# BIZness: Bit Invariant Zero-Sum Property based on Division Trail

Shibam Ghosh[1], Anup Kumar Kundu[2], Mostafizar Rahman[3], and Dhiman Saha[4]

[1]Department of Computer Science, University Of Haifa
[2]Indian Statistical Institute, Kolkata
[3]University of Hyogo, Kobe, Japan
[4]de.ci.phe.red Lab, Department of Computer Science & Engineering
Indian Institute of Technology, Bhilai

Indocrypt
Chennai, India
December, 2024

# DiFA [KGSR23] The Source

- Bit invariance $\implies$ same output division property appears regardless of the position of active bits in the input division property.

## Results (5 rounds)

- For PRESENT, BIZ set: $\{0, 4, 8, 12\}$.
- For GIFT-64, BIZ set: $\{0, 4, 8, \cdots, 60\}$.
- For GIFT-128, $\exists$ 2 distinct sets. $|\text{BIZ}| = 80$.

| Primitive | round | # sets | #BIZ |
|---|---|---|---|
| PRESENT-80/128 | 4 | 1 | 64 |
| | 5 | 1 | 4 |
| GIFT-64 | 4 | 1 | 64 |
| | 5 | | 16 |
| GIFT-128 | 4 | 1 | 128 |
| | 5 | 2 | 80 |

# Background

## Definition (Bit-Based Division Property [Tod15])

A multi-set $X \subseteq \mathbb{F}_2^n$ is said to have the bit-based division property $\mathcal{D}_{\mathbb{K}}^n$ for some set of $n$-dimensional vectors $\mathbb{K}$, if it fulfills the following conditions:

$$\bigoplus_{\boldsymbol{x} \in X} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} \text{unknown} & : \text{if there is } \boldsymbol{k} \in \mathbb{K} \text{ s.t } \boldsymbol{u} \succeq \boldsymbol{k} \\ 0 & : \text{otherwise} \end{cases}$$

# Background

## Definition (Bit-Based Division Property [Tod15])

A multi-set $X \subseteq \mathbb{F}_2^n$ is said to have the bit-based division property $\mathcal{D}_{\mathbb{K}}^n$ for some set of $n$-dimensional vectors $\mathbb{K}$, if it fulfills the following conditions:

$$\bigoplus_{\boldsymbol{x} \in X} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} \text{unknown} & : \text{if there is } \boldsymbol{k} \in \mathbb{K} \text{ s.t } \boldsymbol{u} \succeq \boldsymbol{k} \\ 0 & : \text{otherwise} \end{cases}$$

## Definition (Balanced Bit [Tod15])

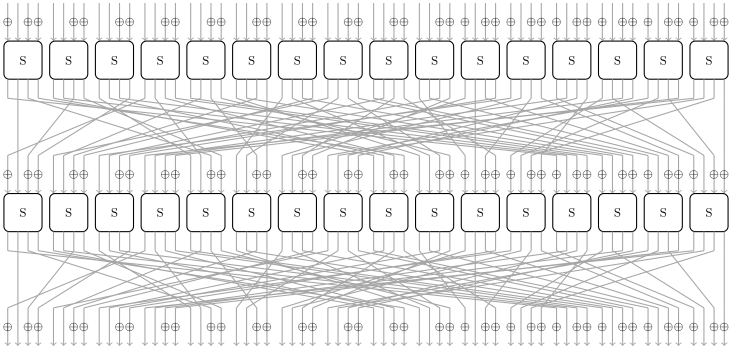
Let $Y \subseteq \mathbb{F}_2^n$ be a multi-set of vectors. A bit position $0 \leq i < n$ is called balanced bit position if $\bigoplus_{y \in Y} y_i = 0$

## Zero-sum property

If for any boolean function $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$, $\exists\ I \subseteq \mathbb{F}_2^n$ s.t.

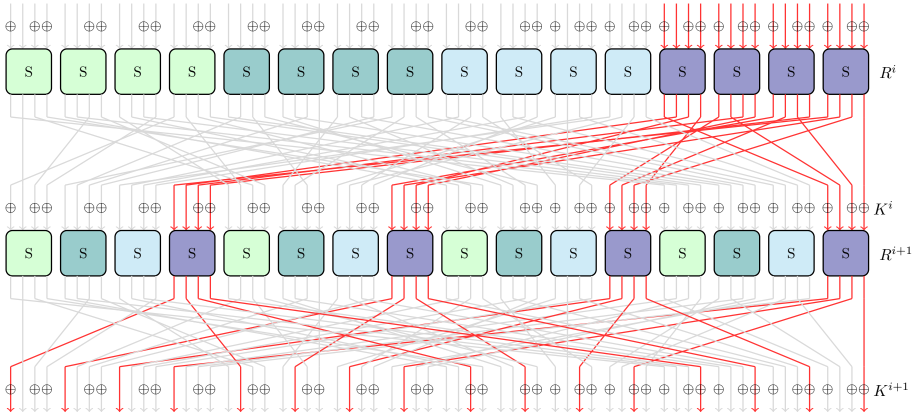
$$\bigoplus_{x \in I} x = \bigoplus_{x \in I} f(x) = 0$$

# GIFT [BPP+17] Recall




- Two versions GIFT-64 and GIFT-128 with state size 64 and 128 bits respectively.
- Consists of 28 and 40 rounds respectively.
- SubCells, Bit Permutation, Addition of round keys in each of the round function.
- SBox has branch number 2.
- Introduces the BOGI permutation (bad outputs goes to good inputs).

# QR-Structure of GIFT Permutation




- Property of two consequitive rounds.
- Permutation within 16-bits.
- Maps the output bits of SBoxes from a quotient group to the input bits of SBoxes in the corresponding remainder group.

# Our Contribution

- We develop a framework for appearing bit invariance zerosum property.
- The property manifests due to both the Sbox and the permutation layer.
- We give the explanation using Algebraic Normal Form (ANF).
- The charecterization of BIZ is done by combining different SBoxes with GIFT and PRESENT permutation layer.
- The property remains unchanged after replacing the GIFT BOGI permutaion with any 16-bit BOGI permutation.

# Division Trail Induced Linear Structures - DiviLS

# DiviLS

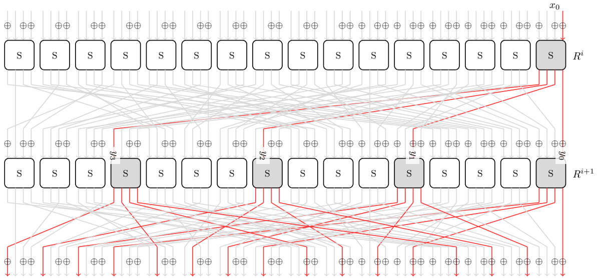
- Its a configuration of the input set for a division property that limits the maximum algebraic degree of the output-sum to one.

# DiviLS

- Its a configuration of the input set for a division property that limits the maximum algebraic degree of the output-sum to one.

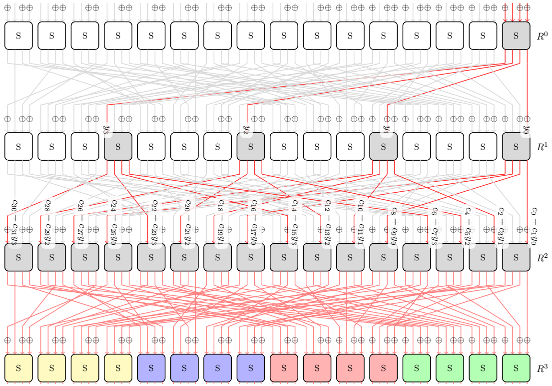
## Observations

- **Sbox-DiviLS:** One input bit of an SBox has DiviLS $\implies$ every SBox output exhibit a DiviLS.
- **QR-DiviLS:** One SBox of a quotient group has a DiviLS $\implies$ each of the four SBoxes in corresponding remainder group has a DiviLS.



# 2.5-Round DiviLS in GIFT

- For the input set corresponding to a division property limited to a nibble at $R^i$, the DiviLS property holds up to $R^{i+2}$.




- This holds for bit permutation layer of PRESENT also.

# BIZ in GIFT-64

# BIZ in GIFT-64

- BIZ = Bit Invariance Zerosum

**Property [KGSR23]**

Irrespective of the position of the active nibble in the input division set of GIFT-64, after 5 rounds, ∃ specific 16 bits, one in every nibble, that are balanced.

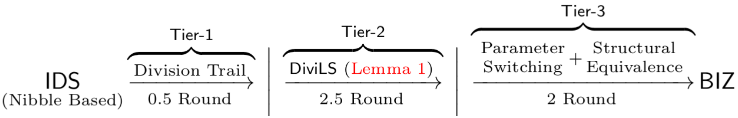
# BIZ in GIFT-64

- BIZ = Bit Invariance Zerosum

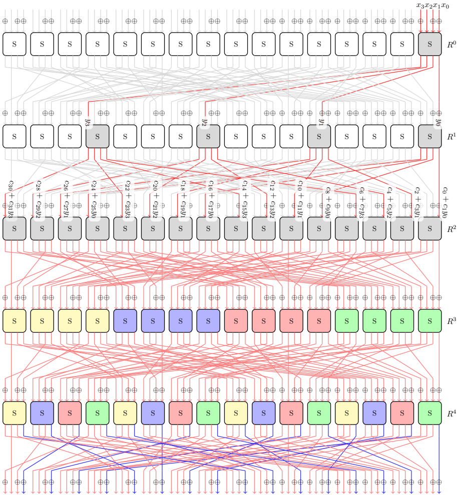
**Property [KGSR23]**

Irrespective of the position of the active nibble in the input division set of GIFT-64, after 5 rounds, $\exists$ specific 16 bits, one in every nibble, that are balanced.

- Try to give a theoretical argument behind this Property.
- Use a three-tier approach:



# BIZ in GIFT-64




- **Tier-2:**
  - Suppose $\{x_3^0, x_2^0, x_1^0, x_0^0\}$ := input bit variables at $R^0$.
  - Consider the variables $\nu = \{y_3^0, y_2^0, y_1^0, y_0^0\}$, after the permutation layer of $R_0$.
  - At input of $R^3$, variables are linear over $\nu$ (2.5 round DiviLS of GIFT-64).
  - Any SBox before $R^3$ can have degree at most 3.

# BIZ in GIFT-64

- **Parameter Switching ($\mathcal{P}_f(\mathbf{p} \to \mathbf{q})$):**
  - Change the parameters $\mathbf{p} \to \mathbf{q}$ in a boolean funtion $f(\mathbf{p}, \mathbf{x})$ by $p_i$ with $q_i$, s.t. $\mathbf{p} \neq \mathbf{q}$ in the ANF of $f$.
  - **Example:** Consider,

$$f(c_2, c_1, c_0, d_2, d_1, d_0, x_3, x_2, x_1, x_0) = d_0x_0 + d_2d_1x_2x_1 + c_2d_1x_1 + c_1c_0d_0x_3,$$

with $\mathbf{c}$, $\mathbf{d}$ are parameters.
Take, $g = \mathcal{P}_f(\mathbf{c} \to \mathbf{a}, \mathbf{d} \to \mathbf{b})$, then,

$$g(a_2, a_1, a_0, b_2, b_1, b_0, x_3, x_2, x_1, x_0) = b_0x_0 + b_2b_1x_2x_1 + a_2b_1x_1 + a_1a_0b_0x_3.$$

# BIZ in GIFT-64

- **Parameter Switching ($\mathcal{P}_f(\mathbf{p} \to \mathbf{q})$):**
  - Change the parameters **p** → **q** in a boolean funtion $f(\mathbf{p}, \mathbf{x})$ by $p_i$ with $q_i$, s.t. **p** ≠ **q** in the ANF of $f$.
  - **Example:** Consider,

$$f(c_2, c_1, c_0, d_2, d_1, d_0, x_3, x_2, x_1, x_0) = d_0x_0 + d_2d_1x_2x_1 + c_2d_1x_1 + c_1c_0d_0x_3,$$

    with $\mathbf{c}$, $\mathbf{d}$ are parameters.
    Take, $g = \mathcal{P}_f(\mathbf{c} \to \mathbf{a}, \mathbf{d} \to \mathbf{b})$, then,

$$g(a_2, a_1, a_0, b_2, b_1, b_0, x_3, x_2, x_1, x_0) = b_0x_0 + b_2b_1x_2x_1 + a_2b_1x_1 + a_1a_0b_0x_3.$$

- **Structural Equivalence ($f \equiv g$):**
  - For parameterized Boolean functions $f(\mathbf{p}, \mathbf{x})$ and $g(\mathbf{q}, \mathbf{x})$,

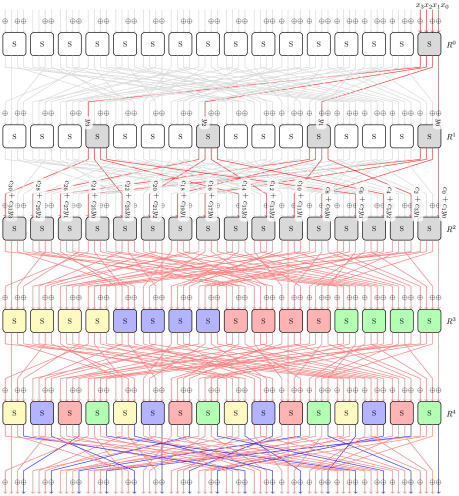
$$\text{structurally equivalent} \iff \mathcal{P}_f(\mathbf{p} \to \mathbf{q}) = g.$$

  - **Example:** Consider the following functions:

$$f(c_1, c_0, x_3, x_2, x_1, x_0) = x_0 + x_2x_1 + c_0x_1 + c_1x_2$$
$$g(d_1, d_0, x_3, x_2, x_1, x_0) = x_0 + x_2x_1 + d_0x_1 + d_1x_2.$$

    Here, $\mathcal{P}_f(\mathbf{c} \to \mathbf{d}) = g(\mathbf{d}, \mathbf{x})$ and so $f \equiv g$.

# BIZ in GIFT-64

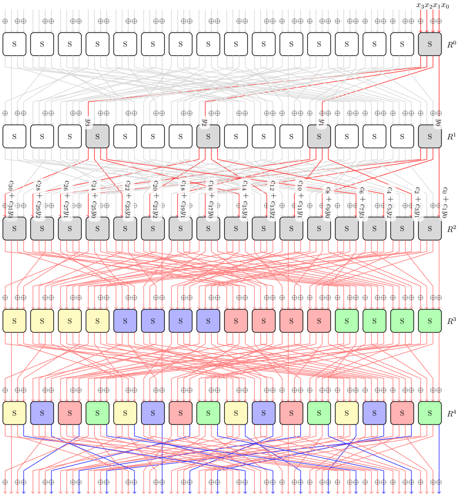



- **Tier-3:**
  - From Tier-2, before $R^3$, all the inputs to the SBoxes in a specific quotient group becomes structurally equivalent.
  - The outputs of four SBoxes in each quotient group are also structurally equivalent to each other.

$$\begin{cases} y^3_{4(4j+0)+0} \equiv y^3_{4(4j+1)+0} \equiv y^3_{4(4j+2)+0} \equiv y^3_{4(4j+3)+0} \\ y^3_{4(4j+0)+1} \equiv y^3_{4(4j+1)+1} \equiv y^3_{4(4j+2)+1} \equiv y^3_{4(4j+3)+1} \\ y^3_{4(4j+0)+2} \equiv y^3_{4(4j+1)+2} \equiv y^3_{4(4j+2)+2} \equiv y^3_{4(4j+3)+2} \\ y^3_{4(4j+0)+3} \equiv y^3_{4(4j+1)+3} \equiv y^3_{4(4j+2)+3} \equiv y^3_{4(4j+3)+3} \end{cases}$$

  - Apply a parameter switching to get structurally equivalent inputs to the SBoxes of $R^4$.

# BIZ in GIFT-64




- **Tier-3:**
  - Consider $\mathbf{u} = (u_0, u_1, u_2, u_3)$, the four output bits of any SBox in $R^4$ i.e. Then,

$$(a_0^i y_0 + a_1^i, b_0^i y_0 + b_1^i,$$
$$c_0^i y_0 + c_1^i, \ \ d_0^i y_0 + d_1^i)$$
$$\downarrow (SBox \circ \mathcal{F} \circ SBox)$$
$$(u_0, u_1, u_2, u_3)$$

where,

$$\mathcal{F} = \begin{cases} \mathcal{P}_{y^3_{4(4j+0)+0}}(\mathbf{a}^i \to \mathbf{a}^i, \mathbf{b}^i \to \mathbf{b}^i, \mathbf{c}^i \to \mathbf{c}^i, \mathbf{d}^i \to \mathbf{d}^i) \\ \mathcal{P}_{y^3_{4(4j+1)+1}}(\mathbf{a}^i \to \mathbf{a}^{i+1}, \mathbf{b}^i \to \mathbf{b}^{i+1}, \mathbf{c}^i \to \mathbf{c}^{i+1}, \mathbf{d}^i \to \mathbf{d}^{i+1}) \\ \mathcal{P}_{y^3_{4(4j+2)+2}}(\mathbf{a}^i \to \mathbf{a}^{i+2}, \mathbf{b}^i \to \mathbf{b}^{i+2}, \mathbf{c}^i \to \mathbf{c}^{i+2}, \mathbf{d}^i \to \mathbf{d}^{i+2}) \\ \mathcal{P}_{y^3_{4(4j+3)+3}}(\mathbf{a}^i \to \mathbf{a}^{i+3}, \mathbf{b}^i \to \mathbf{b}^{i+3}, \mathbf{c}^i \to \mathbf{c}^{i+3}, \mathbf{d}^i \to \mathbf{d}^{i+3}) \end{cases}$$

# BIZ in GIFT-64

- **Tier-1:**
  - Til now, we have considered the propagation of the variables $y_i^0$ for $i \in \{0, \cdots, 3\}$.
  - Actual input variable $:= (x_3, x_2, x_1, x_0)$ before $R^0$.
  - Use the division property table to check if any of the output bits contain the monomial $x_3x_2x_1x_0$ or not.

**Lemma**

If from the input property 1111, $\exists$ only 1111 in the division property table of the SBox and the $i$-th output bit does not contain the monomial $y_3y_2y_1y_0$, then after 5 rounds the $i$-th output bit of GIFT is balanced.

# BIZ in GIFT-64

- **Tier-1:**
  - Til now, we have considered the propagation of the variables $y_i^0$ for $i \in \{0, \cdots, 3\}$.
  - Actual input variable $:= (x_3, x_2, x_1, x_0)$ before $R^0$.
  - Use the division property table to check if any of the output bits contain the monomial $x_3x_2x_1x_0$ or not.

**Lemma**

If from the input property 1111, $\exists$ only 1111 in the division property table of the SBox and the $i$-th output bit does not contain the monomial $y_3y_2y_1y_0$, then after 5 rounds the $i$-th output bit of GIFT is balanced.

- Algorithm for checking the conditions:

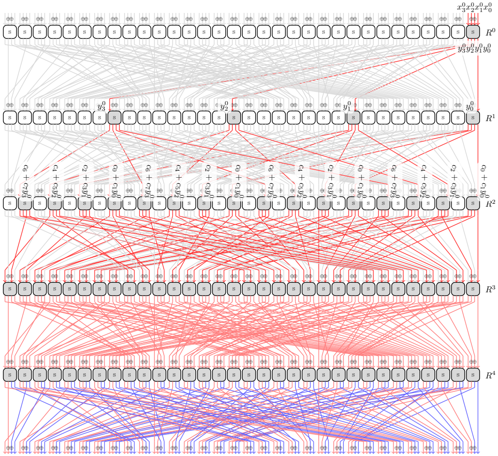
**Algorithm 1** CHECK BIZ FOR GIFT-64

Input: An SBox $\mathcal{S}$, the division property table $DPT_{\mathcal{S}}$ where $(b, a) \in DPT_{\mathcal{S}}$ $\iff$ $(b, a)$ is a valid division trail for $\mathcal{S}$.
Output: coordinates with BIZ property

1: $(u_0, u_1, u_2, u_3) = SBox \circ \mathcal{F} \circ SBox(a_0^i y_0 + a_1^i, b_0^i y_0 + b_1^i, c_0^i y_0 + c_1^i, d_0^i y_0 + d_1^i)$
2: $B = \phi$
3: **if** $(1111, a) \notin DPT_{\mathcal{S}}$ such that $a \neq 1111$ **then**
4:     **for** $i = 0$ to $4$ **do**
5:         **if** $deg(u_i) < 4$ **then**
6:             $B = B \cup \{i\}$
7: **return** B

# BIZ in GIFT-128 and PRESENT

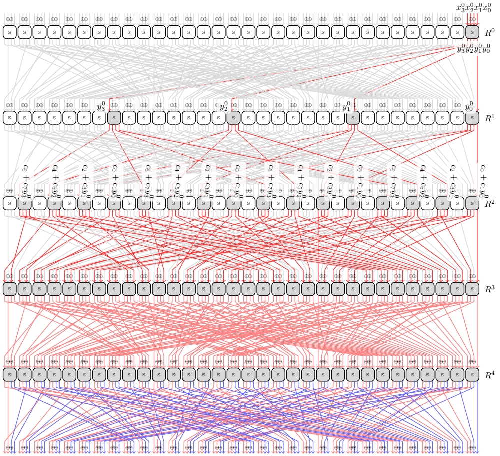
# BIZ in GIFT-128




## Result [KGSR23]

- The two sets of bit invariance balanced bits for GIFT-128.
- Two set appears depending upon the initial active nibble, say $\mathcal{A}$ and $\mathcal{B}$.
- $|\mathcal{A} \cap \mathcal{B}| = 64$

# BIZ in GIFT-128



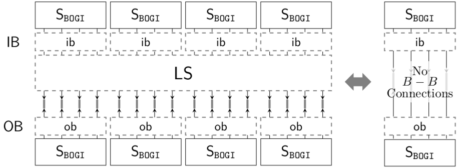

## Result [KGSR23]

- The two sets of bit invariance balanced bits for GIFT-128.
- Two set appears depending upon the initial active nibble, say $\mathcal{A}$ and $\mathcal{B}$.
- $|\mathcal{A} \cap \mathcal{B}| = 64$

- $\mathcal{A}$, $\mathcal{B}$ and BIZ set can be obtained by previously discussed similar procedure.
- For PRESENT also, the same technique can be used to show the BIZ property for 5 rounds of the cipher.

# BIZ in other Permutation Layer [KHSH22]

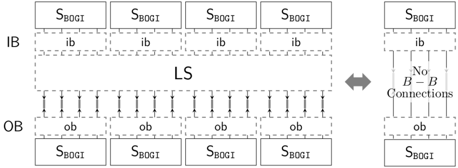
- Proposed a clasification of BOGI-based ciphers.
- Decomposed, BOGI $\equiv OB \circ LS \circ IB$, where $IB$ and $OB$ are 4-bit permutations and $LS$ is 16-bit permutation.




- For BOGI permutation, $LS$ satisfy the following conditions:
  - The four input bits of LS in each SBox position go to four distinct SBox positions.
  - Each bit order of the four input bits of LS in each SBox position is invariant on the four output bits of LS in an SBox position.
- Total BOGI-based cipher can be categorized into 24 classes.

# BIZ in other Permutation Layer [KHSH22]

- Proposed a clasification of BOGI-based ciphers.
- Decomposed, BOGI $\equiv OB \circ LS \circ IB$, where $IB$ and $OB$ are 4-bit permutations and $LS$ is 16-bit permutation.




- For BOGI permutation, $LS$ satisfy the following conditions:
  - The four input bits of LS in each SBox position go to four distinct SBox positions.
  - Each bit order of the four input bits of LS in each SBox position is invariant on the four output bits of LS in an SBox position.
- Total BOGI-based cipher can be categorized into 24 classes.

## Our Result

- We replace the bit permutation of GIFT-64 with each of the 24 permutations.
- The BIZ property remains invariant and appears in the exact same bit positions for each permutation after 5 rounds.

# BIZ after varying SBox

| SBox | PRESENT Permutation | | | GIFT-64 Permutation | | | GIFT-128 Permutation | | |
|---|---|---|---|---|---|---|---|---|---|
| | Round | #bb | BIZ | Round | #bb | BIZ | Round | #bb | BIZ |
| Craft | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 16 | ✓ | 5 | 0 | ✗ | 5 | 32 | ✗ |
| Midori-128 | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 0 | ✗ | 5 | 0 | ✗ | 5 | 0 | ✗ |
| Prince | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 0 | ✗ | 5 | 0 | ✗ | 5 | 0 | ✗ |
| Rectangle | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 16 | ✓ | 5 | 16 | ✓ | 5 | 80 | ✓ |
| Small AES | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 0 | ✗ | 5 | 0 | ✗ | 5 | 0 | ✗ |
| PRESENT | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 4 | ✓ | 5 | 0 | ✗ | 5 | 0 | ✗ |
| GIFT | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 16 | ✓ | 5 | 16 | ✓ | 5 | 80 | ✓ |
| Piccolo | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 32 | ✓ | 5 | 32 | ✓ | 5 | 64 | ✓ |
| Warp | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 16 | ✓ | 5 | 0 | ✗ | 5 | 32 | ✗ |
| Default | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 48 | ✓ | 5 | 32 | ✓ | 5 | 128 | ✓ |
| Baksheesh | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 4 | 64 | ✓ | 5 | 64 | ✓ | 5 | 128 | ✓ |
| ULBC | 4 | 64 | ✓ | 4 | 64 | ✓ | 4 | 128 | ✓ |
| | 5 | 24 | ✓ | 5 | 16 | ✓ | 5 | 64 | ✓ |

# Conclusion

- We know the reason for appearing the BIZ property.
- The designers should consider this before proposing a new GIFT-like structure.


thank you
danke
gracias
merci
obrigado
bedankt
dziękuję
teşekkür ederim
go raibh maith agat
ngiyabonga
謝謝
спасибо
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
mochchakkeram
tapadh leat
dank je
takk
мерси
감사합니다
ありがとう
Баярлалаа
рахмат
хвала
mersi
barka
tack
spas
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
asante
manana
obrigada
tenki
murakoze
chokrane
mamnun
djiere dieuf
tau
дякую
trugarez
shukriya
merce
diolch
dhanyavadagalu
tanemirt
rahmet
xiexie
ευχαριστώ
taiku
sulpay
gracies
gratias ago
chnorakaloutioun
najis tuke
kam sah hamnida
rahmat
didi madloba
sagolun
mèsi
dėkuji
sobodi
enkosi
bayarlalaa
gracie
hvala
maururu
koszonom
nanni
nandri
kiitos
dankie
dhanyavad
faafetai lava
vinaka
спасибі
kia ora
welalin
dankon
akun
aciu

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 *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

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 *CHES*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.

Joan Daemen, Lars Knudsen, and Vincent Rijmen.
The block cipher SQUARE.
In *Proceedings of FSE 1997*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 1997.

Anup Kumar Kundu, Shibam Ghosh, Dhiman Saha, and Mostafizar Rahman.
Divide and rule: Difa - division property based fault attacks on PRESENT and GIFT.
In Mehdi Tibouchi and Xiaofeng Wang, editors, *Applied Cryptography and Network Security - 21st International Conference, ACNS 2023, Kyoto, Japan, June 19-22,*

*2023, Proceedings, Part I*, volume 13905 of *Lecture Notes in Computer Science*, pages 89–116. Springer, 2023.

Seonggyeom Kim, Deukjo Hong, Jaechul Sung, and Seokhie Hong.
Accelerating the best trail search on aes-like ciphers.
*IACR Trans. Symmetric Cryptol.*, 2022(2):201–252, 2022.

Lars Knudsen and David Wagner.
Integral cryptanalysis.
In *Proceedings of FSE 2002*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer.

Yosuke Todo and Masakatu Morii.
Bit-based division property and application to Simon family.
In *FSE*, volume 9783 of *Lecture Notes in Computer Science*, pages 357–377. Springer, 2016.

Yosuke Todo.
Structural evaluation by generalized integral property.
In *Advances in Cryptology – EUROCRYPT*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314. Springer, 2015.

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 *Advances in Cryptology – ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 648–678, 2016.