V = k_1||k_0 = u_{s-1} \cdots u_0||v_{s-1} \cdots v_0$ is obtained from the key state. For GIFT-64, the 128-bit key state is updated as follows,

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

For GIFT-128, $RK = U||V = (k_5||k_4)||(k_1||k_0) = u_{s-1} \cdots u_0||v_{s-1} \cdots v_0$

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

The 128-bit key state is updated as follows,

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

The total number of rounds in GIFT-64 is 28, while the 128-bit version has 40 rounds.

Differential Property. The notable feature of GIFT is that the maximum differential probability for the S-box is $2^{-1.4}$, which is higher than 2^{-2} ensured by many other lightweight block ciphers. In fact, in Table 2, two entries have the value 6, which implies that the transition is satisfied with probability $6/16 \approx 2^{-1.4}$. This contributes relatively larger numbers in BCT, in particular it includes one non-trivial entry that is propagated with probability 1.

3 Automatic Search of (Related-key) Boomerang Based on Boomerang Connectivity Table

In this section, we transform the mathematical description of the boomerang connectivity table into an automatic search model for boomerang distinguishers in block ciphers.

The boomerang connectivity table shares some similarity with difference distribution tables, therefore, it is possible to convert BCT tables into constraints, similar to several previous techniques for DDT tables when dealing with S-boxes in automatic search. As a typical technique which is proposed by Sun *et al.* [11], legal transitions of the differences are modeled as a convex hull and described by a set of linear inequalities. To include the probability information to the model, an additional variable can be allocated to represent the abstract binary logarithm of the probability. As a result, this will probably lead to an increased number of linear inequalities in the model of the Sbox. We notice that a BCT table often encompass more values than the corresponding DDT table, for instance, a differentially 4-uniform S-box may have entries being 6 in its BCT. As a result, it takes more conditions to accurately describe the propagation rules and the corresponding probabilities in a BCT than the corresponding DDT.

In the following, we propose an alternative method to model the BCT table of an S-box with boolean constraints. Assume that for an input difference $\varDelta$, there exist l possible output differences $\{\nabla_0, \ldots, \nabla_{l-1}\} = \mathtt{D}_t(\varDelta)$ where the BCT entries equal to t. We describe the transition $(x \to y)$ with the following logic expression, which evaluates to 1 when $x = \varDelta$ and $y \in \mathtt{D}_t(\varDelta)$, otherwise 0.

$$(x = \varDelta) \wedge ((y = \nabla_0) \vee \cdots \vee (y = \nabla_{l-1})) = (x = \varDelta) \wedge (\bigvee_{\nabla \in \mathtt{D}_t(\varDelta)} (y = \nabla)).$$

In addition, a binary variable w_t is allocated to store the probability information for the BCT entry t. To be specific, when the difference transition is $(x \to y)$, we define w_t as

$$w_t = \bigvee_{\varDelta} ((x = \varDelta) \wedge (\bigvee_{\nabla \in \mathtt{D}_t(\varDelta)} (y = \nabla))).$$

From the expression, w_t evaluates to 1 if one of the possible transitions with BCT value being t is taken.

For instance, in the BCT table of the GIFT S-box (Table 3), when the BCT value t equals 10, there are two possible transitions, namely, $(4 \to 3)$ and $(6 \to 7)$.

So we have

$$w_{10} = ((x = 4) \wedge (y = 3)) \vee ((x = 6) \wedge (y = 7)).$$

It means that if any of the two possible transitions is taken, the variable w_{10} evaluates to 1, which indicates a probability of 10/16 through the S-box.

It is clear that the number of clauses in describing an S-box depends on the nonzero entries of the BCT, corresponding to the variables w_t. In the case of the GIFT Sbox, the number of clauses is 7, where $t = 0, 2, 4, 6, 8, 10, 16$. Therefore, the transitions and their probabilities may be modeled with fewer conditions with our encoding method than before. This is beneficial especially when the number of rounds and the block size are large enough.

To search for a boomerang distinguisher in a block cipher E which is decomposed into three parts E_0, E_m, E_1, one first sets the conditions for valid difference transitions in E_0 and E_1 through the round functions. For the middle round E_m, the propagation through the S-box layer can be modelled with the encoding of BCT discussed above; and we take the linear layer into consideration to connect the characteristics in E_0 and E_1. The probability of the difference propagation through an Sbox can be deduced from the binary variables w_t, which is

$$\sum_t w_t * (t/16).$$

Take the abstract binary logarithm being its weight, and assume that the total weights of the characteristics in E_0, E_1 and E_m are W_0, W_1 and W_m, respectively. The weight of the boomerang is

$$2 * (W_0 + W_1) + W_m.$$

By optimising it, we can directly find a boomerang distinguisher with optimal probability in E.

Remark 1. With related-key differential characteristics, we are able to find related-key boomerang distinguishers. The distinguisher involves four different keys: k and $k \oplus \Delta k$ for a related-key differential characteristic in E_0, and $k \oplus \Delta k'$ and $k \oplus \Delta k \oplus \Delta k'$ in E_1, as shown in Figure 1.

4 Automatic Search of Boomerang Distinguishers in GIFT

In this section, our aim is to apply the automatic search model to search for related-key boomerang distinguishers in GIFT-64 and GIFT-128.

Intuition: Why Boomerang Attacks Can be Strong? We start with finding optimal related-key differential characteristics. Due to the design of the key schedule in GIFT-64, the first four round keys are independent of each other. Thus the number of active S-boxes can be 0 up to 3 rounds by canceling the

Table 4. The minimum number of active S-boxes in related-key differential characteristics of GIFT-64.

#rounds	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
#AS	1	1	2	3	4	6	9	11	13	15	17	19	21	23	25	27

plaintext difference with the first round key. Table 4 shows the minimum number of active S-boxes in related-key differential characteristics of GIFT-64 from 4 rounds.

We observe that the number of active S-boxes slowly increases when the number of rounds is small, especially up to 8 rounds. In contrast, the number of active S-boxes rapidly increases when the number of rounds is large. This is a typical case that the related-key boomerang distinguisher may have a much higher probability than the related-key differential characteristics covering the same number of rounds, by concatenating two short characteristics with high probabilities. Let p_i be the probability of the differential propagation in round i. Then the probability of the differential distinguisher for x rounds is denoted by $\prod_{r=0}^{x} p_i$. In contrast, the boomerang distinguisher basically concatenates two $x/2$-round trail by considering the squared probability, namely $\left(\prod_{r=0}^{x/2} p_i^2\right)^2$. From Table 4, when we increase the number of attacked rounds by 1, the boomerang distinguisher will involve 1 more active S-box with the squared probability and the differential distinguisher will involve 2 more active S-boxes with the normal probability. Those would give almost the same impact to the attack complexity. As a result, the boomerang distinguisher can be more efficient than the differential distinguisher because the boomerang distinguisher can include 3 blank rounds twice (in E_0 and in E_1) and the middle rounds E_m do not require the squared probability.

Finding Boomerang Distinguishers. In this section, we focus on boomerang distinguishers that divide the entire encryption into three parts E_0, E_m and E_1, denoted by $X + 1 + Y$ where X and Y stands for the number of round covered by the differential characteristics in E_0 and E_1, respectively. For instance, an optimal 4-round related-key differential characteristic in GIFT-64 has a probability of $2^{-1.4}$, and it is possible to find a related-key boomerang distinguisher covering 9 rounds with the form $4 + 1 + 4$, where the total probability of the boomerang distinguisher is $(2^{-1.4})^2 \times 1 \times (2^{-1.4})^2 = 2^{-5.6}$.

The strategy of finding boomerang distinguishers follows the theory of the boomerang connectivity table and the model of BCT tables in automatic search techniques. In order to find boomerang distinguishers automatically, our search techniques are based on the model of searching related-key differential characteristics and the translation of BCT table into a solver-friendly language with respect to SMT solvers as explained in Section 3.

The boomerang connectivity table of GIFT S-box is shown in Table 3. For each value in the table, we describe the constraints for valid difference transitions in BCT. For instance, for all the entries $(a \to b)$ taking the value 6, the constraint in SMTLIB-2 language is

```
(= w (bvor (bvand (= a #x2) (= b #x5))
(bvor (bvand (= a #x4) (bvor (= b #x5) (= b #x6)))
(bvor (bvand (= a #x6) (bvor (= b #x2) (= b #x5)))
(bvor (bvand (= a #x8) (bvor (= b #x3) (bvor (= b #x7)
                                        (bvor (= b #xb) (= b #xf)))))
(bvor (bvand (= a #xa) (= b #x1))
(bvor (bvand (= a #xc) (= b #x3))
(bvand (= a #xe) (= b #x7))
))))))),
```

where one of the transitions is taken if $w = 1$.

With the transitions of differences in boomerang distinguishers characterised, we execute the model of GIFT-64 for searching boomerang distinguishers with the form $X+1+X$, where $X = 4, 5, 6, 7, 8, 9, 10$. The probability of the optimal related-key boomerang distinguishers in GIFT-64 which takes the form $X+1+X$ can be found in the following Table 5.

Table 5. The probability of the optimal related-key boomerang distinguishers in GIFT-64 which takes the form $X+1+X$, with a comparison to the probability of the optimal related-key differential characteristics.

#rounds	9	11	13	15	17	19	21
Pr. of RK-boomerang	$2^{-5.6}$	$2^{-5.6}$	$2^{-13.6}$	$2^{-21.6}$	2^{-32}	$2^{-53.6}$	$2^{-79.2}$
Pr. of RK-differential	$2^{-13.4}$	$2^{-28.8}$	2^{-39}	2^{-50}	2^{-61}	2^{-78}	2^{-89}

It can be seen that the distinguishers cover up to 19 rounds of GIFT-64 with a probability larger than 2^{-64}, whereas the probability of the optimal 19-round differential characteristic might be much lower, given that 27 S-boxes are active. We actually searched for the maximum differential characteristic probability for 19 rounds, which was turned out to be 2^{-78}. In Figure 3, we illustrate the comparison between the probabilities of related-key boomerangs and related-key differential characteristics.

Note that we confirmed that the distinguisher does not reach 20 rounds even by relaxing the search space to $X + 1 + Y, X \neq Y$.

Details of the Detected Trail. In Figure 4, we show the detail of a 19-round related-key boomerang distinguisher in GIFT-64. We concatenate two 9-round characteristics of probability $2^{-13.4}$. The transition in the middle round E_m has

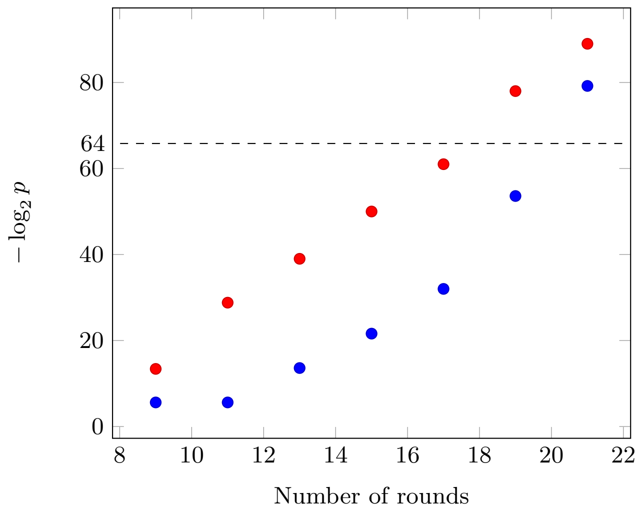


Fig. 3. The comparison between the probabilities of related-key boomerangs and related-key differential characteristics in GIFT-64. The probabilities are shown as the abstract binary logarithm $-\log_2(p)$.

a probability of 2^{-5}, due to the propagation of differences in the BCT table. It is interesting to notice that the transitions $(1 \to 8)$ and $(4 \to 1)$ take advantage of the new properties predicted by the BCT than previous techniques of finding boomerang distinguishers.

Application to GIFT-128. Similarly, we are able to search for boomerang distinguishers in GIFT-128. Usually, the complexity of the problem is proportional to the size of constraints and variables. It is generally more difficult to find characteristics for ciphers with large block size. Therefore, we terminate the program and return the best found solution if necessary. Table 6 shows the probability of the best-found boomerang distinguishers up to 19-rounds for GIFT-128.

Table 6. The probability of the related-key boomerang distinguishers in GIFT-128 which takes the form $X + 1 + X$. Only the 19-round one is not optimal.

#rounds	9	11	13	15	17	19
Pr. of RK-boomerang	$2^{-13.6}$	2^{-24}	2^{-40}	$2^{-59.2}$	$2^{-83.2}$	$2^{-121.2}$

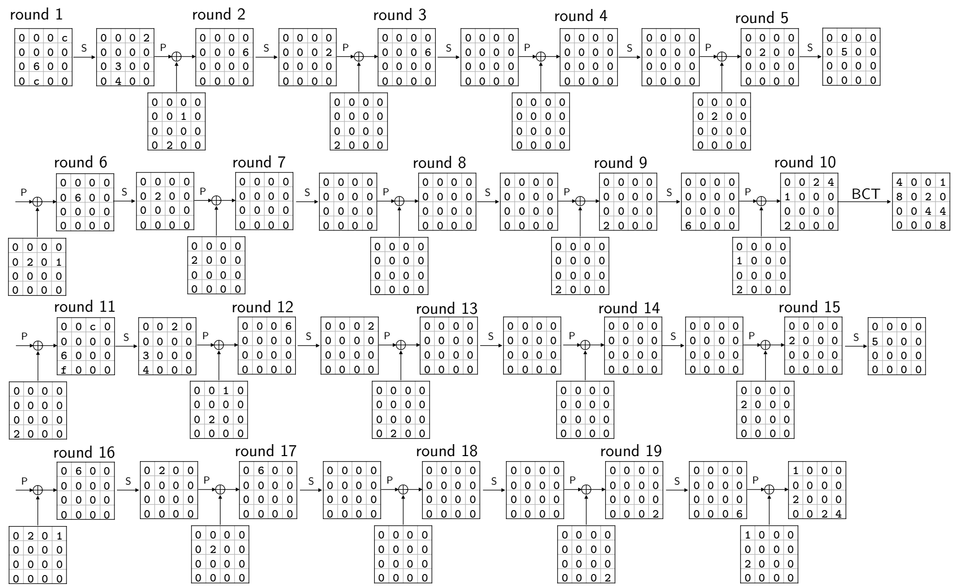


Fig. 4. A 19-round boomerang distinguisher with the form $X + 1 + X$ in GIFT-64, where $X = 9$. The probability is $2^{-58.6}$.

5 Boomerang attack on GIFT-64 and GIFT-128

5.1 Extension of the Distinguisher

As shown by the automatic search, the optimal boomerang distinguisher that covers 19-round GIFT-64 has the probability $2^{-53.6}$, which is obtained by connecting two 9-round related-key characteristics of probability $2^{-13.4}$. The transition probability in the middle round is 1, which largely depends on the output and input differences in E_0 and E_1. For instance, the probability of the middle round in the characteristic in Figure 4 is 2^{-5}.

We extend the 19-round distinguisher for more rounds by using an experimental approach. We enumerate all 9-round characteristics in GIFT-64 with probability $2^{-13.4}$. There are in total 120 such characteristics $\Omega_0, \cdots, \Omega_{119}$. We consider using $\Omega_i, i \in \{0, 1, \ldots, 119\}$ for the first 9 rounds of E_0 and $\Omega_j, j \in \{0, 1, \ldots, 119\}$ for the last 9 rounds of E_1. We have $14,400$ combinations. For each combination, the input and output differences for the middle part E_m are fixed, thus the connecting probability in the middle round(s) can be experimentally found. Notice that many characteristics share the same input and output differences. After removing the duplicated patterns, there are 16 distinct output differences from E_0 and 58 distinct input difference to E_1. Hence, the total number of patterns to be checked is reduced to $16 \times 58 = 928$.

Table 7. All Distinct Input and Output Differences of $\Omega_0, \cdots, \Omega_{119}$

ID	Output Diff from E_0	ID	Output Diff from E_0	ID	Output Diff from E_0	ID	Output Diff from E_0
01	0100040000000102	05	4000000010000201	09	0040000000120100	13	2010004000200000
02	0100040002000002	06	4000200000000201	10	0040002000020100	14	2010004000000010
03	0004000200002010	07	1000400000001020	11	0201000400020000	15	0400020000201000
04	0004000000012010	08	1000400020000020	12	0201000400000001	16	0400000001201000
ID	**Input Diff to E_1**	**ID**	**Input Diff to E_1**	**ID**	**Input Diff to E_1**	**ID**	**Input Diff to E_1**
01	0000600e00000006	16	000c0000d6000000	31	600c0000000c0000	46	000000600000600d
02	0000600f00000006	17	0000000c0000d600	32	600d0000000c0000	47	000000c00000600c
03	0000600e0000000c	18	0000000c0000f600	33	0000e60000006000	48	000000c00000600d
04	0000600f0000000c	19	0000000c0000e600	34	0000f60000006000	49	00c00000600c0000
05	0000600c00000006	20	0000000c0000c600	35	0000e6000000c000	50	00c00000600d0000
06	0000600d00000006	21	000000060000e600	36	0000f6000000c000	51	00c00000600e0000
07	0000600c0000000c	22	000000060000c600	37	0000c6000000c000	52	00c00000600f0000
08	0000600d0000000c	23	000000060000d600	38	0000d6000000c000	53	00600000600c0000
09	00060000e6000000	24	000000060000f600	39	0000c60000006000	54	00600000600d0000
10	00060000f6000000	25	600e000000060000	40	0000d60000006000	55	00600000600e0000
11	000c0000e6000000	26	600f000000060000	41	000000600000600e	56	00600000600f0000
12	000c0000f6000000	27	600e0000000c0000	42	000000600000600f	57	c600000060000000
13	00060000c6000000	28	600f0000000c0000	43	000000c00000600e	58	c6000000c0000000
14	00060000d6000000	29	600c000000060000	44	000000c00000600f	59	
15	000c0000c6000000	30	600d000000060000	45	000000600000600c	60	

For each of the patterns, we generate $2^{13}(= 8,192)$ random keys and state values to experimentally check the probability that the middle round is satisfied. The number of rounds for E_m is a parameter. When we set the number of rounds for E_m is 1, namely when the boomerang characteristic has the form $9+1+9$, we have 34 combinations such that the probability of the middle round is 1.

The experiment can be extended for boomerang distinguishers with the form $9+Y+9$, where the middle part contains $Y = 2, 3$ rounds. Only 10 combinations result in a probability larger than 2^{-10} when $Y = 2$, while all combinations have a probability lower than 2^{-15} for $Y = 3$. As a consequence, we are able to push the 19-round boomerang distinguisher for one round more, and obtain 20-round distinguishers with probability $2^{-62.6}$ as shown in Table 8.

5.2 Key Recovery Attacks

The boomerang distinguisher found above can be extended to a 23-round key-recovery attack against Gift-64 by adding one round in the beginning and two rounds at the end.

The linear layer in the last round does not impact to our attack. We omit in order to keep the description of the attack procedure as simple as possible. Note that the bit positions of the key injection need to change accordingly to the BitPerm operation. However, BitPerm is designed to be closed in each register in the bit-slice implementation. Namely, the first and the second bits of each S-box is XORed by the round key. Indeed, bit-positions $4i$ for $i = 0, 1, ..., 15$ move to bit-position $4j$ for $j = 0, 1, ..., 15$ and the same applies to bit-positions from $4i + 1$ to $4j + 1$.

Table 8. A 20-round boomerang distinguisher of the form $9+2+9$ by concatenating two 9-round characteristics with probability $2^{-13.4}$. The probability of the middle connection is $2^{-8.34}$. The difference nibbles $\mathtt{x} \in \{\mathtt{6}, \mathtt{c}\}$, $\mathtt{y} \in \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$, $(\mathtt{w}, \mathtt{z}) \in (\mathtt{2}, \mathtt{0}), (\mathtt{0}, \mathtt{1})$. The key differences in the two middle rounds follow those in E_1.

Round	Characteristic	Key difference k_7 k_6 $\cdots$ k_1 k_0
0	00x00000600y0000	0040 0000 0000 0000 0004 0000 0008 0020
1	0000006000000000	0002 0200 0040 0000 0000 0000 0004 0000
2	0000000000000000	0001 0000 0002 0200 0040 0000 0000 0000
3	0000000000000000	0000 0000 0001 0000 0002 0200 0040 0000
4	0000000002000000	0010 0000 0000 0000 0001 0000 0002 0200
5	0000000000000060	8000 2000 0010 0000 0000 0000 0001 0000
6	0000000000000000	4000 0000 8000 2000 0010 0000 0000 0000
7	0000000000000000	0000 0000 4000 0000 8000 2000 0010 0000
8	0000000000020000	0004 0000 0000 0000 4000 0000 8000 2000
9	2010004000200000	2000 0002 0004 0000 0000 0000 4000 0000
10	2-round BCT	1000 0000 2000 0002 0004 0000 0000 0000
11	0000600d00000006	0400 0000 0000 0000 4000 0000 0010 0040
12	0000060000000000	0004 0400 0400 0000 0000 0000 4000 0000
13	0000000000000000	1000 0000 0004 0400 0400 0000 0000 0000
14	0000000000000000	0000 0000 1000 0000 0004 0400 0400 0000
15	0000020000000000	0100 0000 0000 0000 1000 0000 0004 0400
16	0000000000000600	0001 4000 0100 0000 0000 0000 1000 0000
17	0000000000000000	0400 0000 0001 4000 0100 0000 0000 0000
18	0000000000000000	0000 0000 0400 0000 0001 4000 0100 0000
19	0000000200000000	0040 0000 0000 0000 0400 0000 0001 4000
20	010004000w000z02	

The distinguisher covers the segment from round 2 to round 21. We prepare the plaintext quartets with the desired input difference at the first round, and perform 2-round partial decryptions on the ciphertexts under the guessed key. To produce the output difference as predicted, we need to make $Q = 2^n p_b^{-2}$ quartets, where n is the block size and p_b is the probability of the boomerang distinguisher. By birthday paradox, the quartets can be generated by making pairs between p_1 and p_2 as well as p_3 and p_4, separately. Each case requires $Q^{1/2}$ queries. After combining them, we get Q quartets with $2 \times (Q^{1/2} + Q^{1/2})$ queries in total, where a pair requires 2 queries. Unfortunately, a direct estimation of the data complexity turns out to exceed the total data available. Therefore, we need to utilise the input differences of the boomerang distinguishers in Table 8, and generate the required quartets with fewer queries. In the following, let the output difference be `0100040000000102`.

The detail of the attack procedure is as follows.

Step 1: (Offline) We have SubCells, BitPerm and AddKey before the 20-round distinguisher. Since the round-key difference can be derived through the linear key schedule, the difference after SubCells in the first round is known. When we choose plaintext, we choose the internal state values after SubCells in the first round to satisfy this difference. We then compute the inverse of SubCells offline to generate the plaintext.

Step 2: (Online) The goal of this step is to make $D = 2^{63.3}$ queries to generate $Q = 2^{126.6}$ quartets. With a probability of 2^{-64}, the encryptions with E_0 of the quartets match the intermediate difference γ, thus we can expect one right quartet satisfying the boomerang distinguisher. The procedure is shown below.

2.(a): At the beginning of the boomerang distinguisher, fix x to `6`. Then the truncated differences is `00600000600y0000`, where $\mathtt{y} \in \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. Notice that the difference on the 16-th and 17-th bit can take any value.

2.(b): Fix a plaintext value p_1 and take all four cases of the 16-th and 17-th bits. Query those 4 plaintexts to the oracle with key K.

2.(c): Compute p_2 by $p_2 = p_1 \oplus \alpha$. Then, make 4 queries to the oracle with $K^{\oplus}\Delta_k$ by testing all the four cases for the 16-th and 17-th bits.

2.(d): Generate $4 \times 4 = 16$ pairs from the above 8 queries.

2.(e): Repeat the process for $2^{59.3}$ different values of p_1 ($2^{62.3}$ queries in total) to generate $2^{63.3}$ pairs of p_1, p_2.

2.(f): Prepare the pairs between p_3 and p_4 analogously, with $2^{62.3}$ queries we generate $2^{63.3}$ pairs of p_3, p_4. By birthday paradox, we get Q quartets p_1, p_2, p_3, p_4 by combining the pairs p_1, p_2 and p_3, p_4.

Step 3: The differential propagation for the extended two rounds after the 20-round distinguisher is shown in Fig. 5.

Collect right quartet candidates where the outputs after 23-rounds of encryption have inactive nibbles at the 1st, 5th, 11th and 13th nibbles for both pairs of c_1, c_3 and c_2, c_4.

Step 4: Guess 8 key-bits at round 22 and 24 key-bits at round 23 for the partial decryption of the ciphertext quartets c_1, c_2, c_3, c_4, which leads to the middle states m_1, m_2, m_3, m_4 having the output difference from the 20-round distinguisher. The positions of the involved key-bits are shown in Figure 5.

Step 5: Exhaustively search for the remaining $128 - 32 = 96$ bits of the key.

From the procedure of Step 2, the data complexity of the attack D is $2^{62.3} + 2^{62.3} = 2^{63.3}$ queries in total. After the filter by the ciphertext difference at Step 3, we obtain $Q \times 2^{-16-16} = 2^{94.6}$ right quartet candidates. At Step 4, we guess $8 + 24 = 32$ key bits and apply partial decryption for all $2^{94.6}$ candidates, it will take $2^{94.6} \times 2^{32} = 2^{126.6}$ 2-round decryptions. Step 4 involves 16 S-boxes and the probability that all the 16 S-boxes will behave as expected is 2^{-128} for each wrong guess. Hence, we expect the only 1 key survives after Step 4.

5.3 21-Round Key Recovery on GIFT-128.

Note that the optimal boomerang distinguisher we obtained in the previous section for GIFT-128 covers the same number of rounds as that of GIFT-64 even though the attacker can make queries up to 2^{128} plaintexts. Such inefficiency

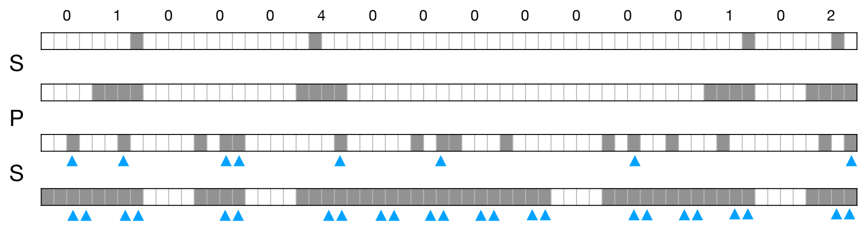


Fig. 5. The difference propagation in the final two rounds when the output difference of the boomerang is `0100040000000102`. The blue triangles label the positions of the guessed key bits.

in GIFT-128 comes from the larger round key size. GIFT-128 injects 64 key bits in every round, which is double of the GIFT-64. This significantly improves the speed of differential diffusion, which only allows the attack up to the same number of rounds as GIFT-64.

We present the 21-round attack on GIFT-128 based on the 19-round boomerang distinguisher found in the previous section. Table 9 shows the 9-round differential characteristic used for the concatenation of the 19-round boomerang. The probability of the 19-round boomerang distinguisher is $2^{-121.2}$, where the middle round switch takes a probability of 2^{-2} as predicted by the BCT.

Table 9. A 9-round differential characteristic of probability $2^{-29.8}$ which can be extended into a 19-round boomerang distinguisher with the form $9+1+9$. The column of the key differences shows the values (k_5, k_4, k_1, k_0) for generating the differences used in round keys.

Round	Characteristic	Key difference (k_5 k_4 k_1 k_0)
0	`000006000000e0000000000000000060`	`1000 0000 4000 0001`
1	`00000000000000000000000000000000`	`0008 0000 0000 0000`
2	`00000000000040000000000000000000`	`0000 1000 0010 4000`
3	`00000000000000000205000000000000`	`0000 0008 0000 0000`
4	`00000000000010000000200000000000`	`0400 0000 0004 0010`
5	`000000000000000000000000000a0000`	`0002 0000 0000 0000`
6	`00000000000000000000002000000000`	`0000 0400 0100 0004`
7	`00000002000000000000000000000000`	`0000 0002 0000 0000`
8	`00000000040000000200000000000040`	`0100 0000 0040 0100`
9	`00200005021010000000000600404002`	

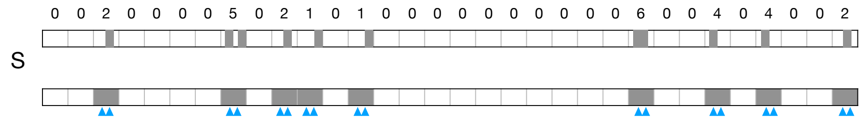


Fig. 6. The difference propagation in the final round when the output difference of the boomerang is `00200005021010000000000600404002`. The blue triangles label the positions of the guessed key bits in 9 S-boxes with nonzero differences.

The distinguisher can be extended to a 21-round attack (one round before and one round after the distinguisher, the final round has no permutation layer) on Gift-128 with the following procedure.

Step 1: (Offline) This stage is similar to the attack on Gift-64, where the attacker prepares the input quartets offline to extend the distinguisher by one round at the beginning.

Step 2: (Online) We make $2^{126.6}$ queries to generate $2^{249.2}$ quartets. With a probability of 2^{-128}, the encryptions with E_0 of the quartets match the intermediate difference γ, and it is sufficient to produce one right quartet satisfying the boomerang distinguisher.

2.(a): Take the difference `000006000000e000000000000000000060` at the beginning of the boomerang distinguisher.

2.(b): We need $2^{125.6}$ queries to generate $2^{124.6}$ pairs between p_1 and p_2. Similarly for p_3 and p_4.

2.(c): By birthday paradox, we get $2^{249.2}$ quartets p_1, p_2, p_3, p_4 by combining the pairs p_1, p_2 and p_3, p_4.

Step 3: Collect the outputs after 23-rounds of encryption. Guess 18 key-bits at round 21 for the partial decryption of the ciphertext quartets c_1, c_2, c_3, c_4, and we obtain the middle states m_1, m_2, m_3, m_4. The guessed key bits are located in those 9 S-boxes with a nonzero difference in the output difference as shown in Figure 6. With the ciphertext filtering technique, we have a gain of 2^{92} since there are 23 nibbles with no difference after the S-box layer.

Step 4: Check the differences among the quartets of the middle states, if the difference match the boomerang distinguisher, the guessed key bits are the candidates for the right keys.

Step 5: The remaining $128 - 18 = 110$ bits of the key is recovered by an exhaustive search.

The data complexity of the attack is $2^{126.6}$. And the time complexity is $2^{126.6} \times 2^{18} \times 2^{-92} + 2^{110} \approx 2^{110}$ partial encryptions. Hence the bottleneck of the complexity is the memory accesses to $2^{126.6}$ queried data.

6 Conclusion

In this paper, we study the automatic search model of boomerang connectivity table and its applications. By converting the boomerang connectivity table into

SMT language, we are able to directly model the propagations in boomerang distinguishers with an automatic search based on the search of differential characteristics. It enables us to find optimal switches in the middle round(s) which may not be predictable by previous techniques. As an application, our target is a recently proposed block ciphers family Gift, and related-key boomerang distinguishers covering 19 rounds of Gift-64 and Gift-128 are found with the automatic search model. Moreover, we experimentally extended the 19-round distinguisher of Gift-64 into a 20-round one, and launched a key-recovery attack against Gift-64 reduced to 23 rounds. Our analysis shows that Gift-64 seems to have a smaller security margin than that of Gift-128.

Acknowledgement

The authors would like to thank the reviewers for their valuable comments. Yunwen Liu is supported by National Natural Science Foundation (No. 61672530, No. 61702537) and Research Fund KU Leuven grant C16/18/004.

References

1. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: a small PRESENT. In: International Conference on Cryptographic Hardware and Embedded Systems - CHES 2017. pp. 321–345. Springer (2017)
2. Biham, E., Dunkelman, O., Keller, N.: The rectangle attack - rectangling the serpent. In: Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding. pp. 340–357 (2001)
3. Biryukov, A., Cannière, C.D., Dellkrantz, G.: Cryptanalysis of SAFER++. In: Advances in Cryptology - CRYPTO 2003. pp. 195–211 (2003)
4. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Advances in Cryptology - ASIACRYPT 2009. pp. 1–18 (2009)
5. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2007. pp. 450–466. Springer (2007)
6. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: A security analysis of deoxys and its internal tweakable block ciphers. IACR Trans. Symmetric Cryptology 2017(3), 73–107 (2017)
7. Cid, C., Huang, T., Peyrin, T., Sasaki, Y., Song, L.: Boomerang connectivity table: A new cryptanalysis tool. In: Advances in Cryptology - EUROCRYPT 2018. pp. 683–714 (2018)
8. Dunkelman, O., Keller, N., Shamir, A.: A practical-time related-key attack on the KASUMI cryptosystem used in GSM and 3G telephony. J. Cryptology 27(4), 824–849 (2014)
9. 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. pp. 227–243 (2018)

10. 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. pp. 185–215 (2017)
11. 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: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. Lecture Notes in Computer Science, vol. 8873, pp. 158–178. Springer (2014)
12. Wagner, D.A.: The boomerang attack. In: Fast Software Encryption, FSE '99. pp. 156–170 (1999)
13. Zhou, C., Zhang, W., Ding, T., Xiang, Z.: Improving the MILP-based security evaluation algorithms against differential cryptanalysis using divide-and-conquer approach. https://eprint.iacr.org/2019/019.pdf
14. Zhu, B., Dong, X., Yu, H.: Milp-based differential attack on round-reduced gift. https://eprint.iacr.org/2018/390.pdf