IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2022, No. 2, pp. 201–252. DOI:10.46586/tosc.v2022.i2.201-252

Accelerating the Best Trail Search on AES-Like Ciphers

Seonggyeom Kim[1], Deukjo Hong[2], Jaechul Sung[3] and Seokhie Hong[1](✉)

[1] Korea University, Seoul, South Korea
{jeffgyeom,shhong}@korea.ac.kr
[2] Jeonbuk National University, Jeonju, South Korea
deukjo.hong@jbnu.ac.kr
[3] University of Seoul, Seoul, South Korea
jcsung@uos.ac.kr

Abstract. In this study, we accelerate Matsui's search algorithm to search for the best differential and linear trails of AES-like ciphers. Our acceleration points are twofold. The first exploits the structure and branch number of an AES-like round function to apply strict pruning conditions to Matsui's search algorithm. The second employs permutation characteristics in trail search to reduce the inputs that need to be analyzed. We demonstrate the optimization of the search algorithm by obtaining the best differential and linear trails of existing block ciphers: `AES`, `LED`, `MIDORI-64`, `CRAFT`, `SKINNY`, `PRESENT`, and `GIFT`. In particular, our search program finds the full-round best differential and linear trails of `GIFT-64` (in approx. 1 s and 10 s) and `GIFT-128` (in approx. 89 h and 452 h), respectively.
For a more in-depth application, we leverage the acceleration to investigate the optimal DC/LC resistance that `GIFT`-variants, called BOGI-based ciphers, can achieve. To this end, we identify all the BOGI-based ciphers and reduce them into 41,472 representatives. Deriving 16-, 32-, 64-, and 128-bit BOGI-based ciphers from the representatives, we obtain their best trails until 15, 15, 13, and 11 rounds, respectively. The investigation shows that 12 rounds are the minimum threshold for a 64-bit BOGI-based cipher to prevent efficient trails for DC/LC, whereas `GIFT-64` requires 14 rounds. Moreover, it is shown that `GIFT` can provide better resistance by only replacing the existing bit permutation. Specifically, the bit permutation variants of `GIFT-64` and `GIFT-128` require fewer rounds, one and two, respectively, to prevent efficient differential and linear trails.

Keywords: Substitution-Permutation Network (SPN) · Matsui's Search Algorithm · Best Differential Trail · Best Linear Trail · Bad Output must go to Good Input (BOGI)

1 Introduction

Nowadays, differential cryptanalysis (DC) [BS91] and linear cryptanalysis (LC) [Mat94] are two of the most fundamental attacks employed during the design stage of new ciphers. In particular, DC/LC resistance is regarded as the main factor to determine the number of rounds for an iterative block cipher, while the other known attacks are applied later and cause more minor modifications of new ciphers.

DC and LC each start by constructing a corresponding distinguisher with high probability because the probability mainly determines their attack complexity and success rate. Differential and linear distinguishers are derived from differential and linear trails (i.e., characteristics), respectively. Thus, block cipher designers try to demonstrate the (non-)existence of high-probability trails to show their ciphers' DC/LC resistance. Although

Received: 2022-03-01 Accepted: 2022-05-01 Published: 2022-06-10

various studies have indicated that single trails are not enough to represent distinguishers owing to strong differential and linear clustering effects [AK19], the probability of a single trail or its upper bound are still the main concern of providing the lower bounds for the complexity of DC/LC and estimating the required number of rounds during the design phase.

There are two main approaches for evaluating the upper bound for the probability of trails. The first involves enumerating the least number of active S-boxes over all trails, and the second is to traverse all the concrete trails and obtain *best trails*, which have the maximum probability. Generally, the former approach exploits the property of S-boxes, the wide trail strategy, or the mixed integer linear programming (MILP)-aided search and provides the useful bounds in the proposals of various block ciphers, such as PRESENT [BKL+07], AES, and SKINNY [BJK+16]. However, it should be noted that such an approach only provides the upper bound for the probability rather than concrete trails and thus may not guarantee the tight upper bound. The tightness significantly affects some cases. An example can be found in GIFT [BPP+17]. The least number of active S-boxes over 22-round GIFT-128 differential trails amounts to 54 and evaluates the probability upper bound[1] as $2^{-75.6}$, whereas the best differential trail has the probability of $2^{-132.4}$ [SWW21]. This considerable gap may lead to designers using more than a compact number of rounds, resulting in an unnecessary drop in performance of the cipher. Therefore, finding the best trail is still considered one of the main concerns not only when conducting attacks but also when designing block ciphers.

At EUROCRYPT'94 [Mat95], Matsui proposed a dedicated search algorithm for the best differential and linear trails. Subsequently, Matsui's search algorithm was improved in [OMA95, AKM97]. The main idea of these studies was to introduce a pre-computation phase in which search patterns for each round are discarded if proper conditions are not satisfied. This approach was further optimized in [BZL15] by beginning from an efficient search point in each pattern and grouping search patterns. As another direction for improving Matsui's search algorithm, tailoring the pruning conditions for bit permutation-based substitution permutation network was proposed in [BBF15]. The study improved on Matsui's original pruning condition, and further suggested an additional pruning condition to prune search trees more strictly.

A large number of lightweight block ciphers adopt bit permutation owing to its negligible hardware implementation cost. Among these block ciphers, GIFT outperforms the others with its state-of-the-art design approach. Therefore, GIFT is widely used as the underlying primitive of various candidates on the ongoing NIST Lightweight Cryptography standardization process [CDJ+20, BCI+20, CDJN19, CDJ+19, BBP+19]. Moreover, one of the candidates, GIFT-COFB [BCI+20], was chosen as a finalist [BCD+]. The main novelty of GIFT is its design logic: "Bad Output must go to Good Input" (BOGI). This logic prevents differential and linear trails consisting of only one active S-box in each round, even though the round function is composed of a bit permutation and an S-box with differential and linear branch numbers of two. This simple but effective idea enhanced the design strategy of PRESENT [BKL+07], allowing GIFT to provide better DC/LC resistance in fewer rounds.

1.1 Our Contributions

In this study, we accelerate Matsui's search algorithm by tailoring it for AES-like ciphers and utilize this acceleration to investigate the best trails of GIFT-variants. Specifically, our contributions are threefold, as described below.

[1]This upper bound is solely derived from the least number of active S-boxes and the maximum differential probability ($2^{-1.4}$) over the S-box of GIFT.

Table 1: Summary of Best Trail Searches on the Considered Ciphers

Cipher	Best Trail Type	Range of Analysis Rounds	Total Elapsed Time
`GIFT-64`	Differential	**2 ∼ 28 (full-round)**	**0.390 s***
	Linear	**2 ∼ 28 (full-round)**	**9.755 s***
`GIFT-128`	Differential	**2 ∼ 40 (full-round)**	**89.0 h***
	Linear	**2 ∼ 40 (full-round)**	**451.3 h***
`PRESENT`	Differential	2 ∼ 31 (full-round)	5.131 s
`AES`	Differential	2 ∼ 3	17.452 s
	Linear	2 ∼ 3	21.009 s
`LED`	Differential	2 ∼ 3	0.008 s
	Linear	2 ∼ 3	0.013 s
`MIDORI-64`	Differential	2 ∼ 12	210.5 h
	Linear	**2 ∼ 16 (full-round)**	**74.2 h**
`CRAFT`	Differential	2 ∼ 8	456.9 h
	Linear	2 ∼ 7	3.2 h
`SKINNY-64`	Differential	2 ∼ 7	27.6 h
	Linear	2 ∼ 7	256.1 h
`SKINNY-128`	Differential	2 ∼ 6	24.998 s
	Linear	2 ∼ 6	0.5 h

*The total elapsed times for `GIFT` were measured with a dedicated implementation of `GIFT`. They are detailed in Suppl. E, whereas the other results can be found in Suppl. D.

Strengthening the Pruning Conditions of Matsui's Search Algorithm. We improve Matsui's search algorithm [Mat95] by devising strict pruning conditions. Our proposed pruning conditions are based on [BBF15]. However, we extend the existing conditions to be available for *non-bit permutation-based AES-like ciphers*, and strengthen them for *bit permutation-based AES-like ciphers*. The main idea is to leverage the fact that the mixing layer of AES-like round function can be decomposed into a number of matrix multiplications that operate independently. Moreover, the branch number of the matrix multiplication allows the pruning conditions to become stricter. To evaluate the impact of strengthened pruning conditions on trail search, we set up two implementations that were identical except for the application of pruning conditions, and compared their elapsed times. According to the experimental results[2], our pruning conditions can reduce the time by up to factors of 463 and 22 for non-bit permutation-based and bit-permutation-based AES-like ciphers, respectively.

Employing Permutation Characteristics in Trail Search. We propose an approach for removing duplicate candidates of first round input differences/masks by utilizing (word-wise) permutation characteristics. Permutation characteristics were initially suggested for invariant subspace attack [LMR15], but we adjust the notion of permutation characteristic into trail search. To this end, we show that two trails derived from each other through a permutation characteristic have the same probability. In addition, we propose a method to find permutation characteristics of AES-like ciphers. It should be noted that our adoption of permutation characteristics can consist of independent interests in search for other types of distinguishers. For example, the consideration of permutation characteristics can reduce the initial input division properties considered for division trail search [XZBL16].

By integrating the strengthened pruning conditions and the duplicate removal, our final optimized search program required only 0.39 s to provide the full-round best differential trail of `GIFT-64` without any prerequisites. Compared with the most recent result for `GIFT-64` [SWW21], which required 2,210.6 s, our improved approach performs approximately 5,668 times faster. Including bit permutation-based AES-like ciphers – `GIFT` and `PRESENT`, we

[2]All the experiments were conducted on a system with Intel® Xeon® Gold 6230 CPU @ 2.10 GHz, and we used one core for each case.

Table 2: Summary of Minimum Required Rounds for BOGI-Based Ciphers

Block Size (b-bit)	Minimum Required Rounds to Prevent Efficient Trails for DC/LC		
	GIFT-b*	With Replacement of Bit Permutation	With Replacement of Bit Perm. and S-box
16-bit	6 rounds	6 rounds	5 rounds
32-bit	10 rounds	10 rounds	8 rounds
64-bit	14 rounds	**13 rounds**	**12 rounds**
128-bit	22 rounds	**20 rounds**	**19 rounds****

*GIFT-16 and GIFT-32 consist of GIFT's S-box and the reduced version of GIFT's bit permutation.
**For the 128-bit versions, alternatives may exist, which require fewer rounds.

also investigated the best trails of non-bit permutation-based AES-like ciphers – AES, LED[3], MIDORI-64, CRAFT, and SKINNY. Table 1 summarizes the trail search results.

Investigating the Most DC/LC Resistant BOGI-Based Cipher. We take advantage of the above acceleration to examine the most DC/LC resistant combinations of S-box and bit permutation for BOGI-based construction, that is, GIFT-variants, called *BOGI-based ciphers*. To achieve this, we revisit the BOGI design and identify the entire combinations of S-box and mixing layer that can compose a BOGI-based round function. We refer to the analysis results of *BOGI-applicable* S-boxes in [KHSH20], and then consider mixing layers that follow BOGI logic for each BOGI-applicable S-box. We consider not only the existing mixing layer of GIFT but also its variants derived from 4×4 Latin squares. These initially provide us approximately $2^{37.09}$ combinations. Considering the relations among inner components, we deduce equivalent combinations in terms of DC/LC resistance, and reduce the total combinations into 41,472 representatives.

We derive 16-, 32-, 64-, and 128-bit BOGI-based ciphers from the representatives, and search for their best trails. According to the results, 16-, 32-, and 64-bit BOGI-based ciphers require at least 5, 8, and 12 rounds, respectively, to prevent efficient trails for DC/LC. For the 128-bit versions, we could not obtain the minimum required rounds because of increased analysis time. Alternatively, we analyze the best trails up to 22 rounds by concentrating on promising combinations with 11-round optimal DC/LC resistance. According to this analysis, 19 rounds are sufficient for the promising variants of GIFT-128 to prevent efficient trails, which is 3 rounds fewer than GIFT-128.

Moreover, we investigate whether better resistance can be obtained by only replacing the existing mixing layer of GIFT. The analysis shows that each best choice of mixing layer is distinct, depending on block size, and allows a savings of 1 and 3 rounds for GIFT-64 and GIFT-128, respectively. Because such modifications do not present additional implementation costs compared with GIFT, they are expected to outperform the existing ones. Table 2 summarizes the analysis results for BOGI-based ciphers.

1.2 Organizations

The remainder of this paper is organized as follows. Section 2 introduces the relevant contents and notations used in this paper, and outlines Matsui's search algorithm. In Section 3, we demonstrate how the pruning conditions are strengthened by tailoring them to the structure of an AES-like round function. The methods of employing permutation characteristics in trail search and finding them are described in Section 4. In Section 5, we discuss the BOGI design and investigate the optimal DC/LC resistance of BOGI-based ciphers. We conclude the paper in Section 6.

[3]Although LED applies the key-addition every four rounds, we assume that it is applied in each round.

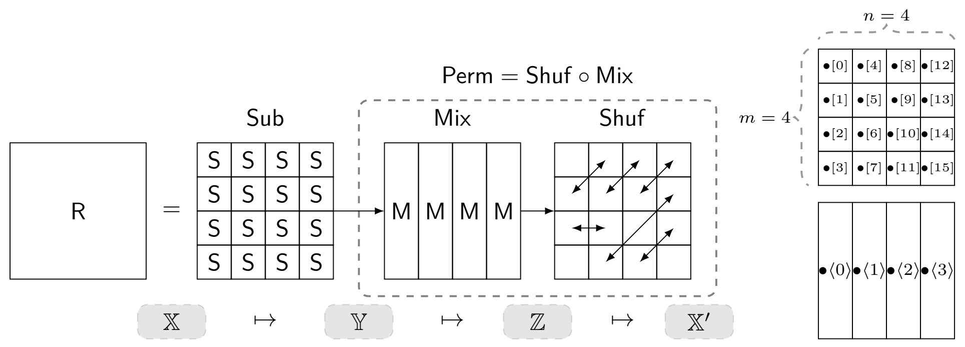

Figure 1: Example of an AES-like round function and two partitions of inputs

2 Preliminaries

2.1 AES-Like Ciphers

In this study, we consider AES-like ciphers, assuming that round keys within each round are chosen to be uniformly random and mutually independent. An *AES-like cipher* is a key-alternating block cipher that applies an AES-like round function R in all rounds. Thus, an AES-like cipher consisting of R rounds is defined as

$$\mathsf{E}^{(R)} = \oplus_{k_R} \circ \mathsf{R} \circ \cdots \circ \oplus_{k_2} \circ \mathsf{R} \circ \oplus_{k_1} \circ \mathsf{R} \circ \oplus_{k_0},$$

where the round key additions $\oplus_{k_i}(x) = x \oplus k_i$. The round function R is decomposed into three functions, denoted by Sub, Mix, and Shuf. While Shuf is a word-wise permutation (transposition) determined by a permutation σ over the index space, Sub and Mix are the parallel applications of an S-box S and a matrix multiplication M, respectively, on smaller inputs. To specify the smaller inputs, we use two partitions of inputs (states) $\bullet$ as the square brackets $\bullet[i]$ and angles $\bullet\langle k\rangle$ for S and M, respectively. Figure 1 presents an example of an AES-like round function and the two partitions. We refer to a transposition over the partition $\bullet\langle k\rangle$ as an M-wise permutation (transposition).

Definition 1. An *AES-like round function* is $\mathsf{R} : \mathbf{F}_2^{wmn} \to \mathbf{F}_2^{wmn}$ which is parameterized by the state dimension $m \times n$ and word size w, and is the composition of *S-layer* Sub and *P-layer* Perm = Shuf ∘ Mix; R = Shuf ∘ Mix ∘ Sub. For $\mathsf{Sub}(\mathbb{X}) = \mathbb{Y}$, an *S-box* $\mathsf{S} : \mathbf{F}_2^w \to \mathbf{F}_2^w$ is concurrently applied to all mn words $\mathbb{X}[j]$ as

$$(\mathbb{Y}[0], ..., \mathbb{Y}[mn-1]) = (\mathsf{S}(\mathbb{X}[0]), ..., \mathsf{S}(\mathbb{X}[mn-1])).$$

Analogously, the *mixing layer* Mix is the parallel application of a *matrix multiplication* $\mathsf{M} : \mathbf{F}_2^{wm} \to \mathbf{F}_2^{wm}$. For $\mathsf{Mix}(\mathbb{Y}) = \mathbb{Z}$, we have

$$(\mathbb{Z}\langle 0\rangle, ..., \mathbb{Z}\langle n-1\rangle) = (\mathsf{M}(\mathbb{Y}\langle 0\rangle), ..., \mathsf{M}(\mathbb{Y}\langle n-1\rangle)),$$

where M is a linear function satisfying

$$\mathsf{M}(\mathbb{Y}\langle k\rangle)^\top = M \cdot \mathbb{Y}\langle k\rangle^\top,$$

for a binary matrix $M \in \mathrm{GL}(wm, \mathbf{F}_2)$. The *shuffle layer* Shuf shuffles the words using the corresponding permutation σ of mn elements. For $\mathsf{Shuf}(\mathbb{Z}) = \mathbb{X}'$, we have

$$(\mathbb{X}'[\sigma(0)], ..., \mathbb{X}'[\sigma(mn-1)]) = (\mathbb{Z}[0], ..., \mathbb{Z}[mn-1]).$$

Moreover, R is called a *bit permutation-based AES-like round function* if M is a bit permutation (transposition), and a *non-bit permutation-based AES-like cipher* otherwise.

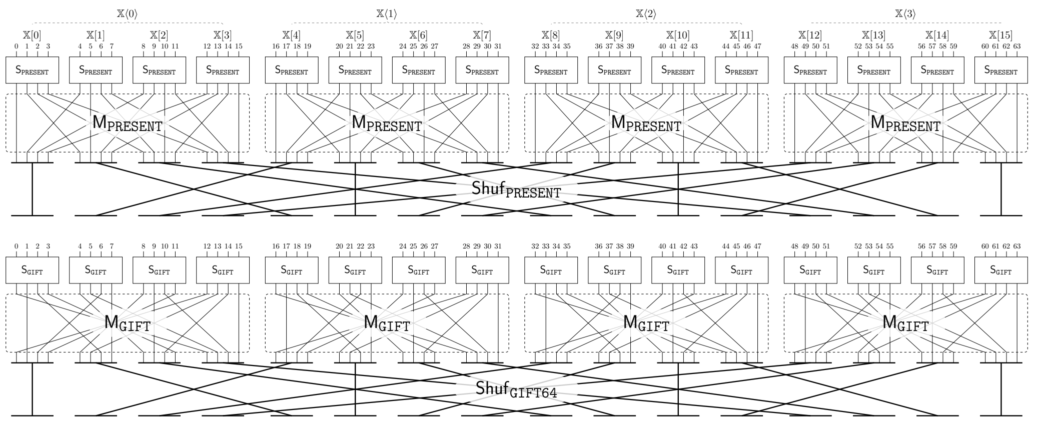
$\mathbb{X}\langle 0\rangle$
$\mathbb{X}\langle 1\rangle$
$\mathbb{X}\langle 2\rangle$
$\mathbb{X}\langle 3\rangle$
$\mathsf{S}_{\texttt{PRESENT}}$
$\mathsf{M}_{\texttt{PRESENT}}$
$\mathsf{Shuf}_{\texttt{PRESENT}}$
$\mathsf{S}_{\texttt{GIFT}}$
$\mathsf{M}_{\texttt{GIFT}}$
$\mathsf{Shuf}_{\texttt{GIFT64}}$

Figure 2: `PRESENT` and `GIFT-64`'s round functions

Examples of non-bit permutation-based AES-like ciphers[4] include `MIDORI`, `SKINNY`, `LED` [GPPR11], and `CRAFT` [BLMR19], whereas `PRESENT` and `GIFT` are bit permutation-based AES-like ciphers.

Example 1. `PRESENT` has a bit permutation-based AES-like round function with $w = 4$ and $m \times n = 4 \times 4$. $\mathsf{Perm}_{\texttt{PRESENT}}$ is a 64-bit permutation and can be decomposed into $\mathsf{Mix}_{\texttt{PRESENT}}$ and $\mathsf{Shuf}_{\texttt{PRESENT}}$, as shown in Figure 2. $\mathsf{Mix}_{\texttt{PRESENT}}$ consists of four 16-bit permutations $\mathsf{M}_{\texttt{PRESENT}}$. $\sigma_{\texttt{PRESENT}}(j)$ for $\mathsf{Shuf}_{\texttt{PRESENT}}$ is defined as $\sigma_{\texttt{PRESENT}}(j) = 4 \times (j \mod 4) + \lfloor \frac{j}{4} \rfloor$.

`GIFT-64` also has a bit permutation-based AES-like round function. $\mathsf{Perm}_{\texttt{GIFT64}}$ consists of $\mathsf{Shuf}_{\texttt{GIFT64}} = \mathsf{Shuf}_{\texttt{PRESENT}}$ and $\mathsf{M}_{\texttt{GIFT}}$, which is determined by the BOGI design of `GIFT`. The detailed discussion on the design is provided in Section 5.

`GIFT-128` adopts the same $\mathsf{M}_{\texttt{GIFT}}$ for $\mathsf{Mix}_{\texttt{GIFT128}}$, while $\mathsf{Shuf}_{\texttt{GIFT128}}$ is derived from $\sigma_{\texttt{GIFT128}}(j) = 8 \times (j \mod 4) + \lfloor \frac{j}{4} \rfloor$.

When considering linear trails, we may use the notation $M^{-\top} = (M^{-1})^\top$ and the corresponding functions; $\mathsf{M}^{-\top}$ and $\mathsf{Perm}^{-\top} = \mathsf{Shuf} \circ \mathsf{M}^{-\top}$. When both types of trail are considered, M^*, M^*, and Perm^* may be used[5].

2.2 Resistance against Differential and Linear Cryptanalysis

The probabilities of differential and linear trails quantify DC/LC resistance. The probability of a trail is derived from difference and linear propagations over rounds of the trail. Therefore, to discuss DC/LC resistance of AES-like block ciphers, we begin with the definition of difference and linear propagations.

Definition 2. Let $f : \mathbf{F}_2^n \to \mathbf{F}_2^n$ be a function. A *difference propagation* $\Delta \xrightarrow{f} \Delta'$ has a *differential probability* $\Pr[\Delta \xrightarrow{f} \Delta']$, which is defined as

$$\Pr[\Delta \xrightarrow{f} \Delta'] = |\{x : f(x) \oplus f(x \oplus \Delta) = \Delta'\}| / 2^n.$$

Analogously, a *linear propagation* $\Gamma \xrightarrow{f} \Gamma'$ has a *linear correlation* $\mathrm{Cr}[\Gamma \xrightarrow{f} \Gamma']$, which is defined as

$$\mathrm{Cr}[\Gamma \xrightarrow{f} \Gamma'] = \left|\left\{x : \Gamma^\top \cdot x = \Gamma'^\top \cdot f(x)\right\}\right| / 2^{n-1} - 1.$$

[4]Note that one can easily exchange the orders of Shuf and Mix with the modified round-keys because Shuf is commutative with Sub.

[5]Since a bit permutation M satisfies $\mathsf{M} = \mathsf{M}^{-\top}$, we simply use the corresponding notations without the superscript $*$ if M is a bit permutation.

Moreover, we define a *linear probability* (i.e., *linear potential*) as the square of the linear correlation, $\mathrm{Cr}^2[\Gamma \xrightarrow{f} \Gamma']$.

The *weight of a difference propagation* $\Delta \xrightarrow{f} \Delta'$ is the negative of the binary logarithm of $\Pr[\Delta \xrightarrow{f} \Delta']$, i.e., $\mathrm{W}(\Delta \xrightarrow{f} \Delta') = -\log_2 \Pr[\Delta \xrightarrow{f} \Delta']$. Analogously, the *weight of a linear propagation* $\Gamma \xrightarrow{f} \Gamma'$ is defined as $\mathrm{W}(\Gamma \xrightarrow{f} \Gamma') = -\log_2 \mathrm{Cr}^2[\Gamma \xrightarrow{f} \Gamma']$.

We also define *the minimum/maximum differential and linear weights* of f as

$$\begin{aligned}
\underline{\Delta}_f &= \min\{\mathrm{W}(\Delta \xrightarrow{f} \Delta') : \Delta \neq \mathbf{0}\},\\
\underline{\Gamma}_f &= \min\{\mathrm{W}(\Gamma \xrightarrow{f} \Gamma') : \Gamma' \neq \mathbf{0}\},\\
\overline{\Delta}_f &= \max\{\mathrm{W}(\Delta \xrightarrow{f} \Delta') : \Delta \neq \mathbf{0}, \Pr[\Delta \xrightarrow{f} \Delta'] \neq \mathbf{0}\},\\
\overline{\Gamma}_f &= \max\{\mathrm{W}(\Gamma \xrightarrow{f} \Gamma') : \Gamma' \neq \mathbf{0}, \mathrm{Cr}^2[\Gamma \xrightarrow{f} \Gamma'] \neq \mathbf{0}\}
\end{aligned}$$

When f is an S-box and both types of the weights are considered simultaneously, the notations $\underline{W}$ and $\overline{W}$ denote the minimum and maximum weights, respectively.

Notations for Differential and Linear Trails. The following table presents the notations for differential and linear trails over AES-like ciphers.

$\mathbb{T}$	Differential trail or linear trail
$\mathbb{T}[i].\mathbb{X}$	Input difference (or mask) of Sub in the i-th round
$\mathbb{T}[i].\mathbb{Y}$	Output difference (or mask) of Sub in the i-th round; This also denotes the input difference (or mask) of Mix in the i-th round.
$\mathbb{T}[i].\mathbb{Z}$	Output difference (or mask) of Mix in the i-th round; This also denotes the input difference (or mask) of Shuf in the i-th round.
$\mathbb{T}[i].\mathbb{X}'$	Output difference (or mask) of Shuf in the i-th round; It is always satisfied that $\mathbb{T}[i].\mathbb{X}' = \mathbb{T}[i+1].\mathbb{X}$.
$\mathbb{T}[i].\bullet[j]$	Difference (or mask) of the j-th word in $\mathbb{T}[i].\bullet$
$\mathbb{T}[i].\bullet\langle k\rangle$	Difference (or mask) of the k-th M's position in $\mathbb{T}[i].\bullet$
$\mathbb{T}[i].\mathbb{W}$	Weight of the difference (linear) propagation $\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[i].\mathbb{Y}$
$\mathbb{T}[i].\mathbb{W}[j]$	Weight of the difference (linear) propagation $\mathbb{T}[i].\mathbb{X}[j] \xrightarrow{\mathsf{S}} \mathbb{T}[i].\mathbb{Y}[j]$
$\mathbb{T}[i].\mathbb{A}[j]$	Activity pattern of $\mathbb{T}[i].\mathbb{X}$; $\mathbb{T}[i].\mathbb{A}[j] = 0$ if $\mathbb{T}[i].\mathbb{X}[j] = \mathbf{0}$ or otherwise, $\mathbb{T}[i].\mathbb{A}[j] = 1$.
$\mathrm{ACT}(\bullet)$	Number of active words (S-boxes) in the full or partial state $\bullet$, where active word has a non-zero value

Differential Trail and Its Weight. An R-round *differential trail* over an AES-like cipher is a sequence of difference propagations over each round. Because round key additions are considered as identities by trails, the output differences of propagations are always equal to the input differences of the subsequent round propagations; i.e., $\mathbb{T}[i].\mathbb{X}' = \mathbb{T}[i+1].\mathbb{X}$. Assuming the independence of uniformly random round keys, the (expected) *probability of a differential trail*, denoted by $\mathrm{EDP}(\mathbb{T})$, can be derived as

$$\mathrm{EDP}(\mathbb{T}) = \prod_{i=1}^{R} \Pr\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{R}} \mathbb{T}[i].\mathbb{X}'\right].$$

The probabilities of difference propagation over each round can also be decomposed into them over Sub and Perm as

$$\Pr\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{R}} \mathbb{T}[i].\mathbb{X}'\right] = \Pr\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[i].\mathbb{Y}\right] \times \Pr\left[\mathbb{T}[i].\mathbb{Y} \xrightarrow{\mathsf{Perm}} \mathbb{T}[i].\mathbb{X}'\right].$$

Because $\Pr\big[\mathbb{T}[i].\mathbb{Y} \xrightarrow{\mathsf{Perm}} \mathbb{T}[i].\mathbb{X}'\big] = 1$ if $\mathbb{T}[i].\mathbb{X}' = \mathsf{Perm}(\mathbb{T}[i].\mathbb{Y})$ or 0 otherwise, the difference propagation $\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[i].\mathbb{Y}$ determines $\Pr\big[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{R}} \mathbb{T}[i].\mathbb{X}'\big]$ as long as $\mathbb{T}[i].\mathbb{X}' = \mathsf{Perm}(\mathbb{T}[i].\mathbb{Y})$. $\Pr\big[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[i].\mathbb{Y}\big]$ is derived from the probabilities of difference propagations over mn S-boxes as

$$\Pr\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[i].\mathbb{Y}\right] = \prod_{j=0}^{mn-1} \Pr\left[\mathbb{T}[i].\mathbb{X}[j] \xrightarrow{\mathsf{S}} \mathbb{T}[i].\mathbb{Y}[j]\right].$$

As a result, assuming that $\mathbb{T}[i].\mathbb{X}' = \mathsf{Perm}(\mathbb{T}[i].\mathbb{Y})$ for all i, the probability of differential trail can be derived as

$$\mathrm{EDP}(\mathbb{T}) = \prod_{i=1}^{R}\prod_{j=0}^{mn-1} \Pr\left[\mathbb{T}[i].\mathbb{X}[j] \xrightarrow{\mathsf{S}} \mathbb{T}[i].\mathbb{Y}[j]\right].$$

Here, we can define the weight of a differential trail from the weights of difference propagations over each S-box as in Definition 3.

Definition 3. Let $\mathbb{T}$ be an R-round differential trail over an AES-like cipher. The *weight of differential trail* $\mathrm{W}(\mathbb{T})$ is defined as the negative of the binary logarithm of $\mathrm{EDP}(\mathbb{T})$. If $\mathbb{T}$ satisfies $\mathbb{T}[i].\mathbb{X}' = \mathsf{Shuf}(\mathbb{T}[i].\mathbb{Z}) = \mathsf{Perm}(\mathbb{T}[i].\mathbb{Y})$ for $1 \le i \le R$, $\mathrm{W}(\mathbb{T})$ can be derived from the weights of each round with each representing the sum of weights of difference propagations over the corresponding S-boxes; i.e.,

$$\mathrm{W}(\mathbb{T}) = -\log_2 \mathrm{EDP}(\mathbb{T}) = \sum_{i=1}^{R} \mathbb{T}[i].\mathbb{W} = \sum_{i=1}^{R}\sum_{j=0}^{mn-1} \mathbb{T}[i].\mathbb{W}[j].$$

Linear Trail and Its Weight. An R-round *linear trail* is a sequence of linear propagations over each round function. Assuming the independence of uniformly random round keys, the (expected) *probability of a linear trail* can be derived as $\mathrm{ELP}(\mathbb{T}) = \prod_{i=1}^{R} \mathrm{Cr}^2\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{R}} \mathbb{T}[i].\mathbb{X}'\right]$. Therefore, similar to the case of differential trail, the weight of linear trail can be evaluated as in Definition 4.

Definition 4. Let $\mathbb{T}$ be an R-round linear trail over an AES-like cipher. The *weight of linear trail* $\mathrm{W}(\mathbb{T})$ is defined as the negative of the binary logarithm of $\mathrm{ELP}(\mathbb{T})$. If $\mathbb{T}$ satisfies $\mathbb{T}[i].\mathbb{X}' = \mathsf{Shuf}(\mathbb{T}[i].\mathbb{Z}) = \mathsf{Perm}^{-\top}(\mathbb{T}[i].\mathbb{Y})$ for $1 \le i \le R$, $\mathrm{W}(\mathbb{T})$ can be derived from the weights of each round with each representing the sum of weights of linear propagations over the corresponding S-boxes; i.e.,

$$\mathrm{W}(\mathbb{T}) = -\log_2 \mathrm{ELP}(\mathbb{T}) = \sum_{i=1}^{R} \mathbb{T}[i].\mathbb{W} = \sum_{i=1}^{R}\sum_{j=0}^{mn-1} \mathbb{T}[i].\mathbb{W}[j].$$

The necessary conditions for differential and linear trails $\mathbb{T}$ to have a non-zero probability are $\mathbb{T}[i].\mathbb{X}' = \mathsf{Shuf}(\mathbb{T}[i].\mathbb{Z}) = \mathsf{Perm}(\mathbb{T}[i].\mathbb{Y})$ and $\mathbb{T}[i].\mathbb{X}' = \mathsf{Shuf}(\mathbb{T}[i].\mathbb{Z}) = \mathsf{Perm}^{-\top}(\mathbb{T}[i].\mathbb{Y})$, respectively. If a trail's propagations over P-layers satisfy this equality, we refer to the trail as a *non-trivial trail.* Consequently, the non-trivial feature of a trail determines $\mathbb{Z}$ and $\mathbb{X}'$ for a given $\mathbb{Y}$.

Minimum Required Rounds for Resistance against DC/LC. Generally, the complexity of single trail-based DC/LC increases as the weight of the used trail increases. In particular, single trail-based DC/LC is possible only if the considered trail has a smaller weight than the block size ($< wmn$). Although such *efficient trails* for DC/LC can almost always be

Algorithm 1: Matsui's Search Algorithm ($\mathcal{M}$)

```
    Input: $R \geq 2$ and $\mathbb{B}[1, ..., R-1]$
    Output: $\overline{\mathbb{T}}$ and $\mathbb{B}[R] = \mathrm{W}(\overline{\mathbb{T}})$
 1  $B_{step} \leftarrow$ a constant positive weight
 2  $\mathsf{Perm}^* \leftarrow \mathsf{Perm}$ or $\mathsf{Perm}^{-\top}$
 3  $B_{init}, B_{set}, \mathbb{T}, \mathbb{T}_{out}$, Found $\leftarrow False$
 4  EstimateBoundAndStart()
 5  Procedure EstimateBoundAndStart()
 6      $B_{init} \leftarrow \mathbb{B}[R-1] + B_{step}$
 7      while Found is $False$ do
 8          $B_{set} \leftarrow B_{init}$
 9          FirstRound()
10          $B_{init} \leftarrow B_{init} + B_{step}$
11      return $\mathbb{B}[R]$, $\mathbb{T}_{out}$
12  Procedure FirstRound()
13      for each $\mathbb{Y} \in \mathbf{F}_2^{wmn} - \{\mathbf{0}\}$ do
14          $\mathbb{T}[1].\mathbb{X} \leftarrow \arg\min_{\mathbb{X}} \mathrm{W}\left(\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{Y}\right)$
15          $\mathbb{T}[1].\mathbb{Y} \leftarrow \mathbb{Y}$
16          $\mathbb{T}[1].\mathbb{W} \leftarrow \mathrm{W}\left(\mathbb{T}[1].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[1].\mathbb{Y}\right)$
17          if $\mathbb{T}[1].\mathbb{W} + \mathbb{B}[R-1] \leq B_{set}$ then
18              if $R = 2$ then
19                  LastRound()
20              else
21                  MiddleRound(2)
22  Procedure MiddleRound($r$)
23      $\mathbb{T}[r].\mathbb{X} \leftarrow \mathsf{Perm}^*\left(\mathbb{T}[r-1].\mathbb{Y}\right)$
24      for each $\mathbb{Y}$ such that $\Pr\left[\mathbb{T}[r].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{Y}\right] \neq 0$ do
25          $\mathbb{T}[r].\mathbb{Y} \leftarrow \mathbb{Y}$
26          $\mathbb{T}[r].\mathbb{W} \leftarrow \mathrm{W}\left(\mathbb{T}[r].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[r].\mathbb{Y}\right)$
27          if $\sum_{i=1}^{r} \mathbb{T}[i].\mathbb{W} + \mathbb{B}[R-r] \leq B_{set}$ then
28              if $r + 1 = R$ then
29                  LastRound()
30              else
31                  MiddleRound($r + 1$)
32  Procedure LastRound()
33      $\mathbb{T}[R].\mathbb{X} \leftarrow \mathsf{Perm}^*\left(\mathbb{T}[R-1].\mathbb{Y}\right)$
34      $\mathbb{T}[R].\mathbb{Y} \leftarrow \arg\min_{\mathbb{Y}} \mathrm{W}\left(\mathbb{T}[R].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{Y}\right)$
35      $\mathbb{T}[R].\mathbb{W} \leftarrow \mathrm{W}\left(\mathbb{T}[R].\mathbb{X} \xrightarrow{\mathsf{Sub}} \mathbb{T}[R].\mathbb{Y}\right)$
36      if $\sum_{i=1}^{R} \mathbb{T}[i].\mathbb{W} \leq B_{set}$ then
37          Found $\leftarrow True$
38          $B_{set} \leftarrow \sum_{i=1}^{R} \mathbb{T}[i].\mathbb{W}$
39          $\mathbb{B}[R] \leftarrow B_{set}$
40          $\mathbb{T}_{out} \leftarrow \mathbb{T}$
```

prevented by taking enough rounds, unnecessarily many rounds occur the overhead of encryption. Therefore, designers are interested in the minimum required rounds beyond which the minimum differential and linear weights are both larger than or equal to the block size ($\geq wmn$). For example, when two candidates of round functions have the same or similar implementation costs, the minimum required rounds can be considered an additional criterion for choosing the best one.

2.3 Matsui's Search Algorithm

This subsection outlines Matsui's search algorithm, denoted by $\mathcal{M}$. Algorithm 1 describes $\mathcal{M}$ briefly. This algorithm is dedicated to searching for the best differential and linear trails based on depth-first search (DFS) with branch-and-bound technique. We denote an R-round best trail by $\overline{\mathbb{T}}$ and its (best) weight by $\mathbb{B}[R] = \mathrm{W}(\overline{\mathbb{T}})$. For consistency, the 0-round best weight is defined as $\mathbb{B}[0] = 0$.

Requirements for Matsui's Search Algorithm. To provide an R-round best trail $\overline{\mathbb{T}}$ and its (best) weight $\mathbb{B}[R]$, $\mathcal{M}$ requires the $(1 \sim R-1)$-round best weights $\mathbb{B}[1, ..., R-1]$ as inputs. The 1-round best weight $\mathbb{B}[1]$ of an AES-like cipher is intuitively evaluated as the minimum weight of S-box $\underline{W}$, whereas the others $\mathbb{B}[2, ..., R-1]$ cannot be trivially determined. However, they can be obtained by applying $\mathcal{M}$ inductively from 2 to $R-1$ rounds. Therefore, hereafter we can assume that the best weights $\mathbb{B}[1, ..., R-1]$ are given for $\mathcal{M}$, and focus on how $\mathcal{M}$ provides an R-round best trail $\overline{\mathbb{T}}$ from the knowledge of $\mathbb{B}[1, ..., R-1]$. Algorithm 1 divides $\mathcal{M}$ into four procedures; `EstimateBoundAndStart()` and the main search procedures – `FirstRound()`, `MiddleRound()`, and `LastRound()`.

Estimating the Initial Bound for $\mathbb{B}[R]$. As shown below, the main search procedures of $\mathcal{M}$ traverse all non-trivial trails whose weights are smaller than or equal to B_{set}. B_{set} is initialized as B_{init}, and is updated only when a trail is found. This implies that the main search procedures provide nothing if $B_{init} < \mathbb{B}[R]$. Therefore, `EstimateBoundAndStart()`

increases the initial bound B_{init} until $B_{init} \geq \mathbb{B}[R]$. The method for increasing B_{init} can be dedicated for each cipher, such as the approach in [OMA95].

Pruning Unnecessary Search Tree. To verify that $\mathcal{M}$ of Algorithm 1 works correctly, we need to demonstrate that the intermediate bound checks in each main search procedure do not drop the sub-part $\mathbb{T}[1, ..., r]$ of trails, whose weights are smaller than or equal to B_{set}; $\mathrm{W}(\mathbb{T}) \leq B_{set}$. This can be easily shown by the fact that AES-like ciphers consist of the same round functions, and thus, $\mathrm{W}(\mathbb{T}[r+1, ..., R]) \geq \mathbb{B}[R-r]$ for any remaining parts $\mathbb{T}[r+1, ..., R]$. Therefore, the intermediate bound checks allow $\mathcal{M}$ to work well and efficiently by skipping unnecessary sub trails.

However, it remains infeasible when analyzing all the candidates for round outputs; c.f., the line 13 and 24 in Algorithm 1. Matsui proposes to determine $\mathbb{Y}$ S-box by S-box instead of once and to apply the pruning conditions based on Proposition 1.

Proposition 1 ([Mat95])**.** *Let $R \geq 1$ and $\mathbb{T}$ be an R-round non-trivial trail. For any $1 \leq r \leq R$ and $0 \leq c < mn$, if $\mathrm{W}(\mathbb{T}) \leq B_{set}$, it is satisfied that*

$$\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \mathbb{B}[R-r] \leq B_{set}.$$

Algorithm 3 presents this approach using the sub-procedure version of `MiddleRound()`. The other search procedures can be derived in a similar way. Note that in addition to these frequent bound checks, sorting the candidates of $\mathbb{T}[r].\mathbb{Y}[c]$ by their weight is also possible owing to the smaller number of candidates, and thus, an early abortion is facilitated at line 17 of Algorithm 3.

2.4 Optimizing Matsui's Search Algorithm

In this subsection, we introduce the optimizations presented in [BBF15]. The optimizations tighten the left side of inequality in Proposition 1. If the left side becomes bigger, unnecessary trees can be pruned earlier, resulting in a faster search. Another optimization of [BBF15] devises a new pruning condition considering the feature of bit permutation.

Filling the Undetermined. The first optimization uses the information of activity pattern $\mathbb{T}[r].\mathbb{A}$ to make up the undetermined output differences (masks) of S-box. At the search tree on `MiddleSubRound()` in Algorithm 3, although $\mathbb{T}[r].\mathbb{Y}[c+1, ..., mn-1]$ are not determined yet, the lower bound for $\sum_{j=c+1}^{mn-1} \mathbb{T}[r].\mathbb{W}[j]$ can be obtained from the number of non-zero words in $\mathbb{T}[r-1].\mathbb{X}'[c+1, ..., mn-1] = \mathbb{T}[r].\mathbb{X}[c+1, ..., mn-1]$. By adding the lower bound to the left side of inequality in Proposition 1, one can derive a more strict pruning condition as Proposition 2. We denote the condition by PC1.

Proposition 2 ([BBF15], PC1)**.** *Let $R \geq 1$ and $\mathbb{T}$ be an R-round non-trivial trail. For any $1 \leq r \leq R$ and $0 \leq c < mn$, if $\mathrm{W}(\mathbb{T}) \leq B_{set}$ it is satisfied that*

$$\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \mathbb{T}[r].\mathbb{A}[j] \times \underline{W} + \mathbb{B}[R-r] \leq B_{set}.$$

It seems that this strengthened bound check is not applicable to the first round because $\mathbb{T}[1].\mathbb{A}$ cannot be obtained from the previous output state. However, with the next optimization, PC1 can also be available for the first round.

Active S-boxes in the First Round Input. The second optimization determines the activity pattern $\mathbb{T}[1].\mathbb{A}$ of the first round before starting the search. The pre-determined activity pattern allows to obtain the lower bound for $\sum_{j=c+1}^{mn-1} \mathbb{T}[1].\mathbb{W}[j]$ even at the first round. Algorithm 4 presents `FirstRound()` and `FirstSubRound()` with PC1. PC1 is checked at line 7 and 20 of Algorithm 4.

Another advantage of using the pre-determined activity pattern is that the early-abortion of `FirstRound()` can be possible by starting the pre-determined activity patterns in the order of the patterns with less active S-boxes. In Algorithm 4, the sets $\mathcal{A}$ of active S-box positions for $\mathbb{T}[1].\mathbb{A}$ are generated in $\mathcal{A}tab$ and sorted by the number of active S-boxes. This enables to preemptively break the loop of $\mathcal{A}tab$ if the bound check at line 7 of Algorithm 4 is not satisfied.

Exploiting the Feature of Bit Permutation. The third optimization can only be used for bit permutation-based AES-like ciphers. It exploits the feature that partial input bits can determine the corresponding output bits exactly over bit permutation, which means the number of active S-boxes in the subsequent round increases as additional input bits of bit permutation are determined.

Lemma 1. *Let* Perm *be a bit permutation. For any* $0 \le c < mn$*, it is satisfied that*

$$\mathrm{ACT}\left(\mathsf{Perm}\left(\mathbb{Y}[0, ..., c] \parallel \mathbf{0}\right)\right) \le \mathrm{ACT}\left(\mathsf{Perm}\left(\mathbb{Y}\right)\right),$$

where $\mathbf{0}$ *denotes a zero-bit padding.*

Proposition 3 ([BBF15], PC2)**.** *Let* $R \ge 2$ *and* $\mathbb{T}$ *be an* R*-round non-trivial trail over a* ***bit permutation-based AES-like cipher****. For any* $1 \le r < R$ *and* $0 \le c < mn$*, if* $\mathrm{W}(\mathbb{T}) \le B_{set}$ *it is satisfied that*

$$\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \mathbb{T}[r].\mathbb{A}[j] \times \underline{W}$$
$$+ \mathrm{ACT}\big(\mathsf{Perm}(\mathbb{T}[r].\mathbb{Y}[0, ..., c] \parallel \mathbf{0})\big) \times \underline{W} + \mathbb{B}[R-1-r] \le B_{set}.$$

Proof. See Appendix C.1. □

Since Lemma 1 is the main factor for Proposition 3, the pruning condition PC2 of Proposition 3 cannot be applied to "non"-bit permutation-based AES-like ciphers.

3 Strengthening Pruning Condition

In this section, we strengthen the previous pruning conditions – PC1 of Proposition 2 and PC2 of Proposition 3. The main idea is to thoroughly use the structure of AES-like round functions and the properties of S and M during trail search.

3.1 Unbalance of the Minimum Weights of S-box

During the trail search for $r \ge 2$, the r-th input $\mathbb{T}[r].\mathbb{X}$ is determined by the $(r-1)$-th output $\mathbb{T}[r-1].\mathbb{X}'$. Therefore, in Propositions 2 and 3, the term $\sum_{j=c+1}^{mn-1} \mathbb{T}[r].\mathbb{A}[j] \times \underline{W}$ for $r \ge 2$ can be tightened by replacing it with $\sum_{j=c+1}^{mn-1} \min_Y \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y)$. In particular, this replacement is influential if S has unbalanced weights.

Definition 5. If $\min_{\Delta' \neq \mathbf{0}} \mathrm{W}(\Delta \xrightarrow{\mathsf{S}} \Delta')$ for each fixed input difference $\Delta \neq \mathbf{0}$ are not same, S has *unbalanced differential weights*. Analogously, If $\min_{\Gamma' \neq \mathbf{0}} \mathrm{W}(\Gamma \xrightarrow{\mathsf{S}} \Gamma')$ for each fixed input mask $\Gamma \neq \mathbf{0}$ are not same, S has *unbalanced linear weights*.

The following table presents an example with the S-box $\mathsf{S}_{\texttt{GIFT}}$ of `GIFT`.

Δ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\min_{\Delta'\neq\mathbf{0}} \mathrm{W}(\Delta \xrightarrow{\mathsf{S}_{\texttt{GIFT}}} \Delta')$	3	2	3	*1.4*	2	*1.4*	2	2	3	2	3	2	2	2	2
Γ	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
$\min_{\Gamma'\neq\mathbf{0}} \mathrm{W}(\Gamma \xrightarrow{\mathsf{S}_{\texttt{GIFT}}} \Gamma')$	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Only two input differences $\Delta = \mathbf{4}, \mathbf{6}$ can have the minimum differential weight 1.4. In such a case, depending on the input difference state $\mathbb{T}[r].\mathbb{X}$ for $r \geq 2$, $\sum_{j=c+1}^{mn-1} \min_Y \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y)$ is strictly bigger than $\sum_{j=c+1}^{mn-1} \mathbb{T}[r].\mathbb{A}[j] \times \underline{W}$. This implies that PC1 of Proposition 2 and PC2 of Proposition 3 can become more strict by using $\mathbb{T}[r].\mathbb{X}$ instead of the activity pattern $\mathbb{T}[r].\mathbb{A}$ for $r \geq 2$ if S-box has unbalanced weights. Table 3 presents the unbalanced weights of the considered ciphers.

3.2 Exploiting Branch Number and Structure of Mixing Layer

Although Perm^* is not a bit permutation, the partial input $\mathbb{T}[r].\mathbb{Y}[0, ..., c]$ of Perm^* can provide a lower bound for the $(r+1)$-th round's weight. The lower bound is derived from the branch number of M^* and the structure of Mix^*.

Branch Number. The branch numbers of matrix multiplication M in an AES-like round function quantify the mixing power of M. Therefore, most ciphers adopt M, which has the highest possible differential and linear branch numbers. The *differential and linear branch numbers* of M are defined as

$$\mathcal{B} = \min_{\boldsymbol{x}\neq\mathbf{0}}\{\mathrm{ACT}(\boldsymbol{x}) + \mathrm{ACT}(\mathsf{M}\,(\boldsymbol{x}))\} \text{ and } \mathcal{B}^{-\top} = \min_{\boldsymbol{x}\neq\mathbf{0}}\{\mathrm{ACT}(\boldsymbol{x}) + \mathrm{ACT}(\mathsf{M}^{-\top}\,(\boldsymbol{x}))\}.$$

We also use $\mathcal{B}^*$ to indicate both branch numbers. Table 3 shows the branch numbers of the considered ciphers.

The knowledge of branch number can allow the lower bound for number of active S-boxes in the subsequent round as Lemma 2.

Lemma 2. *Let $R \geq 2$ and $\mathbb{T}$ be an R-round non-trivial trail. For any $1 \leq r < R$ and $0 \leq k < n$, it is satisfied that*

$$\phi(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]) \leq \sum_{j=0}^{m-1} \mathbb{T}[r+1].\mathbb{A}[\sigma(mk+j)],$$

where $\phi(x) = \begin{cases} \max(1, \mathcal{B}^* - x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$ *and σ is the shuffle function of* Shuf.

Therefore, we can replace Lemma 1 with Lemma 2 to devise PC2 for non-bit permutation-based AES-like cipher. Moreover, the structure of Mix^* further strengthens PC2.

Table 3: Unbalanced Weights and Branch Numbers of the Considered Ciphers

Cipher	dif. wt.	lin. wt.	$\mathcal{B}$	$\mathcal{B}^{-\top}$	Cipher	dif. wt.	lin. wt.	$\mathcal{B}$	$\mathcal{B}^{-\top}$
`AES`	Bal.	Bal.	5	5	`SKINNY-64`	Unbal.	Bal.	2	2
`LED`	Unbal.	Bal.	5	5	`SKINNY-128`	Unbal.	Unbal.	2	2
`MIDORI-64`	Unbal.	Bal.	4	4	`GIFT`	Unbal.	Bal.	2	2
`CRAFT`	Unbal.	Bal.	2	2	`PRESENT`	Unbal.	Bal.	2	2

Structure of Mixing Layer. We leverage the fact that Mix^* is the parallel application of a matrix multiplication M^*. This structure allows that the output of a round (i.e., the input of the subsequent round $\mathbb{T}[r+1].\mathbb{X}$) can be partially determined by the corresponding partial part of $\mathbb{T}[r].\mathbb{Y}$.

Lemma 3. *Let $R \geq 2$ and $\mathbb{T}$ be an R-round non-trivial trail. For any $1 \leq r < R$ and $0 \leq c < mn$, it is satisfied that*

$$\sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}^*\big(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\big)[j] \xrightarrow{\mathsf{S}} Y\big) \leq \sum_{j=0}^{mn_{\mathsf{M}}-1} \mathbb{T}[r+1].\mathbb{W}[\sigma(j)],$$

where $n_{\mathsf{M}} = \lfloor (c+1)/m \rfloor$ and σ is the shuffle function of Shuf.

Proof. See Appendix C.2. □

3.3 Proposed Pruning Conditions

As mentioned in Subsection 3.1, we can strengthen $\mathsf{PC1}$ of Proposition 2 if the S-box has unbalanced weights.

Proposition 4 ($\mathsf{PC1}$)**.** *Let $R \geq 1$, $0 \leq c < mn$, and $\mathbb{T}$ be an R-round non-trivial trail. If $\mathrm{W}(\mathbb{T}) \leq B_{set}$, it is satisfied that*

$$\sum_{j=0}^{c} \mathbb{T}[1].\mathbb{W}[j] + \mathbb{T}[1].\mathbb{A}[j] \times \underline{W} + \mathbb{B}[R-1] \leq B_{set},$$

and that for $2 \leq r \leq R$,

$$\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \min_{Y} \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y) + \mathbb{B}[R-r] \leq B_{set}.$$

We suggest that one chooses $\mathsf{PC1}$ of Proposition 2 or 4 depending on whether the S-box has unbalanced weights. This is because Proposition 2 can be more efficiently implemented. Algorithm 7 compares the applications of $\mathsf{PC1}$ for $r \geq 2$. While the application of Proposition 2 enumerates the number of active S-boxes by simply obtaining the cardinality $|\mathcal{A}_{c<j<mn}|$, that of Proposition 4 consists of a **For** loop at line 6 of Algorithm 7, which causes a non-negligible overhead whenever $\mathsf{PC1}$ is checked.

For Non-bit Permutation-based AES-like Cipher. Based on Lemma 2 and 3, a new pruning condition is available for non-bit permutation-based AES-like cipher.

Proposition 5 ($\mathsf{PC2}$)**.** *Let $R \geq 2$, $0 \leq c < mn$, and $\mathbb{T}$ be an R-round non-trivial trail over a* ***non-bit permutation-based AES-like cipher****. If $\mathrm{W}(\mathbb{T}) \leq B_{set}$, it is satisfied that*

$$\begin{aligned}&\sum_{j=0}^{c} \mathbb{T}[1].\mathbb{W}[j] + \mathbb{T}[1].\mathbb{A}[j] \times \underline{W} \\ &\quad + \sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}^*\big(\mathbb{T}[1].\mathbb{Y}\langle k\rangle\big)[j] \xrightarrow{\mathsf{S}} Y\big) + \sum_{k=n_{\mathsf{M}}}^{n-1} \phi\Big(\sum_{j=0}^{m-1} \mathbb{T}[1].\mathbb{A}[mk+j]\Big) \times \underline{W} \\ &\quad + \mathbb{B}[R-2] \leq B_{set},\end{aligned}$$

and that for $2 \leq r < R$,

$$\begin{aligned}&\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \min_{Y} \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y)\\ &\quad + \sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}^{*}\left(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\right)[j] \xrightarrow{\mathsf{S}} Y\big) + \sum_{k=n_{\mathsf{M}}}^{n-1} \phi(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]) \times \underline{W}\\ &\quad + \mathbb{B}[R-1-r] \leq B_{set},\end{aligned}$$

where $\phi(x) = \begin{cases} \max(1, \mathcal{B}^{*} - x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$ *and* $n_{\mathsf{M}} = \lfloor (c+1)/m \rfloor$.

Except for the duplicated terms of Proposition 4, it is enough to evaluate the different terms to apply PC2 by storing the duplicated terms. The different terms can be evaluated as Algorithm 9.

For Bit Permutation-based AES-like Cipher. One can notice that Proposition 5 can be also applied to bit permutation-based AES-like ciphers. However, Proposition 5 does not use the entire determined part $\mathbb{T}[r].\mathbb{Y}[0,...,c]$ but only the partial part $\mathbb{T}[r].\mathbb{Y}[0,...,m(n_{\mathsf{M}}-1)-1]$. The remaining part can still be used with Lemma 1. Therefore, we can derive Proposition 6 for bit permutation-based AES-like ciphers from Lemma 1 and Proposition 5.

Proposition 6 (PC2)**.** *Let $R \geq 2$, $0 \leq c < mn$ and $\mathbb{T}$ be an R-round non-trivial trail over a* ***bit permutation based AES-like cipher****. If* $\mathrm{W}(\mathbb{T}) \leq B_{set}$, *it is satisfied that*

$$\begin{aligned}&\sum_{j=0}^{c} \mathbb{T}[1].\mathbb{W}[j] + \mathbb{T}[1].\mathbb{A}[j] \times \underline{W}\\ &+ \sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}\big(\mathbb{T}[1].\mathbb{Y}\langle k\rangle\big)[j] \xrightarrow{\mathsf{S}} Y\big)\\ &+ \mathrm{ACT}\left(\mathsf{M}\big(\mathbb{T}[1].\mathbb{Y}\langle n_{\mathsf{M}}\rangle[0,...,m_{\mathsf{M}}-1] \parallel \mathbf{0}\big)\right) \times \underline{W}\\ &+ \sum_{k=n_{\mathsf{M}}+\phi(m_{\mathsf{M}})}^{n-1} \phi(\sum_{j=0}^{m-1} \mathbb{T}[1].\mathbb{A}[mk+j]) \times \underline{W} + \mathbb{B}[R-2] \leq B_{set},\end{aligned}$$

and that for $2 \leq r < R$,

$$\begin{aligned}&\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \min_{Y} \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y)\\ &+ \sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}\big(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\big)[j] \xrightarrow{\mathsf{S}} Y\big)\\ &+ \mathrm{ACT}\left(\mathsf{M}\big(\mathbb{T}[r].\mathbb{Y}\langle n_{\mathsf{M}}\rangle[0,...,m_{\mathsf{M}}-1] \parallel \mathbf{0}\big)\right) \times \underline{W}\\ &+ \sum_{k=n_{\mathsf{M}}+\phi(m_{\mathsf{M}})}^{n-1} \phi(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]) \times \underline{W} + \mathbb{B}[R-1-r] \leq B_{set},\end{aligned}$$

where $\phi(x) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$, $n_{\mathsf{M}} = \lfloor (c+1)/m \rfloor$, *and* $m_{\mathsf{M}} = c + 1 - mn_{\mathsf{M}}$.

The term $\mathrm{ACT}(\mathsf{M}(\mathbb{T}[r].\mathbb{Y}\langle n_\mathsf{M}\rangle[0,...,m_\mathsf{M}-1] \parallel \mathbf{0}))\times \underline{W}$ is added and replaces $\phi(\sum_{j=0}^{m-1}\mathbb{T}[r].\mathbb{A}[mn_\mathsf{M}+j])\times\underline{W}$ of Proposition 5. Because the replaced term is smaller than or equal to the current term, Proposition 6 provides a more strict pruning condition than Proposition 5.

The following two terms support that Proposition 6 provides a more strict pruning condition than Proposition 3.

1. $\sum_{k=0}^{n_\mathsf{M}-1}\sum_{j=0}^{m-1}\min_Y \mathrm{W}\big(\mathsf{M}^*(\mathbb{T}[r].\mathbb{Y}\langle k\rangle)[j] \xrightarrow{\mathsf{S}} Y\big)$: As in Proposition 4, this term is influential if S has unbalanced weights.

2. $\sum_{k=n_\mathsf{M}+\phi(m_\mathsf{M})}^{n-1}\phi(\sum_{j=0}^{m-1}\mathbb{T}[r].\mathbb{A}[mk+j])\times\underline{W}$: This term is an additional part to increase the left side of inequality.

3.4 Implementation of Search Algorithm

The search algorithm that we implement to compare the pruning conditions of [BBF15] with ours is described in Algorithm 5 and 6 of Appendix A.

3.4.1 Utility Procedures

$\mathcal{A}tab$ and the utility procedures – `EstimateBoundAndStart()`, `BoundCheck()` are presented in Algorithm 5.

$\mathcal{A}tab$. Similar to the existing method of [BBF15], the pre-determined indices of active S-boxes for $\mathbb{T}[1].\mathbb{A}$ are used in our algorithm. However, by introducing permutation characteristics, the size of $\mathcal{A}tab$ can be reduced. This reduction will be discussed in Section 4. When comparing the pruning conditions of [BBF15] and ours in Subsection 3.5, we do not apply this reduction. The experimental results with the reduced $\mathcal{A}tab$ will be separately given in Subsection 4.4.

`EstimateBoundAndStart()`. Instead of the constant step B_{step} in Algorithm 1, the minimum and maximum weights of the S-box $\underline{W},\overline{W}$ are used to update B_{init}. This procedure updates B_{init} as $\mathbb{B}[R-1]+\overline{W}$ for the first trial, and then increases it by $\underline{W}$ for each trial.

`BoundCheck()`. This procedure checks the pruning conditions after adding the lower bound for weight of remaining part (i.e., $\mathbb{B}[R-r]$ at the r-th round search tree). Because searching for a single best trail suffices, trails whose weights are expected to be equal to B_{set} are excluded after a trail is found.

3.4.2 Main Search Procedures

The main search procedures; `First`, `Middle`, and `LastRound()` and their sub-procedures; `First`, `Middle`, and `LastSubRound()` are presented in Algorithm 5, 6 respectively.

Accumulating the Weights. Instead of computing the weights $\sum_{i=1}^{r}\sum_{j=0}^{c}\mathbb{T}[i].\mathbb{W}[j]$ at each search tree, they are gradually evaluated from the previous weight $\mathbb{W}_{cum}$ to reduce duplicate computations. Moreover, as mentioned, it may be observed that the weight of PC2 ($\mathbb{W}_{\mathsf{PC2}}$ in Algorithm 5, 6) can be computed with a part of PC1's weight ($\mathbb{W}_{\mathsf{PC1}}$ in Algorithm 5, 6). Therefore, we reuse the evaluated weight $\mathbb{W}_{\mathsf{PC1}}$ to obtain the weight $\mathbb{W}_{\mathsf{PC2}}$ for PC2.

Table 4: Performance Comparisons of Best Trail Searches on the Considered Ciphers

Cipher	$\frac{\lvert\mathcal{A}tab\rvert}{\lvert Opt\mathcal{A}tab\rvert}$	Round*	Trail	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
PRESENT	1	$2 \sim 31$	Dif.	9.777 s	5.131 s		1.91	1	1.91
GIFT-64	15.77	$2 \sim 28$	Dif.	436.242 s	57.235 s	5.627 s	7.62	10.17	77.52
			Lin.	1.0 h	0.5 h	177.506 s	1.92	10.31	19.79
GIFT-128	1	$2 \sim 19$	Dif.	68.2 h	4.5 h		15.00	1	15.00
			Lin.	354.9 h	62.0 h		5.72	1	5.72
AES	15.77	$2 \sim 2$	Dif.	1.0 h	<0.001 s	<0.001 s	∞	2.00	∞
			Lin.	1.4 h	<0.001 s	<0.001 s	∞	2.00	∞
LED	3.98	$2 \sim 3$	Dif.	7.393 s	0.033 s	0.008 s	221.34	3.98	880.11
			Lin.	24.191 s	0.051 s	0.013 s	474.34	4.02	1904.83
MIDORI-64	15.77	$2 \sim 2$	Dif.	0.535 s	0.004 s	0.001 s	130.51	6.83	891.83
			Lin.	0.084 s	0.002 s	<0.001 s	33.44	6.25	209.00
CRAFT	3.95	$2 \sim 7$	Dif.	235.3 h	4.6 h	1.4 h	50.94	3.30	168.10
			Lin.	171.3 h	3.3 h	3.2 h	51.75	1.05	54.26
SKINNY-64	3.98	$2 \sim 6$	Dif.	291.702 s	11.468 s	2.139 s	25.44	5.36	136.39
			Lin.	0.9 h	164.916 s	35.964 s	19.81	4.59	90.86
SKINNY-128	3.98	$2 \sim 6$	Dif.	446.164 s	52.572 s	24.998 s	8.49	2.10	17.85
			Lin.	7.9 h	1.0 h	0.5 h	8.17	2.15	17.54

*For the comparisons, the range of analysis rounds is determined by the existing method of [BBF15]. The range our method can analyze is summarized in Table 1 and detailed in Suppl. D.
$\mathcal{M}_{prev}$: The existing method of [BBF15].
$\mathcal{M}_{pc}$: Our method with the strengthened pruning conditions.
$\mathcal{M}_{our}$: Our method combining the strengthened pruning conditions and permutation characteristics.
$\frac{\lvert\mathcal{A}tab\rvert}{\lvert Opt\mathcal{A}tab\rvert}$: Reduction in input differences (masks) by employing permutation characteristics.

Searching from the Promising Output First. As the early abortion of Algorithm 3, the same technique is applied to `FirstSubRound()` and `MiddleSubRound()`. This search approach can be efficiently implemented with the pre-computed tables that consist of properly reordered propagations by their weights [JZD20].

Concerning Active S-box Only. When applying PC2, the index c_{nxt} of the subsequent active word can be considered the number of determined words in $\mathbb{T}[r].\mathbb{Y}$ because zero-input words must have zero outputs. In Algorithm 6, the index c_{nxt} is evaluated at lines 7 and 36 and then used for obtaining the weight $\mathbb{W}_{nxt}$ to check PC2 at lines 17 and 44. This consideration gives a larger n_{M}, making Propositions 6 and 5 more influential.

3.5 Performance Comparisons

In this subsection, we justify the improvement of our pruning conditions by comparing the search programs that have distinct procedures for checking PC1 and PC2. Algorithm 7 shows the differences for checking PC1 while Algorithm 8 and 9 show the differences for checking PC2. Note that the others except for these parts are identical as Algorithm 5 and 6.

The performance comparisons are presented in the $\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$ column in Table 4, and the detailed results for each round are given in Suppl. D. Our method outperforms the existing method of [BBF15] on all considered ciphers.

GIFT. In `GIFT-64`, the existing method and ours found the full-round best differential and linear weights within an hour. However, our method was faster by a factor of approximately 7.62.

In `GIFT-128`, there is a significant difference between the execution times. Our method provided the $(20 \sim 24)$-round best differential weights and the $(18 \sim 21)$-round best linear weights, whereas the existing method was unable to do so within the execution time (details are provided in Suppl. D.10). The obtained best weights are in accordance with those of [SWW21].

Remark 1. For simplicity of application, we did not consider the decomposition of Perm^* to implement `NextRoundLowerBound()`, and thus, simply used Perm^* instead of M^* for Algorithm 9. This implies that a dedicated implementation should provide better performances than the experimental results obtained on our method. As shown in Subsection 5.4, our dedicated search program for BOGI-based ciphers considered the decomposition of Perm^* and performed better on `GIFT` than what Table 4 shows. With the dedicated search program, the full-round best trail of `GIFT-128` can be obtained, and to the best of our knowledge, these results are new. The experimental results obtained on the dedicated search program are detailed in Suppl. E.

LED. The state of `LED` [GPPR11] is parameterized by $m = 4, n = 4, w = 4$. Our method provided the 3-round best differential and linear weights in less than 0.1 s, whereas the existing method required several seconds. Consequently, our method finds the $(2 \sim 3)$-round best linear weights 474.34 times faster. The obtained best weights are in accordance with the lower bounds derived from the minimum active S-boxes in [GPPR11].

MIDORI-64. The state of `MIDORI-64` [BBI+15] is parameterized by $m = 4, n = 4, w = 4$. Our method provided the 9-round best differential weight and the 10-round linear weight, whereas the existing method provided only the 3-round best differential and linear weights. Moreover, the required time for obtaining the $(2 \sim 3)$-round best differential weights is reduced by a factor of 130.51. The obtained best weights are in accordance with the lower bounds derived from the minimum active S-boxes in [BBI+15].

4 Employing Permutation Characteristics in Trail Search

In this section, we suggest an approach for reducing the inputs that need to be analyzed in trail search. The main idea is to utilize permutation characteristics, which were originally presented in [LMR15] for invariant subspace attack. It can be noted that our approach can be of independent interest for obtaining other types of distinguishers, such as division trails [XZBL16].

The simplest case of our approach is to employ a word-wise permutation A such that $\mathsf{Perm} \circ \mathsf{A} = \mathsf{A} \circ \mathsf{Perm}$. Since $\mathrm{W}(\Delta \xrightarrow{\mathsf{Sub}} \Delta') = \mathrm{W}(\mathsf{A}(\Delta) \xrightarrow{\mathsf{Sub}} \mathsf{A}(\Delta'))$, it is easy to show that $\mathrm{W}(\Delta \xrightarrow{\mathsf{R}} \Delta') = \mathrm{W}(\mathsf{A}(\Delta) \xrightarrow{\mathsf{R}} \mathsf{A}(\Delta'))$. Let $\mathbb{T}$ and $\mathbb{T}'$ be R-round differential trails satisfying

$$\mathbb{T}[i].\mathbb{X} = \Delta_i, \mathbb{T}'[i].\mathbb{X} = \mathsf{A}(\Delta_i) \text{ for } 1 \leq i \leq R, \text{ and } \mathbb{T}[R].\mathbb{X}' = \Delta_f, \mathbb{T}'[R].\mathbb{X}' = \mathsf{A}(\Delta_f).$$

$\mathrm{W}(\Delta \xrightarrow{\mathsf{R}} \Delta') = \mathrm{W}(\mathsf{A}(\Delta) \xrightarrow{\mathsf{R}} \mathsf{A}(\Delta'))$ yields that $\mathbb{T}[i].\mathbb{W} = \mathbb{T}'[i].\mathbb{W}$ for $1 \leq i \leq R$ and $\mathrm{W}(\mathbb{T}) = \mathrm{W}(\mathbb{T}')$. Therefore, when it comes to searching for a best differential trail, one does not have to consider other trails starting from the input difference $\mathsf{A}(\Delta_1)$ after all trails from an input difference Δ_1 are traversed. Analogously, when searching for a best linear trail, one can employ a word-wise permutation A such that $\mathsf{Perm}^{-\top} \circ \mathsf{A} = \mathsf{A} \circ \mathsf{Perm}^{-\top}$.

In this manner, our approach finds A and reduces $\mathcal{A}tab$ of Algorithm 5 by removing the duplicate $\mathcal{A}$ under A. Moreover, we generalize the commutative case to obtain more A by considering $\mathsf{Perm}^* \circ \mathsf{A} = \mathsf{B} \circ \mathsf{Perm}^*$, denoted by $\mathsf{A} \xRightarrow{\mathsf{Perm}^*} \mathsf{B}$.

Remark 2. One can notice a general linear permutation A such that $\mathsf{R} \circ \mathsf{A} = \mathsf{A} \circ \mathsf{R}$ can also be used for trail search because A yields

$$|\{\mathbb{X} \mid \mathsf{R}(\mathbb{X}) \oplus \mathsf{R}(\mathbb{X} \oplus \Delta) = \Delta'\}| = |\{\mathbb{X} \mid \mathsf{R}(\mathbb{X}) \oplus \mathsf{R}(\mathbb{X} \oplus \mathsf{A}(\Delta)) = \mathsf{A}(\Delta')\}|, \text{ and}$$
$$|\{\mathbb{X} \mid \Gamma \cdot \mathbb{X} = \Gamma' \cdot \mathsf{R}(\mathbb{X})\}| = \left|\{\mathbb{X} \mid \mathsf{A}^{-\top}(\Gamma) \cdot \mathbb{X} = \mathsf{A}^{-\top}(\Gamma') \cdot \mathsf{R}(\mathbb{X})\}\right|.$$

However, we concentrate on word-wise permutations because of their easy adjustments into $\mathcal{A}tab$ and the high complexity of obtaining general permutation characteristics. Therefore, one may further consider the general linear permutations, such as the self-equivalences of S-layer, which are presented in Corollary 1 of [RP21].

4.1 Permutation Characteristics for Trail Search

Unlike the commutative case $\mathsf{A} \xrightarrow{\mathsf{Perm}^*} \mathsf{A}$, the general case $\mathsf{A} \xrightarrow{\mathsf{Perm}^*} \mathsf{B}$ does not simply allow two distinct trails to have the same weight. To derive the same notion as above, we require a permutation characteristic for a given number of rounds.

Definition 6. An R-round *differential permutation characteristic* of an AES-like cipher is a sequence of word-wise permutations $\mathsf{D}^{(1)} \Rightarrow \mathsf{D}^{(2)} \Rightarrow \cdots \Rightarrow \mathsf{D}^{(R+1)}$ such that $\mathsf{D}^{(i)} \xrightarrow{\mathsf{Perm}} \mathsf{D}^{(i+1)}$ for $1 \le i \le R$. Analogously, an R-round *linear permutation characteristic* of an AES-like cipher is a sequence of word-wise permutations $\mathsf{L}^{(1)} \Rightarrow \mathsf{L}^{(2)} \Rightarrow \cdots \Rightarrow \mathsf{L}^{(R+1)}$ such that $\mathsf{L}^{(i)} \xrightarrow{\mathsf{Perm}^{-\top}} \mathsf{L}^{(i+1)}$ for $1 \le i \le R$.

Proposition 7. *Let $\mathbb{T}$ be an R-round non-trivial differential trail over an AES-like cipher. If $\mathsf{D}^{(1)} \Rightarrow \mathsf{D}^{(2)} \Rightarrow \cdots \Rightarrow \mathsf{D}^{(R+1)}$ is an R-round differential permutation characteristic of the AES-like cipher, an R-round differential trail $\mathbb{T}'$ such that*

$$\mathbb{T}'[i].\mathbb{X} = \mathsf{D}^{(i)}\left(\mathbb{T}[i].\mathbb{X}\right), \quad \mathbb{T}'[i].\mathbb{Y} = \mathsf{D}^{(i)}\left(\mathbb{T}[i].\mathbb{Y}\right),$$
$$\mathbb{T}'[i].\mathbb{Z} = \mathsf{Mix}(\mathbb{T}'[i].\mathbb{Y}), \quad \mathbb{T}'[i].\mathbb{X}' = \mathsf{D}^{(i+1)}\left(\mathbb{T}[i].\mathbb{X}'\right),$$

for $1 \le i \le R$ is also non-trivial and has the same weight as $\mathrm{W}(\mathbb{T})$.

Proof. See Appendix C.3. □

Proposition 7 implies that $\mathbb{T}'$ need not be considered after the equivalent trail $\mathbb{T}$ is traversed in the best trail search. This allows the reduction of input differences that need to be considered.

Let $\mathcal{D}$ be the set of the first word-wise permutation $\mathsf{D}^{(1)}$ of each permutation characteristic. To search for a best trail, one can only consider trails beginning from inputs that are not derived from each other via a word-wise permutation in $\mathcal{D}$. The representative of each Δ can be defined as $\min_{\mathsf{D}\in\mathcal{D}} \mathsf{D}(\Delta)$ in lexicographical order. In this regard, the set of representatives can be obtained as $\{\Delta : \Delta = \min_{\mathsf{D}\in\mathcal{D}} \mathsf{D}(\Delta)\}$. We adjust such removal into activity patterns for the first round, and reduce $\mathcal{A}tab$ of Algorithm 5. We denote the reduced table by $Opt\mathcal{A}tab$.

Example 2. $\mathcal{A}tab$ for `AES` is

$$\{\underbrace{\{0\},\{1\},\cdots,\{15\}}_{16\ 1-combinations},\underbrace{\{0,1\},\{0,2\},\cdots,\{14,15\}}_{120\ 2-combinations},\underbrace{\{0,1,2\},\{0,1,3\},\cdots,\{13,14,15\}}_{560\ 3-combinations},\cdots\}$$

and has the cardinality $|\mathcal{A}tab|$ of 65,535. The set $\mathcal{D}$ of $\mathsf{D}^{(1)}$ presented in Subsection 4.3 reduces $\mathcal{A}tab$ as

$$Opt\mathcal{A}tab = \{\{0\}\} \cup \{\underbrace{\{0,1\},\{0,2\},\{0,4\},\{1,4\},\{2,4\},\{3,4\},\{0,8\},\{1,8\},\{2,8\}}_{9\ 2-combinations}\}$$
$$\cup \{\underbrace{\{0,1,2\},\{0,1,4\},\{0,2,4\},\cdots,\{2,7,8\},\{3,7,8\}}_{35\ 3-combinations}\}\cdots,$$

It is expected that the removal allows the best trail search to be faster as the reduction rate $|\mathcal{A}tab|/|Opt\mathcal{A}tab| \approx 15.77$.

Because Proposition 7 can also be applied to linear trails by replacing differential permutation characteristics with linear permutation characteristics, the same knowledge can be used when searching for a best linear trail.

It may be observed that such reductions can be directly applied to search not only for variant differential/linear trails (e.g., impossible differential trail) but also for other types of distinguisher. In particular, the considered initial division properties for division trail search [XZBL16] can be reduced by the same approach.

Iterative Permutation Characteristic. It is better to consider *iterative permutation characteristics* $\mathsf{D}^{(1)} \Rightarrow \mathsf{D}^{(2)} \Rightarrow \cdots \Rightarrow \mathsf{D}^{(R+1)} = \mathsf{D}^{(1)}$ for an arbitrary number of rounds. Moreover, $\mathcal{D}$ can include all the word-wise permutations $\mathsf{D}^{(i)}$ since any $\mathsf{D}^{(i)}$ can be the first word-wise permutation. In this regard, we only considered iterative permutation characteristics when obtaining *Opt$\mathcal{A}$tab* of the considered ciphers.

4.2 Obtaining Permutation Characteristics of AES-like Cipher

In this subsection, we demonstrate how to obtain permutation characteristics of an AES-like cipher. Due to a vast number of word-wise permutations, the exhaustive search of $\mathsf{A} \xRightarrow{\mathsf{Perm}^*} \mathsf{B}$ is computationally infeasible. Therefore, we consider the decomposition $\mathsf{Perm}^* = \mathsf{Shuf} \circ \mathsf{Mix}^*$ and derive $\mathsf{A} \xRightarrow{\mathsf{Perm}^*} \mathsf{B}$ from $\mathsf{A} \xRightarrow{\mathsf{Mix}^*} \mathsf{B}$.

Let $\mathsf{A}_k \xRightarrow{\mathsf{M}^*} \mathsf{B}_k$ for $0 \leq k < n$ and C be a M-wise permutation (transposition). Because Mix^* consists of the same matrix multiplications M^* and Shuf also a word-wise permutation, it is satisfied that

$$(\mathsf{A}_0 \parallel \cdots \parallel \mathsf{A}_{n-1}) \circ \mathsf{C} \xRightarrow{\mathsf{Mix}^*} (\mathsf{B}_0 \parallel \cdots \parallel \mathsf{B}_{n-1}) \circ \mathsf{C}, \text{ and}$$
$$(\mathsf{A}_0 \parallel \cdots \parallel \mathsf{A}_{n-1}) \circ \mathsf{C} \xRightarrow{\mathsf{Perm}^*} \mathsf{Shuf} \circ (\mathsf{B}_0 \parallel \cdots \parallel \mathsf{B}_{n-1}) \circ \mathsf{C} \circ \mathsf{Shuf}^{-1}.$$

We denote the set of such pairs over Perm^* by DWSE(Perm^*).

Now, one can obtain R-round permutation characteristics and $\mathcal{D}$ by regarding the pairs $(\mathsf{A}, \mathsf{B}) \in$ DWSE(Perm^*) as the edges of a directed graph G. Algorithm 2 presents this approach. It may be noted that the connected subgraphs of G are cyclic or linear subgraphs. The cyclic subgraphs at line 4 represent iterative permutation characteristics. Therefore, all the vertices are added into $\mathcal{D}$ at line 7. On the other hand, the linear subgraphs at line 5 represent permutation characteristics that are not iterative. Therefore, some vertices from the head vertex are only added into $\mathcal{D}$ at line 9.

Moreover, in Algorithm 2, DWSE(Perm^*) is reduced into $\overleftrightarrow{\text{DWSE}}(\mathsf{Perm}^*)$ as we are interested in obtaining a permutation characteristic whose length is $R \geq 2$. Such a consideration leads to a better performance when obtaining the subgraphs of G.

4.3 Permutation Characteristics of Considered Ciphers

In this subsection, we present the permutation characteristics of the considered ciphers and demonstrate the reduced table *Opt$\mathcal{A}$tab*. It can be noted that we solely consider the connected cyclic subgraphs in Algorithm 2 so that the number of rounds is not restricted. However, the connected linear subgraphs tend to be too small for long trail search. Table 5 summarizes the results of the analysis, and Figure 3 describes some of the obtained directed graphs G in Algorithm 2. All the considered ciphers have DWSE(Perm) = DWSE($\mathsf{Perm}^{-\top}$). Therefore, the same reduction of $\mathcal{A}tab$ is applied to both differential and linear trail searches.

AES. The matrices $M^*_{\mathtt{AES}}$ of $\mathsf{M}^*_{\mathtt{AES}}$ are equivalent to 4×4 circulant matrices over $\mathbf{F}_{2^8}$. Therefore, it is satisfied that $\text{ROL}_k \xRightarrow{\mathsf{M}^*_{\mathtt{AES}}} \text{ROL}_{4-k}$, where ROL_k denotes a word-wise left-rotation with the amount k. This provides DWSE($\mathsf{Perm}^*_{\mathtt{AES}}$) of 6,144 pairs, and generates

Algorithm 2: R-Round Permutation Characteristics of AES-like Cipher

```
   Input: $\mathsf{Perm}^*$ and Rounds $R \geq 2$
   Output: $\mathcal{D}$
1  $\mathcal{D} \leftarrow \varnothing$
2  $\overleftrightarrow{\mathrm{DWSE}}(\mathsf{Perm}^*) \leftarrow \{(\mathsf{A}, \mathsf{B}) \in \mathrm{DWSE}(\mathsf{Perm}^*) : \exists \mathsf{X}$ such that $(\mathsf{B}, \mathsf{X}) \in \mathrm{DWSE}(\mathsf{Perm}^*)\}$
3  $G \leftarrow$ a directed graph regarding the pairs in $\overleftrightarrow{\mathrm{DWSE}}(\mathsf{Perm}^*)$ as the edges
4  $\mathcal{C} \leftarrow$ the set of connected cyclic subgraphs in $G$
5  $\mathcal{L} \leftarrow$ the set of connected linear subgraphs in $G$
6  for $H \in \mathcal{C}$ do
7      $\mathcal{D} \leftarrow \mathcal{D} \cup \{$all vertices (word-wise permutations) in $H\}$
8  for $H \in \mathcal{L}$ do
       // Note that there is one more removed vertex for $H$.
9      $\mathcal{D} \leftarrow \mathcal{D} \cup \{$from the first head vertex, $\min(0, |H| + 1 - R)$ vertices in $H\}$
10 return $\mathcal{D}$
```

Table 5: Comparisons Between the Sizes of $\mathcal{A}tab$ and $Opt\mathcal{A}tab$

Cipher	$\lvert\mathcal{A}tab\rvert$	$\lvert\mathrm{DWSE}(\mathsf{Perm}^*)\rvert$	$\lvert\overleftrightarrow{\mathrm{DWSE}}(\mathsf{Perm}^*)\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
AES	65,535	6,144	16	16	4,155	15.77
LED	65,535	20	4	4	16,455	3.98
MIDORI-64	65,535	7,962,624	576	16	4,155	15.77
CRAFT	65,535	20	4	4	16,575	3.95
SKINNY	65,535	20	4	4	16,455	3.98
PRESENT	65,535	24	1	1	65,535	1
GIFT-64	65,535	6,144	16	16	4,155	15.77
GIFT-128	$2^{32}-1$	$\approx 2^{31.3}$	512	1	$2^{32}-1$	1

a directed graph with 16 edges and 16 vertices[6]. The directed graph consists of ten cycles – (four 1-cycles and six 2-cycles). This gives $|\mathcal{D}| = 16$ and allows the decrease of $|\mathcal{A}tab|$ from 65,535 to 4,155.

MIDORI-64. According to our exhaustive analysis, $\{(\mathsf{A}, \mathsf{B}) : \mathsf{A} \xRightarrow{\mathsf{M}^*_{\mathtt{MIDORI64}}} \mathsf{B}\} = \{(\mathsf{A}, \mathsf{A}^{-1})\}$, where A is a word-wise permutation on four words. This provides $\mathrm{DWSE}(\mathsf{Perm}^*_{\mathtt{MIDORI64}})$ of 7,962,624 pairs, and generates a directed graph with 576 edges and 1,072 vertices. The directed graph consists of 6 cycles – (two 1-cycles, one 2-cycle, and three 4-cycles) and 496 linear subgraphs. All linear subgraphs have a size of 1. Although the linear subgraphs can be used up to two rounds, we consider only six cycles for an arbitrary number of rounds. This allows the decrease of $|\mathcal{A}tab|$ from 65,535 to 4,155.

GIFT. According to our exhaustive analysis, $\{(\mathsf{A}, \mathsf{B}) : \mathsf{A} \xRightarrow{\mathsf{M}_{\mathtt{GIFT}}} \mathsf{B}\}$ has four pairs. This provides $|\mathrm{DWSE}(\mathsf{Perm}_{\mathtt{GIFT64}})| = 6{,}144$ and $|\mathrm{DWSE}(\mathsf{Perm}_{\mathtt{GIFT128}})| = 10{,}321{,}920 \approx 2^{23.3}$.

$\mathrm{DWSE}(\mathsf{Perm}_{\mathtt{GIFT64}})$ gives a corresponding directed graph consisting of 16 edges and 16 vertices. In the directed graph, there exist 6 cycles – (two 1-cycles, one 2-cycle, and three 4-cycles), which drives $|\mathcal{D}| = 16$. This allows the decrease of $|\mathcal{A}tab|$ from 65,535 to 4,155.

For `GIFT-128`, $|\mathcal{D}| = 1$. This yields no reduction in $|\mathcal{A}tab|$. However, there exist non-trivial linear subgraphs – (498 1-linear subgraphs, 5 2-linear subgraphs, and 1 3-linear subgraphs). As mentioned above, they can be used to obtain the 2-, 3-, and 4-round best weights.

[6] This implies that the remaining 6,128 pairs cannot be used for more than one round.

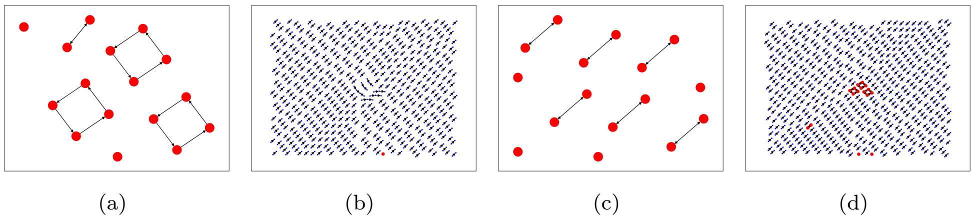

Figure 3: The directed graphs G in Algorithm 2 : (a) `GIFT-64`, (b) `GIFT-128`, (c) `AES`, and (d) `MIDORI-64`.

4.4 Impact of Permutation Characteristics on Trail Search

This subsection discusses the impacts of the reduced table *OptAtab* on trail search. They are evaluated by comparing the elapsed time after applying *OptAtab* to our method with the strengthened pruning conditions. The comparisons are presented in the $\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$ column in Table 4, and the detailed results for each round are given in Suppl. D.

GIFT-64. As shown in Subsection 3.5, our pruning conditions already provide a considerably accelerated search on `GIFT-64`. Moreover, the adoption of *OptAtab* assists the full-round best differential and linear trail searches to end in 5.627 s and 177.506 s, respectively. Our approach presented better performances compared to the method of [BBF15] by factors of 77.52 and 19.79 on differential and linear trail searches, respectively. This outstanding performance enables us to exhaustively investigate the most DC/LC resistant BOGI-based ciphers, which is discussed in Section 5.

MIDORI-64. The $(2 \sim 9)$-round best trail search was improved by a factor of approximately 7 by applying *OptAtab*. Moreover, *OptAtab* allows to provide the full-round best linear weight and 12-round best differential weight in 74.2 h and 210.5 h, respectively.

5 Toward the Most Resistant BOGI-based Cipher

In this section, we take advantage of the above acceleration to exhaustively investigate the optimal DC/LC resistance of BOGI-based ciphers. Our evaluations show that `GIFT` is not optimal in terms of DC/LC resistance. Moreover, even if we consider the implementation cost of round function, some variants provide better resistance than `GIFT`.

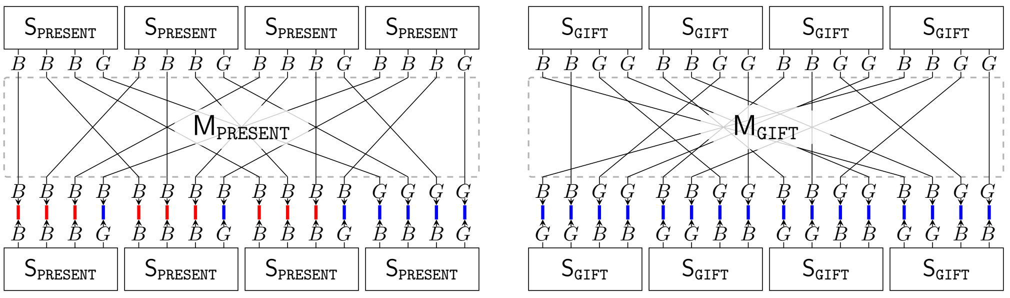

Figure 4: Consecutive single active bit propagations of `PRESENT` and `GIFT`

5.1 BOGI Logic

BOGI logic is a design approach proposed for the fundamental prevention of consecutive single active bit propagations over a trail. Let $e_i \in \mathbf{F}_2^w$ be the vector with 1 in i-th coordinate and 0 elsewhere. The i-th input bit of an S-box S is *bad*, denoted by B, if $\Pr[e_i \xrightarrow{\mathsf{S}} e_j] \neq 0$ or $\mathrm{Cr}[e_i \xrightarrow{\mathsf{S}} e_j] \neq 0$ for some e_j. Otherwise, the i-th input bit is *good*, denoted by G. The bad/good output bits are determined in a similar way. Assume that a bad output bit of S is connected to a bad input bit of S over the bit permutation-based AES-like round function. One can easily derive a propagation that has only four active S-boxes over four successive rounds, and could extend the propagation to obtain a longer trail on which only a single active S-box exists for each round.

The existence of such B-B connections over Perm of bit permutation-based AES-like round function can be checked by the *Super Box* [DR20], which consists of S and M, because Shuf does not affect the connections. Figure 4 presents the Super Boxes of PRESENT and GIFT. As can be observed in Figure 4, there exist B-B connections over $\mathsf{M}_{\mathtt{PRESENT}}$. This raises a number of efficient linear trails and allows multidimensional linear cryptanalysis on PRESENT up to 28 rounds out of 31 rounds [Cho10, FN20]. On the other hand, GIFT tackles this weakness with the well-crafted combination $\{\mathsf{S}_{\mathtt{GIFT}}, \mathsf{M}_{\mathtt{GIFT}}\}$. Therefore, although GIFT-64 adopts the same structure as PRESENT and takes a smaller number of rounds[7], it is believed that GIFT-64 provides better security margin against DC/LC than PRESENT.

BOGI-applicable S-box. For GIFT's prevention to be available, a necessary condition is required on the S-box, as given below.

Lemma 4. *[BPP+17, KHSH20] Let S be a 4-bit S-box. B-B connections can be prevented by a proper 16-bit permutation, if and only if S has at least four good bits.*

We refer to the S-boxes satisfying Lemma 4 as *BOGI-applicable* S-boxes and denote their set by $\mathcal{BS}$. All the BOGI-applicable S-boxes were identified in [KHSH20]. The total number $|\mathcal{BS}|$ of 4-bit BOGI-applicable S-boxes is 186,392,448. Among them, only 2,654,208 S-boxes have the minimum differential uniformity and linearity as 6 and 8, respectively.

5.2 Mixing Layer of BOGI Logic

In this subsection, we demonstrate how to construct the proper 16-bit permutations $\mathsf{M}_{\mathtt{BOGI}}$ of Lemma 4 for a given BOGI-applicable S-box $\mathsf{S}_{\mathtt{BOGI}} \in \mathcal{BS}$. Merely finding a 16-bit permutation that prevents B-B connections with a given S-box is not reasonable due to a number of 16-bit permutations and their low mixing power. Therefore, we construct $\mathsf{M}_{\mathtt{BOGI}}$ considering a proper decomposition.

In our decomposition, $\mathsf{M}_{\mathtt{BOGI}}$ consists of three layers; $\mathsf{M}_{\mathtt{BOGI}} = \mathsf{OB} \circ \mathsf{LS} \circ \mathsf{IB}$ where $\mathsf{IB} = (\mathsf{ib} \parallel \mathsf{ib} \parallel \mathsf{ib} \parallel \mathsf{ib})$ (resp. $\mathsf{OB} = (\mathsf{ob} \parallel \mathsf{ob} \parallel \mathsf{ob} \parallel \mathsf{ob})$) is the application of the four same 4-bit permutations ib (resp. ob), and LS is a 16-bit permutation. Figure 5 describes the decomposition.
Additionally, the 16-bit permutation LS satisfies the following conditions.

1. The four input bits of LS in each S-box position go to four distinct S-box positions.

2. Each bit order of the four input bits of LS in each S-box position is invariant on the four output bits of LS in an S-box position.

We denote $\mathcal{LS}$ as the set of LS. It is easy to demonstrate that $\mathcal{LS}$ can be derived from 4×4 Latin squares; therefore, the size $|\mathcal{LS}|$ is 576.

[7]Since GIFT-64 does not have the whitening key, the cipher is cryptographically considered to have 27 rounds.

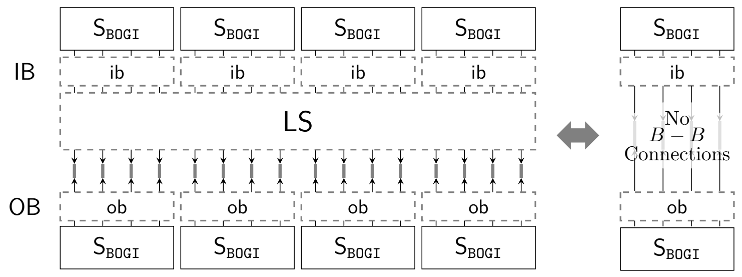

Figure 5: Our Decomposition of $\mathsf{M}_{\mathtt{BOGI}}$

Because of the second condition of LS, the distribution of B and G in each S-box position is invariant over LS. Thus, finding the pair $(\mathsf{ib}, \mathsf{ob})$ that prevents the B-B connections over $\mathsf{M}_{\mathtt{BOGI}}$ is simplified into finding the composition $\mathsf{ob} \circ \mathsf{ib}$ that gives no B-B connections with the given $\mathsf{S}_{\mathtt{BOGI}}$.

We denote by $\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ the set of such pairs $(\mathsf{ib}, \mathsf{ob})$ for $\mathsf{S}_{\mathtt{BOGI}}$. As shown in [KHSH20], all BOGI-applicable S-boxes with differential uniformity of 6 and linearity of 8 have two Bs and two Gs on their input and output bits. Thus, one can easily deduce that $|\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})| = 24 \times 4 = 96$ if $\mathsf{S}_{\mathtt{BOGI}}$ has a differential uniformity of 6 and a linearity of 8.

Example 3. $\mathsf{M}_{\mathtt{GIFT}}$ of `GIFT` can be decomposed into a 16-bit permutation $\mathsf{LS}_{\mathtt{GIFT}}$ and a pair of 4-bit identity permutations (i, i) as described in Figure 5. $\mathsf{LS}_{\mathtt{GIFT}}$ places the l-th input bit (MSB is the 0-th bit) in the $\pi_{\mathtt{GIFT}}(l)$-th output bit, where $\pi_{\mathtt{GIFT}}$ is defined as follows.

l	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi_{\mathtt{GIFT}}(l)$	12	1	6	11	8	13	2	7	4	9	14	3	0	5	10	15

Therefore, $\mathsf{LS}_{\mathtt{GIFT}}$ satisfies the above two conditions. Since $\mathsf{S}_{\mathtt{GIFT}}$ has the input/output bits of $GGBB/BBGG$ (c.f., Figure 4), the composition $\mathsf{i} \circ \mathsf{i}$ can prevent B-B connections. One may notice that 96 pairs of 4-bit permutations are included in $\mathcal{BP}(\mathsf{S}_{\mathtt{GIFT}})$.

Remark 3. Although we focus on the case where IB and OB are the parallel applications of ib and ob, respectively, IB and OB with distinct four 4-bit permutations may also prevent the B-B connections. However, the current decomposition can be expected to fit into bit-slice implementations such as Fixslicing [ANP20].

5.3 BOGI-based Ciphers

In this subsection, we evaluate the number of non-equivalent BOGI-based ciphers in terms of DC/LC resistance. First, we define BOGI-based ciphers.

Definition 7. A $(16{\cdot}n)$-bit *BOGI-based cipher*, denoted by $\mathtt{BOGI\text{-}16}{\cdot}n$, is a bit permutation-based AES-like cipher that is parameterized by the state dimension $(m = 4) \times n$ and the word size $w = 4$. Each component of the AES-like round function is given as follows.

- $\mathsf{Sub}_{\mathtt{BOGI16}\cdot n}$: The parallel application of a BOGI-applicable S-box $\mathsf{S}_{\mathtt{BOGI}} \in \mathcal{BS}$ that has differential uniformity of 6 and linearity of 8.
- $\mathsf{Mix}_{\mathtt{BOGI16}\cdot n}$: The parallel application of a 16-bit permutation $\mathsf{M}_{\mathtt{BOGI}}$ which is derived from $\mathsf{LS} \in \mathcal{LS}$ and $(\mathsf{ib}, \mathsf{ob}) \in \mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ as described in Figure 5.
- $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$: The shuffle layer with $\sigma_{\mathtt{BOGI16}\cdot n}(j) = n \times (j \mod 4) + \left\lfloor \frac{j}{4} \right\rfloor$.

Note that the number of considered $\mathsf{S}_{\mathtt{BOGI}}$ is 2,654,208, $|\mathcal{LS}| = 576$, and $|\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})| = 96$ as we mentioned. Therefore, the number of $(16 \cdot n)$-bit BOGI-based ciphers is about $2^{37.09}$.

Reducing Considered S-boxes. We first reduce the considered S-boxes $\mathsf{S}_{\mathtt{BOGI}} \in \mathcal{BS}$ by introducing DDT-equivalence relation. Because DDT-equivalent S-boxes have exactly the same DDT and extended-LAT (absolute version of LAT) [DH19], we can choose a single representative in each equivalent class in terms of DC/LC resistance. The number of such classes is 10,368 according to [KHSH20].

Reduction in $|\mathcal{LS}|$. For any 4-bit permutation p, if $\mathsf{LS} \in \mathcal{LS}$, then $\mathsf{LS}' = (p^{-1} \parallel p^{-1} \parallel p^{-1} \parallel p^{-1}) \circ \mathsf{LS} \circ (p \parallel p \parallel p \parallel p)$ is also included in $\mathcal{LS}$. This yields $\mathsf{OB} \circ \mathsf{LS} \circ \mathsf{IB} = \mathsf{OB}' \circ \mathsf{LS}' \circ \mathsf{IB}'$ where $(\mathsf{IB}', \mathsf{OB}') = ((p^{-1} \parallel p^{-1} \parallel p^{-1} \parallel p^{-1}) \circ \mathsf{IB}, \mathsf{OB} \circ (p \parallel p \parallel p \parallel p))$ for any 4-bit permutation p. Therefore, when all the pairs in $\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ are concerned, it suffices to consider the 24 representatives LS_i which are not represented by the other as the above. The 24 representatives are presented in Appendix B.

Restricting $\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$. The BOGI-applicability of S-box is invariant up to the permutation equivalence. This gives a reduction in the number of considered combinations $\{\mathsf{S}_{\mathtt{BOGI}}, (\mathsf{ib}, \mathsf{ob})\}$ for a given $\mathsf{LS} \in \mathcal{LS}$.

Lemma 5. *If $(\mathsf{a}, \mathsf{b}) \in \mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ and $\mathsf{S}'_{\mathtt{BOGI}} = \mathsf{a} \circ \mathsf{S}_{\mathtt{BOGI}} \circ \mathsf{b}$, a pair of identities $(\mathsf{i}, \mathsf{i}) \in \mathcal{BP}(\mathsf{S}'_{\mathtt{BOGI}})$. Moreover, for a given $\mathsf{LS} \in \mathcal{LS}$, two $(16 \cdot n)$-bit BOGI-based ciphers derived from $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{a}, \mathsf{b})\}$ and $\{\mathsf{S}'_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ have the same best differential and linear weights for any rounds.*

Proof. See Appendix C.4 □

According to Lemma 5, it is sufficient to analyze the combinations of $\{\mathsf{S}_{\mathtt{BOGI}}, (\mathsf{i}, \mathsf{i})\}$ for a given $\mathsf{LS} \in \mathcal{LS}$. Among the 10,368 representatives of $\mathsf{S}_{\mathtt{BOGI}}$, 1,728 representatives have $(\mathsf{i}, \mathsf{i}) \in \mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$. As a result, the final considered combinations of $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ amount to $1{,}728 \times 24 \times 1 = 41{,}472$.

Further Reduction. The consideration of $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$ can further reduce the considered combinations. A brief reduction can be derived from the fact that two ciphers with $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ and $\{\mathsf{S}^{-1}_{\mathtt{BOGI}}, \mathsf{LS}^{-1}, (\mathsf{i}, \mathsf{i})\}$ have the same best weights if $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$ is involutory. $\mathsf{Shuf}_{\mathtt{BOGI16}}$ and $\mathsf{Shuf}_{\mathtt{BOGI64}}$ satisfy the property.

Moreover, for some $\mathsf{LS} \in \mathcal{LS}$, there may exist (a, b) such that two ciphers derived from $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ and $\{\mathsf{a} \circ \mathsf{S}_{\mathtt{BOGI}} \circ \mathsf{b}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ have the same best weights. This equivalence is briefly given in Appendix C.5. The reduction depends on block size – `BOGI-16`, `-128`, `-256` : 25,920 and `BOGI-32` : 29,376.

However, we do not apply both reductions when searching for the best trails of BOGI-based ciphers.

5.4 Best Trail Search on BOGI-based Ciphers

Pruning Conditions. Because all the BOGI-based ciphers are bit permutation-based AES-like ciphers, the applied pruning conditions are derived from Propositions 4 and 6. Since all the considered $\mathsf{S}_{\mathtt{BOGI}}$ have unbalanced differential weights, the impact of $\mathsf{PC1}$ and $\mathsf{PC2}$ of Propositions 4 and 6 on the differential trail search is more influential.

Permutation Characteristics. Because we restrict $\mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ as a pair of the identities (i, i), the 16-bit permutation $\mathsf{M}_{\mathtt{BOGI}}$ is derived solely from LS. Therefore, for simplicity, we use the notation LS instead of $\{\mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ to indicate $\mathsf{M}_{\mathtt{BOGI}}$. We classify the 24 representatives of $\mathcal{LS}$ depending on $\mathcal{D}$ obtained from the connected cyclic subgraphs of Algorithm 2. Since $\mathcal{D}$ is affected by the shuffle layer $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$, the classifications depend on each version of $\mathtt{BOGI\text{-}16} \cdot n$.

64-bit BOGI-based ciphers have the reduction in $|\mathcal{A}tab|$ is up to 15.77 factors.

`BOGI-16` $\cdot n$	i of LS_i	$\lvert\mathcal{A}tab\rvert$	$\lvert\mathcal{D}\rvert$	Reduce?	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
`BOGI-64`	6, 7, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21	65,535	1	✗	65,535	1
	0, 5, 9, 14, 18, 22		16	✓	4,335	15.12
	1, 4		16	✓	4,155	15.77
	2, `GIFT`		16	✓	4,155	15.77
	3, 8		16	✓	4,155	15.77

Although `GIFT-128` has no reduction in $\lvert\mathcal{A}tab\rvert$, 128-bit BOGI-based ciphers have a reduction of up to 32 factors.

`BOGI-16` $\cdot n$	i of LS_i	$\lvert\mathcal{A}tab\rvert$	$\lvert\mathcal{D}\rvert$	Reduce?	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
`BOGI-128`	1, 2, 4, 6, 7, 12, 15, 16, 19, 20, 21, `GIFT`	$2^{32}-1$	1	✗	$2^{32}-1$	1
	3, 8		2	✓	$\approx 2^{31.00}$	2
	11, 13		4	✓	$\approx 2^{30.00}$	4
	10, 17		8	✓	$\approx 2^{29.00}$	8
	0, 5, 9, 14, 18, 22		32	✓	$\approx 2^{27.00}$	32

The permutation characteristics of `BOGI-16`, `BOGI-32`, and `BOGI-256` can be found in Suppl. F.

Considering the Decomposition of P-layer. At line 6 and 20 of Algorithm 9, $Z = \mathsf{M}^*(\mathbb{T}[r].\mathbb{Y}\langle k\rangle)$ are computed whenever $\mathsf{PC2}$ is checked. These duplicate evaluations can be removed by storing Z in $\mathbb{T}[r].\mathbb{Z}\langle k\rangle$. Moreover, since M^* is a linear function, $\mathbb{T}[r].\mathbb{Z}\langle k\rangle$ can be gradually obtained by XORing it with $\mathsf{M}^*(\mathbf{0} \parallel \mathbb{T}[r].\mathbb{Y}[c] \parallel \mathbf{0})$, which is similar to the use of *T-table* in `AES` implementations. Another advantage of using $\mathbb{T}[r].\mathbb{Z}$ is to simplify the computation $\mathsf{Perm}^*(\mathbb{T}[r].\mathbb{Y})$ for the next input state $\mathbb{T}[r+1].\mathbb{X}$ into the simple word-wise permutation $\mathsf{Shuf}(\mathbb{T}[r].\mathbb{Z})$.

We adjust such approaches into our search program for BOGI-based ciphers. The dedicated implementation requires standard memory loads and XOR instructions to evaluate $\mathsf{M}_{\mathtt{BOGI}}(\mathbf{0} \parallel \mathbb{T}[r].\mathbb{Y}[c] \parallel \mathbf{0})$, which is more efficient than SIMD instructions for $\mathsf{Perm}_{\mathtt{BOGI16}\cdot n}$ in [JZD20]. It is noted that the final implementation for the full-round best differential and linear trail searches takes 0.390 and 9.755 s (resp. 89 and 451.3 h) on `GIFT-64` (resp. `GIFT-128`), which is detailed in Suppl. E.

5.5 Optimal DC/LC Resistance of BOGI-based Ciphers

This subsection investigates the DC/LC resistance of BOGI-based ciphers. The considered block sizes include 16-, 32-, 64-, 128-, and 256-bit. Table 6 summarizes the range of analysis rounds and the elapsed time on each version of BOGI-based ciphers. Among these versions, we demonstrate the resistance of 16-, 32-, 64-, and 128-bit versions. Table 2 summarizes the analysis results, whereas the detailed analysis results can be found in Suppl. G.

Table 6: Summary of the Best Trail Searches on BOGI-Based Ciphers

`BOGI-16` $\cdot n$	Combinations	Trail Type	Rounds	Min Elapsed	Avg Elapsed	Max Elapsed
`BOGI-16`	41,472	Differential	$2 \sim 15$	0.001 s	0.016 s	1.261 s
		Linear		<0.001 s	0.003 s	0.11 s
`BOGI-32`		Differential	$2 \sim 15$	0.001 s	0.072 s	4.622 s
		Linear		0.001 s	0.03 s	1.229 s
`BOGI-64`		Differential	$2 \sim 13$	0.004 s	29.508 s	0.5 h
		Linear		0.002 s	40.618 s	1.3 h
`BOGI-128`		Differential	$2 \sim 11$	0.011 s	1.5 h	450.9 h
		Linear		0.004 s	0.9 h	156.4 h
`BOGI-256`	3,456*	Differential	$2 \sim 9$	13.842 s	5.4 h	93.5 h
		Linear	$2 \sim 11$	0.82 s	289.452 s	5.7 h

*For 256-bit versions, we consider BOGI-based ciphers with LS that supports $\lvert\mathcal{D}\rvert = 32$. The considered combinations amount to $1{,}728 \times 2 = 3{,}456$ out of 41,472.

16-bit BOGI-based ciphers require **at least 5 rounds** for both the best differential and linear weights to become larger than or equal to 16. The combination of $\{\mathsf{S}_{\mathtt{GIFT}}, \mathsf{LS}_{\mathtt{GIFT}}\}$ with

$\mathsf{Shuf}_{\texttt{BOGI16}}$, denoted by GIFT-16, has the 5-round differential and linear weights as $(15.8, 14)$, whereas the 5-round optimal resistance $(17, 16)$ can be obtained by 240 combinations.

32-bit BOGI-based ciphers require **at least 8 rounds** for both the best differential and linear weights to become larger than or equal to 32. The combination of $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{\texttt{GIFT}}\}$ with $\mathsf{Shuf}_{\texttt{BOGI32}}$, denoted by GIFT-32, has the 8-round best differential and linear weights as $(28.8, 24)$, whereas the 8-round optimal resistance $(33, 32)$ can be obtained by 24 combinations.

64-bit BOGI-based ciphers require **at least 12 rounds** for both the best differential and linear weights to become larger than or equal to 64. The 12-round optimal resistance $(68, 64)$ can be obtained by 96 combinations. Among them, two examples are given as follows:

1. $\{\mathsf{S}_{\texttt{BOGI}} = (3, 10, 4, 7, 12, 1, 9, 14, 6, 13, 11, 2, 0, 15, 5, 8), \mathsf{LS}_6\}$
2. $\{\mathsf{S}_{\texttt{BOGI}} = (5, 2, 10, 12, 9, 7, 4, 11, 3, 13, 0, 6, 14, 1, 15, 8), \mathsf{LS}_{12}\}$

Considering that GIFT-64 has the 12-round differential and linear weights as $(58, 62)$ and requires 14 rounds to prevent efficient trails for DC/LC, these combinations outperform GIFT-64.

GIFT-128 requires 22 rounds to prevent efficient trails for DC/LC. However, we evaluate the best weights of 128-bit BOGI-based ciphers up to 11 rounds instead of 22 rounds because of the increased analysis time. The 11-round optimal resistance $(72, 72)$ can be obtained using 8 combinations:

1. $\{\mathsf{S}_{\texttt{BOGI}} = (2, 6, 11, 8, 15, 9, 4, 3, 1, 12, 13, 7, 10, 5, 0, 14), \mathsf{LS}_0\}$
2. $\{\mathsf{S}_{\texttt{BOGI}} = (1, 7, 9, 4, 2, 14, 12, 11, 15, 8, 6, 3, 5, 0, 10, 13), \mathsf{LS}_0\}$
3. $\{\mathsf{S}_{\texttt{BOGI}} = (8, 15, 4, 10, 9, 6, 3, 5, 14, 1, 7, 0, 2, 12, 13, 11), \mathsf{LS}_0\}$
4. $\{\mathsf{S}_{\texttt{BOGI}} = (4, 8, 15, 5, 13, 11, 2, 0, 6, 3, 9, 10, 1, 14, 12, 7), \mathsf{LS}_0\}$
5. $\{\mathsf{S}_{\texttt{BOGI}} = (11, 0, 5, 10, 2, 13, 12, 1, 9, 4, 6, 15, 14, 7, 3, 8), \mathsf{LS}_0\}$
6. $\{\mathsf{S}_{\texttt{BOGI}} = (13, 9, 4, 7, 10, 6, 3, 12, 0, 2, 11, 14, 5, 15, 8, 1), \mathsf{LS}_0\}$
7. $\{\mathsf{S}_{\texttt{BOGI}} = (14, 8, 6, 11, 0, 7, 1, 13, 5, 3, 9, 12, 10, 4, 15, 2), \mathsf{LS}_0\}$
8. $\{\mathsf{S}_{\texttt{BOGI}} = (7, 10, 0, 5, 6, 9, 8, 15, 1, 12, 14, 2, 13, 3, 11, 4), \mathsf{LS}_0\}$

Based on the above promising combinations, we further analyze the best weights up to 22 rounds. The result shows that all the combinations require **19 rounds** for their best weights to be $(134, 132)$. This implies that they require 3 rounds fewer than GIFT-128 to prevent efficient trails for DC/LC.

5.6 Considering the Implementation Cost of Round Function

Because replacing $\mathsf{M}_{\texttt{GIFT}}$ $(= \mathsf{LS}_{\texttt{GIFT}})$ with other variants does not present an additional implementation cost, we ascertain whether better combinations $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_i\}$ exist by only replacing the existing bit permutation $\mathsf{LS}_{\texttt{GIFT}}$ in GIFT. The evaluation shows that such replacements allow for GIFT-64 and -128 to prevent efficient differential and linear trails in at most 1 and 2 rounds fewer, respectively. All the optimal replacements out of 24 are as follows.

- $\mathsf{S}_{\texttt{GIFT}} = (1, 10, 4, 12, 6, 15, 3, 9, 2, 13, 11, 7, 5, 0, 8, 14)$
- GIFT-64 : the 13-round, 14-round best weights as $(62, 68)$, $(64, 74)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{12}\}$, $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{19}\}$: the 13-round best weights as $(65.4, 64)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{16}\}$: the 13-round best weights as $(69.4, 64)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{7}\}$, $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{20}\}$: the 13-round best weights as $(70, 64)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{13}\}$: the 13-round best weights as $(69.4, 68)$
- GIFT-128[8] : the 21-round, 22-round best weights as $(126.4, 136)$, $(132.4, 148)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{15}\}$: the 20-round best weights as $(130.8, 132)$
  - $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{20}\}$: the 20-round best weights as $(136.4, 128)$

[8] $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_6\}$ prevents efficient differential trails with 20 rounds since it has the 20-round best differential weight as 133.4. However, we could not obtain the minimum required rounds for LC resistance. Instead, we obtained the 18-round best linear weight as 122.

- $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{18}\}$: the 20-round best weights as $(138.8, 128)$
- $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{4}\}$: the 20-round best weights as $(131.4, 136)$
- $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{5}\}$: the 20-round best weights as $(129, 140)$
- $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{22}\}$: the 20-round best weights as $(132, 138)$
- $\{\mathsf{S}_{\texttt{GIFT}}, \mathsf{LS}_{16}\}$: the 20-round best weights as $(133.8, 138)$

We also consider the replacement of S-box, as 384 S-boxes out of the 1,728 S-boxes have the same software/hardware implementation costs as those of $\mathsf{S}_{\texttt{GIFT}}$ [KHSH20]. However, these further replacements do not result in an improvement to `GIFT-64` compared with only replacing $\mathsf{M}_{\texttt{GIFT}}$ (details are given in Suppl. H).

6 Conclusion

In this study, we attempted to optimize Matsui's search algorithm for AES-like ciphers by strengthening the pruning conditions and employing permutation characteristics. The main idea of the former is to take advantage of the structure of AES-like ciphers, whereas the latter stems from the fact that two trails that can be derived from each other via a permutation characteristic have the same weight. Moreover, we applied our algorithm to investigate the optimal DC/LC resistance that BOGI-based ciphers can achieve, and suggested combinations of S-box and bit permutation that are superior to `GIFT`.

Software and Experimental Result. Our codes and more detailed results can be found in https://github.com/jeffgyeom/Best-Trail-Search-on-AES-Like-Ciphers. The codes could be easily adapted to other ciphers or modified for clustering. Moreover, the obtained best trails of BOGI-based ciphers will be helpful for future design considerations.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. This work was supported as part of Military Crypto Research Center (UD210027XD) funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

References

[AK19] Ralph Ankele and Stefan Kölbl. Mind the gap - A closer look at the security of block ciphers against differential cryptanalysis. In Carlos Cid and Michael J. Jacobson Jr:, editors, *SAC 2018*, volume 11349 of *LNCS*, pages 163–190. Springer, Heidelberg, August 2019.

[AKM97] Kazumaro Aoki, Kunio Kobayashi, and Shiho Moriai. Best differential characteristic search of FEAL. In Eli Biham, editor, *FSE'97*, volume 1267 of *LNCS*, pages 41–53. Springer, Heidelberg, January 1997.

[ANP20] Alexandre Adomnicai, Zakaria Najm, and Thomas Peyrin. Fixslicing: A new gift representation. *IACR TCHES*, 2020(3):402–427, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8595.

[BBF15] Arnaud Bannier, Nicolas Bodin, and Eric Filiol. Automatic search for a maximum probability differential characteristic in a substitution-permutation network. In *2015 48th Hawaii International Conference on System Sciences*, pages 5165–5174, 2015.

[BBI$^+$15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 411–436. Springer, Heidelberg, November / December 2015.

[BBP$^+$19] Subhadeep Banik, Andrey Bogdanov, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, Elmar Tischhauser, and Yosuke Todo. SUNDAE-GIFT. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates, 2019. [Online; accessed 10-May-2022].

[BCD$^+$] Lawrence Bassham, Donghoon Chang, Tyler Diamond, Jinkeon Kang, John Kelsey, Kerry McKay, Meltem Sönmez Turan, and Noah Waller. Lightweight Cryptography. https://csrc.nist.gov/Projects/lightweight-cryptography/finalists. [Online; accessed 10-May-2022].

[BCI$^+$20] Subhadeep Banik, Avik Chakraborti, Tetsu Iwata, Kazuhiko Minematsu, Mridul Nandi, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT-COFB. Cryptology ePrint Archive, Report 2020/738, 2020. https://eprint.iacr.org/2020/738.

[BJK$^+$16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 123–153. Springer, Heidelberg, August 2016.

[BKL$^+$07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466. Springer, Heidelberg, September 2007.

[BLMR19] Christof Beierle, Gregor Leander, Amir Moradi, and Shahram Rasoolzadeh. CRAFT: Lightweight tweakable block cipher with efficient protection against DFA attacks. *IACR Trans. Symm. Cryptol.*, 2019(1):5–45, 2019.

[BPP$^+$17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *CHES 2017*, volume 10529 of *LNCS*, pages 321–345. Springer, Heidelberg, September 2017.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, January 1991.

[BZL15] Zhenzhen Bao, Wentao Zhang, and Dongdai Lin. Speeding up the search algorithm for the best differential and best linear trails. In Dongdai Lin, Moti Yung, and Jianying Zhou, editors, *Information Security and Cryptology*, pages 259–285, Cham, 2015. Springer International Publishing.

[CDJ$^+$19] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancilias Lopez, Mridul Nandi, and Yu Sasaki. LOTUS Mode of Authenticated Encryption. https://www.isical.ac.in/~lightweight/lotus/, 2019. [Online; accessed 10-May-2022].

[CDJ+20] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, Cuauhtemoc Mancillas-López, Mridul Nandi, and Yu Sasaki. ESTATE: A lightweight and low energy authenticated encryption mode. *IACR Trans. Symm. Cryptol.*, 2020(S1):350–389, 2020.

[CDJN19] Avik Chakraborti, Nilanjan Datta, Ashwin Jha, and Mridul Nandi. HyENA. https://www.isical.ac.in/~lightweight/hyena/, 2019. [Online; accessed 10-May-2022].

[Cho10] Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 302–317. Springer, Heidelberg, March 2010.

[DH19] Orr Dunkelman and Senyang Huang. Reconstructing an s-box from its difference distribution table. *IACR Transactions on Symmetric Cryptology*, 2019(2):193âĂŞ217, Jun. 2019.

[DR20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael.* Springer Berlin Heidelberg, 2020.

[FN20] Antonio Flórez-Gutiérrez and María Naya-Plasencia. Improving key-recovery in linear attacks: Application to 28-round PRESENT. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 221–249. Springer, Heidelberg, May 2020.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 326–341. Springer, Heidelberg, September / October 2011.

[JZD20] Fulei Ji, Wentao Zhang, and Tianyou Ding. Improving Matsui's Search Algorithm for the Best Differential/Linear Trails and its Applications for DES, DESL and GIFT. *The Computer Journal*, 64(4):610–627, 08 2020.

[KHSH20] S. Kim, D. Hong, J. Sung, and S. Hong. Classification of 4-bit s-boxes for bogi permutation. *IEEE Access*, 8:210935–210949, 2020.

[LMR15] Gregor Leander, Brice Minaud, and Sondre Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iSCREAM and Zorro. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 254–283. Springer, Heidelberg, April 2015.

[Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 386–397. Springer, Heidelberg, May 1994.

[Mat95] Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 366–375. Springer, Heidelberg, May 1995.

[OMA95] Kazuo Ohta, Shiho Moriai, and Kazumaro Aoki. Improving the search algorithm for the best linear expression. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 157–170. Springer, Heidelberg, August 1995.

[RP21] Adrián Ranea and Bart Preneel. On self-equivalence encodings in white-box implementations. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O'Flynn, editors, *Selected Areas in Cryptography*, pages 639–669, Cham, 2021. Springer International Publishing.

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Accelerating the search of differential and linear characteristics with the SAT method. *IACR Trans. Symm. Cryptol.*, 2021(1):269–315, 2021.

[XZBL16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 648–678. Springer, Heidelberg, December 2016.

A Algorithms

Algorithm 3: The sub-procedure version of `MiddleRound()` in Algorithm 1

```
Procedure MiddleRound($r$)
    $\mathbb{T}[r].\mathbb{X} \leftarrow \mathsf{Perm}^*(\mathbb{T}[r-1].\mathbb{Y})$
    MiddleSubRound($r$, 0)
Procedure MiddleSubRound($r$, $c$)
    for each $Y$ sorted in ascending order according to $\mathrm{W}\big(\mathbb{T}[r].\mathbb{X}[c] \xrightarrow{\mathsf{S}} Y\big)$ do
        $\mathbb{T}[r].\mathbb{Y}[c] \leftarrow Y$
        $\mathbb{T}[r].\mathbb{W}[c] \leftarrow \mathrm{W}\big(\mathbb{T}[r].\mathbb{X}[c] \xrightarrow{\mathsf{S}} \mathbb{T}[r].\mathbb{Y}[c]\big)$
        if $\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \mathbb{B}[R-r] \leq B_{set}$ then
            if $c = mn - 1$ then
                if $r + 1 = R$ then
                    LastRound()
                else
                    MiddleRound($r+1$)
            else
                MiddleSubRound($r$, $c+1$)
        else
            break
```

Algorithm 4: The optimized version of `FirstRound()` with $\mathcal{A}tab$

```
$\underline{W} \leftarrow \underline{\Delta}_{\mathsf{S}}$ or $\underline{\Gamma}_{\mathsf{S}}$
$\mathcal{A}tab \leftarrow \left\{ \begin{array}{c} |\mathcal{A}|\text{-combinations of the word indices } \{0, ..., mn-1\} \\ \text{for } 1 \leq |\mathcal{A}| < mn \end{array} \right\}$
Sort $\mathcal{A}$ of $\mathcal{A}tab$ in ascending order according to $|\mathcal{A}|$
Procedure FirstRound()
    for $\mathcal{A}$ in $\mathcal{A}tab$ do
        $\mathbb{W}_{\mathsf{PC1}} \leftarrow \underline{W} \times |\mathcal{A}|$
        if $\mathbb{W}_{\mathsf{PC1}} + \mathbb{B}[R-1] \leq B_{set}$ then
            $\mathbb{T}[1].\mathbb{X} \leftarrow \mathbf{0}$, $\mathbb{T}[1].\mathbb{Y} \leftarrow \mathbf{0}$, $\mathbb{T}[1].\mathbb{W} \leftarrow \mathbf{0}$
            $c_{init} \leftarrow \min \mathcal{A}$
            FirstSubRound($c_{init}$, $\mathcal{A}$)
        else
            break
Procedure FirstSubRound($c$, $\mathcal{A}$)
    $\mathcal{A}_{c<j<mn} \leftarrow \{\ c < j < mn : j \in \mathcal{A}\}$
    $\mathbb{W}_{rem} \leftarrow \underline{W} \times |\mathcal{A}_{c<j<mn}|$
    for each nonzero $Y$ sorted in ascending order according to $\min_X \mathrm{W}\big(X \xrightarrow{\mathsf{S}} Y\big)$ do
        $\mathbb{T}[1].\mathbb{X}[c] \leftarrow \arg\min_X \mathrm{W}\big(X \xrightarrow{\mathsf{S}} Y\big)$
        $\mathbb{T}[1].\mathbb{Y}[c] \leftarrow Y$, $\mathbb{T}[1].\mathbb{W}[c] \leftarrow \mathrm{W}\big(X \xrightarrow{\mathsf{S}} Y\big)$
        $\mathbb{W}_{\mathsf{PC1}} \leftarrow \sum_{j=0}^{c} \mathbb{T}[1].\mathbb{W}[j] + \mathbb{W}_{rem}$
        if $\mathbb{W}_{\mathsf{PC1}} + \mathbb{B}[R-1] \leq B_{set}$ then
            if $|\mathcal{A}_{c<j<mn}| = 0$ then
                if $R = 2$ then
                    LastRound()
                else
                    MiddleRound(2)
            else
                $c_{nxt} \leftarrow \min \mathcal{A}_{c<j<mn}$
                FirstSubRound($c_{nxt}$, $\mathcal{A}$)
```

Algorithm 5: Our Search Algorithm

```
Input: $R \geq 2$ and $\mathbb{B}[1, \ldots, R-1]$
Output: $\overline{\mathbb{T}}$ and $\mathbb{B}[R] = \mathrm{W}(\overline{\mathbb{T}})$
$\mathcal{A}tab \leftarrow \left\{ \begin{array}{c} |\mathcal{A}|\text{-combinations of the word indices} \\ \{0, \ldots, mn-1\} \text{ for } 1 \leq |\mathcal{A}| < mn \end{array} \right\}$
Reduce $\mathcal{A}tab$ by permutation characteristics
Sort $\mathcal{A}$ of $\mathcal{A}tab$ in ascending order according to $|\mathcal{A}|$
$\underline{W} \leftarrow \underline{\Delta}_{\mathsf{S}}$ or $\underline{\Gamma}_{\mathsf{S}}$
$\overline{W} \leftarrow \overline{\Delta}_{\mathsf{S}}$ or $\overline{\Gamma}_{\mathsf{S}}$
$\mathsf{Perm}^* \leftarrow \mathsf{Perm}$ or $\mathsf{Perm}^{-\top}$
$B_{init}, B_{set}, \mathbb{T}, \mathbb{T}_{out}$
Found $\leftarrow$ FALSE
EstimateBoundAndStart()
Procedure EstimateBoundAndStart()
    $B_{init} \leftarrow \mathbb{B}[R-1] + \overline{W}$
    while Found is FALSE do
        $B_{set} \leftarrow B_{init}$
        FirstRound()
        $B_{init} \leftarrow B_{init} + \underline{W}$
    return $\mathbb{B}[R]$, $\mathbb{T}_{out}$
Procedure BoundCheck($r$, $\mathbb{W}$)
    $\mathbb{W}_{ck} \leftarrow \mathbb{W} + \mathbb{B}[R-r]$
    if Found is TRUE then
        if $\mathbb{W}_{ck} \geq B_{set}$ then
            return EXCEED
        else
            return UNDERBOUND
    else
        if $\mathbb{W}_{ck} > B_{set}$ then
            return EXCEED
        else
            return UNDERBOUND
// FillUndetermined() and NextRoundLowerBound()
   are presented in Algorithm 7 and 9,
   respectively.
// FirstSubRound(), MiddleSubRound(), and
   LastSubRound() are presented in Algorithm 6.
Procedure FirstRound()
    for $\mathcal{A}$ in $\mathcal{A}tab$ do
        $\mathbb{W}_{\mathsf{PC1}} \leftarrow |\mathcal{A}| \times \underline{W}$
        if BoundCheck(1, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
            break
        $\mathbb{W}_{nxt} \leftarrow$ NextRoundLowerBound(1, 0, NULL, $\mathcal{A}$)
        $\mathbb{W}_{\mathsf{PC2}} \leftarrow \mathbb{W}_{\mathsf{PC1}} + \mathbb{W}_{nxt}$
        if BoundCheck(2, $\mathbb{W}_{\mathsf{PC2}}$) returns EXCEED then
            continue
        $\mathbb{T}[1].\mathbb{X} \leftarrow \mathbf{0}$
        $\mathbb{T}[1].\mathbb{Y} \leftarrow \mathbf{0}$
        $\mathbb{T}[1].\mathbb{W} \leftarrow \mathbf{0}$
        $\mathbb{W}_{cum} \leftarrow 0$
        $c_{init} \leftarrow \min \mathcal{A}$
        FirstSubRound($c_{init}$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
Procedure MiddleRound($r$, $\mathbb{W}_{cum}$)
    $\mathbb{T}[r].\mathbb{X} \leftarrow \mathsf{Perm}^*\left(\mathbb{T}[r-1].\mathbb{Y}\right)$
    $\mathcal{A} \leftarrow \{j : \mathbb{T}[r].\mathbb{X}[j] \neq \mathbf{0}\}$
    $\mathbb{W}_{\mathsf{PC1}} \leftarrow \mathbb{W}_{cum}+$ FillUndetermined($c$, $\mathbb{T}[r].\mathbb{X}$, $\mathcal{A}$)
    if BoundCheck($r$, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
        return
    $\mathbb{W}_{nxt} \leftarrow$ NextRoundLowerBound($r$, 0, NULL, $\mathcal{A}$)
    $\mathbb{W}_{\mathsf{PC2}} \leftarrow \mathbb{W}_{\mathsf{PC1}} + \mathbb{W}_{nxt}$
    if BoundCheck($r+1$, $\mathbb{W}_{\mathsf{PC2}}$) returns EXCEED then
        return
    $\mathbb{T}[r].\mathbb{Y} \leftarrow \mathbf{0}$
    $\mathbb{T}[r].\mathbb{W} \leftarrow \mathbf{0}$
    $c_{init} \leftarrow \min \mathcal{A}$
    MiddleSubRound($r$, $c_{init}$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
Procedure LastRound($\mathbb{W}_{cum}$)
    $\mathbb{T}[R].\mathbb{X} \leftarrow \mathsf{Perm}^*\left(\mathbb{T}[R-1].\mathbb{Y}\right)$
    $\mathcal{A} \leftarrow \{j : \mathbb{T}[R].\mathbb{X}[j] \neq \mathbf{0}\}$
    $\mathbb{W}_{\mathsf{PC1}} \leftarrow \mathbb{W}_{cum}+$ FillUndetermined($c$, $\mathbb{T}[r].\mathbb{X}$, $\mathcal{A}$)
    if BoundCheck($R$, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
        return
    $\mathbb{T}[R].\mathbb{Y} \leftarrow \mathbf{0}$
    $\mathbb{T}[R].\mathbb{W} \leftarrow \mathbf{0}$
    $c_{init} \leftarrow \min \mathcal{A}$
    LastSubRound($c_{init}$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
```

Algorithm 6: `FirstSubRound()`, `MiddleSubRound()`, and `LastSubRound()`

```
Procedure FirstSubRound($c$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
    $\mathcal{A}_{c<j<mn} \leftarrow \{\ c < j < mn : j \in \mathcal{A}\}$
    $\mathbb{W}_{rem} \leftarrow \mathbb{W}_{cum} + |\mathcal{A}_{c<j<mn}| \times \underline{W}$
    if $|\mathcal{A}_{c<j<mn}| = 0$ then
        $c_{nxt} \leftarrow mn$
    else
        $c_{nxt} \leftarrow \min \mathcal{A}_{c<j<mn}$
    for each nonzero $Y$ sorted in ascending order according to $\min_X \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$ do
        $X \leftarrow \arg\min_X \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$
        $W \leftarrow \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$
        $\mathbb{W}_{\mathsf{PC1}} \leftarrow \mathbb{W}_{rem} + W$
        if BoundCheck($1$, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
            break
        $\mathbb{T}[1].\mathbb{X}[c] \leftarrow X$
        $\mathbb{T}[1].\mathbb{Y}[c] \leftarrow Y$
        $\mathbb{T}[1].\mathbb{W}[c] \leftarrow W$
        $\mathbb{W}_{nxt} \leftarrow$ NextRoundLowerBound($1$, $c_{nxt}$, $\mathbb{T}[1].\mathbb{Y}$, $\mathcal{A}$)
        $\mathbb{W}_{\mathsf{PC2}} \leftarrow \mathbb{W}_{\mathsf{PC1}} + \mathbb{W}_{nxt}$
        if BoundCheck($2$, $\mathbb{W}_{\mathsf{PC2}}$) returns EXCEED then
            continue
        $\mathbb{W}_{det} \leftarrow \mathbb{W}_{cum} + W$
        if $c_{nxt} = mn$ then
            if $R = 2$ then
                LastRound($\mathbb{W}_{det}$)
            else
                MiddleRound($2$, $\mathbb{W}_{det}$)
        else
            FirstSubRound($c_{nxt}$, $\mathcal{A}$, $\mathbb{W}_{det}$)

Procedure MiddleSubRound($r$, $c$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
    $\mathcal{A}_{c<j<mn} \leftarrow \{\ c < j < mn : j \in \mathcal{A}\}$
    $\mathbb{W}_{rem} \leftarrow \mathbb{W}_{cum} +$ FillUndetermined($c$, $\mathbb{T}[r].\mathbb{X}$, $\mathcal{A}$)
    $X \leftarrow \mathbb{T}[r].\mathbb{X}[c]$
    if $|\mathcal{A}_{c<j<mn}| = 0$ then
        $c_{nxt} \leftarrow mn$
    else
        $c_{nxt} \leftarrow \min \mathcal{A}_{c<j<mn}$
    for each nonzero $Y$ sorted in ascending order according to $\mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$ do
        $W \leftarrow \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$
        $\mathbb{W}_{\mathsf{PC1}} \leftarrow \mathbb{W}_{rem} + W$
        if BoundCheck($r$, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
            break
        $\mathbb{T}[r].\mathbb{Y}[c] \leftarrow Y$
        $\mathbb{T}[r].\mathbb{W}[c] \leftarrow W$
        $\mathbb{W}_{nxt} \leftarrow$ NextRoundLowerBound($r$, $c_{nxt}$, $\mathbb{T}[r].\mathbb{Y}$, $\mathcal{A}$)
        $\mathbb{W}_{\mathsf{PC2}} \leftarrow \mathbb{W}_{\mathsf{PC1}} + \mathbb{W}_{nxt}$
        if BoundCheck($r + 1$, $\mathbb{W}_{\mathsf{PC2}}$) returns EXCEED then
            continue
        $\mathbb{W}_{det} \leftarrow \mathbb{W}_{cum} + W$
        if $c_{nxt} = mn$ then
            if $r = R - 1$ then
                LastRound($\mathbb{W}_{det}$)
            else
                MiddleRound($r + 1$, $\mathbb{W}_{det}$)
        else
            MiddleSubRound($r$, $c_{nxt}$, $\mathcal{A}$, $\mathbb{W}_{det}$)
    return

Procedure LastSubRound($c$, $\mathcal{A}$, $\mathbb{W}_{cum}$)
    $\mathcal{A}_{c<j<mn} \leftarrow \{\ c < j < mn : j \in \mathcal{A}\}$
    $X \leftarrow \mathbb{T}[R].\mathbb{X}[c]$
    $Y \leftarrow \arg\min_Y \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$
    $W \leftarrow \mathrm{W}\Big(X \overset{\mathsf{S}}{\to} Y\Big)$
    $\mathbb{W}_{det} \leftarrow \mathbb{W}_{cum} + W$
    $\mathbb{W}_{\mathsf{PC1}} \leftarrow \mathbb{W}_{det} +$ FillUndetermined($c$, $\mathbb{T}[r].\mathbb{X}$, $\mathcal{A}$)
    if BoundCheck($R$, $\mathbb{W}_{\mathsf{PC1}}$) returns EXCEED then
        return
    $\mathbb{T}[R].\mathbb{Y}[c] \leftarrow Y$
    $\mathbb{T}[R].\mathbb{W}[c] \leftarrow W$
    if $|\mathcal{A}_{c<j<mn}| = 0$ then
        Found $\leftarrow$ TRUE
        $B_{set} \leftarrow \mathbb{W}_{det}$
        $\mathbb{B}[R] \leftarrow B_{set}$
        $\mathbb{T}_{out} \leftarrow \mathbb{T}$
    else
        $c_{nxt} \leftarrow \min \mathcal{A}_{c<j<mn}$
        LastSubRound($c_{nxt}$, $\mathcal{A}$, $\mathbb{W}_{det}$)
```

Algorithm 7: `FillUndetermined`$(c, \mathbb{T}[r].\mathbb{X}, \mathcal{A})$ for the Previous and Our Methods

```
// PC1 based on Proposition 2
Procedure FillUndetermined($c, \mathbb{T}[r].\mathbb{X}, \mathcal{A}$)
    $\mathcal{A}_{c<j<mn} \leftarrow \{\, c < j < mn : j \in \mathcal{A}\}$
    return $|\mathcal{A}_{c<j<mn}| \times \underline{W}$

// PC1 based on Proposition 4
Procedure FillUndetermined($c, \mathbb{T}[r].\mathbb{X}, \mathcal{A}$)
    $\mathbb{W}_u \leftarrow 0$
    for $j \leftarrow c+1$ to $mn-1$ do
        $\mathbb{W}_u \leftarrow \mathbb{W}_u + \min_Y \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y)$
    return $\mathbb{W}_u$
```

Algorithm 8: `NextRoundLowerBound`$(r, n_{det}, \mathbb{T}[r].\mathbb{Y}, \mathcal{A})$ for the Previous Method

```
$c \leftarrow n_{det} - 1$
if $n_{det} = 0$ then
    return 0
// No PC2 is applied
if $\mathsf{Perm}^*$ is Not a Bit Permutation then
    return 0
// PC2 based on Proposition 3
if $\mathsf{Perm}^*$ is a Bit Permutation then
    return $\mathrm{ACT}\Big(\mathsf{Perm}^*(\mathbb{T}[r].\mathbb{Y}[0,\ldots,c] \parallel \mathbf{0})\Big) \times \underline{W}$
```

Algorithm 9: `NextRoundLowerBound`$(r, n_{det}, \mathbb{T}[r].\mathbb{Y}, \mathcal{A})$ for Our Method

```
$c \leftarrow n_{det} - 1$
$n_{\mathsf{M}} \leftarrow \lfloor (c+1)/m \rfloor$
$\mathbb{W}_1 \leftarrow 0,\ \mathbb{W}_2 \leftarrow 0,\ \mathbb{W}_3 \leftarrow 0$
if $n_{\mathsf{M}} \neq 0$ then
    for $k \leftarrow 0$ to $n_{\mathsf{M}} - 1$ do
        $Z \leftarrow \mathsf{M}^*\Big(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\Big)$
        for $j \leftarrow 0$ to $m-1$ do
            $\mathbb{W}_1 \leftarrow \mathbb{W}_1 + \min_Y \mathrm{W}\Big(Z[j] \xrightarrow{\mathsf{S}} Y\Big)$
if $n_{\mathsf{M}} = n$ then
    return $\mathbb{W}_1$
// PC2 based on Proposition 5
if $\mathsf{Perm}^*$ is Not a Bit Permutation then
    for $k \leftarrow n_{\mathsf{M}}$ to $n-1$ do
        $x \leftarrow |\{\, mk \le j < m(k+1) : j \in \mathcal{A}\}|$
        if $x \neq 0$ then
            $\mathbb{W}_2 \leftarrow \mathbb{W}_2 + \max(1, \mathcal{B}^* - x) \times \underline{W}$
    return $\mathbb{W}_1 + \mathbb{W}_2$
// PC2 based on Proposition 6
if $\mathsf{Perm}^*$ is a Bit Permutation then
    $m_{\mathsf{M}} \leftarrow c + 1 - mn_{\mathsf{M}}$
    if $m_{\mathsf{M}} \neq 0$ then
        $Z \leftarrow \mathsf{M}^*\Big(\mathbb{T}[r].\mathbb{Y}\langle n_{\mathsf{M}}\rangle[0,\ldots,m_{\mathsf{M}}-1] \parallel \mathbf{0}\Big)$
        $\mathbb{W}_2 \leftarrow \mathrm{ACT}(Z) \times \underline{W}$
        if $n_{\mathsf{M}} + 1 \neq n$ then
            for $k \leftarrow n_{\mathsf{M}} + 1$ to $n-1$ do
                if $\mathbb{T}[r].\mathbb{Y}\langle k\rangle \neq \mathbf{0}$ then
                    $\mathbb{W}_3 \leftarrow \mathbb{W}_3 + \underline{W}$
    else
        for $k \leftarrow n_{\mathsf{M}}$ to $n-1$ do
            if $\mathbb{T}[r].\mathbb{Y}\langle k\rangle \neq \mathbf{0}$ then
                $\mathbb{W}_3 \leftarrow \mathbb{W}_3 + \underline{W}$
    return $\mathbb{W}_1 + \mathbb{W}_2 + \mathbb{W}_3$
```

B Representative Latin Squares for BOGI-based Ciphers

The following table presents the 24 representative 16-bit permutations $\mathsf{LS}_i \in \mathcal{LS}$. LS_i place the l-th input bit (MSB is the 0-th bit) in the $\pi_i(l)$-th output bit.

	l	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LS_0	$\pi_0(l)$	12	9	6	3	8	13	2	7	4	1	14	11	0	5	10	15
LS_1	$\pi_1(l)$	8	13	6	3	12	9	2	7	4	1	14	11	0	5	10	15
LS_2	$\pi_2(l)$	4	9	14	3	8	13	2	7	12	1	6	11	0	5	10	15
LS_3	$\pi_3(l)$	12	5	10	3	8	13	2	7	4	1	14	11	0	9	6	15
LS_4	$\pi_4(l)$	12	9	6	3	8	13	2	7	4	1	10	15	0	5	14	11
LS_5	$\pi_5(l)$	8	13	6	3	12	9	2	7	4	1	10	15	0	5	14	11
LS_6	$\pi_6(l)$	4	13	10	3	12	9	2	7	8	1	6	15	0	5	14	11
LS_7	$\pi_7(l)$	8	5	14	3	12	9	2	7	4	1	10	15	0	13	6	11
LS_8	$\pi_8(l)$	12	9	6	3	4	13	2	11	8	1	14	7	0	5	10	15
LS_9	$\pi_9(l)$	12	5	10	3	4	13	2	11	8	1	14	7	0	9	6	15
LS_{10}	$\pi_{10}(l)$	4	13	10	3	12	5	2	11	8	1	14	7	0	9	6	15
LS_{11}	$\pi_{11}(l)$	8	5	14	3	4	13	2	11	12	1	10	7	0	9	6	15
LS_{12}	$\pi_{12}(l)$	8	13	6	3	4	9	2	15	12	1	10	7	0	5	14	11
LS_{13}	$\pi_{13}(l)$	12	5	10	3	4	9	2	15	8	1	14	7	0	13	6	11
LS_{14}	$\pi_{14}(l)$	8	5	14	3	4	9	2	15	12	1	10	7	0	13	6	11
LS_{15}	$\pi_{15}(l)$	4	9	14	3	8	5	2	15	12	1	10	7	0	13	6	11
LS_{16}	$\pi_{16}(l)$	8	13	6	3	12	5	2	11	4	1	10	15	0	9	14	7
LS_{17}	$\pi_{17}(l)$	12	5	10	3	4	13	2	11	8	1	6	15	0	9	14	7
LS_{18}	$\pi_{18}(l)$	4	13	10	3	12	5	2	11	8	1	6	15	0	9	14	7
LS_{19}	$\pi_{19}(l)$	4	9	14	3	12	5	2	11	8	1	6	15	0	13	10	7
LS_{20}	$\pi_{20}(l)$	4	13	10	3	8	5	2	15	12	1	6	11	0	9	14	7
LS_{21}	$\pi_{21}(l)$	8	5	14	3	4	9	2	15	12	1	6	11	0	13	10	7
LS_{22}	$\pi_{22}(l)$	4	9	14	3	8	5	2	15	12	1	6	11	0	13	10	7
$\mathsf{LS}_{\mathtt{GIFT}}$	$\pi_{\mathtt{GIFT}}(l)$	12	1	6	11	8	13	2	7	4	9	14	3	0	5	10	15

C Proof

C.1 Proof of Proposition 3

It is sufficient to show that $\mathrm{ACT}\big(\mathsf{Perm}(\mathbb{T}[r].\mathbb{Y}[0,...,c] \parallel \mathbf{0})\big) \times \underline{W} \leq \mathbb{T}[r+1].\mathbb{W}$. According to Lemma 1, it is satisfied that

$$\begin{aligned}\mathrm{ACT}(\mathsf{Perm}(\mathbb{T}[r].\mathbb{Y}[0,...,c] \parallel \mathbf{0})) \times \underline{W} &\leq \mathrm{ACT}(\mathsf{Perm}(\mathbb{T}[r].\mathbb{Y})) \times \underline{W} \\ &= \mathrm{ACT}(\mathbb{T}[r+1].\mathbb{X}) \times \underline{W} \\ &\leq \mathbb{T}[r+1].\mathbb{W}.\end{aligned}$$

C.2 Proof of Lemma 3

Since Mix^* consists of n independent M^*s, it is satisfied that $\mathbb{T}[r].\mathbb{Z}\langle k\rangle = \mathsf{M}^*\big(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\big)$ for $0 \leq k < n_{\mathsf{M}}$. This yields that

$$\begin{aligned}\sum_{j=0}^{mn_{\mathsf{M}}-1} \mathbb{T}[r+1].\mathbb{W}[\sigma(j)] &\geq \sum_{j=0}^{mn_{\mathsf{M}}-1} \min_Y \mathrm{W}\big(\mathbb{T}[r+1].\mathbb{X}[\sigma(j)] \xrightarrow{\mathsf{S}} Y\big) \\ &= \sum_{j=0}^{mn_{\mathsf{M}}-1} \min_Y \mathrm{W}\big(\mathbb{T}[r].\mathbb{Z}[j] \xrightarrow{\mathsf{S}} Y\big) \\ &= \sum_{k=0}^{n_{\mathsf{M}}-1}\sum_{j=0}^{m-1} \min_Y \mathrm{W}\big((\mathbb{T}[r].\mathbb{Z}\langle k\rangle)[j] \xrightarrow{\mathsf{S}} Y\big) \\ &= \sum_{k=0}^{n_{\mathsf{M}}-1}\sum_{j=0}^{m-1} \min_Y \mathrm{W}\big(\mathsf{M}^*\big(\mathbb{T}[r].\mathbb{Y}\langle k\rangle\big)[j] \xrightarrow{\mathsf{S}} Y\big).\end{aligned}$$

C.3 Proof of Proposition 7

It is sufficient to show that $\mathbb{T}'$ is also a non-trivial differential trail because if then,

$$\begin{aligned}\mathrm{W}(\mathbb{T}') &= \sum_{i=1}^{R}\sum_{j=0}^{mn-1} \mathrm{W}(\mathbb{T}'[i].\mathbb{X}[j] \xrightarrow{\mathsf{S}} \mathbb{T}'[i].\mathbb{Y}[j]) \\ &= \sum_{i=1}^{R}\sum_{j=0}^{mn-1} \mathrm{W}(\mathbb{T}[i].\mathbb{X}[d^{(i)}(j)] \xrightarrow{\mathsf{S}} \mathbb{T}[i].\mathbb{Y}[d^{(i)}(j)]) = \mathrm{W}(\mathbb{T}),\end{aligned}$$

where $d^{(i)}$ is the corresponding shuffle function of $\mathsf{D}^{(i)}$. To do so, we will show that $\mathbb{T}'[i].\mathbb{X}' = \mathsf{Perm}\,(\mathbb{T}'[i].\mathbb{Y})$ for $1 \leq i \leq R$. Since $\mathbb{T}$ is non-trivial and $\mathsf{D}^{(i)} \xRightarrow{\mathsf{Perm}} \mathsf{D}^{(i+1)}$, it is satisfied that

$$\begin{aligned}\mathbb{T}'[i].\mathbb{X}' &= \mathsf{D}^{(i+1)}\,(\mathbb{T}[i].\mathbb{X}') = \mathsf{D}^{(i+1)}\,(\mathsf{Perm}\,(\mathbb{T}[i].\mathbb{Y})) \\ &= \mathsf{Perm}(\mathsf{D}^{(i)}\,(\mathbb{T}[i].\mathbb{Y})) = \mathsf{Perm}\,(\mathbb{T}'[i].\mathbb{Y})\,.\end{aligned}$$

C.4 Proof of Lemma 5

Since $(\mathsf{a},\mathsf{b}) \in \mathcal{BP}(\mathsf{S}_{\mathtt{BOGI}})$ and $\mathsf{S}'_{\mathtt{BOGI}} \circ \mathsf{i} \circ \mathsf{i} \circ \mathsf{S}'_{\mathtt{BOGI}} = \mathsf{a} \circ (\mathsf{S}_{\mathtt{BOGI}} \circ \mathsf{b} \circ \mathsf{a} \circ \mathsf{S}_{\mathtt{BOGI}}) \circ \mathsf{b}$, $(\mathsf{i},\mathsf{i}) \in \mathcal{BP}(\mathsf{S}'_{\mathtt{BOGI}})$.

Next, we will show that two $(16{\cdot}n)$-bit BOGI-based ciphers derived from $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{a},\mathsf{b})\}$ and $\{\mathsf{S}'_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i},\mathsf{i})\}$, denoted by BE and BE', are permutation equivalent up to round key additions for any rounds. Let $\mathsf{B} = (\underbrace{\mathsf{b} \parallel \cdots \parallel \mathsf{b}}_{4n \text{ times}})$, $\mathsf{A} = (\underbrace{\mathsf{a} \parallel \cdots \parallel \mathsf{a}}_{4n \text{ times}})$, and $\mathsf{LS}^\times = (\underbrace{\mathsf{LS} \parallel \cdots \parallel \mathsf{LS}}_{n \text{ times}})$.

For simplicity, we denote $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$ by Shuf without the subscript, and denote by Sub and Sub' the S-layers consisting of $\mathsf{S}_{\mathtt{BOGI}}$ and $\mathsf{S}'_{\mathtt{BOGI}}$, respectively. Note that $\mathsf{Sub}' = \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{B}$, and A and B are commutative with Shuf.

If BE and BE' have a single round, it is satisfied that

$$\begin{aligned}
&\mathsf{BE}' \\
&= \oplus_{k_1} \circ \mathsf{Shuf} \circ \mathsf{LS}^{\times} \circ \mathsf{Sub}' \circ \oplus_{k_0} \\
&= \oplus_{k_1} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{A}^{-1} \circ \mathsf{Sub}' \circ \oplus_{k_0} \\
&= \oplus_{k_1} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{A}^{-1} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{B} \circ \oplus_{k_0} \\
&= \oplus_{k_1} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{B} \circ \oplus_{k_0} \\
&= \oplus_{k_1} \circ \mathsf{B}^{-1} \circ \mathsf{Shuf} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{B} \circ \oplus_{k_0} \\
&= \mathsf{B}^{-1} \circ \oplus_{\mathsf{B}(k_1)} \circ \mathsf{Shuf} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \oplus_{\mathsf{B}(k_0)} \circ \mathsf{B} \\
&= \mathsf{B}^{-1} \circ \mathsf{BE} \circ \mathsf{B}
\end{aligned}$$

up to round key additions.

Assume that if $\mathsf{BE}^{(t)}$ and $\mathsf{BE}'^{(t)}$ have t rounds, $\mathsf{BE}'^{(t)} = \mathsf{B}^{-1} \circ \mathsf{BE}^{(t)} \circ \mathsf{B}$ up to round key additions. If $\mathsf{BE}^{(t+1)}$ and $\mathsf{BE}'^{(t+1)}$ have $t+1$ rounds, it is satisfied that

$$\begin{aligned}
&\mathsf{BE}'^{(t+1)} \\
&= \oplus_{k_{t+1}} \circ \mathsf{Shuf} \circ \mathsf{LS}^{\times} \circ \mathsf{Sub}' \circ \{\oplus_{k_i} \circ \mathsf{Shuf} \circ \mathsf{LS} \circ \mathsf{Sub}'\}_{i=1}^{t} \circ \oplus_{k_0} \\
&= \oplus_{k_{t+1}} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{A}^{-1} \circ \mathsf{Sub}' \circ \mathsf{B}^{-1} \circ \mathsf{BE}^{(t)} \circ \mathsf{B} \\
&= \oplus_{k_{t+1}} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{A}^{-1} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{B} \circ \mathsf{B}^{-1} \circ \mathsf{BE}^{(t)} \circ \mathsf{B} \\
&= \oplus_{k_{t+1}} \circ \mathsf{Shuf} \circ \mathsf{B}^{-1} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{BE}^{(t)} \circ \mathsf{B} \\
&= \oplus_{k_{t+1}} \circ \mathsf{B}^{-1} \circ \mathsf{Shuf} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{BE}^{(t)} \circ \mathsf{B} \\
&= \mathsf{B}^{-1} \circ \oplus_{\mathsf{B}(k_{t+1})} \circ \mathsf{Shuf} \circ \mathsf{B} \circ \mathsf{LS}^{\times} \circ \mathsf{A} \circ \mathsf{Sub} \circ \mathsf{BE}^{(t)} \circ \mathsf{B} \\
&= \mathsf{B}^{-1} \circ \mathsf{BE}^{(t+1)} \circ \mathsf{B}
\end{aligned}$$

up to round key additions. Therefore, by mathematical induction, it is true that BE and BE' are permutation equivalent up to round key additions for any rounds. This implies that BE and BE' have the same best weights for any rounds.

C.5 Consideration of $\mathsf{Shuf}_{\mathtt{BOGI16}\cdot n}$

This subsection shows that for some $\mathsf{LS} \in \mathcal{LS}$, there may exist (a, b) such that two ciphers derived from $\{\mathsf{S}_{\mathtt{BOGI}}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ and $\{\mathsf{b} \circ \mathsf{S}_{\mathtt{BOGI}} \circ \mathsf{a}, \mathsf{LS}, (\mathsf{i}, \mathsf{i})\}$ have the same best weights. Let $\mathsf{B} = (\underbrace{\mathsf{b} \parallel ... \parallel \mathsf{b}}_{4n \text{ times}})$, $\mathsf{A} = (\underbrace{\mathsf{a} \parallel ... \parallel \mathsf{a}}_{4n \text{ times}})$, and $\mathsf{LS}^{\times} = (\underbrace{\mathsf{LS} \parallel ... \parallel \mathsf{LS}}_{n \text{ times}})$. The equivalence is obtained by checking if

$$\mathsf{P}^{(t)} = \{\mathsf{B} \circ \mathsf{Shuf}_{\mathtt{BOGI16}\cdot n} \circ \mathsf{LS}^{\times} \circ \mathsf{A}\}^{t} \circ \{(\mathsf{LS}^{\times})^{-1} \circ \mathsf{Shuf}^{-1}_{\mathtt{BOGI16}\cdot n}\}^{t}$$

is a word-wise permutation for any $t \geq 1$. The proof can be obtained from the fact that $\mathsf{P}^{(t)}$ are commutative with $\mathsf{Sub}_{\mathtt{BOGI16}\cdot n}$.

>>>>>> Supplementary Material <<<<<<

D Analysis Results and Performance Comparisons

D.1 Experimental Environment and Notations

All the experiments were conducted on a system with Intel® Xeon® Gold 6230 CPU @ 2.10 GHz, and we used one core for each case. The following notations are used through result tables.

$\mathbb{B}[R]$	R-round best weight.
B^g_{init}	Initial bound weight B_{init} for the first trial.
B^f_{init}	Initial bound weight B_{init} updated for the last trial. It is always satisfied that $B^g_{init} \leq \mathbb{B}[R] \leq B^f_{init}$.
$\mathcal{M}_{prev}$	Elapsed time the existing method of [BBF15] requires from the initial bound B^g_{init}.
$\mathcal{M}_{pc}$	Elapsed time with our strengthened pruning conditions.
$\mathcal{M}_{our}$	Elapsed time with our strengthened pruning conditions and permutation characteristics applied.
$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	Improvement rate between $\mathcal{M}_{prev}$ and $\mathcal{M}_{pc}$.
$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	Improvement rate between $\mathcal{M}_{pc}$ and $\mathcal{M}_{our}$.
$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$	Final improvement rate.

D.2 AES

• Best Differential Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	6.0								
2	30.0	12.0	33.0	1.0 h	<0.001 s	<0.001 s	∞	∞	∞
3	54.0	36.0	57.0	-	258.654 s	17.452 s	-	14.82	-

• Best Linear Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	6.0								
2	30.0	12.0	36.0	1.4 h	<0.001 s	<0.001 s	∞	∞	∞
3	54.0	36.0	60.0	-	333.918 s	21.008 s	-	15.89	-

D.3 LED

• Best Differential Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	10.0	4.0	10.0	0.402 s	<0.001 s	<0.001 s	∞	∞	∞
3	18.0	12.0	18.0	6.991 s	0.033 s	0.008 s	210.58	4.00	842.34

• Best Linear Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	10.0	4.0	12.0	0.704 s	<0.001 s	<0.001 s	∞	∞	∞
3	18.0	12.0	20.0	23.488 s	0.051 s	0.013 s	463.27	4.02	1864.10

D.4 MIDORI-64

• Best Differential Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	8.0	4.0	10.0	0.011 s	<0.001 s	<0.001 s	∞	∞	∞
3	14.0	10.0	16.0	0.524 s	0.004 s	<0.001 s	131.12	∞	∞
4	32.0	16.0	34.0	-	1.1 h	455.930 s	-	8.42	-
5	46.0	34.0	46.0	-	3.2 h	607.661 s	-	18.97	-
6	60.0	48.0	60.0	-	73.8 h	6.9 h	-	10.68	-
7	70.0	62.0	71.0	-	147.7 h	16.7 h	-	8.83	-
8	76.0	72.0	78.0	-	1.8 h	686.880 s	-	9.56	-
9	82.0	78.0	84.0	-	4.634 s	4.459 s	-	1.04	-
10	100.0	84.0	102.0	-	53.6 h	5.9 h	-	9.07	-
11	114.0	102.0	114.0	-	-	138.4 h	-	-	-
12	124.0	116.0	125.0	-	-	42.0 h	-	-	-

• Best Linear Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	8.0	4.0	8.0	0.005 s	<0.001 s	<0.001 s	∞	∞	∞
3	14.0	10.0	14.0	0.079 s	0.002 s	<0.001 s	32.79	∞	∞
4	32.0	16.0	32.0	-	0.5 h	252.591 s	-	7.16	-
5	46.0	34.0	46.0	-	2.8 h	664.165 s	-	15.19	-
6	60.0	48.0	60.0	-	66.7 h	6.9 h	-	9.72	-
7	70.0	62.0	70.0	-	47.0 h	8.7 h	-	5.42	-
8	76.0	72.0	76.0	-	0.4 h	169.766 s	-	8.96	-
9	82.0	78.0	82.0	-	2.733 s	2.686 s	-	1.02	-
10	100.0	84.0	100.0	-	3.3 h	0.4 h	-	7.50	-
11	114.0	102.0	114.0	-	255.1 h	14.0 h	-	18.22	-
12	124.0	116.0	124.0	-	1.2 h	377.420 s	-	11.32	-
13	134.0	126.0	134.0	-	-	41.8 h	-	-	-
14	144.0	136.0	144.0	-	-	1.6 h	-	-	-
15	150.0	146.0	150.0	-	-	2.825 s	-	-	-
16	168.0	152.0	168.0	-	-	0.5 h	-	-	-

D.5 CRAFT

• Best Differential Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	8.0	6.0	9.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
4	12.0	10.0	13.0	0.007 s	0.002 s	<0.001 s	3.84	∞	∞
5	20.0	14.0	20.0	0.855 s	0.217 s	0.046 s	3.95	4.70	18.55
6	28.0	22.0	28.0	140.953 s	16.156 s	4.005 s	8.72	4.03	35.20
7	40.0	30.0	42.0	235.3 h	4.6 h	1.4 h	50.98	3.30	168.20
8	52.0	42.0	54.0	-	-	455.5 h	-	-	-

• Best Linear Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	8.0	6.0	10.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
4	12.0	10.0	14.0	0.015 s	0.004 s	0.002 s	3.53	2.53	8.94
5	20.0	14.0	22.0	2.528 s	0.388 s	0.104 s	6.51	3.73	24.28
6	28.0	22.0	30.0	724.806 s	91.097 s	29.274 s	7.96	3.11	24.76
7	40.0	30.0	42.0	171.1 h	3.3 h	3.1 h	52.09	1.04	54.33

D.6 SKINNY-64

• Best Differential Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	10.0	6.0	12.0	0.011 s	0.003 s	<0.001 s	4.24	∞	∞
4	16.0	12.0	18.0	0.249 s	0.039 s	0.010 s	6.46	3.86	24.95
5	24.0	18.0	24.0	3.127 s	0.719 s	0.165 s	4.35	4.36	18.96
6	32.0	26.0	32.0	288.315 s	10.708 s	1.963 s	26.93	5.45	146.86
7	52.0	34.0	52.0	-	210.2 h	27.6 h	-	7.62	-

• Best Linear Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	10.0	6.0	10.0	0.005 s	0.001 s	<0.001 s	3.57	∞	∞
4	16.0	12.0	16.0	0.114 s	0.026 s	0.007 s	4.45	3.76	16.75
5	24.0	18.0	26.0	20.081 s	1.947 s	0.451 s	10.32	4.32	44.57
6	32.0	26.0	34.0	0.9 h	162.942 s	35.506 s	19.93	4.59	91.47
7	52.0	34.0	54.0	-	-	256.1 h	-	-	-

D.7 SKINNY-128

• Best Differential Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	10.0	6.0	13.0	0.027 s	0.004 s	0.002 s	7.60	2.19	16.62
4	16.0	12.0	19.0	0.632 s	0.055 s	0.051 s	11.41	1.08	12.37
5	24.0	18.0	25.0	13.247 s	1.332 s	0.657 s	9.94	2.03	20.16
6	32.0	26.0	33.0	432.258 s	51.181 s	24.288 s	8.45	2.11	17.80

• Best Linear Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	10.0	6.0	16.0	0.072 s	0.014 s	0.004 s	5.12	3.71	19.00
4	16.0	12.0	22.0	1.690 s	0.203 s	0.083 s	8.34	2.44	20.31
5	24.0	18.0	28.0	111.002 s	8.459 s	2.004 s	13.12	4.22	55.39
6	32.0	26.0	36.0	7.9 h	1.0 h	0.5 h	8.16	2.14	17.49

D.8 PRESENT

• Best Differential Weights						
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc} = \mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}} = \frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0					
2	4.0	4.0	4.0	<0.001 s	<0.001 s	∞
3	8.0	6.0	9.0	<0.001 s	<0.001 s	∞
4	12.0	10.0	13.0	0.006 s	0.007 s	0.9
5	20.0	14.0	20.0	0.241 s	0.223 s	1.1
6	24.0	22.0	25.0	0.146 s	0.102 s	1.4
7	28.0	26.0	29.0	0.136 s	0.056 s	2.4
8	32.0	30.0	33.0	0.077 s	0.027 s	2.9
9	36.0	34.0	37.0	0.010 s	0.007 s	1.5
10	41.0	38.0	41.0	0.048 s	0.033 s	1.5
11	46.0	43.0	46.0	0.077 s	0.049 s	1.6
12	52.0	48.0	54.0	0.789 s	0.388 s	2.0
13	56.0	54.0	57.0	0.221 s	0.099 s	2.2
14	62.0	58.0	64.0	2.513 s	1.185 s	2.1
15	66.0	64.0	67.0	0.466 s	0.243 s	1.9
16	70.0	68.0	71.0	0.117 s	0.066 s	1.8
17	74.0	72.0	75.0	0.034 s	0.016 s	2.1
18	78.0	76.0	79.0	0.055 s	0.033 s	1.7
19	82.0	80.0	83.0	0.016 s	0.010 s	1.6
20	86.0	84.0	87.0	0.018 s	0.011 s	1.6
21	90.0	88.0	91.0	0.016 s	0.010 s	1.6
22	96.0	92.0	98.0	0.231 s	0.132 s	1.8
23	100.0	98.0	101.0	0.060 s	0.032 s	1.9
24	106.0	102.0	108.0	0.696 s	0.360 s	1.9
25	110.0	108.0	111.0	0.136 s	0.060 s	2.3
26	116.0	112.0	118.0	3.225 s	1.739 s	1.9
27	120.0	118.0	121.0	0.221 s	0.118 s	1.9
28	124.0	122.0	125.0	0.088 s	0.048 s	1.8
29	128.0	126.0	129.0	0.034 s	0.016 s	2.1
30	132.0	130.0	133.0	0.084 s	0.052 s	1.6
31	136.0	134.0	137.0	0.017 s	0.011 s	1.6
			Total	9.777 s	5.131 s	1.9

D.9 GIFT-64

• Best Differential Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	1.4								
2	3.4	2.8	5.8	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	7.0	4.8	7.8	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
4	11.4	8.4	11.4	0.007 s	0.005 s	<0.001 s	1.46	∞	∞
5	17.0	12.8	18.8	0.104 s	0.056 s	0.004 s	1.85	15.67	28.92
6	22.4	18.4	24.4	0.433 s	0.149 s	0.011 s	2.90	13.21	38.36
7	28.4	23.8	29.8	3.519 s	0.897 s	0.090 s	3.92	9.92	38.88
8	38.0	29.8	38.8	130.440 s	27.586 s	1.916 s	4.73	14.40	68.08
9	42.0	39.4	42.4	14.232 s	1.319 s	0.267 s	10.79	4.94	53.35
10	48.0	43.4	49.4	132.776 s	9.106 s	0.993 s	14.58	9.17	133.78
11	52.0	49.4	52.4	12.931 s	0.899 s	0.177 s	14.38	5.09	73.18
12	58.0	53.4	59.4	76.958 s	5.423 s	0.634 s	14.19	8.56	121.44
13	62.0	59.4	62.4	2.659 s	0.218 s	0.066 s	12.18	3.33	40.54
14	68.0	63.4	69.4	12.130 s	1.573 s	0.175 s	7.71	8.98	69.23
15	72.0	69.4	72.4	0.423 s	0.090 s	0.036 s	4.70	2.51	11.79
16	78.0	73.4	79.4	6.674 s	1.311 s	0.146 s	5.09	8.98	45.71
17	82.0	79.4	82.4	0.421 s	0.091 s	0.037 s	4.64	2.46	11.45
18	88.0	83.4	89.4	6.688 s	1.321 s	0.147 s	5.06	8.96	45.37
19	92.0	89.4	92.4	0.423 s	0.090 s	0.036 s	4.70	2.50	11.74
20	98.0	93.4	99.4	6.709 s	1.331 s	0.147 s	5.04	9.06	45.64
21	102.0	99.4	102.4	0.421 s	0.090 s	0.037 s	4.67	2.46	11.48
22	108.0	103.4	109.4	6.720 s	1.339 s	0.149 s	5.02	8.96	44.98
23	112.0	109.4	112.4	0.423 s	0.090 s	0.036 s	4.68	2.49	11.67
24	118.0	113.4	119.4	6.737 s	1.347 s	0.149 s	5.00	9.04	45.19
25	122.0	119.4	122.4	0.422 s	0.090 s	0.037 s	4.67	2.45	11.43
26	128.0	123.4	129.4	6.774 s	1.355 s	0.151 s	5.00	9.00	44.98
27	132.0	129.4	132.4	0.424 s	0.091 s	0.036 s	4.68	2.48	11.61
28	138.0	133.4	139.4	6.794 s	1.366 s	0.150 s	4.97	9.08	45.17
			Total	436.242 s	57.235 s	5.627 s	7.62	10.17	77.52

• Best Linear Weights									
R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc}$	$\mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}}$	$\frac{\mathcal{M}_{pc}}{\mathcal{M}_{our}}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0								
2	4.0	4.0	4.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
3	6.0	6.0	6.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
4	10.0	8.0	12.0	<0.001 s	<0.001 s	<0.001 s	∞	∞	∞
5	14.0	12.0	16.0	0.002 s	0.002 s	<0.001 s	1.31	∞	∞
6	20.0	16.0	20.0	0.023 s	0.018 s	0.001 s	1.29	13.69	17.69
7	26.0	22.0	26.0	0.119 s	0.077 s	0.008 s	1.55	9.27	14.36
8	32.0	28.0	32.0	1.005 s	0.568 s	0.048 s	1.77	11.77	20.81
9	40.0	34.0	42.0	38.481 s	18.464 s	2.427 s	2.08	7.61	15.86
10	50.0	42.0	50.0	612.174 s	301.243 s	21.548 s	2.03	13.98	28.41
11	58.0	52.0	60.0	0.4 h	899.000 s	63.717 s	1.79	14.11	25.20
12	62.0	60.0	64.0	254.368 s	108.001 s	11.072 s	2.36	9.75	22.97
13	68.0	64.0	68.0	197.336 s	90.373 s	10.632 s	2.18	8.50	18.56
14	74.0	70.0	74.0	265.399 s	121.480 s	18.375 s	2.18	6.61	14.44
15	80.0	76.0	80.0	225.098 s	109.007 s	16.877 s	2.06	6.46	13.34
16	86.0	82.0	86.0	139.898 s	72.981 s	12.304 s	1.92	5.93	11.37
17	92.0	88.0	92.0	57.407 s	34.011 s	5.127 s	1.69	6.63	11.20
18	98.0	94.0	98.0	13.432 s	9.366 s	1.644 s	1.43	5.70	8.17
19	104.0	100.0	104.0	9.524 s	6.329 s	1.180 s	1.51	5.36	8.07
20	110.0	106.0	110.0	9.363 s	5.944 s	1.439 s	1.58	4.13	6.51
21	116.0	112.0	116.0	11.028 s	6.984 s	1.280 s	1.58	5.46	8.62
22	122.0	118.0	122.0	9.851 s	6.193 s	1.471 s	1.59	4.21	6.70
23	128.0	124.0	128.0	11.058 s	7.024 s	1.289 s	1.57	5.45	8.58
24	134.0	130.0	134.0	9.928 s	6.220 s	1.474 s	1.60	4.22	6.74
25	140.0	136.0	140.0	11.120 s	7.055 s	1.300 s	1.58	5.43	8.55
26	146.0	142.0	146.0	9.941 s	6.263 s	1.488 s	1.59	4.21	6.68
27	152.0	148.0	152.0	11.135 s	7.098 s	1.312 s	1.57	5.41	8.49
28	158.0	154.0	158.0	10.002 s	6.295 s	1.492 s	1.59	4.22	6.71
			Total	1.0 h	0.5 h	177.506 s	1.92	10.31	19.79

D.10 GIFT-128

• Best Differential Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc} = \mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}} = \frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	1.4					
2	3.4	2.8	5.8	<0.001 s	<0.001 s	∞
3	7.0	4.8	7.8	0.003 s	0.001 s	2.3
4	11.4	8.4	11.4	0.033 s	0.023 s	1.4
5	17.0	12.8	18.8	0.615 s	0.331 s	1.9
6	22.4	18.4	24.4	1.459 s	0.603 s	2.4
7	28.4	23.8	29.8	10.907 s	2.827 s	3.9
8	39.0	29.8	41.8	1.7 h	833.735 s	7.2
9	45.4	40.4	46.4	0.3 h	111.879 s	9.1
10	49.4	46.8	49.8	74.630 s	3.835 s	19.5
11	54.4	50.8	56.8	506.167 s	23.051 s	22.0
12	60.4	55.8	61.8	896.855 s	40.930 s	21.9
13	67.8	61.8	67.8	0.4 h	77.168 s	20.8
14	79.0	69.2	81.2	34.6 h	2.3 h	14.7
15	85.4	80.4	86.4	4.7 h	0.3 h	14.8
16	90.4	86.8	92.8	1.8 h	499.185 s	13.3
17	96.4	91.8	97.8	3.0 h	710.925 s	15.2
18	103.4	97.8	103.8	5.5 h	0.3 h	16.1
19	110.8	104.8	110.8	15.8 h	0.9 h	17.6
20	121.4	112.2	124.2	-	33.7 h	-
21	126.4	122.8	128.8	-	1.3 h	-
22	132.4	127.8	133.8	-	1.6 h	-
23	139.4	133.8	139.8	-	2.4 h	-
24	146.8	140.8	146.8	-	5.9 h	-
25	157.4	148.2	160.2	-	210.7 h	-
26	162.4	158.8	164.8	-	8.5 h	-
27	168.4	163.8	169.8	-	9.0 h	-
28	174.4	169.8	175.8	-	11.1 h	-
29	181.8	175.8	181.8	-	13.9 h	-

• Best Linear Weights

R	$\mathbb{B}[R]$	B^g_{init}	B^f_{init}	$\mathcal{M}_{prev}$	$\mathcal{M}_{pc} = \mathcal{M}_{our}$	$\frac{\mathcal{M}_{prev}}{\mathcal{M}_{pc}} = \frac{\mathcal{M}_{prev}}{\mathcal{M}_{our}}$
1	2.0					
2	4.0	4.0	4.0	<0.001 s	<0.001 s	∞
3	6.0	6.0	6.0	<0.001 s	<0.001 s	∞
4	10.0	8.0	12.0	<0.001 s	<0.001 s	∞
5	14.0	12.0	16.0	0.005 s	0.004 s	1.3
6	20.0	16.0	20.0	0.082 s	0.067 s	1.2
7	26.0	22.0	26.0	0.383 s	0.248 s	1.5
8	34.0	28.0	36.0	56.153 s	32.719 s	1.7
9	44.0	36.0	44.0	700.208 s	358.827 s	2.0
10	52.0	46.0	54.0	3.3 h	1.1 h	3.1
11	62.0	54.0	62.0	12.9 h	3.9 h	3.3
12	72.0	64.0	72.0	44.9 h	16.1 h	2.8
13	76.0	74.0	78.0	2.7 h	0.6 h	4.5
14	82.0	78.0	82.0	1.9 h	0.5 h	3.8
15	90.0	84.0	92.0	58.0 h	8.8 h	6.6
16	96.0	92.0	96.0	24.7 h	3.1 h	8.1
17	102.0	98.0	102.0	14.4 h	1.9 h	7.5
18	112.0	104.0	112.0	165.8 h	22.4 h	7.39
19	118.0	114.0	118.0	26.0 h	3.6 h	7.26
20	128.0	120.0	128.0	160.9 h	30.4 h	5.28
21	136.0	130.0	138.0	-	40.0 h	-
22	148.0	138.0	150.0	-	360.0 h	-

E Analysis Results of GIFT

The following results are evaluated with a dedicated implementation to GIFT. The main differences from the implementation in Suppl. D are discussed in Subsection 5.4.

E.1 GIFT-64

• Best Differential Weights and Elapsed Times (Total Elapsed Time : 0.390 s)

R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed
1	1.4		8	38.0	0.139 s	15	72.0	0.003 s	22	108.0	0.010 s
2	3.4	<0.001 s	9	42.0	0.018 s	16	78.0	0.010 s	23	112.0	0.003 s
3	7.0	<0.001 s	10	48.0	0.064 s	17	82.0	0.003 s	24	118.0	0.010 s
4	11.4	<0.001 s	11	52.0	0.012 s	18	88.0	0.010 s	25	122.0	0.003 s
5	17.0	<0.001 s	12	58.0	0.041 s	19	92.0	0.003 s	26	128.0	0.010 s
6	22.4	0.001 s	13	62.0	0.005 s	20	98.0	0.010 s	27	132.0	0.003 s
7	28.4	0.008 s	14	68.0	0.012 s	21	102.0	0.003 s	28	138.0	0.011 s

• Best Linear Weights and Elapsed Times (Total Elapsed Time : 9.755 s)

R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed
1	2.0		8	32.0	0.003 s	15	80.0	0.890 s	22	122.0	0.081 s
2	4.0	<0.001 s	9	40.0	0.148 s	16	86.0	0.661 s	23	128.0	0.071 s
3	6.0	<0.001 s	10	50.0	1.280 s	17	92.0	0.273 s	24	134.0	0.081 s
4	10.0	<0.001 s	11	58.0	3.492 s	18	98.0	0.092 s	25	140.0	0.072 s
5	14.0	<0.001 s	12	62.0	0.582 s	19	104.0	0.064 s	26	146.0	0.082 s
6	20.0	<0.001 s	13	68.0	0.599 s	20	110.0	0.079 s	27	152.0	0.072 s
7	26.0	<0.001 s	14	74.0	0.979 s	21	116.0	0.071 s	28	158.0	0.082 s

E.2 GIFT-128

• Best Differential Weights and Elapsed Times (Total Elapsed Time : 89.0 h)

R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed
1	1.4		11	54.4	1.301 s	21	126.4	227.728 s	31	198.4	2.6 h
2	3.4	<0.001 s	12	60.4	2.314 s	22	132.4	275.725 s	32	204.4	1.7 h
3	7.0	<0.001 s	13	67.8	4.356 s	23	139.4	424.719 s	33	210.4	1.3 h
4	11.4	0.002 s	14	79.0	405.529 s	24	146.8	0.3 h	34	217.4	1.7 h
5	17.0	0.018 s	15	85.4	56.977 s	25	157.4	11.2 h	35	224.8	2.2 h
6	22.4	0.041 s	16	90.4	25.514 s	26	162.4	0.5 h	36	234.4	22.3 h
7	28.4	0.191 s	17	96.4	36.356 s	27	168.4	0.5 h	37	240.4	3.9 h
8	39.0	41.122 s	18	103.4	62.416 s	28	174.4	0.6 h	38	246.4	3.0 h
9	45.4	6.729 s	19	110.8	156.618 s	29	181.8	0.7 h	39	253.4	3.6 h
10	49.4	0.216 s	20	121.4	1.6 h	30	193.0	26.7 h	40	260.4	4.2 h

• Best Linear Weights and Elapsed Times (Total Elapsed Time : 451.3 h)

R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed	R	$\mathbb{B}[R]$	Elapsed
1	2.0		11	62.0	645.423 s	21	136.0	1.6 h	31	216.0	0.7 h
2	4.0	<0.001 s	12	72.0	0.7 h	22	148.0	15.7 h	32	224.0	8.8 h
3	6.0	<0.001 s	13	76.0	93.287 s	23	158.0	17.2 h	33	234.0	9.4 h
4	10.0	<0.001 s	14	82.0	78.174 s	24	164.0	1.3 h	34	242.0	9.5 h
5	14.0	<0.001 s	15	90.0	0.4 h	25	172.0	88.9 h	35	252.0	49.1 h
6	20.0	0.004 s	16	96.0	474.670 s	26	182.0	115.8 h	36	260.0	26.5 h
7	26.0	0.016 s	17	102.0	294.646 s	27	188.0	12.9 h	37	266.0	0.5 h
8	34.0	1.752 s	18	112.0	0.9 h	28	196.0	61.4 h	38	274.0	4.5 h
9	44.0	18.271 s	19	118.0	521.480 s	29	202.0	2.9 h	39	280.0	0.7 h
10	52.0	177.917 s	20	128.0	1.2 h	30	210.0	19.9 h	40	286.0	892.746 s

F Permutation Characteristics of BOGI-based Ciphers

16-Bit BOGI-Based Ciphers. The reduction in $|\mathcal{A}tab|$ is up to 3 factors.

BOGI-16 $\cdot\, n$	i of LS_i	$\lvert\mathcal{A}tab\rvert$	$\lvert\mathcal{D}\rvert$	Reduce?	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
BOGI-16	6, 7, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21	15	1	✗	15	1
	0, 5, 9, 14, 18, 22		4	✓	6	2.5
	1, 4		4	✓	5	3
	2, GIFT		4	✓	5	3
	3, 8		4	✓	5	3

32-Bit BOGI-Based Ciphers. The reduction in $|\mathcal{A}tab|$ is up to 6.54 factors.

BOGI-16 $\cdot\, n$	i of LS_i	$\lvert\mathcal{A}tab\rvert$	$\lvert\mathcal{D}\rvert$	Reduce?	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
BOGI-32	1, 2, 4, 6, 7, 12, 15, 16, 19, 20, 21, GIFT	255	1	✗	255	1
	3, 8		2	✓	135	1.89
	10, 17		4	✓	75	3.4
	0, 5, 9, 14, 18, 22		8	✓	45	5.67
	11, 13		8	✓	39	6.54

256-Bit BOGI-Based Ciphers. Since $|\mathrm{DWSE}(\mathsf{Perm}^*_{\mathtt{BOGI256}})|$ becomes infeasible to analyze, we reduce $\mathrm{DWSE}(\mathsf{Perm}^*_{\mathtt{BOGI256}})$ into

$$(\mathsf{A} \parallel \cdots \parallel \mathsf{A}) \circ \mathsf{C} \xrightarrow{\mathsf{Perm}^*_{\mathtt{BOGI256}}} \mathsf{Shuf}_{\mathtt{BOGI256}} \circ (\mathsf{B} \parallel \cdots \parallel \mathsf{B}) \circ \mathsf{C} \circ \mathsf{Shuf}^{-1}_{\mathtt{BOGI256}},$$

where $\mathsf{A} \xrightarrow{\mathsf{M}^*_{\mathtt{BOGI256}}} \mathsf{B}$ and C is a M-position transposition.

The classification is equal to that of 64-bit BOGI-based cipher. Since $|\mathcal{A}tab|$ amounts to $2^{64} - 1$, $|Opt\mathcal{A}tab|$ cannot be computed in a practical time. Therefore, we compare the sizes considering the number of the active S-boxes is from 1 to 4, denoted by $|\mathcal{A}tab_1^4|$ and $|Opt\mathcal{A}tab_1^4|$.

BOGI-16 $\cdot\, n$	i of LS_i	$\lvert\mathcal{A}tab_1^4\rvert$	$\lvert\mathcal{D}\rvert$	Reduce?	$\lvert Opt\mathcal{A}tab_1^4\rvert$	$\lvert\mathcal{A}tab_1^4\rvert/\lvert Opt\mathcal{A}tab_1^4\rvert$
BOGI-256	6, 7, 10, 11, 12, 13, 15, 16, 17, 19, 20, 21	679,120	1	✗	679,120	1
	0, 5, 9, 14, 18, 22		64	✓	10,416	61.01
	1, 4		64	✓	9,996	63.57
	2, GIFT		64	✓	9,996	63.57
	3, 8		64	✓	9,996	63.57

G The Best Weights of BOGI-based Ciphers

Rows : Best differential weights
Columns : Best linear weights
Cells : Number of `BOGI-16`$\cdot n$

G.1 5-Round 16-bit BOGI-based Ciphers

$\mathbb{B}[5]$	14	16
13.8	1920	1344
14.4	1152	-
14.8	6240	1536
15	672	-
15.2	-	1152
15.4	8928	2208
15.8	2928*	2784
16	2832	2688
16.4	1152	3504
16.8	-	192
17	-	240

*5-round `GIFT-16` has the best weights as (15.8, 14).

G.2 8-Round 32-bit BOGI-based Ciphers

$\mathbb{B}[8]$	20	22	24	26	28	30	32
23.4	-	-	48	128	104	-	-
24	-	128	176	224	32	-	-
24.2	-	16	-	-	-	-	-
24.4	40	192	232	464	272	144	-
24.8	-	-	48	920	280	-	-
25	-	48	400	568	96	8	-
25.4	-	136	976	2424	1720	336	16
25.8	-	16	32	64	48	48	-
26	-	8	168	720	800	184	16
26.4	40	64	296	2136	2392	816	56
26.8	-	-	-	-	128	16	-
27	8	24	216	944	1064	240	8
27.4	-	88	472	1440	1384	888	192
27.8	-	-	16	16	152	72	-
28	-	-	192	912	656	536	184
28.1	-	-	-	40	48	-	-
28.2	-	-	128	-	128	-	-
28.4	-	-	288	536	1256	400	32
28.7	-	-	-	40	48	8	-
28.8	-	-	72*	120	224	32	16
29	-	-	208	264	504	152	72
29.1	-	-	32	-	48	-	-
29.2	-	-	-	-	64	-	-
29.4	-	-	104	528	768	704	208
29.7	-	-	-	-	8	-	-
29.8	-	-	48	104	152	32	8
30	-	-	32	152	584	248	168
30.1	-	-	-	-	40	-	-
30.2	-	-	8	56	96	8	-
30.4	-	-	32	296	560	384	128
30.5	-	-	-	-	16	-	-
30.7	-	-	-	16	40	-	-
30.8	-	-	16	88	96	192	16
31	-	-	-	128	184	208	152
31.2	-	-	-	-	8	24	16
31.4	-	-	-	448	352	128	16
31.8	-	-	-	168	128	200	48
32	-	-	-	40	80	120	8
32.2	-	-	-	-	16	8	-
32.4	-	-	-	8	104	64	56
32.8	-	-	-	-	8	-	16
33	-	-	-	-	-	40	24
34	-	-	-	-	-	48	-

*8-round `GIFT-32` has the best weights of (28.8, 24).

G.3 12-Round 64-bit BOGI-based Ciphers

$\mathbb{B}[12]$	42	44	46	48	50	52	54	56	58	60	62	64	66	68
46.8	-	-	-	528	-	96	144	336	288	144	96	96	-	-
48	-	-	-	96	-	-	-	-	96	96	96	-	-	-
48.4	-	-	-	192	-	384	192	192	-	-	-	-	-	-
49.4	-	-	-	-	-	144	48	-	-	-	-	-	-	-
50.4	-	-	-	192	-	192	-	192	-	-	-	-	-	-
50.8	-	-	-	-	-	-	48	-	-	-	-	-	-	-
51.8	-	-	-	-	-	-	-	-	48	-	-	-	-	-
52	-	-	-	-	-	-	192	-	-	-	-	-	-	-
52.4	-	-	-	384	-	48	384	48	48	96	144	96	96	-
52.8	-	48	48	288	144	48	96	144	48	192	48	-	-	-
53.4	-	-	-	48	-	336	96	96	-	48	96	-	-	-
53.8	-	-	-	-	48	144	-	192	192	528	96	-	-	-
54	-	-	144	-	-	48	48	48	48	48	48	48	-	-
54.4	-	-	-	48	-	-	-	-	-	-	-	-	-	-
54.8	48	-	-	288	48	144	240	432	48	336	240	-	-	-
55.4	192	-	-	96	48	-	-	48	-	-	-	192	-	-
55.8	-	-	-	384	-	144	96	-	-	96	-	-	-	-
56	-	-	-	-	192	96	96	144	192	-	144	48	48	96
56.4	-	-	48	240	-	240	-	144	192	48	96	-	-	-
56.8	-	-	96	1584	96	432	288	1344	96	336	816	144	-	-
57	-	-	-	-	-	-	-	-	96	96	-	-	-	-
57.4	-	-	-	528	-	192	-	432	96	816	96	144	-	-
57.8	-	-	-	48	96	144	48	-	96	192	-	-	-	-
58	-	-	-	624	-	-	480	432	96	96	432*	48	-	-
58.4	-	-	-	-	-	-	288	144	192	384	-	-	-	-
58.8	-	-	96	1104	-	240	-	432	192	192	-	-	-	-
59	-	-	-	96	-	-	-	-	192	48	336	96	-	-
59.2	-	-	-	-	-	-	-	-	96	-	-	-	-	-
59.4	-	-	96	96	-	192	240	672	-	240	288	48	-	-
59.8	-	96	-	48	192	48	48	96	-	96	-	-	-	-
60	-	-	48	192	-	96	48	192	144	144	-	-	-	-
60.4	-	-	48	-	-	48	192	288	240	288	96	48	-	-
60.7	-	-	-	-	-	-	-	-	-	96	-	-	-	-
60.8	-	-	96	96	-	240	192	384	96	96	96	240	-	-
61	-	96	-	-	-	-	-	-	-	-	96	-	-	-
61.4	-	-	-	96	-	-	144	-	240	144	48	288	-	-
61.8	-	-	-	-	-	-	192	96	-	96	96	96	-	-
62	-	-	-	-	-	-	-	-	192	192	48	96	-	-
62.4	-	-	96	96	-	144	96	288	96	240	144	96	-	-
62.8	-	-	-	48	-	96	96	-	48	-	144	48	-	-
63	-	-	-	-	-	-	-	96	96	-	48	-	-	-
63.2	-	-	-	-	-	-	48	-	-	-	48	-	-	-
63.4	-	-	-	-	-	-	-	144	-	96	144	48	-	-
63.8	-	-	-	48	-	-	-	-	-	-	-	-	-	-
64	-	-	-	48	-	-	-	48	-	-	-	-	-	-
64.2	-	-	-	48	-	-	-	-	-	-	-	-	-	-
64.4	-	-	-	-	-	-	-	-	-	-	192	-	-	-
64.8	-	-	-	-	-	-	-	-	-	96	-	-	-	-
65	-	-	-	-	-	-	-	-	96	192	48	-	-	-
65.4	-	-	-	48	-	-	-	-	-	384	48	-	-	-
65.8	-	-	-	48	-	-	-	-	-	-	48	-	-	-
67.2	-	-	-	-	-	-	-	-	-	-	48	-	-	-
68	-	-	-	-	-	-	-	-	-	-	-	96	-	-

*12-round `GIFT-64` has the best weights of (58, 62).

G.4 11-Round 128-bit BOGI-based Ciphers

$\mathbb{B}[11]$	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72
42.8	128	-	-	40	40	-	-	-	48	-	-	-	-	-	-
44	-	-	-	-	8	32	88	-	-	-	-	-	-	-	-
45.4	120	-	-	40	16	-	56	56	160	72	120	-	-	-	-
47.4	-	-	-	-	16	-	-	16	-	-	48	48	-	-	-
47.8	-	-	-	-	-	-	32	8	8	-	-	-	-	-	-
48.4	192	-	-	-	224	-	56	168	280	160	136	64	-	-	-
48.8	-	-	-	-	-	16	32	16	-	32	-	-	-	-	-
49	-	-	-	-	-	-	-	-	-	32	24	8	-	-	-
49.4	-	-	8	-	8	-	16	32	-	-	16	-	-	-	-
49.8	-	-	-	-	56	16	80	96	88	80	24	32	-	-	-
50.4	-	-	-	-	-	-	-	-	64	64	16	16	-	-	-
50.7	-	-	-	-	-	-	8	-	8	-	-	-	-	-	-
50.8	192	40	16	40	112	96	376	208	96	64	16	8	-	-	-
51	-	-	-	-	-	-	8	-	16	-	-	-	-	-	-
51.4	-	-	-	-	64	-	40	16	8	16	-	48	-	-	-
51.7	256	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51.8	-	-	-	16	-	112	-	8	24	-	-	-	-	-	-
52	64	40	-	40	216	48	160	184	168	8	16	32	-	-	-
52.2	-	-	-	-	-	-	-	-	-	-	32	-	-	-	-
52.4	-	-	32	-	32	-	16	48	80	16	16	32	-	-	-
52.7	-	-	-	-	-	-	-	-	32	-	-	-	-	-	-
52.8	448	-	520	168	336	8	256	168	48	24	48	-	-	-	-
53	-	-	-	-	-	-	-	32	32	-	32	48	-	-	-
53.2	-	-	-	-	-	-	16	-	-	-	32	-	-	-	-
53.4	128	16	16	-	16	48	160	-	32	-	-	-	-	-	-
53.7	-	-	-	-	-	-	-	-	-	8	-	-	-	-	-
53.8	-	-	16	-	16	-	80	-	48	24	112	16	8	-	-
54	128	128	128	-	-	-	16	144	16	32	-	-	-	-	-
54.4	-	-	64	-	64	128	40	16	16	32*	24	-	-	-	-
54.8	8	-	144	32	104	104	352	144	144	168	200	24	32	-	-
55	-	-	-	-	-	-	-	16	-	-	-	-	-	-	-
55.2	-	-	-	-	-	-	-	-	-	32	32	-	-	-	-
55.4	-	-	-	-	-	16	-	32	-	-	-	16	32	-	-
55.7	-	-	-	-	-	-	8	-	8	16	-	-	-	-	-
55.8	-	-	-	-	-	16	128	112	64	112	32	64	-	-	-
56	-	16	72	48	8	-	24	72	-	-	16	-	-	-	-
56.2	-	-	-	32	64	96	48	16	48	-	-	-	-	-	-
56.4	-	16	-	88	-	56	96	192	96	64	48	16	-	-	-
56.5	-	-	-	-	-	-	-	-	16	-	-	-	-	-	-
56.7	-	-	-	-	-	-	16	-	64	32	-	-	-	-	-
56.8	56	8	56	64	208	72	328	48	208	120	432	80	8	-	-
57	-	-	-	-	-	64	32	32	88	16	32	16	-	-	-
57.2	-	-	112	-	-	-	-	16	64	16	-	-	-	-	-
57.4	24	-	16	56	-	16	128	160	80	80	112	16	-	-	-
57.5	-	-	8	-	-	-	-	-	-	-	-	-	-	-	-
57.8	-	8	16	8	-	-	64	8	-	32	64	-	-	-	-
58	-	-	-	16	24	192	32	16	200	24	-	96	-	-	-
58.2	-	-	-	-	-	-	128	112	16	-	16	-	-	-	-
58.4	-	-	64	32	-	16	48	80	48	16	16	48	16	-	-
58.7	-	-	8	-	32	-	-	-	-	-	-	-	-	-	-
58.8	64	-	96	64	112	40	144	144	224	112	72	40	8	-	-
59	-	-	-	-	24	-	32	-	-	16	-	-	-	-	-
59.2	-	-	-	-	-	-	-	64	8	-	-	-	-	-	-
59.4	16	-	64	48	64	-	200	96	160	40	56	40	-	-	-
59.8	16	-	16	-	-	64	144	264	72	112	160	104	24	-	-
60	-	-	-	32	48	56	40	40	240	16	24	32	8	-	-
60.2	64	-	-	-	32	64	-	80	-	24	16	-	-	-	-
60.4	-	-	16	96	-	40	32	168	72	56	112	32	-	-	-
60.7	-	-	-	-	-	-	-	-	-	8	8	-	16	-	-
60.8	160	8	72	40	72	72	224	552	440	144	88	32	-	-	-
61	-	-	-	-	-	-	96	104	136	40	-	16	-	-	-
61.2	-	-	-	-	-	-	-	8	8	-	24	8	-	-	-
61.4	16	16	16	48	64	64	112	96	144	24	40	16	-	16	-
61.7	8	-	8	8	-	-	8	-	-	8	-	-	-	-	-

*11-round `GIFT-128` has the best weights of (54.4, 62).

$\mathbb{B}$[11]	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72
61.8	-	32	32	16	-	16	88	72	104	48	136	8	-	8	-
62	80	16	-	16	40	24	8	40	88	-	32	40	40	-	-
62.2	-	16	-	-	-	-	16	16	8	8	16	40	8	-	-
62.4	32	8	-	-	8	112	80	96	8	64	88	-	8	-	-
62.7	-	-	-	-	-	16	48	16	24	48	-	8	-	-	-
62.8	32	24	16	56	112	136	96	64	224	144	184	144	16	-	-
63	16	-	16	8	32	-	-	8	24	-	24	32	16	-	-
63.1	-	-	-	-	-	-	-	-	8	-	-	-	-	-	-
63.2	-	-	-	-	-	-	-	-	40	32	32	40	-	-	-
63.4	16	16	-	48	40	40	136	24	192	32	160	48	32	-	-
63.7	-	-	-	-	-	-	-	-	16	-	24	-	-	-	-
63.8	-	-	32	8	-	24	48	40	96	40	64	8	-	-	-
64	-	-	-	-	16	48	24	16	16	8	16	72	40	24	-
64.1	-	-	-	-	-	-	-	8	-	-	-	-	-	-	-
64.2	-	-	-	-	-	-	8	-	40	16	16	40	-	-	-
64.4	16	32	16	40	32	128	104	64	208	48	88	64	-	-	-
64.5	-	-	-	-	-	-	-	-	-	-	16	-	-	-	-
64.7	-	-	-	-	-	-	-	16	-	-	-	40	8	-	-
64.8	-	24	-	32	152	40	176	88	224	232	248	48	24	64	8
65	-	8	-	16	24	8	56	8	48	-	8	16	8	-	-
65.1	-	-	-	-	-	-	-	-	16	16	32	-	-	-	-
65.2	-	-	-	-	8	8	8	-	-	-	48	8	-	-	-
65.4	24	-	-	56	88	56	72	144	168	32	64	24	-	-	16
65.5	-	-	-	-	-	-	-	16	-	-	24	-	-	-	-
65.7	-	8	-	-	-	-	16	-	24	-	8	-	-	-	-
65.8	-	-	24	16	24	-	80	40	96	72	40	80	32	-	-
66	-	-	-	8	48	-	40	48	24	56	24	8	88	16	8
66.2	-	-	-	-	16	32	32	16	16	40	-	16	-	-	-
66.4	8	-	72	8	152	24	40	264	72	16	-	-	-	-	-
66.5	-	-	-	-	-	-	8	-	-	-	-	-	-	-	-
66.7	-	8	-	-	-	-	8	-	-	32	80	32	-	-	-
66.8	-	-	40	16	120	72	112	40	160	80	88	-	-	-	-
67	-	-	-	8	8	88	48	32	32	8	24	-	32	-	8
67.2	-	-	-	-	-	-	32	16	-	16	16	8	-	-	-
67.4	-	24	-	16	80	64	120	40	128	64	72	32	8	16	-
67.7	-	-	-	-	16	-	8	-	-	8	-	8	-	-	-
67.8	-	32	-	8	40	24	160	72	80	48	40	72	16	-	-
68	-	8	-	8	40	48	48	16	24	16	40	48	8	16	-
68.2	-	-	-	-	8	16	8	8	8	16	-	-	-	-	-
68.4	-	-	-	16	48	56	40	56	40	-	-	16	-	-	-
68.7	-	-	-	8	-	-	16	-	24	8	-	-	-	-	-
68.8	-	-	-	8	8	56	56	48	24	16	24	24	48	8	-
69	-	-	-	-	40	-	16	-	112	40	56	40	-	32	-
69.2	-	-	-	-	16	-	-	-	16	-	-	8	-	-	-
69.4	-	24	-	-	-	24	48	24	-	8	32	48	-	-	-
69.7	-	-	-	-	8	16	24	-	8	-	8	-	-	-	-
69.8	-	16	-	-	-	40	16	16	16	16	8	-	-	-	-
70	-	-	-	8	-	-	40	8	32	16	32	24	-	-	-
70.2	-	-	-	-	-	16	-	-	8	16	-	16	-	-	-
70.4	-	-	-	16	8	-	-	24	16	16	-	-	-	-	-
70.8	-	8	-	-	-	48	16	16	8	8	8	16	-	-	-
71	-	-	-	-	-	-	16	24	-	-	8	8	16	-	-
71.2	-	-	-	16	8	16	-	8	-	-	-	-	-	-	-
71.4	-	-	-	-	-	24	-	-	-	-	8	-	-	-	-
71.7	-	32	-	-	16	16	-	8	16	-	-	-	-	-	-
71.8	-	-	-	-	-	8	8	-	-	16	-	-	16	-	8
72	-	-	-	8	8	-	32	-	8	8	-	8	8	-	8
72.2	-	-	-	16	-	-	-	8	-	-	-	-	-	-	-
72.4	-	-	-	-	-	32	8	-	8	8	-	-	-	-	-
72.8	-	8	16	16	-	8	-	-	-	16	-	-	-	-	-
73	-	8	-	-	-	-	-	-	-	-	-	-	-	-	-
73.2	-	-	-	-	-	-	8	-	-	-	-	8	-	-	-
73.4	-	-	-	-	-	-	-	8	-	-	-	-	-	-	-
73.7	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-
74.2	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-
74.4	-	-	-	-	-	8	-	-	-	-	-	-	-	-	-

H BOGI-based Ciphers Considering Implementation Cost

This section gives the best weights of BOGI-based ciphers that can support the same software/hardware implementation costs as those of `GIFT`'s S-box. Among the 1,728 BOGI-applicable S-boxes, 384 S-boxes have the same implementation costs. Therefore, the number of BOGI-based ciphers considered in this section is 384×24 = 9,216.

Rows : Best differential weights
Columns : Best linear weights
Cells : Number of `BOGI-16`·n

H.1 13-Round 64-bit BOGI-based Ciphers Considering Implementation Cost

$\mathbb{B}[13]$	50	52	54	56	58	60	62	64	66	68	70	72
50.8	-	192	-	-	-	48	-	-	-	96	48	-
52	-	96	-	-	-	-	48	144	48	48	-	-
56	-	-	-	-	-	96	-	-	-	-	-	-
56.4	-	-	-	-	-	192	-	-	-	-	144	48
56.8	-	96	96	48	-	144	-	48	48	-	-	-
57.4	-	-	-	-	-	192	-	-	192	-	-	-
58	-	-	-	-	-	-	-	-	48	48	-	-
58.8	48	-	-	48	-	-	-	-	96	-	-	-
59.4	96	-	-	-	-	-	-	-	-	-	-	-
59.7	-	-	-	-	-	-	-	-	48	48	-	-
59.8	-	-	-	-	-	-	-	-	-	192	-	-
60.4	-	-	-	-	-	-	-	-	-	-	48	144
60.8	-	-	48	-	48	288	-	48	240	192	96	-
61	-	-	-	-	-	-	-	96	96	-	-	-
61.2	48	48	-	-	-	-	-	-	-	-	-	-
61.4	-	-	-	-	-	-	192	-	96	-	-	-
61.8	-	-	-	-	-	96	-	-	-	-	-	-
62	-	-	-	-	-	-	-	96	96	288*	-	-
62.4	-	-	-	-	-	96	96	96	96	-	-	-
62.8	-	192	-	-	-	288	-	-	96	96	96	-
63.2	-	-	-	96	-	-	-	-	-	-	-	-
63.4	-	-	-	-	96	96	-	288	96	-	-	-
63.8	-	-	-	-	-	-	96	-	-	-	-	-
64	-	-	-	-	-	48	-	-	-	-	-	-
64.2	-	-	-	-	-	-	-	96	-	-	-	-
64.4	-	-	-	-	-	96	96	96	-	-	-	-
64.8	-	-	-	-	96	-	96	-	-	-	-	-
65.4	-	-	-	-	-	48	-	192	-	-	-	-
65.8	-	96	-	-	-	96	96	-	-	-	-	-
66	48	-	-	-	-	48	-	-	-	-	-	-
66.2	-	-	-	-	-	96	-	-	-	-	-	-
66.4	48	96	-	-	-	-	-	-	-	-	-	-
67.8	-	-	-	-	96	-	-	-	-	-	-	-
68	48	-	-	-	-	-	-	-	-	-	-	-
68.2	-	-	-	-	-	48	-	-	-	-	-	-
68.4	48	-	-	-	-	-	-	-	-	-	-	-
69	-	-	-	-	-	96	-	-	-	-	-	-
69.4	-	-	-	-	-	-	-	48	96	48	-	-
70	-	-	-	-	-	-	-	192	-	-	-	-

*13-round `GIFT-64` has the best weights of (62, 68).

H.2 11-Round 128-bit BOGI-based Ciphers Considering Implementation Cost

$\mathbb{B}[11]$	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72
42.8	64	-	-	-	40	-	-	-	24	-	-	-	-	-	-
44	-	-	-	-	8	32	88	-	-	-	-	-	-	-	-
47.4	-	-	-	-	16	-	-	16	-	-	48	48	-	-	-
48.4	-	-	-	-	16	-	-	16	-	-	48	48	-	-	-
48.8	-	-	-	-	-	-	-	-	-	16	-	-	-	-	-
50.8	-	-	-	-	-	72	160	-	24	-	-	-	-	-	-
51.4	-	-	-	-	-	-	-	-	-	16	-	48	-	-	-
52	-	-	-	-	8	40	32	88	88	-	-	-	-	-	-
52.4	-	-	-	-	32	-	16	32	64	-	16	32	-	-	-
52.8	128	-	392	-	64	8	56	48	16	-	8	-	-	-	-
53.4	-	16	16	-	-	-	-	-	-	-	-	-	-	-	-
53.8	-	-	16	-	16	-	16	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	16	-	32	-	-	-	-	-
54.4	-	-	48	-	-	-	-	-	-	32*	-	-	-	-	-
54.8	-	-	56	24	16	64	48	16	48	32	48	16	16	-	-
55	-	-	-	-	-	-	-	16	-	-	-	-	-	-	-
55.4	-	-	-	-	-	16	-	-	-	-	-	16	32	-	-
55.7	-	-	-	-	-	-	8	-	8	-	-	-	-	-	-
55.8	-	-	-	-	-	-	-	-	16	16	-	-	-	-	-
56	-	-	16	-	-	-	24	-	-	-	16	-	-	-	-
56.2	-	-	-	16	32	-	16	16	48	-	-	-	-	-	-
56.4	-	-	-	-	-	16	48	112	-	-	16	16	-	-	-
56.8	-	-	-	-	48	16	-	16	-	-	88	8	-	-	-
57	-	-	-	-	-	32	-	-	32	-	16	-	-	-	-
57.2	-	-	112	-	-	-	-	-	-	-	-	-	-	-	-
57.4	-	-	-	-	-	-	-	64	-	-	-	-	-	-	-
57.8	-	-	-	8	-	-	-	8	-	-	-	-	-	-	-
58	-	-	-	-	-	8	16	16	16	8	-	-	-	-	-
58.4	-	-	-	32	-	-	16	-	-	-	-	16	16	-	-
58.8	-	-	-	16	-	-	8	24	8	24	-	16	-	-	-
59	-	-	-	-	-	-	16	-	-	16	-	-	-	-	-
59.4	-	-	-	-	16	-	8	8	-	-	16	16	-	-	-
59.8	-	-	16	-	-	-	-	-	16	-	-	16	16	-	-
60	-	-	-	32	-	8	8	-	8	-	-	24	8	-	-
60.2	-	-	-	-	32	64	-	-	-	-	-	-	-	-	-
60.4	-	-	-	-	-	8	-	-	-	-	32	16	-	-	-
60.7	-	-	-	-	-	-	-	-	-	8	8	-	16	-	-
60.8	-	-	-	-	32	8	80	48	32	56	24	16	-	-	-
61	-	-	-	-	-	-	-	32	8	8	-	-	-	-	-
61.4	-	-	-	16	-	32	-	-	32	-	-	16	-	16	-
61.8	-	-	32	-	-	-	-	40	8	8	16	-	-	8	-
62	-	-	-	-	-	8	-	32	-	-	16	40	32	-	-
62.4	-	-	-	-	-	32	32	-	-	-	8	-	8	-	-
62.7	-	-	-	-	-	-	16	16	8	-	-	8	-	-	-
62.8	-	8	8	-	16	-	-	8	16	8	96	80	16	-	-
63	-	-	16	-	8	-	-	8	-	-	-	16	8	-	-
63.4	-	-	-	32	16	-	32	24	8	-	24	16	24	-	-
63.7	-	-	-	-	-	-	-	-	16	-	-	-	-	-	-
63.8	-	-	-	-	-	16	8	8	40	16	16	8	-	-	-

*11-round `GIFT-128` has the best weights of (54.4, 62).

$\mathbb{B}$[11]	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72
64	-	-	-	-	-	32	24	8	-	-	-	48	32	8	-
64.2	-	-	-	-	-	-	-	-	8	-	-	-	-	-	-
64.4	-	-	-	24	16	112	72	32	56	24	-	8	-	-	-
64.7	-	-	-	-	-	-	-	16	-	-	-	-	-	-	-
64.8	-	-	-	16	48	24	32	32	152	40	56	-	-	-	8
65	-	8	-	16	16	8	48	8	40	-	-	8	-	-	-
65.4	-	-	-	-	16	16	8	40	32	8	-	-	-	-	-
65.8	-	-	16	-	16	-	48	40	64	32	-	-	-	-	-
66	-	-	-	8	32	-	8	48	8	56	16	-	32	16	-
66.2	-	-	-	-	16	16	16	-	-	-	-	16	-	-	-
66.4	-	-	40	-	64	8	32	80	16	16	-	-	-	-	-
66.8	-	-	8	-	32	32	56	32	24	16	16	-	-	-	-
67	-	-	-	8	-	56	16	-	8	-	16	-	16	-	-
67.2	-	-	-	-	-	-	16	16	-	-	-	-	-	-	-
67.4	-	-	-	-	32	8	32	8	32	-	-	16	-	16	-
67.7	-	-	-	-	-	-	-	-	-	8	-	-	-	-	-
67.8	-	-	-	-	8	16	-	8	16	24	16	48	16	-	-
68	-	-	-	-	40	24	16	8	8	8	-	16	-	-	-
68.2	-	-	-	-	8	16	-	8	-	8	-	-	-	-	-
68.4	-	-	-	-	48	8	16	40	24	-	-	16	-	-	-
68.7	-	-	-	-	-	-	-	-	8	-	-	-	-	-	-
68.8	-	-	-	-	-	48	8	16	-	-	-	16	48	8	-
69	-	-	-	-	8	-	-	-	-	-	16	-	-	16	-
69.2	-	-	-	-	16	-	-	-	16	-	-	-	-	-	-
69.4	-	-	-	-	-	24	40	8	-	8	16	32	-	-	-
69.7	-	-	-	-	8	16	-	-	-	-	8	-	-	-	-
70	-	-	-	-	-	-	-	8	-	-	8	-	-	-	-
70.2	-	-	-	-	-	16	-	-	-	-	-	-	-	-	-
70.4	-	-	-	-	8	-	-	24	16	-	-	-	-	-	-
71	-	-	-	-	-	-	-	8	-	-	-	-	16	-	-
71.2	-	-	-	-	8	16	-	-	-	-	-	-	-	-	-
71.4	-	-	-	-	-	16	-	-	-	-	-	-	-	-	-
71.8	-	-	-	-	-	-	-	-	-	-	-	-	16	-	-
72	-	-	-	-	-	-	-	-	-	-	-	8	-	-	-
72.4	-	-	-	-	-	-	8	-	8	8	-	-	-	-	-
73.2	-	-	-	-	-	-	-	-	-	-	-	8	-	-	-