# Improving Matsui's Search Algorithm for the Best Differential/Linear Trails and its Applications for DES, DESL and GIFT

Fulei Ji[1,2], Wentao Zhang[1,2], and Tianyou Ding[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{jifulei, zhangwentao}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** Automatic search methods have been widely used for cryptanalysis of block ciphers, especially for the most classic cryptanalysis methods – differential and linear cryptanalysis. However, the automatic search methods, no matter based on MILP, SMT/SAT or CP techniques, can be inefficient when the search space is too large. In this paper, we propose three new methods to improve Matsui's branch-and-bound search algorithm which is known as the first generic algorithm for finding the best differential and linear trails. The three methods, named *Reconstructing DDT and LAT According to Weight*, *Executing Linear Layer Operations in Minimal Cost* and *Merging Two 4-bit S-boxes into One 8-bit S-box* respectively, can efficiently speed up the search process by reducing the search space as much as possible and reducing the cost of executing linear layer operations. We apply our improved algorithm to DESL and GIFT, which are still the hard instances for the automatic search methods. As a result, we find the best differential trails for DESL (up to 14 rounds) and GIFT-128 (up to 19 rounds). The best linear trails for DESL (up to 16 rounds), GIFT-128 (up to 10 rounds) and GIFT-64 (up to 15 rounds) are also found. To the best of our knowledge, these security bounds for DESL and GIFT under single-key scenario are given for the first time. Meanwhile, it is the longest exploitable (differential or linear) trails for DESL and GIFT. Furthermore, benefiting from the efficiency of the improved algorithm, we do experiments to demonstrate that the clustering effect of differential trails for 13-round DES and DESL are both weak.

**Keywords:** Matsui's search algorithm; Differential trail; Linear trail; Clustering effect; DESL; GIFT-128; GIFT-64; DES

## 1 Introduction

Differential cryptanalysis [6] and linear cryptanalysis [19] are two of the most fundamental methods for cryptanalysis of block ciphers. The first and the most important step of differential cryptanalysis (or linear cryptanalysis) is to find differential trails (or linear trails) with high probabilities (or large correlations). Therefore, how to find effective differential or linear trails has become a hot issue for cryptographers.

At EUROCRYPT'94 [20], Matsui proposed a branch-and-bound depth-first search algorithm, which presents an efficient automatic search algorithm to find the best differential trails and linear trails of DES. Differential trails with highest probability or linear trails with largest correlation are called the best. However, Matsui's method is not efficient enough for some other block ciphers such as DESL and FEAL. At CRYPTO'95 [22], Moriai et al. improved Matsui's algorithm. They introduced the concept of search patterns to reduce unneccessary search candidates and found the best $n$-round linear trails of FEAL ($n \leq 32$). At FSE'97 [3], Aoki et al. further improved the search algorithm by proposing the pre-search strategy to discard unsatisfiable search patterns. They determined all the best differential trails of FEAL up to 32 rounds. In [5], Bao et al. proposed three strategies to speed up the search algorithm. They got good results on the best differential and linear trails of NOEKEON and SPONGENT. Bao's three strategies can efficiently speed up the search process. However, it needs quite complex programming skills to implement their strategies.

The idea of search patterns and the pre-search strategy are hard to implement when the weights of the differential or linear trails of the target cipher are not integers. The definitions of *the weight of difference propagation (or linear correlation)* and *the weight of differential (or linear) trail* are given in Sect.2.1. For example, the weights of the difference propagations of S-boxes in DESL are not all integers. It is not convenient for DESL to perform the pre-search phase, since assigning decimal weights to every single round is difficult. In [3], Aoki et al. provided a version of pre-search algorithm that specially designed for decimal weights, but this algorithm is complicated to implement.

In recent years, Mixed-Integer Linear Programing (MILP) based method has been very popular in constructing automatic search algorithm in differential and linear cryptanalysis [21,26,24,25,23]. MILP-based method is convenient to programing. But when the block size or the round number of the target cipher is large, the size of the corresponding MILP model will be too large. There are some papers [28,17] using MILP-based method to study the security of GIFT-128. But they can only get differential trails under some limitations and they cannot estimate the best difference propagation probabilities when the round number is large. In [23], Sun et al. applied MILP-based method to search differential trails of DESL, but they can only found a 10-round single-key differential trail of DESL with probability $2^{-52.25}$.

The weights of the difference propagations of the S-boxes in DESL and GIFT and the weights of the linear propagations in DESL are not all integers. As mentioned above, the original Matsui's method and the MILP-based method are both not efficient enough to search the best differential trails of DESL or GIFT-128, and it is really complicated to apply the idea of search patterns to accelerate the search process.

How to find multiple differential or linear distinguishers [8] is another problem that researchers pay attention [2,11,14]. In [2], Abdelraheem et al. presented a time-memory trade-off method to search the multipule linear trails of SIMON. They combined the trails found by constructing multi-round correlation submatrixes and by Mixed Integer Programming model. In [14], Mathias et al. proposed a breadth-first search algorithm. The algorithm represents the problem of searching multiple differential or linear trails as the problem of finding many long paths through a multistage graph.

### 1.1 Our Contributions

In this paper, we apply three methods to speed up the original Matsui's algorithm. By these three speeding-up methods, we can efficiently prune unsatisfiable candidates and reduce the cost of executing linear layer operations:

- **Reconstructing DDT and LAT According to Weight** method is very helpful to reduce the complexity of the search process. We apply this method to prune unsatisfiable candidates by sorting the input and output differences according to their weights.
- **Executing Linear Layer Operations in Minimal Cost** method can be used to reduce the cost of P permutation and E expansion. By constructing linear layer table and executing look-up-table operations by SSE instructions, we can implement linear layer operations of each round with 2 XOR operations of 128-bit variables.
- **Merging Two 4-bit S-boxes into One 8-bit S-box** method is used to speed up the search process when the number of S-box is large. In the case of GIFT, by merging two consecutive $4 \times 4$ S-boxes into one $8 \times 8$ S-box, we can further reduce the cost of executing linear layer operations.

We use our improved search algorithm to search the best differential trails and linear trails of DES, DESL, GIFT-64 and GIFT-128. The results are helpful for estimating the security of DESL, GIFT-64 and GIFT-128 against differential and linear cryptanalysis. Besides, we do experiments to estimate the clustering effect of differential trails for 13-round DES and DESL.

We use method 1 and method 2 to accelerate the search process of DES and DESL. Method 1, method 2 and method 3 are all used to accelerate the search process of GIFT-64 and GIFT-128.

For DES and DESL, our experimental results presented in Table 3 show that the first two methods bring an acceleration by a factor of 26-173. For GIFT-64 and GIFT-128, our experimental results presented in Table 4 show that the third method brings an acceleration by a factor of 2-4. All of the experiments and results in this paper are obtained and timed on a PC with Intel(R) Core(TM) i7-6700 3.40 GHz CPU, and 16 GB RAM, using single-thread program in C.

**Results on DES and DESL:**

- **For DES**. We find the best differential trails for up to 18 rounds[3]. The best difference probability of 18-round DES is $2^{-69.84}$. The results are summarized in Table 3. We find the best linear trails of DES for up to 22 rounds. The best linear correlation of 22-round DES is $2^{-32.46}$. The results are summarized in Table 5.
  We confirm the results on the best difference and linear probability of DES provided in [20,22]. As we can see from Table 3, our new methods can efficiently speed up the search process.
- **For DESL**. We find the best differential trails for up to 14 rounds. The best difference probability of 14-round DESL is $2^{-68.78}$. The results are summarized in Table 3. We find the best linear trails of DESL for up to 16 rounds. The best linear correlation of 16-round DESL is $2^{-32.98}$. The results are summarized in Table 5.
  To the best of our knowledge, we find the longest exploitable best differential and linear trails for DESL. In [24], the authors found a 10-round differential trail with probability $2^{-52.25}$, and there is no previous results on linear trails of DESL.
- **Demonstrating the Clustering Effect**. In [12], it has been mentioned that the differential and linear clustering effect of DES are both weak. But no experimental result was provided to demonstrate this conclusion.
  In this paper, we conduct experiments applying the improved Matsui's algorithm to search the clustering effect of differential trails for 13-round DES and DESL. Through the experiments in Sect.6, we find that the clustering effect of differential trails for 13-round DES and DESL are both weak.

**Results on GIFT:**

- **For GIFT-64**. We find the best differential trails for up to 14 rounds. The best difference probability of 14-round GIFT-64 is $2^{-68.000}$. The results are summarized in Table 4. We find the best linear trails of GIFT-64 for up to 15 rounds. The best linear correlation of 13-round GIFT-64 is $2^{34.00}$. The results are summarized in Table 6.
  We find the longest exploitable best linear trails of GIFT-64 and we confirm the results on the best difference probability of GIFT-64 provided in [27].
- **For GIFT-128**. We find the best differential trails for up to 19 rounds. The best difference probability of 19-round GIFT-128 is $2^{-110.830}$. The results are summarized in Table 4.
  We find the best linear trails of GIFT-128 for up to 10 rounds. The best linear correlation of 10-round GIFT-128 is $2^{-26.00}$. The results are summarized in Table 6.
  We find the longest exploitable best differential and linear trails for GIFT-128. The results on the best difference probabilities and best linear correlations provided in this paper are the tightest security bounds of GIFT-128, which are found for the first time.

To illustrate the efficiency of our improved algorithm, we compare our results on the weight of the best differential trails with previous work in Table 1.

**We find no result on the best linear trails of DESL or GIFT from previous work**. Since the weights of the linear propagations of S-boxes in GIFT are integers, Bao's algorithm can be very suitable to search the best linear trails of GIFT. We compare our results on the weight of the best linear trails with the results of Bao's algorithm in Table 2.

As shown in Table 2, for SPN block ciphers, when the weights are all integers, Bao's algorithm is more efficient than our improved Matsui's algorithm. But we need to state that: **The improved**

[3] The round number of DES is 16.

**Table 1.** The weight of the best differential trails

| DESL | | | GIFT-64 | | | GIFT-128 | | |
|---|---|---|---|---|---|---|---|---|
| $n$ | $B_n$ (ours) | $B_n$ ([23]) | $n$ | $B_n$ (ours) | $B_n$ ([27]) | $n$ | $B_n$ (ours) | $B_n$ ([17]) |
| 7 | 29.25 | | 7 | 28.415 | 28.415 | 14 | 79.000 | 79* |
| 8 | 37.59 | | 8 | 38.000 | 38.000 | 15 | 85.415 | 86* |
| 9 | 41.76 | | 9 | 42.000 | 42.000 | 16 | 90.415 | 91* |
| 10 | 49.79 | 52.52* | 10 | 48.000 | 48.000 | 17 | 96.415 | 97* |
| 11 | 52.71 | | 11 | 52.000 | 52.000 | 18 | 103.415 | 103.415* |
| 12 | 58.10 | | 12 | 58.000 | 58.000 | 19 | 110.830 | 115* |
| 13 | 60.78 | | 13 | 62.000 | 62.000 | 20 | 121.415* | 121.415* |
| 14 | 68.78 | | 14 | 68.000 | 68.000 | 21 | 126.415* | 126.415* |

[1] $n$ : the round number. $B_n$ : the weight of the best differential trails of $n$-round.
[2] $*$ : the authors cannot guarantee that it is the best weight.

**Table 2.** The weight of the best linear trails

| GIFT-64 | | | | | GIFT-128 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $n$ | $B_n$ | $t$ (ours) | pre-search workload | $t$ ([5]) | $n$ | $B_n$ | $t$ (ours) | pre-search workload | $t$ ([5]) |
| 9 | 20 | 41.47s | 1 | 0.00s | 5 | 7 | 0.05s | 10 | 47.44s |
| 10 | 25 | 0.85h | 1 | 0.23h | 6 | 10 | 1.77s | 6 | 0.63s |
| 11 | 29 | 8.83h | 2 | 36.54h | 7 | 13 | 19.38s | 5 | 0.01s |
| 12 | 31 | 0.73h | 2 | 1.73s | 8 | 17 | 0.50h | 4 | 24.03s |
| 13 | 34 | 3.15h | 2 | 35.45h | 9 | 22 | 44.09h | 3 | 1.01h |
| 14 | 37 | 5.12h | 1 | 0.18s | 10 | 26 | 20.9d | 3 | 36.72h |
| 15 | 40 | 4.59h | 1 | 24.26s | | | | | |

[1] $n$ : the round number. $B_n$ : the weight of the best linear trails of $n$-round.
[2] $t$ : the search time.

**Matsui's algorithm in this paper is easier to implement, and is more convenient to be applied in situations when the weights are not all integers**.

In addition, the improved Matsui's algorithm can be used to search the best differential trails and best linear trails for other primitives, estimate the clustering effect and search the multiple differential distinguishers or multi-dimensional linear distinguishers.

### 1.2 Organization

The paper is organized as follows. Some definitions and symbolic conventions are presented in Sect.2. Meanwhile, we introduce the three block ciphers: DES, DESL and GIFT and Matsui's search algorithm in Sect.2. In Sect.3, we propose our three speeding-up methods. Sect.4 and Sect.5 give the experimental results on the best differential and linear trails of DES, DESL and GIFT, providing justification for the efficiency of the three speeding-up methods. Sect.6 gives experimental results on estimating the clustering effect of differential trails for 13-round DES and DESL. Sect.7 is the conclusion and discussion.

## 2 Preliminaries

In [20], Matsui illustrated the duality between differential cryptanalysis and linear cryptanalysis. In this paper, **we take differential cryptanalysis as an example to explain our speeding-up methods**.

### 2.1 Notations and Definitions

The notations we use are similar to those in [20]. We use the sum of the weights of the round differential trails to characterize the best differential trails.

$X_i, Y_i$ : the input and the output of the round function $F_i$

$X_i^t, Y_i^t$ : the input and the output of the $t$-th S-box

$\Delta X_i$, $\Delta Y_i$, $\Delta X_i^t$, $\Delta Y_i^t$ : the differential value of $X_i$, $Y_i$, $X_i^t$ and $Y_i^t$

$\Gamma X_i, \Gamma Y_i$ : the masking value of $X_i$ and $Y_i$

$K_i$ : the round key of the round function $F_i$

$\oplus$ , $\bullet$ : the bitwise XOR operation and AND operation

$P(x)$ : P permutation on 0-1 string $x$

$E \circ P(x)$ : composition of P permutation and E expansion on 0-1 string $x$

$parity(x) \stackrel{def}{=} x_0 \oplus x_1 \oplus \cdots \oplus x_{n-1}$, in which $x = x_0||x_1||\cdots||x_{n-1}$ is 0-1 string

**Definition 1** ([12]). The correlation $C(a^{'}, b^{'})$ between the linear propagation $(a^{'}, b^{'})$ over the transformation $h$ is defined as:

$$C(a^{'}, b^{'}) = 2Prob\{parity(a \bullet a^{'}) = parity(h(a) \bullet b^{'})\} - 1 \tag{1}$$

$a^{'}$ is the input mask and $b^{'}$ is the output mask, $a$ traverses all the input values of $h$.

**Definition 2** ([12]). The weight of a difference propagation $(a^{'}, b^{'})$ is the negnative of the binary logarithm of the difference propagation probability over the transformation $h$, i.e.,

$$w_r(a^{'}, b^{'}) = -log_2^{Prob^h(a^{'}, b^{'})} \tag{2}$$

$a^{'}$ is the input difference and $b^{'}$ is the output difference.

**Definition 3** ([12]). The weight of a linear correlation $C(a^{'}, b^{'})$ is the negnative of the binary logarithm of the absolute value of $C(a^{'}, b^{'})$ over the transformation $h$, i.e.,

$$w_r(a^{'}, b^{'}) = -log_2^{|C(a^{'}, b^{'})|} \tag{3}$$

$a^{'}$ is the input mask and $b^{'}$ is the output mask.

**Definition 4** ([12]). The weight of a differential (or linear) trail $Q$ over an iterative transformation $H$ is the sum of the weights of its differential (or linear) steps, i.e.,

$$w_r(Q) = \sum_i w_r^{h^{(i)}}(q^{i-1}, q^i) \tag{4}$$

The iterative transformation $H$ is a sequence of $r$ transformations:

$$H = h^{(r)} \circ \cdots \circ h^{(2)} \circ h^{(1)} \tag{5}$$

The differential (or linear) trail $Q$ over $H$ consists of a sequence of $r+1$ difference (or linear mask) values:

$$Q = (q^{(0)}, q^{(1)}, \cdots, q^{(r-1)}, q^{(r)}) \tag{6}$$

In the case of **differential cryptanalysis**, we calculate the weights of the difference propagation probabilities:

$(\Delta X_i, \Delta Y_i) \stackrel{def}{=} -log_2^{Prob\{F_i(X_i \oplus \Delta X_i, K_i) = F_i(X_i, K_i) \oplus \Delta Y_i\}}$

$[w_1, w_2, \cdots, w_t] \stackrel{def}{=} \Sigma_{i=1}^{t} w_i$

$B_n \stackrel{def}{=} min[(\Delta X_1, \Delta Y_1), (\Delta X_2, \Delta Y_2), \cdots, (\Delta X_n, \Delta Y_n)]$

Note:

$\Delta X_i = \Delta X_{i-2} \oplus E \circ P(\Delta Y_{i-1}), 3 \leq i \leq n$, in the case of DES and DESL;

$\Delta X_i = P(\Delta Y_{i-1}), 2 \leq i \leq n$, in the case of GIFT.

In the case of **linear cryptanalysis**, we calculate the weights of the linear correlations:

$(\Gamma X_i, \Gamma Y_i) \stackrel{def}{=} -1 - log_2^{|Prob\{parity(X_i \bullet \Gamma X_i) = parity(F_i(X_i, K_i) \bullet \Gamma Y_i)\} - 1/2|}$

$[w_1, w_2, \cdots, w_t] \stackrel{def}{=} \Sigma_{i=1}^{t} w_i$

$B_n \stackrel{def}{=} min[(\Gamma X_1, \Gamma Y_1), (\Gamma X_2, \Gamma Y_2), \cdots, (\Gamma X_n, \Gamma Y_n)]$

Note:

$\Gamma Y_i = \Gamma Y_{i-2} \oplus E \circ P(\Gamma X_{i-1}), 3 \leq i \leq n$, in the case of DES and DESL;

$\Gamma X_i = P(\Gamma Y_{i-1}), 2 \leq i \leq n$, in the case of GIFT.

### 2.2 Description of Feistel Block Cipher DES and DESL

The Data Encryption Standard [15] (DES) was developed at IBM and adopted by the U.S. National Bureau of Standards as the standard cryptosystem for sensitive but unclassified data. DES is a Feistel structure block cipher whose block size is 64 bits and key size is 56 bits. The round number of DES is 16. We illustrate the Feistel structure and the round function of DES in Fig.1. We refer readers to [15] for more details of DES.

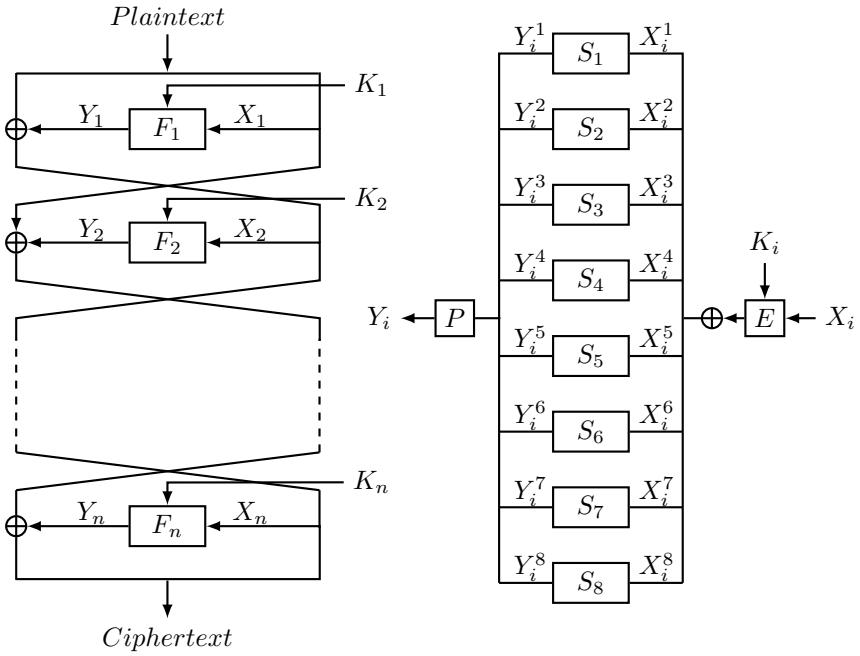


**Fig. 1.** Round function of DES

– There are 8 different S-boxes in DES. We denote them as $S_i, 1 \leq i \leq 8$. For each S-box, it takes an input of 6 bits and gives an output of 4 bits.
– E is a bitwise expansion. It takes an input of 32 bits and gives an output of 48 bits. P is a bitwise permutation. It takes an input of 32 bits and gives an output of 32 bits.

DESL [16] is a lightweight variant of DES. DESL is almost the same as DES, except that it uses a single S-box instead of 8 different S-boxes as DES. The designers adapted the DES S-box design criteria and proposed the S-box used in DESL. Using the well designed S-box, DESL is stronger than DES in resisting differential and linear cryptanalysis. We refer readers to [16] for more details of DESL.

### 2.3 Description of SPN Block Cipher GIFT

GIFT [4] is an SP-network lightweight block cipher. GIFT has two versions named in GIFT-64 and GIFT-128, whose block sizes are 64 and 128 bits respectively, and round numbers are 28 and 40 respectively. The key length of GIFT-64 and GIFT-128 are both 128 bits.

Recently, Banik et al. proposed a lightweight Authenticated Encryption (AE) scheme GIFT-COFB based on GIFT-128, which is one of the Round 1 Candidates of NIST Lightweight Crypto Standardization process [1].

There are 32 same $4 \times 4$ S-boxes in GIFT-128 and 16 same $4 \times 4$ S-boxes in GIFT-64. The round function of GIFT-128 is shown in Fig.2. We refer readers to [4] for more details of GIFT.

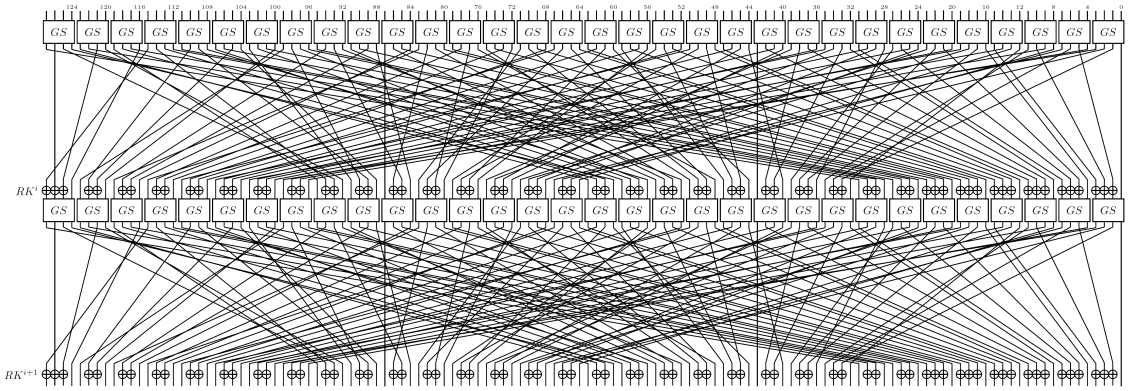


**Fig. 2.** Two rounds of GIFT-128 [1]

### 2.4 Matsui's Search Algorithm

Matsui's algorithm works by induction on the number of rounds and derives the best $n$-round weight $B_n$ from the knowledge of all $i$-round best weight $B_i$ $(1 \leq i \leq n-1)$. The program requires an initial value for $B_n$, which is represented as $Bc_n$. It works correctly for any $Bc_i$ as long as $Bc_i \geq B_i$ $(1 \leq i \leq n-1)$.

The original search algorithm is recursive and targets DES. Alg.1 and Alg.2 show the details of Matsui's search algorithm targets Feistel cipher and SPN cipher respectively.

## 3 Our New Speeding-up Methods

**Overall Strategy** In order to speed up Matsui's search algorithm, we need to further study the inner details of Matsui's algorithm and the objective block ciphers: DES, DESL, GIFT.

We speed up the search process in the following two ways:

**Algorithm 1** Matsui's Algorithm Targets Feistel Cipher

**Input:** $n\ (\geq 4)$; $B_1, B_2, \cdots, B_{n-1}$; $Bc_n$
**Output:** $B_n = Bc_n$; the best differential trails of round $n$

**Procedure** Round-1:
**for** each candidate of $\Delta X_1$ **do**
  $w_1 \leftarrow min_{\Delta Y_1}(\Delta X_1, \Delta Y_1)$
  **if** $[w_1, B_{n-1}] \leq Bc_n$ **then**
    call Round-2
  **end if**
**end for**

**Procedure** Round-2:
**for** each candidate of $\Delta X_2$ and $\Delta Y_2$ **do**
  $w_2 \leftarrow (\Delta X_2, \Delta Y_2)$
  **if** $[w_1, w_2, B_{n-2}] \leq Bc_n$ **then**
    call Round-3
  **end if**
**end for**
**Procedure** Round-$i, 3 \leq i \leq n-1$:
**for** each candidate of $\Delta Y_i$ **do**
  $\Delta X_i \leftarrow \Delta X_{i-2} \oplus E \circ P(\Delta Y_{i-1})$
  $w_i \leftarrow (\Delta X_i, \Delta Y_i)$
  **if** $[w_1, \cdots, w_i, B_{n-i}] \leq Bc_n$ **then**
    call Round-($i$+1)
  **end if**
**end for**

**Procedure** Round-$n$:
$\Delta X_n \leftarrow \Delta X_{n-2} \oplus E \circ P(\Delta Y_{n-1})$
$w_n \leftarrow min_{\Delta Y_n}(\Delta X_n, \Delta Y_n)$
**if** $[w_1, w_2, \cdots, w_n] \leq Bc_n$ **then**
  $Bc_n = [w_1, w_2, \cdots, w_n]$
**end if**
return to the upper procedure

1 As mentioned in [22], the complexity of the search process for the $n$-round best trails is dominated by the number of candidates in Procedure Round-1 and Procedure Round-2. We apply the speeding-up method in Sect.3.1 to prune unsatisfiable candidates as soon as we can;
2 When we get a new candidate of $\Delta Y_i$, we need to calculate $E \circ P(\Delta Y_i)$ or $P(\Delta Y_i)$. We apply the speeding-up methods in Sect.3.2 and Sect.3.3 to reduce the cost of executing linear layer operations.

### 3.1 Reconstructing DDT and LAT According to Weight

In Sect.3.1, **We take Alg.1 and DES as an example to illustrate how to construct new look-up tables**. Alg.1 is a branch-and-bound algorithm, pruning unsatisfiable candidates by the inequalities set in each round. In [20], Matsui suggested that we should try $\Delta X_i^j$ and $\Delta Y_i^j$ in order of the magnitude of $(\Delta X_i^j, \Delta Y_i^j)$. In [5], Bao et al. also searched $\Delta X_i^j$ and $\Delta Y_i^j$ in order of their weights. Inspired by their ideas, we reconstruct the difference distribution table in Procedure Round-1, Round-2 and Round-$i$ $(3 \leq i \leq n-1)$ respectively, ranking the input-output difference pairs in order of their weights. With the help of the new look-up tables, we can efficiently search the input-output difference pairs with large weights and throw away input-output difference pairs with small weights as soon as possible. By doing this, we can significantly accelerate the search process.

There are 9 different weights of the difference propagations for each S-box in DES: 5.00, 4.00, 3.42, 3.00, 2.68, 2.42, 2.20, 2.00, 0.00. We use a table to denote this weight set. The elements are sorted in a descending order:

**WeightTable[9]**$= \{5.00, 4.00, 3.42, 3.00, 2.68, 2.42, 2.20, 2.00, 0.00\}$

**In Procedure Round-1** Since the output difference has no effect on the subsequent search process, we only care about the input difference and their weights. We construct one table to classify the input differences of each S-box according to the corresponding weights:

**DDTwX[SboxN][WeightN][InN]**

**Algorithm 2** Matsui's Algorithm Targets SPN Cipher

**Input:** $n\ (\geq 3)$; $B_1, B_2, \cdots, B_{n-1}$; $Bc_n$
**Output:** $B_n = Bc_n$; the best differential trails of round $n$

**Procedure** Round-1:
**for** each candidate of $\Delta Y_1$ **do**
  $w_1 \leftarrow min_{\Delta X_1}(\Delta X_1, \Delta Y_1)$
  **if** $[w_1, B_{n-1}] \leq Bc_n$ **then**
    call Round-2
  **end if**
**end for**

**Procedure** Round-$i, 2 \leq i \leq n-1$:
$\Delta X_i \leftarrow P(\Delta Y_{i-1})$
**for** each candidate of $\Delta Y_i$ **do**
  $w_i \leftarrow (\Delta X_i, \Delta Y_i)$
  **if** $[w_1, \cdots, w_i, B_{n-i}] \leq Bc_n$ **then**
    call Round-$(i+1)$
  **end if**
**end for**

**Procedure** Round-$n$:
$\Delta X_n \leftarrow P(\Delta Y_{n-1})$
$w_n \leftarrow min_{\Delta Y_n}(\Delta X_n \Delta Y_n)$
**if** $[w_1, w_2, \cdots, w_n] \leq Bc_n$ **then**
  $Bc_n = [w_1, w_2, \cdots, w_n]$
**end if**
return to the upper procedure

- SboxN represents the index of the S-box. It ranges from 1 to 8. WeightN represents the index of the weights. It ranges from 0 to 8. InN represents the index of the input difference. It ranges from 0 to 63.
- DDTwX$[t][j][r]$ represents the $r$-th input difference of the $t$-th S-box with a weight WeightTable$[j]$.

**In Procedure Round-2** We care about the input-output difference pairs and their weights. We construct two tables:

**DDTXorder[SboxN][WeightN][InN]**
**DDTYorder[SboxN][WeightN][OutN]**

- InN represents the index of the input difference. OutN represents the index of the output difference. The range of these two values depends on the choice of the S-box. For DES, InN and OutN range from 0 to 267 respectively.
- DDTXorder$[t][j][r]$ represents the $r$-th input difference of the $t$-th S-box with a weight WeightTable$[j]$. Its corresponding output difference is DDTYorder$[t][j][r]$.

**In Procedure Round-$i$, $3 \leq i \leq n-1$** Since the input difference is fixed, we care about the corresponding output differences and their weights. We construct two tables:

**DDTY[SboxN][InV][OutN]**
**DDTYw[SboxN][InV][OutN]**

- InV represents the value of the input difference. It ranges from 0 to 63. OutN represents the index of the output difference. It ranges from 0 to 15.
- DDTY$[t][j][r]$ represents the $r$-th output difference of the $t$-th S-box whose input difference is $j$. Its corresponding weight is DDTYw$[t][j][r]$.
- We rank the output differences of each input difference in order of their weights: if $r_1 < r_2$ , then DDTYw$[t][j][r_1] \leq$ DDTYw$[t][j][r_2]$.

The accelerated Procedure Round-$i$, $1 \leq i \leq n-1$ using new look-up tables is shown in Alg.3. We omit the constraints between the the input differences of adjacent S-boxes in Alg.3.

In Procedure Round-1. For the $t$-th S-box, we traverse the weights of its difference propagations from WeightTable[8] to WeightTable[0]. If WeightTable[j] does not satisfy the inequality:

$$[w_1, B_{n-1}] \leq Bc_n \tag{7}$$

**Algorithm 3** Our Search Approach for DES and DESL

**Input:** $n$ ($\geq 4$); $B_1, B_2, \cdots, B_{n-1}$; $Bc_n$; WeightTable[9]
**Output:** $B_n = Bc_n$; the best differential trails of $n$-round

**Generate Tables** :
DDTwX[SboxN][WeightN][InN]
DDTXoder[SboxN][WeightN][InN]
DDTYoder[SboxN][WeightN][OutN]
DDTY[SboxN][InV][OutN]
DDTYw[SboxN][InV][OutN]
EPtable[SboxN][OutV][LocN]

**Procedure** Round-1:
$w_1 \leftarrow 0, \Delta X_1 \leftarrow 0, t \leftarrow 1$
**Function** Sbox-1$(t, w_1)$:
**for** $j = 8$ to 0 **do**
  $\alpha \leftarrow w_1 + \text{WeightTable}[j]$
  **if** $[\alpha, B_{n-1}] \geq Bc_n$ **then**
    break
  **else**
    **for** each DDTwX$[t][j][r]$ **do**
      $\Delta X_1^t \leftarrow \text{DDTwX}[t][j][r]$
      **if** $t < 8$ **then**
        call Sbox-1$(t+1, \alpha)$
      **else**
        $w_1 \leftarrow \alpha$
        call Round-2
      **end if**
    **end for**
  **end if**
**end for**

**Procedure** Round-2:
$w_2 \leftarrow 0, \Delta X_2 \leftarrow 0, \Delta Y_2 \leftarrow 0, t \leftarrow 1$
**Function** Sbox-2$(t, w_2)$:
**for** $j = 8$ to 0 **do**
  $\alpha \leftarrow w_2 + \text{WeightTable}[j]$
  **if** $[w_1, \alpha, B_{n-2}] \geq Bc_n$ **then**
    break
  **else**
    **for** each DDTXoder$[t][j][r]$ **do**
      $\Delta X_2^t \leftarrow \text{DDTXoder}[t][j][r]$
      $\Delta Y_2^t \leftarrow \text{DDTYoder}[t][j][r]$
      **if** $t < 8$ **then**
        call Sbox-2$(t+1, \alpha)$
      **else**
        $w_2 \leftarrow \alpha$
        call Round-3
      **end if**
    **end for**
  **end if**
**end for**

**Procedure** Round-$i, 3 \leq i \leq n-1$:
$\Delta X_i \leftarrow \Delta X_{i-2} \oplus E \circ P(\Delta Y_{i-1}), t \leftarrow 1$
$w_i \leftarrow \text{DDTYw}[1][\Delta X_i^1][0] + \cdots + \text{DDTYw}[8][\Delta X_i^8][0]$
$\Delta Y_i^1 \leftarrow \text{DDTY}[1][\Delta X_i^1][0]$
$\cdots$
$\Delta Y_i^8 \leftarrow \text{DDTY}[8][\Delta X_i^8][0]$
**Function** Sbox$(i, t, w_i)$:
$w_i \leftarrow w_i - \text{DDTYw}[t][\Delta X_i^t][0]$
**for** each DDTYw$[t][\Delta X_i^t][r]$ **do**
  $\alpha_i \leftarrow w_i + \text{DDTYw}[t][\Delta X_i^t][r]$
  **if** $[w_1, w_2, \cdots, w_{i-1}, \alpha_i, B_{n-i}] \geq Bc_n$ **then**
    break
  **else**
    $\Delta Y_i^t \leftarrow \text{DDTY}[t][\Delta X_i^t][r]$
    **if** $t < 8$ **then**
      call Sbox$(i, t+1, \alpha_i)$
    **else**
      $w_i \leftarrow \alpha_i$
      call Round-$(i+1)$
    **end if**
  **end if**
**end for**

**Procedure** Round-$n$:
$\Delta X_n \leftarrow \Delta X_{n-2} \oplus E \circ P(\Delta Y_{n-1})$
$w_n \leftarrow min_{\Delta Y_n}(\Delta X_n, \Delta Y_n)$
**if** $[w_1, w_2, \cdots, w_n] \leq Bc_n$ **then**
  $Bc_n = [w_1, w_2, \cdots, w_n]$
**end if**
return to the upper procedure

then we can prune the input differences with weights range from WeightTable[j] to WeightTable[0].

In Procedure Round-2. For the $t$-th S-box, we traverse its weights from WeightTable[8] to WeightTable[0]. For each satisfiable WeightTable[j], we traverse the corresponding input differences and output differences.

In Procedure Round-$i$, $3 \leq i \leq n-1$. For the $t$-th S-box, its input difference is $\Delta X_i^t$. We traverse its weights in the order: DDTYw$[t][\Delta X_i^t][0]$, DDTYw$[t][\Delta X_i^t][1]$, $\cdots$ and the corresponding output differences. If DDTYw$[t][\Delta X_i^t]$[j] does not satisfy the inequality:

$$[w_1, w_2, \cdots, w_i, B_{n-i}] \leq Bc_n \tag{8}$$

then we don't need to traverse DDTYw$[t][\Delta X_i^t][m]$ $(m \geq j)$.

### 3.2 Executing Linear Layer Operations in Minimal Cost

In Sect.3.2, **We take Alg.3 and DES as an example**. The implementation of P and E operations is one of the most costly parts of the search process and it is efficient to execute linear layer operations by looking up tables. In [5], Bao et al. proposed a speeding-up method named *Trailing in Minimal Changes Order Strategy*. They constructed one table to implement nonlinear layer and linear layer operations at once.

Inspired by Bao's method, we construct one table to execute the P permutation and the E expansion at once. According to Alg.3, we find that when we traverse the output differences of the $t$-th S-box of the $i$-th round, the only difference between the new output difference value $\Delta Y_{inew}$ and the old output difference value $\Delta Y_{iold}$ is $\Delta Y_i^t$. So we build the linear-layer table according to each single output difference of each S-box. Further more, in order to use the SSE instructions, we split the 32-bit output of $E \circ P(\cdot)$ into eight 4-bit values.

The new table is:

**EPtable[SboxN][OutV][LocN]**

- SboxN represents the index of the S-box. It ranges from 1 to 8. OutV represents the value of the output difference. It ranges from 0 to 15. LocN represents the index of the output after executing P and E operations. It ranges from 0 to 7.
- EPtable$[t][j][r]$ represents that: for the $t$-th S-box whose output difference is $j$, after executing the P permutation and the E expansion, the $r$-th value of the output is EPtable$[t][j][r]$.

With the help of EPtable[SboxN][OutV][LocN] table and SSE instructions, the cost of executing linear layer operations can be reduced in two sides:

1 The cost of executing P and E operations can be reduced from 7 XOR operations to 2 XOR operations
   - In general, we look up the EPtable[SboxN][OutV][LocN] table to determine the value of $E \circ P(\Delta Y_i)$ when we generate a new value of $\Delta Y_i$. We use the SSE instructions to execute the look-up-table operations. To generate the value of $E \circ P(\Delta Y_i)$, we need to perform **7 XOR operations of 128-bit variables**.
   - When we traverse the output difference of the $t$-th S-box of the $i$-th round, the only difference between the new output difference value $\Delta Y_{inew}$ and the old output difference value $\Delta Y_{iold}$ is $\Delta Y_i^t$. So we can calculate $E \circ P(\Delta Y_{inew})$ using $E \circ P(\Delta Y_{iold})$ and EPtable[SboxN][OutV][LocN]. Then we only need to perform **2 XOR operations of 128-bit variables** to generate the value of $E \circ P(\Delta Y_{inew})$.

   *For example* When we traverse the fourth S-box of round-$i$, if the output difference changes from

$$\Delta Y_{iold} := 0x1, 0x3, 0xe, 0x5, 0xf, 0x2, 0x1, 0x3 \tag{9}$$

   to

$$\Delta Y_{inew} := 0x1, 0x3, 0xe, 0x7, 0xf, 0x2, 0x1, 0x3 \tag{10}$$

then we can calculate $E \circ P(\Delta Y_{inew})$ as follows:

$$E \circ P(\Delta Y_{inew}) = E \circ P(\Delta Y_{iold}) \oplus EPtable[4][0x5]^* \oplus EPtable[4][0x7]^* \tag{11}$$

$$EPtable[i][j]^* := *((\_m128i*)EPtable[i][j] \tag{12}$$

2 The cost of calculating $\Delta X_i^t$ $(1 \le t \le 8)$ from $\Delta X_i$ $(3 \le i \le n)$ can be reduced to only one memory copy operation.
In Procedure Round-$i$ $(3 \le i \le n)$, once we get $\Delta X_i \leftarrow \Delta X_{i-2} \oplus E \circ P(\Delta Y_{i-1})$, we need to determine the value of each $\Delta X_i^t$ $(1 \le t \le 8)$.
- In the original look-up-table method, when we get the value of $\Delta X_i$, we need to perform 7 bitwise shifting operations and 8 bitwise AND operations to get each $\Delta X_i^t$ $(1 \le t \le 8)$.
- In our method, we define $\Delta X_i$ as a 128-bit variable and define $\Delta X_i^1, \Delta X_i^2, \cdots, \Delta X_i^8$ as an array. The length of each $\Delta X_i^t$ $(1 \le t \le 8)$ in the array is 16-bit. Then using the SSE instructions, we can get the values of eight $\Delta X_i^t$ $(1 \le t \le 8)$ by only 1 memory copy operation.

The speeding-up methods presented in Sect.3.1 and Sect.3.2 are also applied in the search process of GIFT. The improved search approach is shown in Alg.4[4].

Compared with Alg.3, there are two new tables in Alg.4:

**DDTwY[SboxN][WeightN][OutN]**
**Ptable[SboxN][OutV][LocN]**

- DDTwY$[t][j][r]$ represents the $r$-th output difference of the $t$-th S-box with a weight WeightTable$[j]$.
- We split the output of $P(\cdot)$ into sixteen 8-bit values. LocN represents the index of the output after executing the P permutation. It ranges from 0 to 15.
- Ptable$[t][j][r]$ represents that: for the $t$-th S-box whose output difference is $j$, after executing the P permutation, the $r$-th value of the output is Ptable$[t][j][r]$.

### 3.3 Merging Two 4-bit S-boxes into One 8-bit S-box

In Sect.3.3, **We take GIFT-128 as an example**. There are 32 $4 \times 4$ S-boxes in GIFT-128. To minimize the cost of linear layer operations, **we merge the two consecutive** $4 \times 4$ **S-boxes into one** $8 \times 8$ **S-box**. The original 32 S-boxes of GIFT-128 is as follows:

$$S_1, S_2, S_3, \cdots, S_{31}, S_{32} \tag{13}$$

We mark the original version of GIFT-128 as the **old-version GIFT-128**. There are 16 $8 \times 8$ S-boxes in the **new-version GIFT-128**:

$$SS_1, SS_2, SS_3, \cdots, SS_{15}, SS_{16} \tag{14}$$

The relationship between $S_1, S_2, \cdots, S_{32}$ and $SS_1, SS_2, \cdots, SS_{16}$ is that, if we have:

$$Y^{2 \cdot i-1} = S_{2 \cdot i-1}[X^{2 \cdot i-1}] \tag{15}$$

$$Y^{2 \cdot i} = S_{2 \cdot i}[X^{2 \cdot i}], \tag{16}$$

then we have:

$$Y^{2 \cdot i-1} || Y^{2 \cdot i} = SS_i[X^{2 \cdot i-1} || X^{2 \cdot i}] \tag{17}$$

in which $1 \le i \le 16$.

Using this speeding-up method, we can efficiently speed up the search process of GIFT-128. There are two advantages of applying this speeding-up method:

1 There are 16 8-bit S-boxes in the new-version GIFT-128. It is very convenient to store 16 8-bit values with a 128-bit variable, so we can easily use the SSE instructions just as in Alg.3;
2 We combine the difference traversal of two 4-bit S-boxes into one 8-bit S-box. When we generate a new output difference of $SS_r$, the cost of executing the P permutation can be reduced.

[4] ns represents the number of the S-boxes. For GIFT-128, ns = 16; for GIFT-64, ns = 8.

**Algorithm 4** Our Search Approach for GIFT

**Input:** $n$ ($\geq 3$); $B_1, B_2, \cdots, B_{n-1}$; $Bc_n$; WeightTable[10]; ns : the number of the S-boxes
**Output:** $B_n = Bc_n$; the best differential trails of $n$-round

**Generate Tables** :
DDTwY[SboxN][WeightN][OutN]
DDTY[SboxN][InV][OutN]
DDTYw[SboxN][InV][OutN]
Ptable[SboxN][OutV][LocN]

**Procedure** Round-1:
$w_1 \leftarrow 0, \Delta Y_1 \leftarrow 0, t \leftarrow 1$
**Function** Sbox-1$(t, w_1)$:
**for** $j = 9$ to 0 **do**
  $\alpha \leftarrow w_1 + \text{WeightTable}[j]$
  **if** $[\alpha, B_{n-1}] \geq Bc_n$ **then**
    break
  **else**
    **for** each DDTwY$[t][j][r]$ **do**
      $\Delta Y_1^t \leftarrow \text{DDTwY}[t][j][r]$
      **if** $t <$ ns **then**
        call Sbox-1$(t+1, \alpha)$
      **else**
        $w_1 \leftarrow \alpha$
        call Round-2
      **end if**
    **end for**
  **end if**
**end for**

**Procedure** Round-$i, 2 \leq i \leq n-1$:
$\Delta X_i \leftarrow P(\Delta Y_{i-1}), t \leftarrow 1$
$w_i \leftarrow \text{DDTYw}[1][\Delta X_i^1][0] + \cdots$
$+\text{DDTYw[ns]}[\Delta X_i^{\text{ns}}][0]$
$\Delta Y_i^1 \leftarrow \text{DDTY}[1][\Delta X_i^1][0]$
$\cdots$
$\Delta Y_i^{\text{ns}} \leftarrow \text{DDTY[ns]}[\Delta X_i^{\text{ns}}][0]$
**Function** Sbox$(i, t, w_i)$:
$w_i \leftarrow w_i - \text{DDTYw}[t][\Delta X_i^t][0]$
**for** each DDTYw$[t][\Delta X_i^t][r]$ **do**
  $\alpha_i \leftarrow w_i + \text{DDTYw}[t][\Delta X_i^t][r]$
  **if** $[w_1, \cdots, w_{i-1}, \alpha_i, B_{n-i}] \geq Bc_n$ **then**
    break
  **else**
    $\Delta Y_i^t \leftarrow \text{DDTY}[t][\Delta X_i^t][r]$
    **if** $t <$ ns **then**
      call Sbox$(i, t+1, \alpha_i)$
    **else**
      $w_i \leftarrow \alpha_i$
      call Round-$(i+1)$
    **end if**
  **end if**
**end for**

**Procedure** Round-$n$:
$\Delta X_n \leftarrow P(\Delta Y_{n-1})$
$w_n \leftarrow min_{\Delta Y_n}(\Delta X_n, \Delta Y_n)$
**if** $[w_1, w_2, \cdots, w_n] \leq Bc_n$ **then**
  $Bc_n = [w_1, w_2, \cdots, w_n]$
**end if**
return to the upper procedure

*For example* If the output difference of $SS_r$ changes from $0x12$ to $0x35$, then we only need 2 XOR operations ($0x12 \rightarrow 0x35$) of 128-bit variables to generate $P(\Delta Y)$. While in the search of the old-version GIFT-128, we need 4 XOR operations ($0x1 \rightarrow 0x3$, $0x2 \rightarrow 0x5$ respectively) of 128-bit variables.

**The idea of merging S-boxes is a time-memory trade-off method**. In [13], Daemen et al. introduced the structure of the (AES) super box. They combined four 8-bit S-boxes into one 32-bit S-box. In Sect.3.3, we combine two 4-bit S-boxes into one 8-bit S-box. **Both of the two methods are aimed at reducing the cost of linear layer operations over the search process**.

## 4 Experimental Results on Best Differential Trails

### 4.1 DES and DESL

We have searched the best differential trails and the weights of the best differential trails of DES and DESL with Alg.3. Table 3 summarizes our experimental results. Table 14 in Appendix A shows one of the best 12-round differential trails of DESL.

**Table 3.** The weights of the best differential trails of DES and DESL

| | DES | | | | | DESL | | | |
|---|---|---|---|---|---|---|---|---|---|
| $n$ | $Bc_n$ | $B_n$ | $t0$ $(s)$ | $t1$ $(s)$ | $t2$ $(s)$ | $Bc_n$ | $B_n$ | $t1$ | $t2$ |
| 4 | 10 | 9.61 | 4.83 | 0.23 | 0.04 | 10 | 9.02 | 0.14s | 0.06s |
| 5 | 14 | 13.22 | 57.34 | 2.91 | 0.88 | 15 | 14.05 | 12.44s | 4.80s |
| 6 | 20 | 19.96 | 303.26 | 20.68 | 2.63 | 23 | 22.19 | 0.67h | 569.55s |
| 7 | 24 | 23.61 | 338.31 | 20.61 | 1.96 | 30 | 29.25 | 11.00h | 2.96h |
| 8 | 31 | 30.48 | 436.82 | 45.00 | 6.31 | 38 | 37.59 | 15.35h | 6.66h |
| 9 | 32 | 31.48 | 3.66 | 0.16 | 0.03 | 42 | 41.76 | 812.45s | 293.78s |
| 10 | 39 | 38.35 | 7.80 | 0.90 | 0.30 | 50 | 49.79 | 2.99h | 1.55h |
| 11 | 40 | 39.35 | 3.72 | 0.16 | 0.03 | 53 | 52.71 | 326.29s | 96.70s |
| 12 | 47 | 46.22 | 7.61 | 0.90 | 0.16 | 59 | 58.10 | 148.37s | 72.81s |
| 13 | 48 | 47.22 | 3.75 | 0.15 | 0.04 | 61 | 60.78 | 0.73s | 0.30s |
| 14 | 55 | 54.10 | 7.61 | 0.90 | 0.16 | 69 | 68.78 | 40.49s | 11.15s |
| 15 | 56 | 55.10 | 3.81 | 0.14 | 0.04 | | | | |
| 16 | 62 | 61.97 | 7.55 | 0.88 | 0.12 | | | | |
| 17 | 63 | 62.97 | 3.83 | 0.16 | 0.03 | | | | |
| 18 | 70 | 69.84 | 7.65 | 0.91 | 0.15 | | | | |

[1] $n$ : the round number. $B_n$ : the weight of the best differential trails of $n$-round. $Bc_n$ : the initial value of $B_n$.
[2] $t_0$ : the search time of the original Matsui's method (Alg.1).
[3] $t_1$ : the search time of applying the speeding-up method in Sect.3.1.
[4] $t_2$ : the search time of applying the speeding-up methods in Sect.3.1 and Sect.3.2.

### 4.2 GIFT

We have searched the best differential trails and the weights of the best differential trails of GIFT-64 and GIFT-128 with Alg.4. Table 4 summarizes our experimental results. Table 15 in Appendix A shows one of the best 19-round differential trails of GIFT-128. Table 16 in Appendix A shows one of the best 14-round differential trails of GIFT-64.

## 5 Experimental Results on Best Linear Trails

### 5.1 DES and DESL

We have searched the best linear trails and the weights of the best linear trails of DES and DESL with the variant of Alg.3[5]. Table 5 summarizes our experimental results. Table 14 in Appendix A shows one of the best 16-round linear trails of DESL.

### 5.2 GIFT

We have searched the best linear trails and the weights of the best linear trails of GIFT-64 and GIFT-128 with the variant of Alg.4. Table 6 summarizes our experimental results. Table 15 in Appendix A shows one of the best 10-round linear trails of GIFT-128. Table 16 in Appendix A shows one of the best 13-round linear trails of GIFT-64.

[5] We refer readers to [20] to get the transformation method.

**Table 4.** the weights of the best differential trails of GIFT

| | GIFT-64 | | | GIFT-128 | | | |
|---|---|---|---|---|---|---|---|
| $n$ | $Bc_n$ | $B_n$ | $t2$ $(s)$ | $Bc_n$ | $B_n$ | $t1$ | $t2$ |
| 3 | 7.0 | 7.000 | 0.01 | 7.0 | 7.000 | 0.02s | 0.00s |
| 4 | 11.5 | 11.415 | 0.00 | 11.5 | 11.415 | 0.05s | 0.02s |
| 5 | 17.0 | 17.000 | 0.02 | 17.0 | 17.000 | 0.60s | 0.21s |
| 6 | 22.5 | 22.415 | 0.09 | 22.5 | 22.415 | 1.65s | 0.62s |
| 7 | 28.5 | 28.415 | 0.64 | 28.5 | 28.415 | 9.93s | 3.74s |
| 8 | 39.0 | 38.000 | 66.01 | 39.0 | 39.000 | 7.58h | 2.23h |
| 9 | 43.0 | 42.000 | 26.75 | 45.5 | 45.415 | 1.88h | 0.66h |
| 10 | 49.0 | 48.000 | 71.85 | 49.5 | 49.415 | 0.39h | 0.13h |
| 11 | 53.0 | 52.000 | 23.22 | 54.5 | 54.415 | 0.50h | 0.17h |
| 12 | 59.0 | 58.000 | 31.62 | 60.5 | 60.415 | 1.11h | 0.38h |
| 13 | 63.0 | 62.000 | 5.15 | 68.0 | 67.830 | 6.67h | 2.24h |
| 14 | 69.0 | 68.000 | 6.99 | 79.0 | 79.000 | - | 257.13h |
| 15 | | | | 86.0 | 85.415 | 435.72h | 144.80h |
| 16 | | | | 90.5 | 90.415 | 24.42h | 8.39h |
| 17 | | | | 97.0 | 96.415 | 66.74h | 21.93h |
| 18 | | | | 103.5 | 103.415 | - | 102.48h |
| 19 | | | | 111.0 | 110.830 | - | 363.90h |
| 20 | | | | 121.5 | $\leq$121.415 | - | - |
| 21 | | | | 126.5 | $\leq$126.415 | - | - |

[1] $t_1$ : the search time of applying the speeding-up methods in Sect.3.1 and Sect.3.2.
[2] $t_2$ : the search time of applying the speeding-up methods in Sect.3.1, Sect.3.2 and Sect.3.2.
[3] - : the values that we have not got.

## 6 Experimental Results on Estimating the Differential Clustering Effect of DES and DESL

Let $PD$ denote the plaintext difference and $CD$ denote the ciphertext difference. Let $Bu_n$ denote the upper bound of $B_n$. In all the experiments of Sect.6, we try to find all the differential trails with weights $B_n$ satisfying $B_n \leq Bu_n$.

### 6.1 DES

In [7], Biham et al. proposed a key recovery attack on 16-round DES using a 13-round differential trail with weight 47.22. In Sect.6.1, we estimate the clustering effect of differential trails for 13-round DES based on Alg.3. We focus on $PD$, $CD$ and the weights of the differential trails.

**Experiment 1** The 13-round differential trail used in [7] is as follows[6]:

$$PD = 19600000\ 00000000,\ CD = 19600000\ 00000000 \tag{18}$$

First of all, we try to find different differential trails satisfying equation (12). Table 7 summarizes the experimental results.

We set $Bu_{13} = 65$ , but there is still only one differential trail satisfying equation (18). Therefore, **the clustering effect of differential trail satisfying equation (18) is weak**.

[6] All the plaintext differences and ciphertext differences given below are in hexadecimal form.

**Table 5.** The weights of the best linear trails of DES and DESL

| | DES | | | DESL | | |
|---|---|---|---|---|---|---|
| $n$ | $Bc_n$ | $B_n$ | $t$ $(s)$ | $Bc_n$ | $B_n$ | $t$ |
| 4 | 4 | 3.03 | 0.003 | 5 | 4.02 | 0.01s |
| 5 | 5 | 4.71 | 0.010 | 6 | 5.22 | 0.00s |
| 6 | 8 | 7.03 | 0.043 | 10 | 9.05 | 6.37s |
| 7 | 9 | 8.03 | 0.020 | 13 | 12.57 | 0.39h |
| 8 | 10 | 9.71 | 0.004 | 16 | 15.53 | 6.66h |
| 9 | 13 | 12.07 | 0.083 | 17 | 16.49 | 3.22s |
| 10 | 14 | 13.39 | 0.009 | 20 | 19.32 | 10.30s |
| 11 | 15 | 14.07 | 0.003 | 22 | 21.32 | 10.92s |
| 12 | 16 | 15.75 | 0.002 | 25 | 24.15 | 37.74s |
| 13 | 18 | 17.42 | 0.019 | 25 | 24.74 | 0.01s |
| 14 | 20 | 19.75 | 0.030 | 28 | 27.57 | 0.29s |
| 15 | 21 | 20.75 | 0.006 | 30 | 29.57 | 0.54s |
| 16 | 23 | 22.42 | 0.008 | 33 | 32.98 | 72.55s |
| 17 | 25 | 24.78 | 0.045 | | | |
| 18 | 27 | 26.10 | 0.011 | | | |
| 19 | 27 | 26.78 | 0.001 | | | |
| 20 | 29 | 28.46 | 0.003 | | | |
| 21 | 31 | 30.14 | 0.037 | | | |
| 22 | 33 | 32.46 | 0.038 | | | |

[1] $n$ : the round number. $B_n$ : the weight of the best linear trails of $n$-round. $Bc_n$ : the initial value of $B_n$.
[2] $t$ : the search time of applying the speeding-up methods in Sect.3.1 and Sect.3.2.

**Table 6.** The weights of the best linear trails of GIFT

| | GIFT-64 | | | GIFT-128 | | |
|---|---|---|---|---|---|---|
| $n$ | $Bc_n$ | $B_n$ | $t1$ | $Bc_n$ | $B_n$ | $t1$ |
| 3 | 3.00 | 3.00 | 0.00s | 3.00 | 3.00 | 0.00s |
| 4 | 5.00 | 5.00 | 0.00s | 5.00 | 5.00 | 0.01s |
| 5 | 7.00 | 7.00 | 0.00s | 7.00 | 7.00 | 0.05s |
| 6 | 10.00 | 10.00 | 0.01s | 10.00 | 10.00 | 1.77s |
| 7 | 13.00 | 13.00 | 0.11s | 13.00 | 13.00 | 19.38s |
| 8 | 16.00 | 16.00 | 1.48s | 17.00 | 17.00 | 0.50h |
| 9 | 20.00 | 20.00 | 41.47s | 22.00 | 22.00 | 44.09h |
| 10 | 25.00 | 25.00 | 0.85h | 26.00 | 26.00 | 20.9d |
| 11 | 29.00 | 29.00 | 8.83h | | | |
| 12 | 31.00 | 31.00 | 0.73h | | | |
| 13 | 34.00 | 34.00 | 3.15h | | | |
| 14 | 37.00 | 37.00 | 5.12h | | | |
| 15 | 40.00 | 40.00 | 4.59h | | | |

[1] $t1$ : the search time of applying the speeding-up methods in Sect.3.1, Sect.3.2 and Sect.3.2.

**Table 7.** Results of Experiment 1

| parameters | | | results | |
|---|---|---|---|---|
| $Bu_{13}$ | $PD$ | $CD$ | number of trails | search time |
| 65 | 19600000 00000000 | 19600000 00000000 | 1 | 2.81h |

**Experiment 2** We try to find other 13-round differential trails with small weight and strong clustering effect, and we hope to find 13-round differential hulls with weight smaller than 47.22.

We increase the value of $Bu_{13}$ and count the number of differential trails with weight satisfying $B_{13} \leq Bu_{13}$. Table 8 summarizes the experimental results.

**Table 8.** Results of Experiment 2

| parameters | results | | |
|---|---|---|---|
| $Bu_{13}$ | number of trails | number of trails with the same plaintext and ciphertext difference | search time |
| 51 | 55 | 0 | 2.17m |
| 52 | 203 | 0 | 11.69m |
| 53 | 531 | 0 | 81.17m |

As we can see from Table 8, setting $Bu_{13} = 53$, we get 531 differential trails, but there is still no two trails sharing the same $PD$ and $CD$.

**Experiment 3** In Experiment 2, setting $Bu_{13} = 51$, we get 55 differential trails. Among the 55 trails, there are four trails with weights smaller than 50. The $PD$ and $CD$ of these four trails are shown in Table 9.

**Table 9.** Four differential trails with the smallest weights of 13-round DES

| number | $PD$ | $CD$ | weight |
|---|---|---|---|
| 1 | 19600000 00000000 | 19600000 00000000 | 47.22 |
| 2 | 1$b$600000 00000000 | 1$b$600000 00000000 | 47.22 |
| 3 | 00196000 00000000 | 00196000 00000000 | 48.00 |
| 4 | 000003$d$4 00000000 | 000003$d$4 00000000 | 48.56 |

**Table 10.** Results of Experiment 3

| parameters | | | results | |
|---|---|---|---|---|
| $Bu_{13}$ | $PD$ | $CD$ | number of trails | search time |
| 56 | 1$b$600000 00000000 | 1$b$600000 00000000 | 1 | 8.53s |
| 56 | 00196000 00000000 | 00196000 00000000 | 1 | 10.03s |
| 56 | 000003$d$4 00000000 | 000003$d$4 00000000 | 1 | 7.28s |

Similar to Experiment 1, we explore the clustering effect of the differential trail No.2 to No.4. Table 10 summarizes the experimental results. Setting $Bu_{13} = 56$ , we still cannot find new differential trails sharing the same $PD$ and $CD$ with differential trail No.2, No.3 or No.4 in Table 9.

The experimental results of Experiment 1 to 3 show that **the clustering effect of differential trails for 13-round DES is weak**.

### 6.2 DESL

As we can see from Table 3, the weight of 12-round best differential trails of DESL is 58.10, which is larger than the key size 56. In order to construct 16-round key recovery attack on DESL, we estimate the differential clustering effect of 13-round DESL.

**Experiment 4** We try to find 13-round differential trails with small propagation weight and strong clustering effect. Hopefully, we want to find 13-round differential hulls with weight smaller than 56.

We increase the value of $Bu_{13}$ and count the number of differential trails with weight satisfying $B_{13} \leq Bu_{13}$. Table 11 summarizes the experimental results.

**Table 11.** Results of Experiment 4

| parameters | results | | |
|---|---|---|---|
| $Bu_{13}$ | number of trails | number of trails with the same plaintext and ciphertext difference | search time |
| 62 | 52 | 0 | 16.78s |
| 63 | 214 | 0 | 73.40s |
| 65 | 1137 | 0 | 1277.17s |

Through Experiment 4, we cannot find two trails having the same $PD$ and $CD$. We get two trails with weight 60.78 and 16 trails with weight 61.37. 61.37 is the second smallest weight of 13-round DESL differential trails. The $PD$ and $CD$ of these trails are shown in Table 12. Considering the symmetry of encryption and decryption process of DESL, there are actually one trail with weight 60.78 and eight trails with weight 61.37.

**Table 12.** Nine differential trails with the smallest weights of 13-round DESL

| $B_{13}$ | number | $PD$ | $CD$ |
|---|---|---|---|
| 60.78 | 1 | 027$c$0400 00000040 | 027$a$0400 00000040 |
| 61.37 | 2 | 027$a$0401 00000040 | 027$c$0400 00000040 |
| 61.37 | 3 | 027$c$0400 00000040 | 026$a$0400 00000040 |
| 61.37 | 4 | 027$c$0400 00000040 | 007$a$0400 00000040 |
| 61.37 | 5 | 027$c$0400 00000040 | 007$a$0401 00000040 |
| 61.37 | 6 | 027$a$0400 00000040 | 026$c$0400 00000040 |
| 61.37 | 7 | 027$a$0400 00000040 | 007$c$0401 00000040 |
| 61.37 | 8 | 027$a$0400 00000040 | 027$c$0401 00000040 |
| 61.37 | 9 | 007$c$0400 00000040 | 027$a$0400 00000040 |

**Experiment 5** We explore the clustering effect of differential trails in Table 12. Table 13 summarizes the experimental results.

We set $Bu_{13} = 70$, and set the value of $PD$ and $CD$ same as the No.1 trail in Table 12:

$$PD = 027c0400\ 00000040,\ CD = 027a0400\ 00000040 \tag{19}$$

We find three trails satisfying equation (19). The weight of these three trails are: 65.17, 67.56 and 60.78.

As we can see from Table 13, we get one differential hull with weight 60.70 and eight differential hulls with weight 61.29. These differential hulls are useless for 16-round key recovery attack of DESL.

The experimental results of Experiment 4 and 5 show that **the clustering effect of differential trails for 13-round DESL is weak**.

## 7 Conclusion and Future Work

In this paper, we improve Matsui's search algorithm of searching the best differential and linear trails by applying three new speeding-up methods. The key idea of our speeding-up methods is to

**Table 13.** Results of Experiment 5

| parameters | | results | | |
|---|---|---|---|---|
| $Bu_{13}$ | number in Table 12 | number of trails | sum of weights | search time |
| 70 | 1 | 3 | 60.70 | 1141.01s |
| 70 | 2 | 3 | 61.29 | 333.46s |
| 70 | 3 | 3 | 61.29 | 1142.66s |
| 70 | 4 | 3 | 61.29 | 1144.72s |
| 70 | 5 | 3 | 61.29 | 1139.02s |
| 70 | 6 | 3 | 61.29 | 601.44s |
| 70 | 7 | 3 | 61.29 | 594.18s |
| 70 | 8 | 3 | 61.29 | 602.26s |
| 70 | 9 | 3 | 61.29 | 624.57s |

[1] Set the value of $PD$ and $CD$ according to Table 12.

prune unsatisfiable candidates as soon as we can and to decrease the cost of linear layer operations. With the help of the improved algorithm, we find the best differential and linear trails of Feistel block cipher DESL and SPN block cipher GIFT-128 and GIFT-64. We also estimate that the clustering effect of differential trails for 13-round DES and DESL are both weak.

In the end, we would like to propose some problems deserving further investigation.

- As shown in Table 4, our improved search algorithm is not efficient enough to find the best differential trails for GIFT-128 when $n \geq 19$. It is because that when the value of $Bc_n - B_{n-1}$ is large, there would be too many candidates. We hope to find some ways to prune unsatisfiable candidates more efficiently.
- Since the speeding-up methods proposed in this paper can help to prune unsatisfiable candidates quickly and decrease the cost of linear layer operations, they can also be used to speed up the search of related-key differential distinguishers.
  In [24,25,10,9,18], some related-key differential distinguishers of GIFT, PRESENT, DESL and LBlock have been found utilizing MILP-based or SMT-based methods. In the following work, we are going to adjust Alg.3 and Alg.4 to adapt for the related-key scenario. We will research whether we can find better related-key differential distinguishers for GIFT, PRESENT, DESL, LBlock and other lightweight block ciphers.

## References


1. https://csrc.nist.gov/Projects/Lightweight-Cryptography/Round-1-Candidates. Last accessed 10 May 2019: NIST Homepage
2. Abdelraheem, M.A., Alizadeh, J., AlKhzaimi, H.A., Aref, M.R., Bagheri, N., Gauravaram, P.: Improved linear cryptanalysis of reduced-round SIMON-32 and SIMON-48. In: Progress in Cryptology - INDOCRYPT 2015 - 16th International Conference on Cryptology in India, Bangalore, India, 6-9 December. pp. 153–179. Springer, Heidelberg (2015)
3. Aoki, K., Kobayashi, K., Moriai, S.: Best differential characteristic search of FEAL. In: Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, 20-22 January. pp. 41–53. Springer, Heidelberg (1997)
4. 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, R.O.C., 25-28 September. pp. 321–345. Springer, Heidelberg (2017)
5. Bao, Z., Zhang, W., Lin, D.: Speeding up the search algorithm for the best differential and best linear trails. In: Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, 13-15 December. pp. 259–285. Springer, Heidelberg (2014)
6. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: CRYPTO 1990, Santa Barbara, California, USA, 11-15 August. pp. 2–21. Springer, Heidelberg (1990)

7. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
8. Blondeau, C., Gérard, B.: Multiple differential cryptanalysis: Theory and practice. In: Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, 13-16 February. pp. 35–54. Springer, Heidelberg (2011)
9. Cao, M., Zhang, W.: Related-key differential cryptanalysis of the reduced-round block cipher GIFT. IEEE Access **7**, 175769–175778 (2019)
10. Chen, L., Wang, G., Zhang, G.: Milp-based related-key rectangle attack and its application to GIFT, khudra, MIBS. Comput. J. **62**(12), 1805–1821 (2019)
11. Chen, S., Liu, R., Cui, T., Wang, M.: Automatic search method for multiple differentials and its application on MANTIS. SCIENCE CHINA Information Sciences **62**(3), 32111:1–32111:15 (2019)
12. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer, Heidelberg (2002)
13. Daemen, J., Rijmen, V.: Understanding two-round differentials in AES. In: Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, 6-8 September. pp. 78–94. Springer, Heidelberg (2006)
14. Hall-Andersen, M., Vejre, P.S.: Generating graphs packed with paths estimation of linear approximations and differentials. IACR Trans. Symmetric Cryptol. **2018**(3), 265–289 (2018)
15. Knudsen, L.R., Robshaw, M.: The Block Cipher Companion. Information Security and Cryptography, Springer, Heidelberg (2011)
16. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New lightweight DES variants. In: Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, 26-28 March. pp. 196–210. Springer, Heidelberg (2007)
17. Li, L., Wu, W., Zheng, Y., Zhang, L.: The relationship between the construction and solution of the MILP models and applications. IACR Cryptology ePrint Archive **2019**, 49 (2019)
18. Liu, Y., Sasaki, Y.: Related-key boomerang attacks on GIFT with automated trail search including BCT effect. In: Information Security and Privacy - 24th Australasian Conference, ACISP 2019, Christchurch, New Zealand, 3-5 July. pp. 555–572. Springer, Heidelberg (2019)
19. Matsui, M.: Linear cryptanalysis method for DES cipher. In: EUROCRYPT 1993, Lofthus, Norway, 23-27 May. pp. 386–397. Springer, Heidelberg (1993)
20. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. In: EUROCRYPT 1994, Perugia, Italy, 9-12 May. pp. 366–375. Springer, Heidelberg (1994)
21. 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, 30 November - 3 December. pp. 57–76. Springer, Heidelberg (2011)
22. Ohta, K., Moriai, S., Aoki, K.: Improving the search algorithm for the best linear expression. In: CRYPTO 1995, Santa Barbara, California, USA, 27-31 August. pp. 157–170. Springer, Heidelberg (1995)
23. Sun, S., Hu, L., Qiao, K., Ma, X., Shan, J., Song, L.: Improvement on the method for automatic differential analysis and its application to two lightweight block ciphers DESL and LBlock-s. In: Advances in Information and Computer Security - 10th International Workshop on Security, IWSEC 2015, Nara, Japan, 26-28 August. pp. 97–111. Springer, Heidelberg (2015)
24. Sun, S., Hu, L., Song, L., Xie, Y., Wang, P.: Automatic security evaluation of block ciphers with S-bP structures against related-key differential attacks. In: Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, 27-30 November. pp. 39–51. Springer, Heidelberg (2013)
25. 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: ASIACRYPT 2014, Kaoshiung, Taiwan, R.O.C., 7-11 December. pp. 158–178. Springer, Heidelberg (2014)
26. Wu, S., Wang, M.: Security evaluation against differential cryptanalysis for block cipher structures. IACR Cryptology ePrint Archive **2011**, 551 (2011)
27. Zhou, C., Zhang, W., Ding, T., Xiang, Z.: Improving the MILP-based security evaluation algorithms against differential cryptanalysis using divide-and-conquer approach. IACR Cryptology ePrint Archive **2019**, 19 (2019)
28. 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, 4-8 March. pp. 372–390. Springer, Heidelberg (2019)

# A Examples of the Best Trails

**Table 14.** A best 12-round differential trail with propagation probability $2^{-58.10}$ and a best 16-round linear trail with linear correlation $2^{-32.98}$ of DESL

| 12-round differential trail | | | | 16-round linear trail | | | |
|---|---|---|---|---|---|---|---|
| $r$ | $\Delta X_r$ | $\Delta Y_r$ | $w_r$ | $r$ | $\Gamma X_r$ | $\Gamma Y_r$ | $w_r$ |
| 1 | 000000000200 | 00000070 | 2.42 | 1 | 440000000011 | $f000000f$ | 2.83 |
| 2 | 000358000000 | 00080000 | 4.68 | 2 | 000000004000 | 00000$c$00 | 2.00 |
| 3 | 000000000000 | 00000000 | 0.00 | 3 | 880000000012 | $d000000f$ | 3.42 |
| 4 | 000358000000 | 00080000 | 4.68 | 4 | 000000000000 | 00000000 | 0.00 |
| 5 | 000000000200 | 00000060 | 3.00 | 5 | 880000000012 | $d000000f$ | 3.42 |
| 6 | 0003$f$8008000 | 00$a$40500 | 9.61 | 6 | 000000004000 | 00000$c$00 | 2.00 |
| 7 | 0000001$a$2$a$58 | 00004002 | 11.83 | 7 | 480000000011 | $f000000f$ | 2.83 |
| 8 | 0003$f$4008000 | 00$a$40500 | 9.42 | 8 | 000000000000 | 00000000 | 0.00 |
| 9 | 000000000200 | 00000060 | 3.00 | 9 | 480000000011 | $f000000f$ | 2.83 |
| 10 | 000354000000 | 00080000 | 4.87 | 10 | 000000004000 | 00000$c$00 | 2.00 |
| 11 | 000000000000 | 00000000 | 0.00 | 11 | 880000000012 | $d000000f$ | 3.42 |
| 12 | 000354000000 | 00480000 | 4.61 | 12 | 000000000000 | 00000000 | 0.00 |
| | | | | 13 | 880000000012 | $d000000f$ | 3.42 |
| | | | | 14 | 000000004000 | 00000$c$00 | 2.00 |
| | | | | 15 | 480000000011 | $f000000f$ | 2.83 |
| | | | | 16 | 000000000000 | 00000000 | 0.00 |

**Table 15.** A best 19-round differential trail with propagation probability $2^{-110.830}$ and a best 10-round linear trail with linear correlation $2^{-26.00}$ of GIFT-128

| 19-round differential trail | | | 10-round linear trail | | |
|---|---|---|---|---|---|
| $r$ | $\Delta X_r$ | $w_r$ | r | $\Gamma X_r$ | $w_r$ |
| 1 | 0$a$000000600$c$00000000000000000000 | 6.000 | 1 | 00000000000001600000000000000160 | 4.00 |
| 2 | 00000000106000000000000000000000 | 5.000 | 2 | 0000000000000000000$c$000$c$00000000 | 2.00 |
| 3 | 00000000000000000000000000$a$00000 | 2.000 | 3 | 00000000000000000000000000001100 | 3.00 |
| 4 | 00000010000000000000000000000000 | 3.000 | 4 | 0000000000000000000000000000000$c$ | 1.00 |
| 5 | 00000000080000000000000000000000 | 2.000 | 5 | 00000000000000000000000200000000 | 1.00 |
| 6 | 00140000000$a$0000000000000000000 | 5.000 | 6 | 00000000000000000000020000000100 | 3.00 |
| 7 | 00000000000000004040000020200000 | 8.000 | 7 | 00000000000000000000080800000000 | 2.00 |
| 8 | 00005050000000000005050000000000 | 12.000 | 8 | 00000000000005000000000000000500 | 4.00 |
| 9 | 00000000000000000000000$a$000$a$00 | 4.000 | 9 | 00000000000000000000000000040004 | 2.00 |
| 10 | 00000000000001100000000000000000 | 6.000 | 10 | 00000000000004400000002200000000 | 4.00 |
| 11 | 00080000000$c$0000000000000000000 | 4.000 | | | |
| 12 | 00000000000000002020000010000000 | 8.000 | | | |
| 13 | 000050400000$a$0200000000000000000 | 9.000 | | | |
| 14 | 05010000000000000505000000000000 | 12.000 | | | |
| 15 | $a$000$a$00000000000000000000000000 | 4.000 | | | |
| 16 | 00000000000000001100000000000000 | 6.000 | | | |
| 17 | 00006000000000000000000000000$c$000 | 4.000 | | | |
| 18 | 00000000020000020000000000000000 | 4.000 | | | |
| 19 | 00200000000400000002000000400000 | 6.830 | | | |

**Table 16.** A best 14-round differential trail with propagation probability $2^{-68.00}$ and a best 13-round linear trail with linear correlation $2^{-34.00}$ of GIFT-64

| 14-round differential trail | | | 13-round linear trail | | |
|---|---|---|---|---|---|
| $r$ | $\Delta X_r$ | $w_r$ | $r$ | $\Gamma X_r$ | $w_r$ |
| 1 | 0000600000006000 | 4.00 | 1 | 0$c$0$c$000000000000 | 2.00 |
| 2 | 0000500000005000 | 6.00 | 2 | 0000100000001000 | 2.00 |
| 3 | 0000020200000000 | 4.00 | 3 | 0000000000000808 | 2.00 |
| 4 | 0000050000000500 | 6.00 | 4 | 0000000500000005 | 2.00 |
| 5 | 0202000000000000 | 4.00 | 5 | 0808000002020000 | 4.00 |
| 6 | 0000500000005000 | 6.00 | 6 | 0000505000005050 | 4.00 |
| 7 | 0000020200000000 | 4.00 | 7 | 00000$a$0$a$00000$a$0$a$ | 6.00 |
| 8 | 0000050000000500 | 6.00 | 8 | 000000000$a$0$a$0000 | 3.00 |
| 9 | 0202000000000000 | 4.00 | 9 | 0000000000$a$00000 | 1.00 |
| 10 | 0000500000005000 | 6.00 | 10 | 0000000000000020 | 2.00 |
| 11 | 0000020200000000 | 4.00 | 11 | 0000000800000000 | 1.00 |
| 12 | 0000050000000500 | 6.00 | 12 | 0000040000000100 | 2.00 |
| 13 | 0202000000000000 | 4.00 | 13 | 0200000000080400 | 3.00 |
| 14 | 0000500000005000 | 4.00 | | | |