Accelerating the Best Trail Search on AES-Like Ciphers

Seonggyeom Kim Deukjo Hong Jaechul Sung Seokhie Hong

Korea University, Seoul, South Korea *jeffgyeom@korea.ac.kr*
Jeonbuk National University, Jeonju, South Korea *deukjo.hong@jbnu.ac.kr*
University of Seoul, Seoul, South Korea *jcsung@uos.ac.kr*
Korea University, Seoul, South Korea *shhong@korea.ac.kr*

FSE 2023
March, 2023

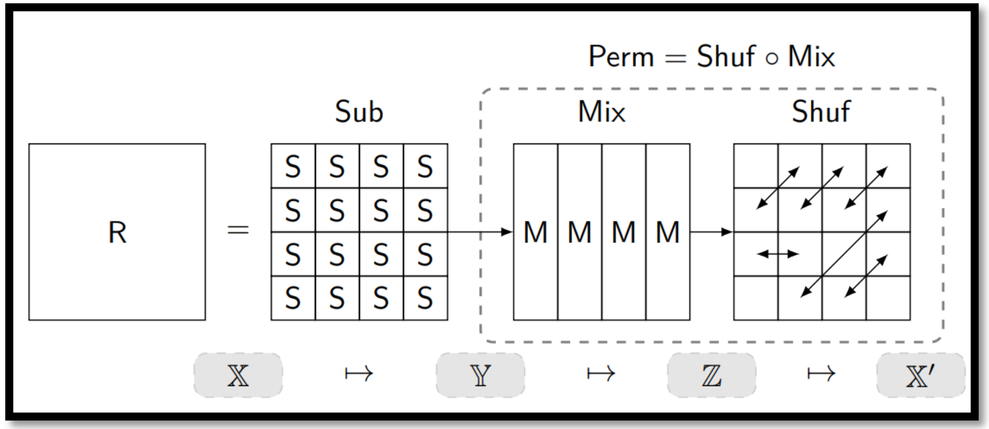
Presentation Overview

1. Contributions
2. Preliminaries
3. Two Accelerating Strategies
4. Analysis Results of BOGI-based Ciphers
5. Conclusion

Contribution-1 : Accelerating the Best Trail Search on AES-like Ciphers

- **We accelerate Matsui's search algorithm [Mat95] with**
  1. Strengthening the Pruning Conditions and
  2. Employing Permutation Characteristics in Trail Search.
- **Strict pruning conditions are derived from**
  - the structure of *AES-like ciphers*.
    - ➔ Two rounds can be represented with Super-Boxes.
- **Employing permutation characteristics in trail Search**
  - allows to reduce the search space.
- **Apply it to `GIFT`, `PRESENT`, `AES`, `LED`, `MIDORI-64`, `CRAFT`, and `SKINNY`.**
  - Trail searches get faster up to 1904 factors.
- **Our implementation and codes can be found publicly in**

`https://github.com/jeffgyeom/Best-Trail-Search-on-AES-Like-Ciphers.`



- An example of AES-like cipher's round function

Contribution-1 : Accelerating the Best Trail Search on AES-like Ciphers

Cipher	Best Trail Type	Range of Analysis Rounds	Total Elapsed Time
GIFT-64	Differential	**2 ∼ 28 (full-round)**	**0.390s***
	Linear	**2 ∼ 28 (full-round)**	**9.755s***
GIFT-128	Differential	**2 ∼ 40 (full-round)**	**89.0h***
	Linear	**2 ∼ 40 (full-round)**	**451.3h***
PRESENT	Differential	2 ∼ 31 (full-round)	7.280s
AES	Differential	2 ∼ 3	17.452s
	Linear	2 ∼ 3	21.009s
LED	Differential	2 ∼ 3	0.008s
	Linear	2 ∼ 3	0.013s
MIDORI-64	Differential	2 ∼ 12	210.5h
	Linear	**2 ∼ 16 (full-round)**	**74.2h**
CRAFT	Differential	2 ∼ 8	456.9h
	Linear	2 ∼ 7	3.2h
SKINNY-64	Differential	2 ∼ 7	27.6h
	Linear	2 ∼ 7	256.1h
SKINNY-128	Differential	2 ∼ 6	24.998s
	Linear	2 ∼ 6	0.5h

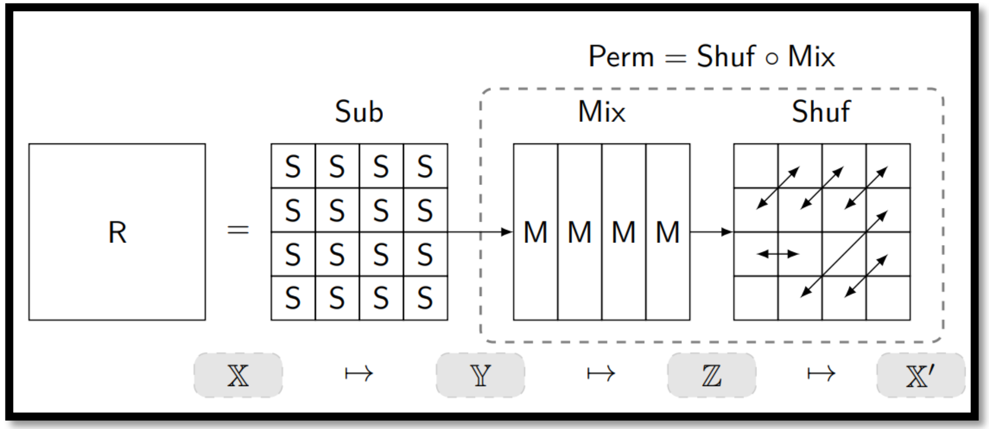
First analysis result on full-round GIFT-128

Contribution-1 : Accelerating the Best Trail Search on AES-like Ciphers

- **We accelerate Matsui's search algorithm [Mat95] with**
  1. Strengthening the Pruning Conditions and
  2. Employing Permutation Characteristics in Trail Search.
- **Strict pruning conditions are derived from**
  - the structure of *AES-like ciphers*.

    ➔ Two rounds can be represented with Super-Boxes.
- **Employing permutation characteristics in trail Search**
  - allows to reduce the search space.
- **Apply it to `GIFT`, `PRESENT`, `AES`, `LED`, `MIDORI-64`, `CRAFT`, and `SKINNY`.**
  - Trail searches get faster up to 1904 factors.
- **Our implementation and codes can be found publicly in**

`https://github.com/jeffgyeom/Best-Trail-Search-on-AES-Like-Ciphers.`



- An example of AES-like cipher's round function

Contribution-2 : Investigating the Most DC/LC Resistant BOGI-based Cipher

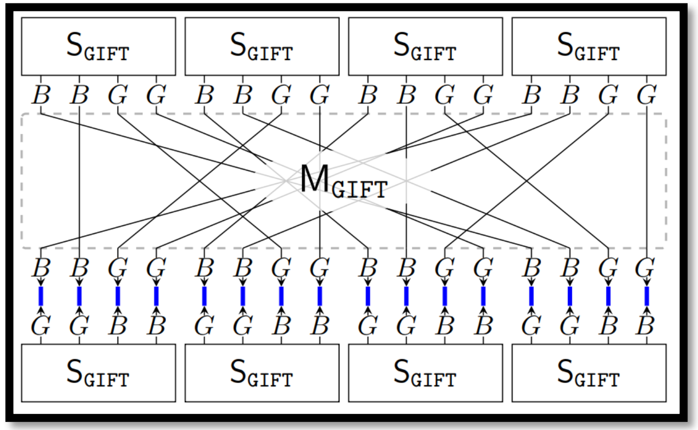
- **GIFT-variants, called *BOGI-based Ciphers*, by replacing the existing S-box and bit permutation.**
  - Considered S-boxes : 2,654,208
  - Considered bit perms : 55,296

  Totally, $2^{37.09}$ variants!

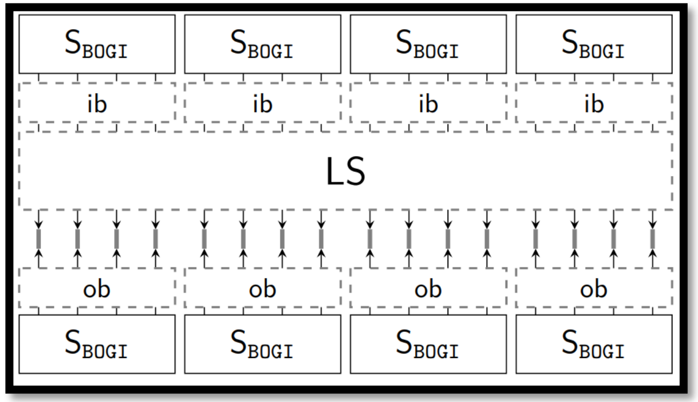
- **Deduce equivalent combinations and take their representatives.**
  - Representatives : $2^{37.09} \rightarrow 41{,}472$

- **Minimum required rounds to prevent effective trails for DC/LC.**
  - Each representation gives 16-,32-,64-,128-bit BOGI-based ciphers.

- **Search results show that there are GIFT-variants allowing fewer rounds than `GIFT`.**
  - To prevent effective differential and linear trails,
  - 2 rounds fewer than `GIFT-64`
  - 3 rounds fewer than `GIFT-128`



- Super-box of `GIFT`



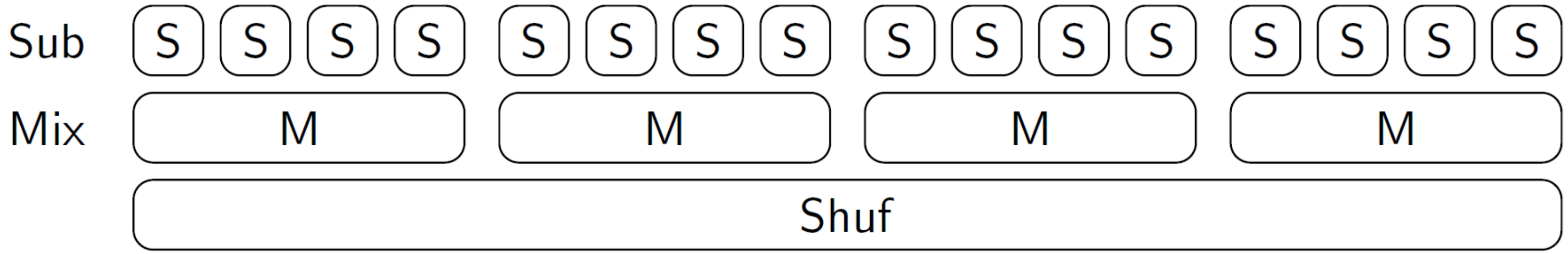
- S-box and Bit-perm of BOGI-based Cipher

AES-like Cipher

- **An AES-like cipher applies an <u>identical</u> round function R in all rounds.**

- **The AES-like round function R $=$ Shuf∘Mix∘Sub consists of**

  1. S-layer **Sub** : nm **<u>identical</u>** w-bit S-boxes S are applied concurrently.
  2. Mixing-layer **Mix** : n **<u>identical</u>** wm-bit matrix multiplications M are applied concurrently.
  3. Shuffle-layer **Shuf** : shuffles nm w-bit words. Thus, **Sub** ∘ **Shuf** $=$ **Shuf** ∘ **Sub**.

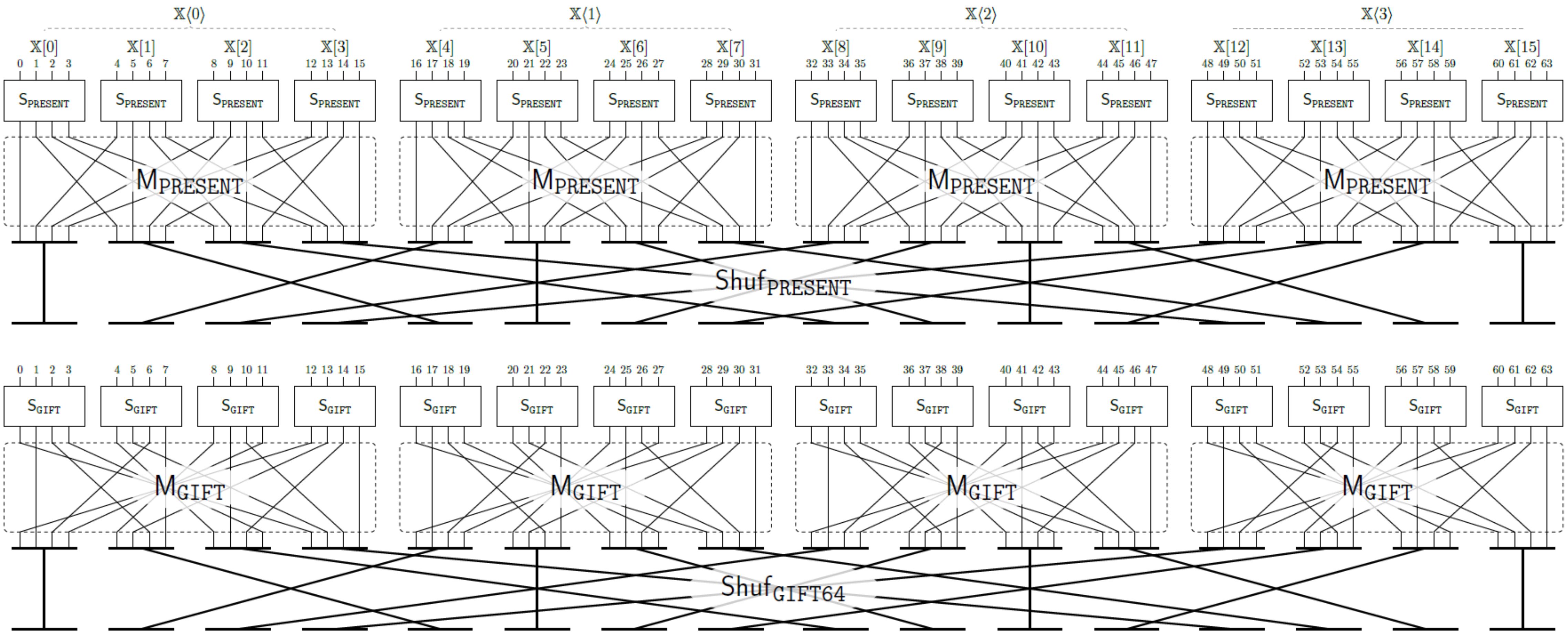


• AES-like Round Function

- **Depending on whether M is a bit permutation or not, we call R and the cipher as**
  - Bit permutation-based AES-like cipher : `PRESENT`, `GIFT`
  - Non-bit permutation-based AES-like cipher : `MIDORI`, `SKINNY`, `LED`, `CRAFT`

Examples – PRESENT, GIFT-64

- PRESENT, GIFT-64, and GIFT-128 are bit permutation-based AES-like ciphers.



Weight and Minimum Required Rounds for DC/LC Resistances

- **Weight**
  - Negative Logarithm of Probability of Differential Trail (or Squared Linear Correlation of Linear Trail)

$$\mathrm{W}(\mathbb{T}) = -\log_2 \prod_{i=1}^{R} \Pr\left[\mathbb{T}[i].\mathbb{X} \xrightarrow{\mathsf{R}} \mathbb{T}[i].\mathbb{X}'\right]$$

- Let n_b be the block size. In general, DC/LC can be mounted with a trail $\mathbb{T}$ such that

$$\mathrm{W}(\mathbb{T}) < n_b.$$

- We are interested in **the minimum rounds** when every trail does not satisfy the above condition.
- Therefore, if the R−round best (minimum) weight as

$$\boxed{\geq n_b,}$$

we can show that the non−existence of distinguisher for DC/LC in R rounds.

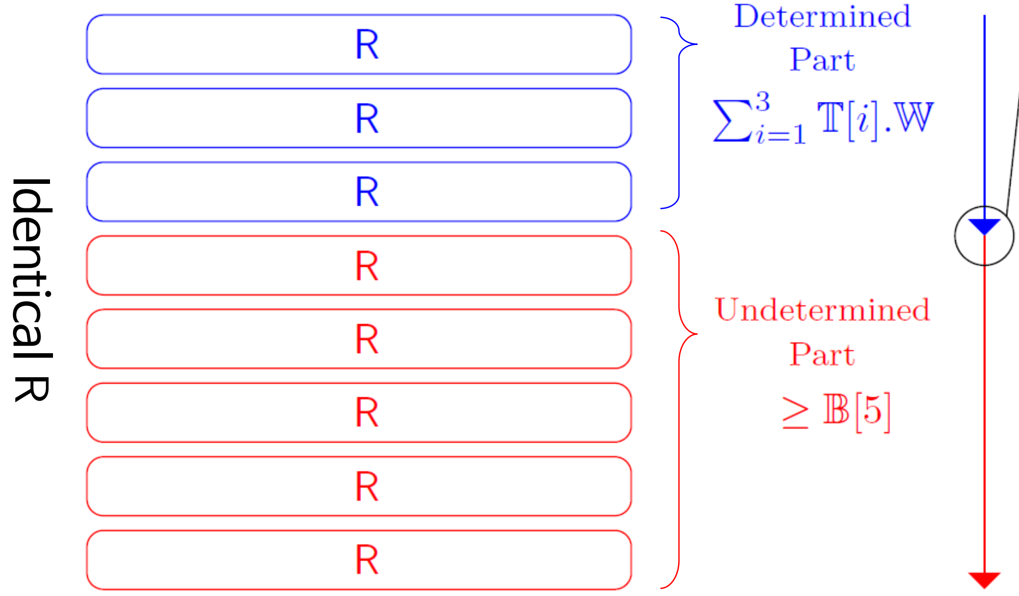
Matsui's Search Algorithm [Mat95]

- **Branch-and-bound Depth-First Search algorithm**
  - When we search for $(R \geq 2)$-**round best trails**,
    it traverses all the possible trails while checking the pruning condition with **the knowledge of**

    1 ~ $(R-1)$-round best weights $\mathbb{B}[1, \dots, R-1]$.
  - This requirement can be satisfied by reclusively applying the algorithm from 2-round.

- **Pruning Condition**
  - Main Speedup Factor

• Application to 8-round best differential trail search,



$\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \mathbb{B}[5] \leq \mathrm{W}(\mathbb{T})$

B_{set}: currently minimum weight.

- Pruning Condition

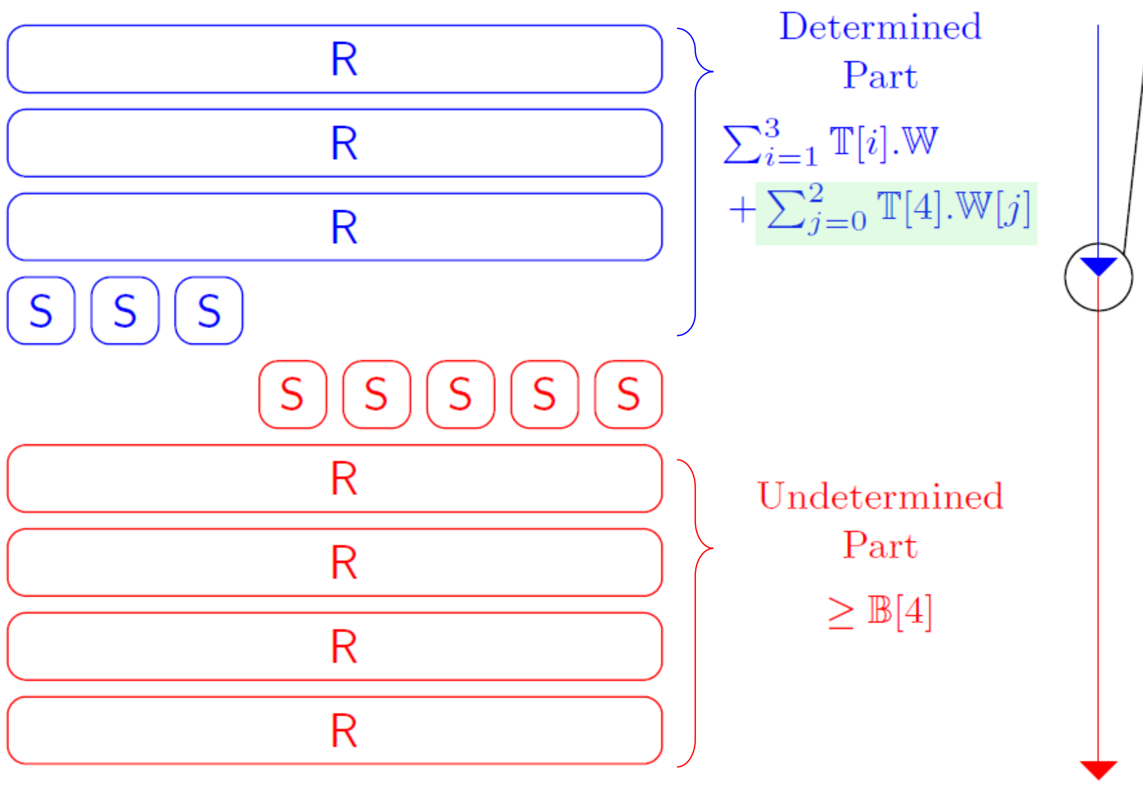
  $\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \mathbb{B}[5] > B_{set}$

  $\Rightarrow \mathrm{W}(\mathbb{T}) > B_{set}$

  $\therefore$ We don't need to go further with $\mathbb{T}[1,2,3]$.

Pruning Condition of Matsui's Search Algorithm

- Check the pruning condition S-box by S-box.



$\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{2} \mathbb{T}[4].\mathbb{W}[j] + \mathbb{B}[4] \leq \mathrm{W}(\mathbb{T})$

B_{set}: currently minimum weight.

- Pruning Condition

  $\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{2} \mathbb{T}[4].\mathbb{W}[j] + \mathbb{B}[4] > B_{set}$

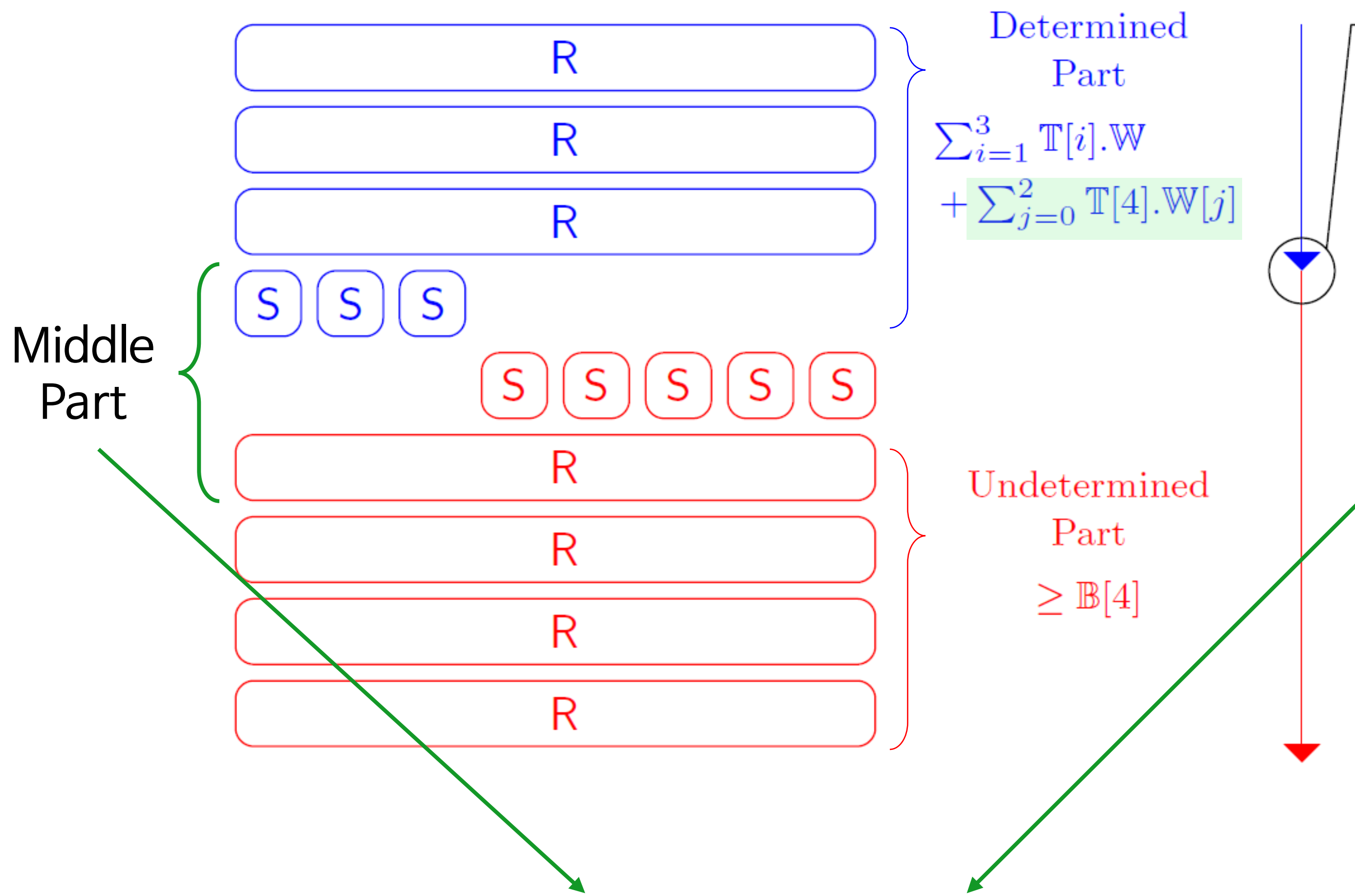
  $\Rightarrow \mathrm{W}(\mathbb{T}) > B_{set}$

  $\therefore$ We don't need to go further!

- Stricter pruning condition: Make the left side of the pruning condition bigger.

Pruning Condition of Matsui's Search Algorithm

- Check the pruning condition S-box by S-box.



$\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{2} \mathbb{T}[4].\mathbb{W}[j] + \mathbb{B}[4] \leq \mathrm{W}(\mathbb{T})$

B_{set}: currently minimum weight.

- Pruning Condition

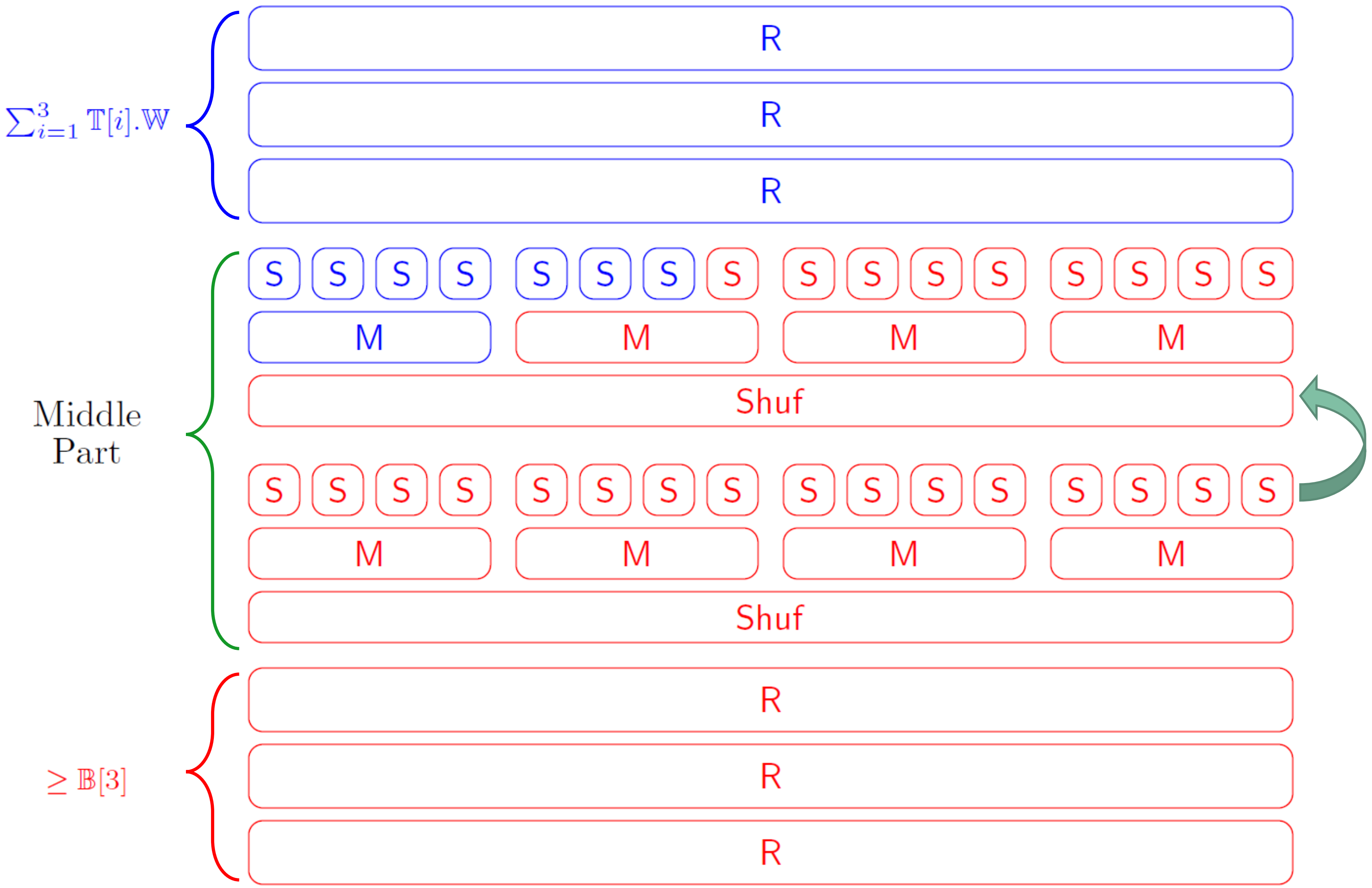
$\sum_{i=1}^{3} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{2} \mathbb{T}[4].\mathbb{W}[j] + \mathbb{B}[4] > B_{set}$

$\Rightarrow \mathrm{W}(\mathbb{T}) > B_{set}$

$\therefore$ We don't need to go further!

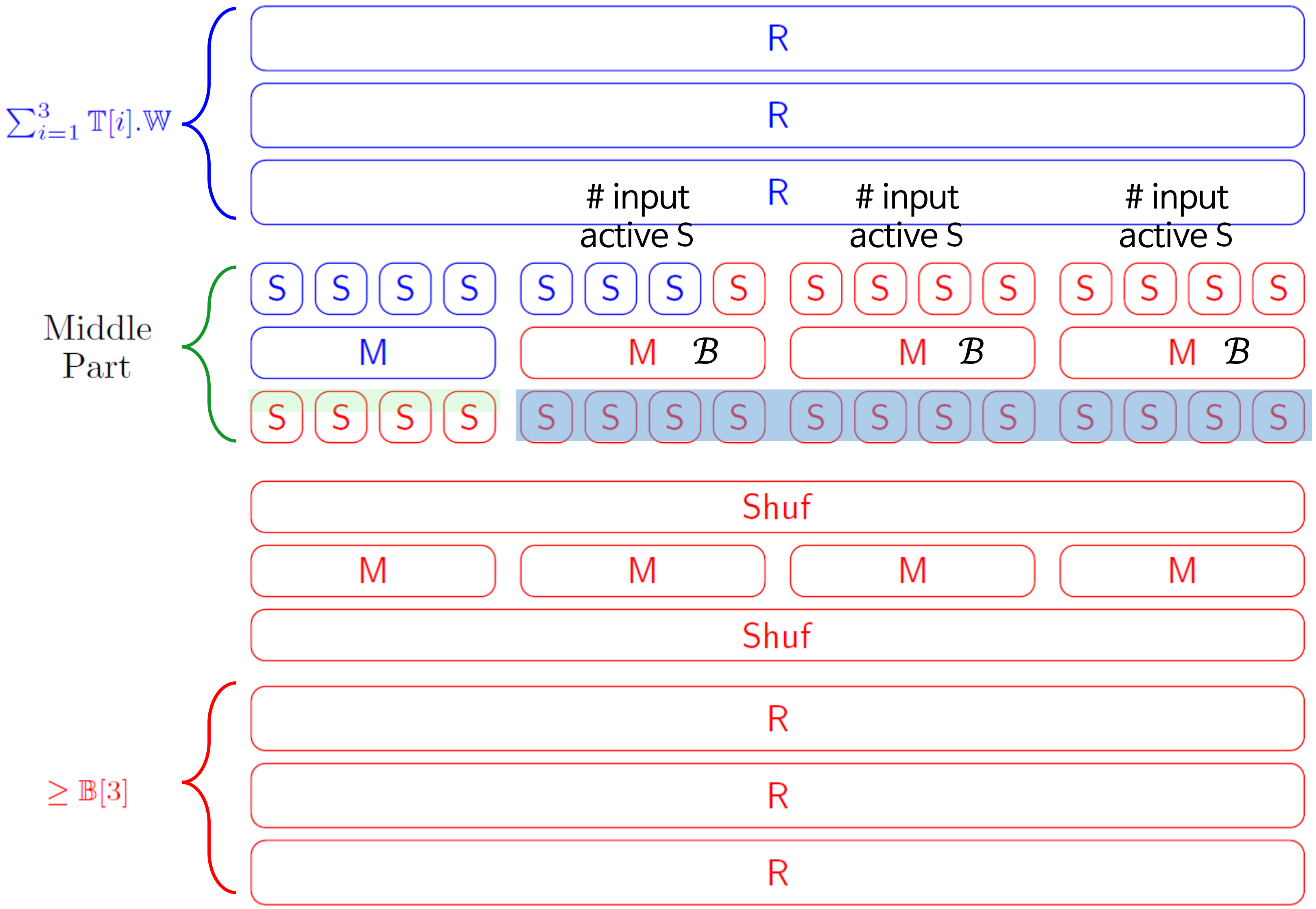
- Stricter pruning condition: Make the left side of the pruning condition bigger.

Strict Pruning Conditions for AES-like cipher

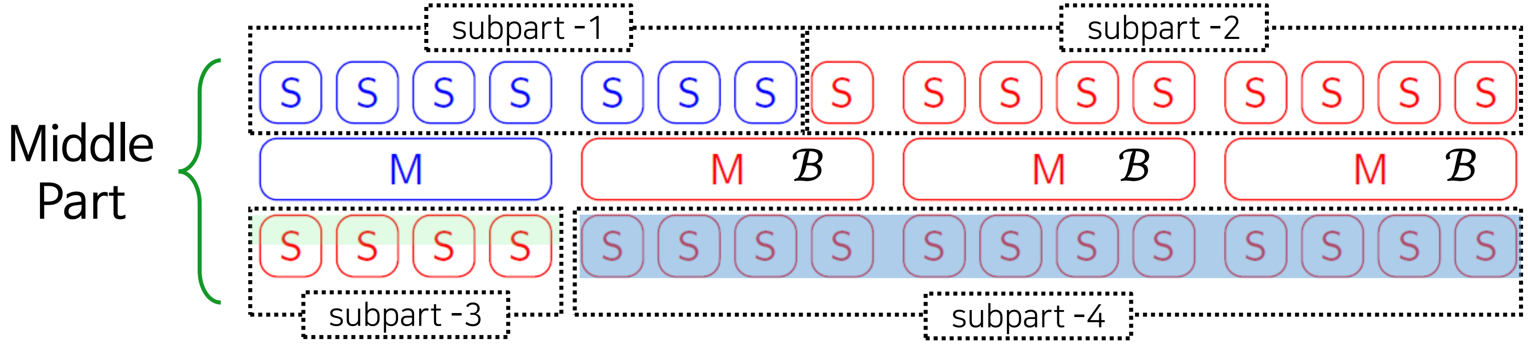


AES-like Structure
➔ **Shuf, Sub**
are Commutative

Strict Pruning Conditions for AES-like cipher



Four Subparts in the Middle Part



subpart -1

$$\sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j]$$

subpart -2

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

subpart -3

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

subpart -4

$$\sum_{k=n_{\mathsf{M}}}^{n-1} \phi\Big(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]\Big) \times \underline{W}$$

- **Non-bit Permutation-based AES-like Cipher**

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

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

$$+ \sum_{k=n_{\mathsf{M}}}^{n-1} \phi\Big(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]\Big) \times \underline{W}$$

$$+ \mathbb{B}[R-1-r] \le B_{set},$$

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

For Bit Permutation-based AES-like Cipher

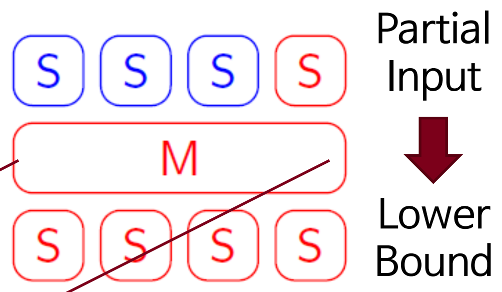
- When M is a bit permutation,

  the partial input can provide the (non−trivial > 1) lower bound for # output active S−boxes.

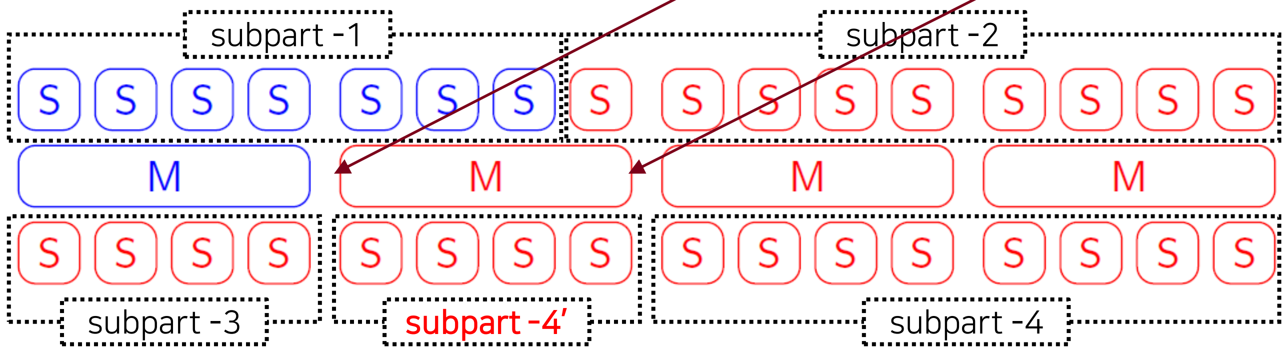
Lemma 1. *Let* Perm *be a bit permutation. For any* $0 \le c < mn$, *it is satisfied that*

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

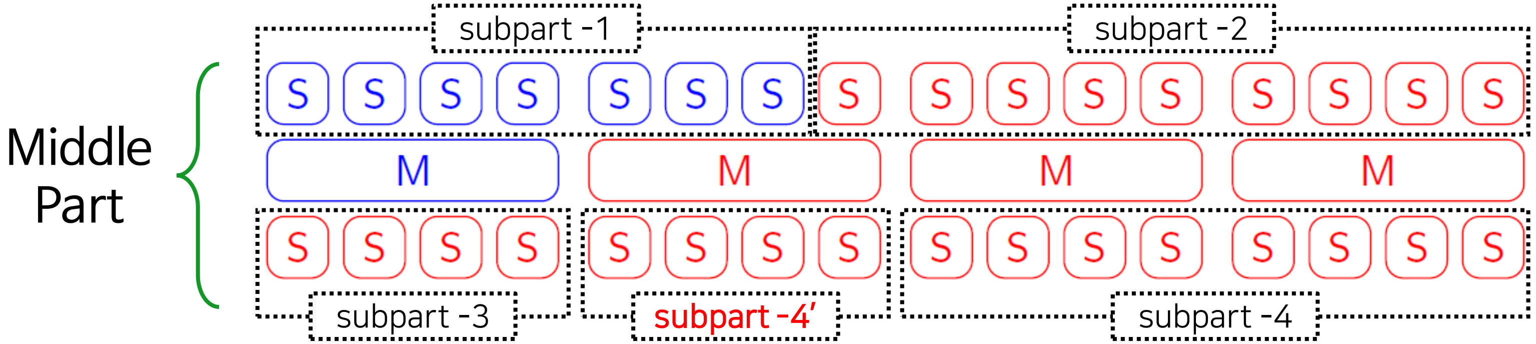
where $\mathbf{0}$ *denotes a zero-bit padding.*



- Thus, we can further consider subpart−4'.



For Bit Permutation-based AES-like Cipher



subpart -1

$$\sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j]$$

subpart -2

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

subpart -3

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

subpart -4'

$$\mathrm{ACT}\left(\mathsf{M}\big(\mathbb{T}[r].\mathbb{Y}\langle n_{\mathsf{M}}\rangle[0,\ldots,m_{\mathsf{M}}-1] \parallel \mathbf{0}\big)\right) \times \underline{W}$$

subpart -4

$$\sum_{k=n_{\mathsf{M}}+\phi(m_{\mathsf{M}})}^{n-1} \phi\Big(\sum_{j=0}^{m-1} \mathbb{T}[r].\mathbb{A}[mk+j]\Big) \times \underline{W} \text{ -}$$

- **Bit Permutation-based AES-like Cipher**

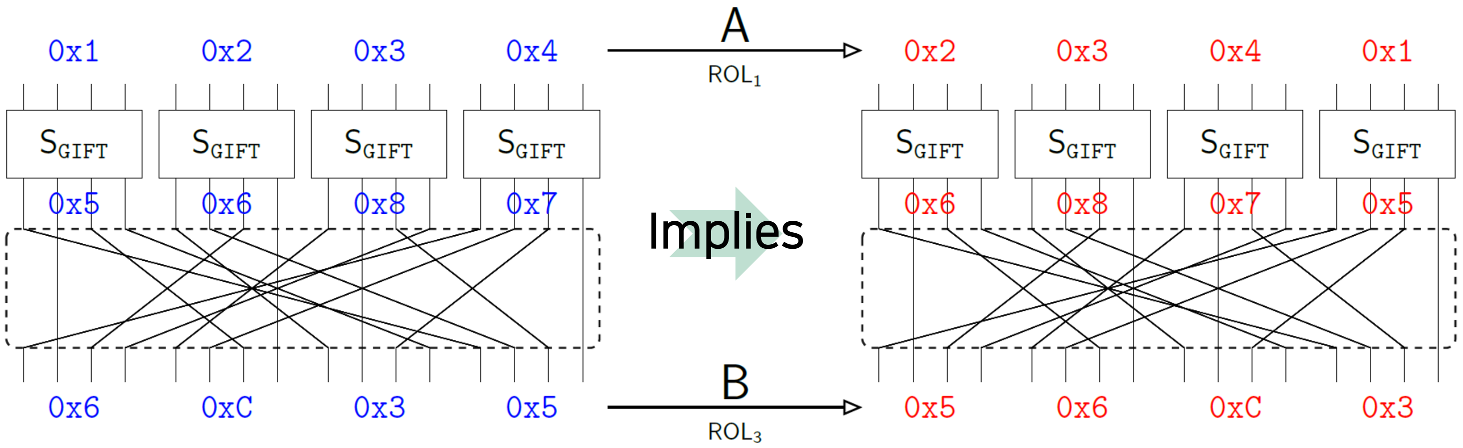
$$\begin{aligned}
&\sum_{i=1}^{r-1} \mathbb{T}[i].\mathbb{W} + \sum_{j=0}^{c} \mathbb{T}[r].\mathbb{W}[j] + \sum_{j=c+1}^{mn-1} \min_{Y} \mathrm{W}(\mathbb{T}[r].\mathbb{X}[j] \xrightarrow{\mathsf{S}} Y) \\
&+ \sum_{k=0}^{n_{\mathsf{M}}-1} \sum_{j=0}^{m-1} \min_{Y} \mathrm{W}\big(\mathsf{M}\big(\mathbb{T}[r].\mathbb{Y}\langle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

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

Employing Permutation Characteristics in Trail Search

- **Permutation Characteristic**
  - It was presented originally for invariant subspace attack [LMR15].
  - Although permutation characteristics can be used for the attack in a limited way due to key(constant)-additions, we can ignore them in trail search.
  - **It ensures that two different trails have the same weight. ➔ reduced search space**

- **Concept**



Same Weight

$$\mathrm{W}(\texttt{0x1} \xrightarrow{S_{GIFT}} \texttt{0x5}) + \mathrm{W}(\texttt{0x2} \xrightarrow{S_{GIFT}} \texttt{0x6}) + \mathrm{W}(\texttt{0x3} \xrightarrow{S_{GIFT}} \texttt{0x8}) + \mathrm{W}(\texttt{0x4} \xrightarrow{S_{GIFT}} \texttt{0x7})$$

$$\mathrm{W}(\texttt{0x2} \xrightarrow{S_{GIFT}} \texttt{0x6}) + \mathrm{W}(\texttt{0x3} \xrightarrow{S_{GIFT}} \texttt{0x8}) + \mathrm{W}(\texttt{0x4} \xrightarrow{S_{GIFT}} \texttt{0x7}) + \mathrm{W}(\texttt{0x1} \xrightarrow{S_{GIFT}} \texttt{0x5})$$

For any sequences of values, there must exist the corresponding word-wise permutated sequence.

∴ After trail search from an input active S-box pattern D, we don't have to consider trails from A(D).

Permutation Characteristics

- R-round permutation characteristic: a sequence of $\mathrm{A}_i \overset{\mathrm{R}}{\Rightarrow} \mathrm{B}_i$ such that

$$\mathrm{A}_1 \xRightarrow{\mathrm{R}} \mathrm{B}_1(=\mathrm{A}_2) \xRightarrow{\mathrm{R}} \mathrm{B}_2(=\mathrm{A}_3) \xRightarrow{\mathrm{R}} \cdots \xRightarrow{\mathrm{R}} \mathrm{B}_R.$$

- R-round permutation characteristic only covers $(t \leq R)$-round trail search.

- Iterative permutation characteristics are needed for the arbitrary round trail search.

$$\mathrm{A}_1 \xRightarrow{\mathrm{R}} \mathrm{B}_1(=\mathrm{A}_2) \xRightarrow{\mathrm{R}} \mathrm{B}_2(=\mathrm{A}_3) \xRightarrow{\mathrm{R}} \cdots \xRightarrow{\mathrm{R}} \mathrm{B}_R(=\mathrm{A}_1).$$

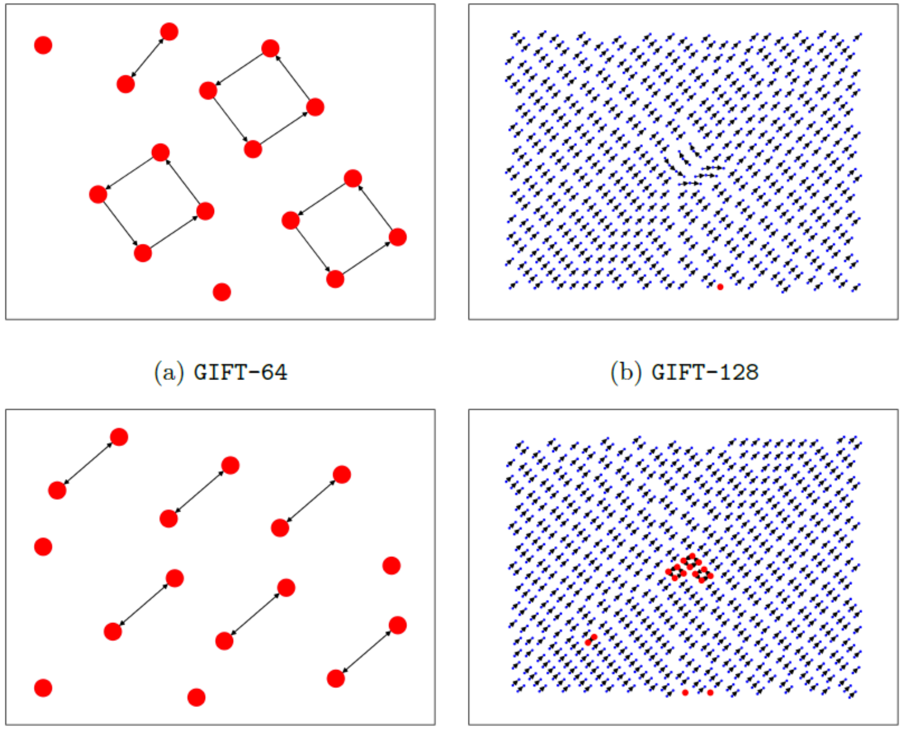
Obtaining Iterative Permutation Characteristics

- **Generate a directed graph and find the cyclic subgraphs**
  - After obtaining $\mathsf{A} \xrightarrow{\mathsf{M}} \mathsf{B}$, extend them as $\mathsf{A} \xrightarrow{\mathsf{Mix}} \mathsf{B}$ and $\mathsf{A} \xrightarrow{\mathsf{Perm}} \mathsf{B}$; $\mathrm{DWSE}(\mathsf{Perm})$.
  - Find cycles in the corresponding directed graph (vertex – A, B, edge - $\xrightarrow{\mathsf{R}}$).
  - All Permutations (vertices) in a cycle can reduce the considered inputs.

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

Input: Perm^* and Rounds $R \geq 2$
Output: $\mathcal{D}$

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



(a) GIFT-64 (b) GIFT-128 (c) AES (d) MIDORI-64

Iterative Permutation Characteristics of Considered Ciphers

- The number of word-wise permutations on iteratives permutation characteristics is up to 16.
- The number of input active S-box patterns decreases up to 15.77 factors.

Cipher	$\lvert\mathcal{A}tab\rvert$	$\lvert\text{DWSE}(\text{Perm}^*)\rvert$	$\lvert\overleftrightarrow{\text{DWSE}(\text{Perm}^*)}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert Opt\mathcal{A}tab\rvert$	$\lvert\mathcal{A}tab\rvert/\lvert Opt\mathcal{A}tab\rvert$
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

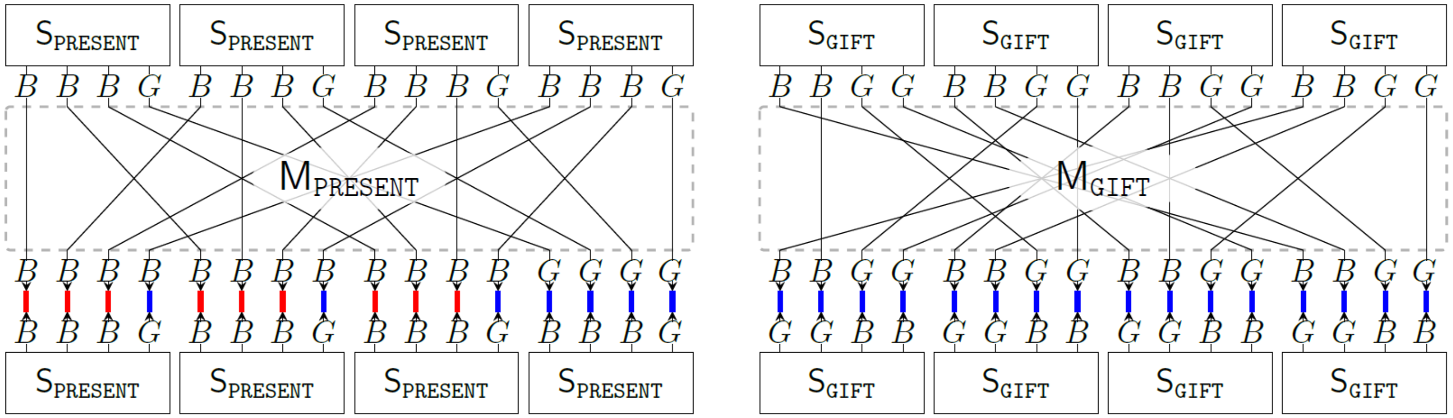
Performance Comparisons

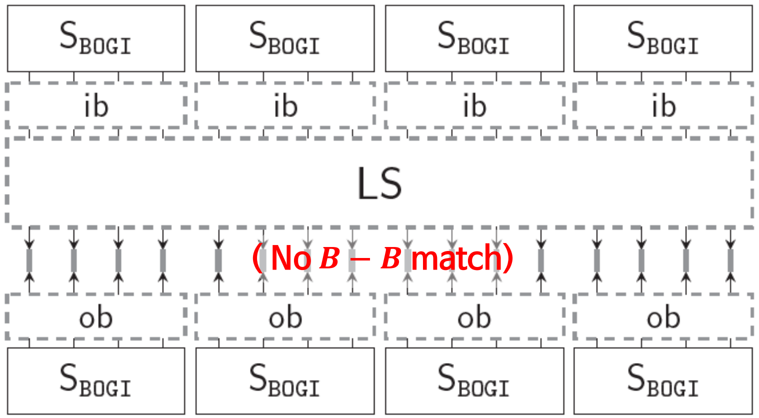
- Strategy - 1 : Accelerate up to 474 factors
- Strategy - 2 : Accelerate up to 10 factors
- Both : Accelerate up to 1904 factors

Cipher	$\frac{\lvert Atab \rvert}{\lvert OptAtab \rvert}$	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 ~ 31	Dif.	9.777 s	5.131 s		1.91	1	1.91
GIFT-64	15.77	2 ~ 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 ~ 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 ~ 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 ~ 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 ~ 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 ~ 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 ~ 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 ~ 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

BOGI Design

- **BOGI (Bad Output must go to Good Input) Design**
  - `GIFT` is based on this design.
  - The fundamental prevention of consecutive single active bit propagations over a trail.
  - **It requires a proper combination of S-box and bit permutation.**





- S-box and Bit-perm of BOGI-based Cipher

  - **Such combinations amount to 2,654,208 × 55,296 = $2^{37.09}$.**
    - S-box : 2,654,208
    - 16-bit permutation ({ib, ob} and {LS}) : 96×576 = 55,296

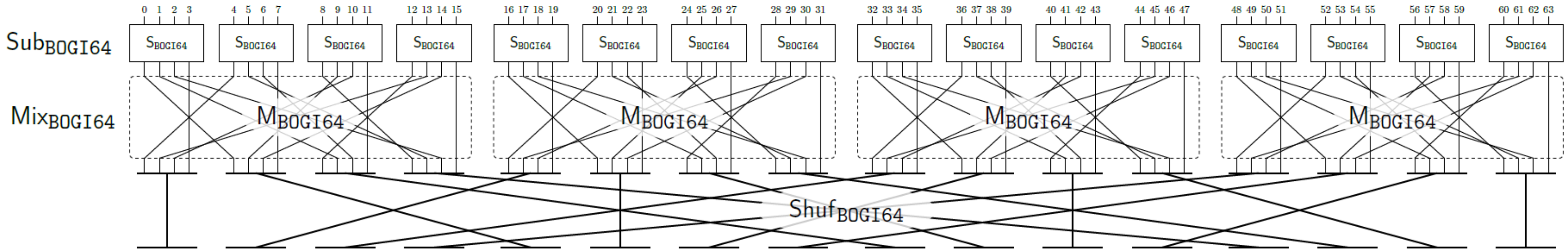
BOGI-based Cipher

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 $\sigma_{\mathtt{BOGI16}\cdot n}(j) = n \times (j \mod 4) + \lfloor \frac{j}{4} \rfloor$.

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 $2^{37.09}$.

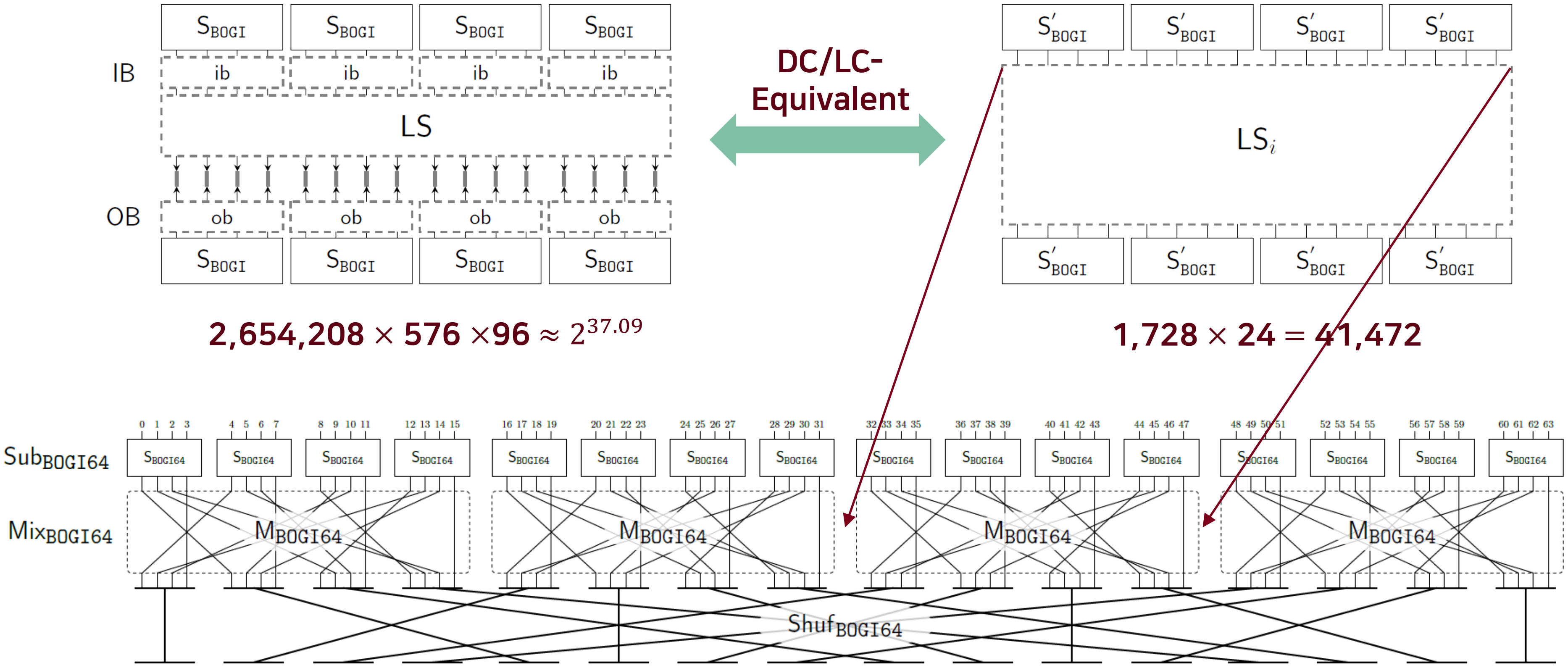
- **Each combination gives a (16$\cdot$ n)-bit BOGI-based cipher with a proper shuffle layer.**
- **Therefore, each version amounts to $2^{37.09}$.**



DC/LC-Equivalent BOGI-based Cipher

- **Deduce DC/LC-equivalence classes between super boxes ➔ # BOGI-based ciphers for each block size is same.**
  - DDT-Equivalence : |Sub| = 2,654,208 → 10,398
  - (LS)-Equivalence : |LS| = 576 → 24
  - (ib,S,ob)-Equivalence : |Sub| × |Mix| = |Sub| × |LS| = 1,728 × 24 = **41,472**

(We don't specify the shuffle layer)



Trail Search on BOGI-based Cipher

- **Strategy - 1 : Use the Pruning Condition for Bit Permutation-based AES-like Cipher**
- **Strategy - 2 : Obtaining Permutation Characteristics for Each Version**
  - `BOGI-64` : Up to 15.77 factor
  - `BOGI-128` : Up to 32 factor

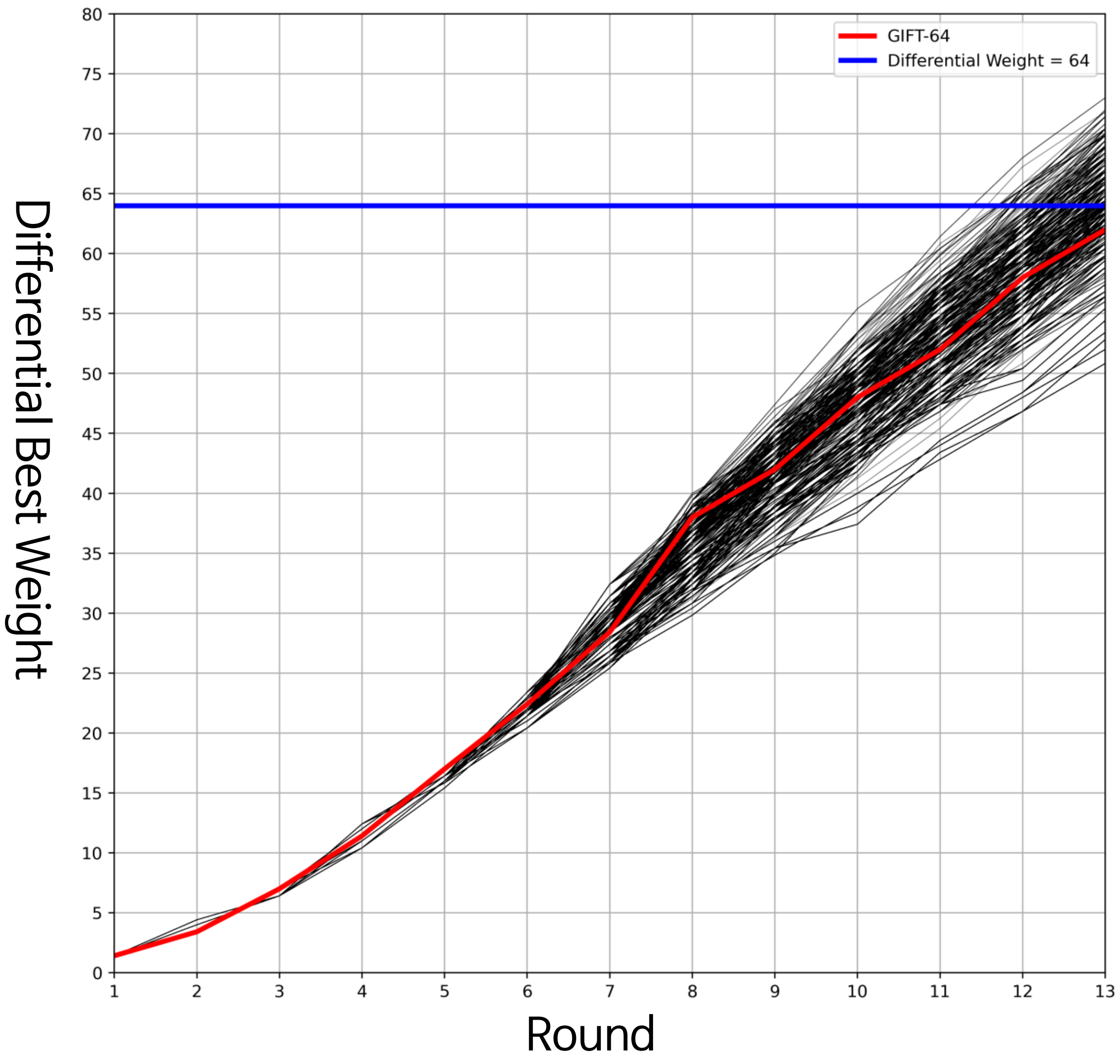
BOGI-16 · n	Combinations	Trail Type	Rounds	Min Elapsed	Avg Elapsed	Max Elapsed
BOGI-16	41,472	Differential Linear	2 ∼ 15	0.001 s <0.001 s	0.016 s 0.003 s	1.261 s 0.11 s
BOGI-32		Differential Linear	2 ∼ 15	0.001 s 0.001 s	0.072 s 0.03 s	4.622 s 1.229 s
BOGI-64		Differential Linear	2 ∼ 13	0.004 s 0.002 s	29.508 s 40.618 s	0.5 h 1.3 h
BOGI-128		Differential Linear	2 ∼ 11	0.011 s 0.004 s	1.5 h 0.9 h	450.9 h 156.4 h

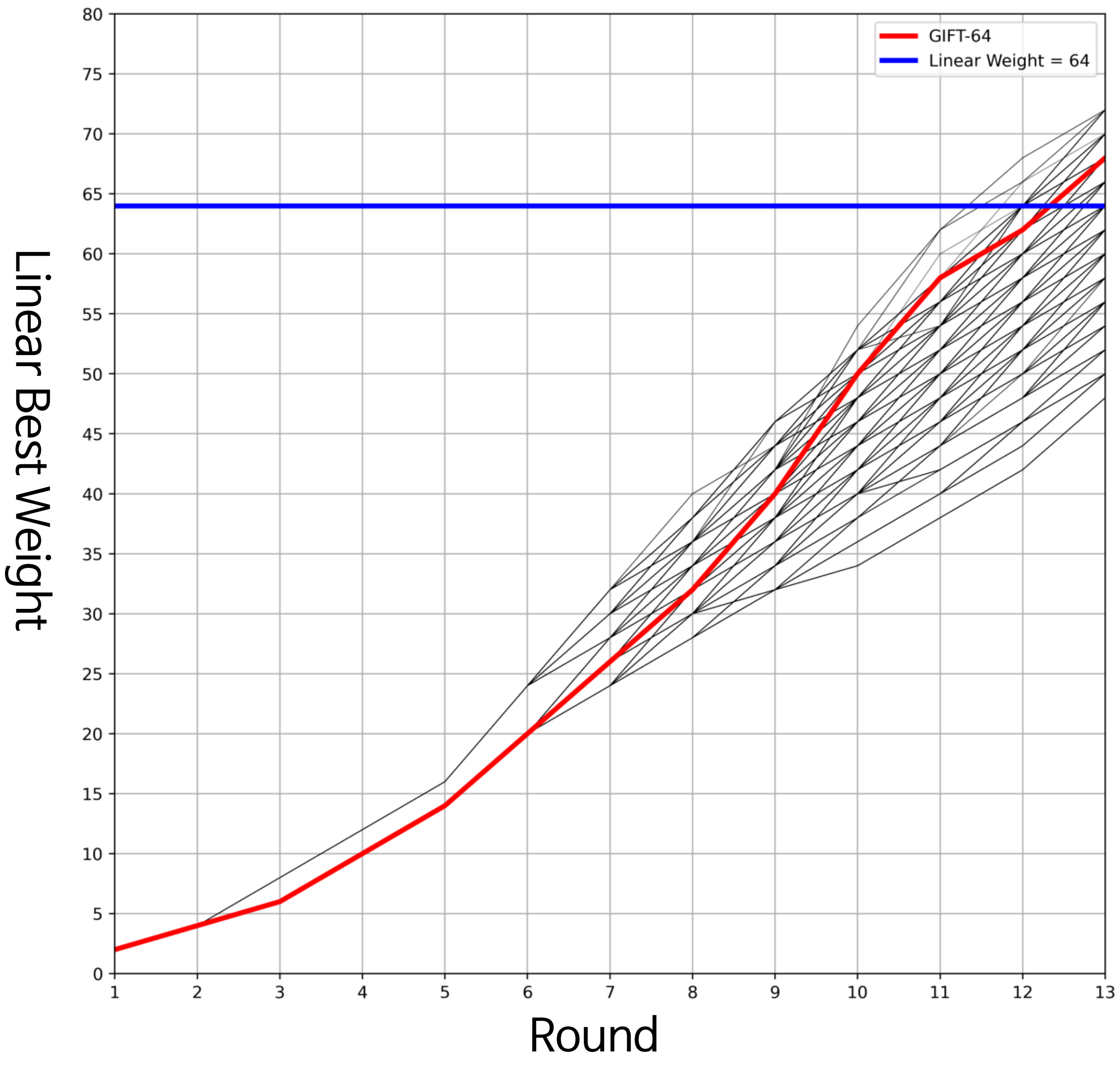
Trail Search on BOGI-based Cipher

- Strategy – 1 : Use the Pruning Condition for Bit Permutation-based AES-like Cipher
- Strategy – 2 : Obtaining Permutation Characteristics for Each Version
  - BOGI-64 : Up to 15.77 factor
  - BOGI-128 : Up to 32 factor
- There exist better combinations than GIFT-64 and GIFT-128 in terms of DC/LC-resistance.
  - Only replacing the existing bit permutation even allows fewer rounds.

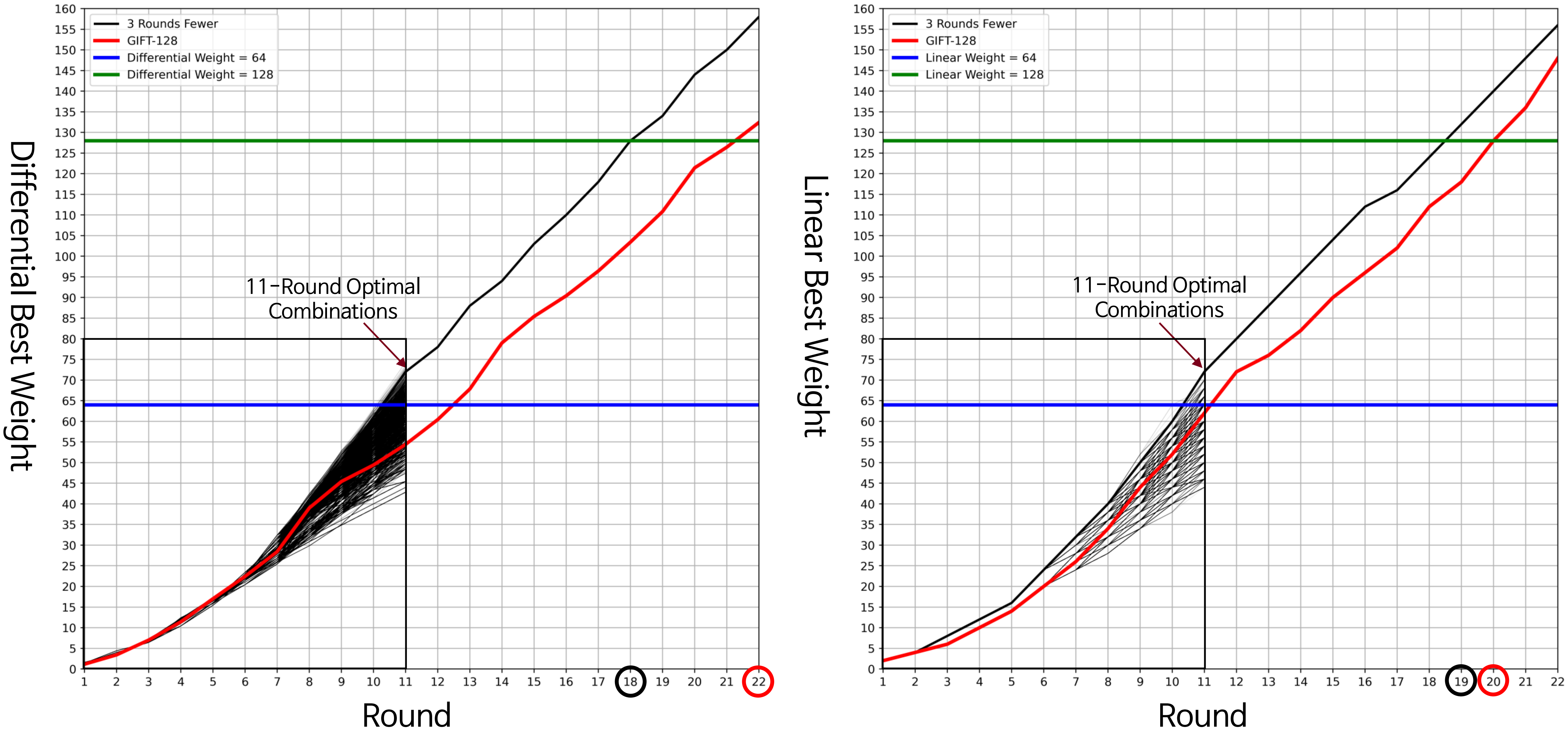
Block Size (b-bit)	Minimum Required Rounds to Prevent Efficient Trails for DC/LC		
	GIFT-b*	With Replacement of Bit Permutation	With Replacement of Bit Perm. and S-box
16-bit	6 rounds	6 rounds	5 rounds
32-bit	10 rounds	10 rounds	8 rounds
64-bit	14 rounds	13 rounds (1 Round)	12 rounds (2 Rounds Fewer)
128-bit	22 rounds	20 rounds (2 Round)	19 rounds** (3 Rounds Fewer)

Best Weights of 64-bit BOGI-based Ciphers





Best Weights of 128-bit BOGI-based Ciphers



Conclusion

- We attempted to optimize Matsui's Search Algorithm with two strategies.
- Moreover, taking advantage of the optimization,
  we obtain the first analysis results of best trail of full-round `GIFT-128` and
  investigate the most DC/LC-resistant BOGI-based ciphers.
- Our implementations, codes, and analysis results can be found publicly in

  `https://github.com/jeffgyeom/Best-Trail-Search-on-AES-Like-Ciphers`.

Q&A

Thanks

KOREA UNIVERSITY

Cryptographic Algorithm Laboratory