Out of Non-linearity: Search Impossible Differentials by the Bitwise Characteristic Matrix

Yunxiao Yang[1], Xuan Shen[2(✉)], and Bing Sun[1,3]

[1] College of Liberal Arts and Sciences, National University of Defense Technology, Changsha, China
yyx23@live.com, happy_come@163.com

[2] College of Information and Communication, National University of Defense Technology, Wuhan, China
shenxuan_08@163.com

[3] Hunan Engineering Research Center of Commercial Cryptography Theory and Technology Innovation, Changsha, China

Abstract. In this paper, we propose the $\mathcal{M}$-method which uses the bitwise characteristic matrix to search impossible differentials. $\mathcal{M}$-method exploits not only the linear components but also partial information of non-linear components. According to the principle of miss-in-the-middle, we construct two different types of contradiction to search the impossible differentials with limited time and memory complexity by calculating $\mathcal{M}_{en}^{r_1}$ and $\mathcal{M}_{de}^{r_2}$ which represent r_1 rounds encryption and r_2 rounds decryption, respectively. Compared with the previous methods, our technique is comprehensible and fast especially for large block size.

As a result, we find the 7-round impossible differentials of GIFT-128, the 5-round impossible differentials of PRIDE, and the 4-round impossible differentials of Pyjamask-96. For GIFT-64, PRESENT, RECTANGLE which are well-analyzed by MILP-method or SAT-method, we construct new impossible differentials. Moreover, the efficiency of our method will not be influenced by the block size, which makes us find the new 5-round impossible differentials of the 320-bit permutation of ASCON.

Keywords: Block cipher · Characteristic matrix · Impossible differential cryptanalysis

1 Introduction

The block cipher is of great importance in the field of cryptology. When designing a block cipher, designers always obey the *diffusion and confusion* principle

This study was funded by the National Natural Science Foundation of China (Grant Nos. 61772545 and 62002370), Scientific Research Plan of National University of Defense Technology (No. ZK21-36).

The original version of this chapter was revised: this chapter contained mistakes. This has been corrected. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-93206-0_24

R. Deng et al. (Eds.): ISPEC 2021, LNCS 13107, pp. 69–89, 2021.
https://doi.org/10.1007/978-3-030-93206-0_6

and guarantee them by iterating the round function which contains linear and non-linear layers. The SPN structure and Feistel structure with SP-type round functions are examples of the design principle. When it comes to lightweight block ciphers, such as PRESENT [4], GIFT [2], and PRIDE [1], the designers tend to use bitwise operations like bit permutation or cyclic shift rather than multiplication with the matrix over the finite field and to use 4-bit S-boxes rather than 8-bit S-boxes for efficient implementation. The diffusion of a bitwise permutation is much slower than the matrix multiplication such as the MDS matrix used in AES, and the confusion is weaker when the size of the S-box is smaller. Moreover, in some S-boxes such as the S-box of PRESENT and GIFT, certain bits of the output difference are determinate if the input difference is fixed with specific bits. In [20], Tezcan named the bits as undisturbed points which can simplify the differential cryptanalysis and the impossible differential cryptanalysis.

As one of the most powerful cryptanalysis techniques, impossible differential cryptanalysis was independently proposed by Knudsen [12] and Biham et al. [3]. Unlike the differential cryptanalysis aiming at finding high-probability differentials, the impossible differential cryptanalysis is to find the differential $(\varDelta_{in}, \varDelta_{out})$, where the input difference $\varDelta_{in}$ can never propagate to $\varDelta_{out}$. Impossible differential cryptanalysis usually has two phases, the first one is to find the impossible differentials covering as many rounds as possible; the second one is to filter the wrong keys by extending the distinguisher several rounds. Therefore, constructing the impossible differentials is the key step that determines the number of attacking rounds.

To search longer impossible differentials efficiently, the automatic searching tools have been developed rapidly in the last decades. In 2003, Kim et al. [11] published the first automatic searching tool named $\mathcal{U}$-method for impossible differentials. The $\mathcal{U}$-method classifies every byte of a block into the $\mathcal{U}$-set and constructs contradictions in the middle state. In 2009, Luo et al. [14] improved the $\mathcal{U}$-method and proposed the UID-method. In 2012, Wu et al. [21] further exploited the properties of linear operations by solving the system of linear equations. Although the searching ability is improved rapidly compared with manual derivation, the above automatic tools cannot make use of the details of S-boxes and they can only cover the word-oriented block ciphers. To alleviate the above limitations, researchers have turned their attention to modeling the impossible differential searching into Mixed Integer Linear Programming (MILP) problem or Boolean Satisfiability Problem (SAT), which have been used maturely for optimization problems. In 2016, Cui et al. [5] extended the applications of MILP-method on searching impossible differentials. At EUROCRYPT 2017, Sasaki et al. [15] presented another MILP-based automatic tool for impossible differentials searching which can cover more structures. At ASIACRYPT 2020, Hu et al. [10] proposed a new automatic search tool based on SAT-method to model the impossible polytopic transitions and key dependent transitions which were not considered by the previous automatic tools. In summary, the more information of a block cipher that the automatic search tools can absorb, the more impossible differentials can be found and in some cases the more rounds can be covered.

In general, the methods based on MILP or SAT can cover more structures and find longer distinguishers, but there are also limitations. One of them is that the computation complexity will increase rapidly while the block size is large. In [15], the authors claimed that even if the size of the S-box is small, it is computationally hard to evaluate a large block size of more than 256 bits. And the automatic tools cannot tell why the differentials are impossible. Because of the heuristic algorithms used in MILP/SAT-solvers, the process of solving is nearly a black box. Therefore, the authors of the automatic tools always manually verify some of the results.

While the above research found impossible differentials by automatic tools, there is another line of research that determines the impossible differentials by theoretical proof. At CRYPTO 2015, Sun et al. [19] proved that without considering the details of S-boxes, the $\mathcal{WW}$-method [21] can find all word-oriented impossible differentials of both Feistel structure with SP-type round functions and SPN structure. Moreover, at EUROCRYPT 2016, Sun et al. [18] utilized the characteristic matrix to prove the upper bound of truncated impossible differentials for SPN structure. Following the line of research, Shen et al. [17] considered the details of the S-boxes and found longer impossible differentials for Russian standard block cipher Kuznyechik [8] and the permutation of PHOTON [9]. After that, at ISPEC 2017, Shen et al. [16] further studied the matrix representation of a block cipher and proposed a more precise matrix representation named diffusion matrix. By utilizing the diffusion matrix, they constructed impossible differentials of SIMON-like block ciphers.

Our Contributions. Along the research line of Sun et al. [18] and Shen et al. [16], we propose the $\mathcal{M}$-method which uses the bitwise characteristic matrix to search impossible differentials for more block ciphers while Shen et al. [16] only considered the SIMON-like block ciphers.

We first calculate the matrix representation of one round encryption which is denoted as $\mathcal{M}_{en}$. The matrix $\mathcal{M}_{en}$ contains not only the information of linear components but also some information of the S-box. After iterating the $\mathcal{M}_{en}$ for r times, i.e. $\mathcal{M}_{en}^{r}$, we get the matrix representation of the r-round encryption. By multiplying the difference with the corresponding matrix, we get the middle state of the block cipher. The decryption is the same. Then we can construct impossible differentials according to the principle of miss-in-the-middle. Moreover, we propose the indirect contradiction where we extend the rounds of impossible differentials by looking up the Difference Distribution Table (DDT) of the S-box. The main results of our technique for searching impossible differentials are listed in Table 1.

Compared with the MILP-based and SAT-based tools, our technique has the following advantages:

(1) Model Large States: Our method models an n-bit block cipher by an $n \times n$ matrix and the only computation is matrix multiplication, which is easy for a laptop. Therefore, our method can function with nearly no compromises no matter how large the block size is. We apply our method to the 320-bit

Table 1. Main results

Block ciphers	Search tool	Rounds	Ref.
GIFT-64	SAT	6	[2]
	MILP	6	[10]
	$\mathcal{M}$-method	6	Ours
GIFT-128	$\mathcal{M}$-method	**7**	Ours
PRIDE	$\mathcal{M}$-method	**5**	Ours
Pyjamask-96	Previous	3	[13]
	$\mathcal{M}$-method	**4**	Ours
Pyjamask-128	$\mathcal{M}$-method	3	Ours
PRESENT	MILP	6	[5]
	$\mathcal{M}$-method	6	Ours
ASCON	Previous	5	[7]
	$\mathcal{M}$-method	5	Ours
RECTANGLE	MILP	8	[15]
	$\mathcal{M}$-method	8	Ours

permutation of ASCON and find new impossible differentials. We also find the 7-round impossible differentials of GIFT-128.

(2) Comprehensible Contradictions: We construct contradictions by determining the middle states with the characteristic matrices. So we are clear about the type and the position of every contradiction. Utilizing the linear correlations between different contradictions, the $\mathcal{M}$-method can construct new impossible differentials for GIFT-64, PRESENT, ASCON, RECTANGLE.

(3) Negligible time and memory complexity: After the bitwise characteristic matrix is determinate, we can construct contradictions by combining the column vectors in the matrix. And we only consider specific columns with determinate entries, which is much less than the search range. During the computation, the only thing we have to store in the memory is several $n \times n$ matrices.

Paper Outline. In Sect. 2, we introduce necessary preliminaries. In Sect. 3, we introduce the bitwise characteristic matrix and demonstrate the mechanism of our searching tool for impossible differentials. In Sect. 4, we apply our technique to some block ciphers. In Sect. 5, we conclude this paper and put forward some future works. And the necessary supplemental material is given in the Appendixes.

2 Preliminaries

2.1 Notation

The notation in this paper is listed in Table 2.

Table 2. Notation

$\mathbb{F}_{2^n}^*$	All non-zero elements in $\mathbb{F}_{2^n}$
e_i	A vector with only the i-th bit being 1, others being 0
$\#(I)$	The number of elements in set I
$\oplus$	Bitwise XOR
$\bigoplus_{i\in\{0,1,2\}} x_i$	$x_0 \oplus x_1 \oplus x_2$
$\alpha[i]$	The i-th bit of α
$f^n(x)$	$\underbrace{f \circ f \circ \cdots \circ f}_{n}(x)$
$\mathcal{M}_F$	The bitwise characteristic matrix of $F(x)$
$\mathcal{M}_{ij}$	The element of $\mathcal{M}$ located at the i-row and the j-th column
$\alpha \xrightarrow{F_1 F_2} \beta$	$\beta = F_2 \circ F_1(\alpha)$
$\alpha \xleftarrow{F_1 F_2} \beta$	$\alpha = F_1 \circ F_2(\beta)$

2.2 The Boolean Function

The n-variable boolean function is a function maps $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $f_0, f_1, \dots f_{m-1}$ be n-variable boolean functions, so the vectorial boolean function maps $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is defined as:

$$F(x) = (f_0(x), f_1(x), \dots, f_{m-1}(x)).$$

For any block cipher with a block size of n bits, we can treat it as a vectorial boolean function that maps $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. ANF (Algebraic Normal Form) is one of the representations for a boolean function.

Let $x \in \mathbb{F}_2^n$, the ANF of a n-variable boolean function is as follows:

$$f(x) = \bigoplus_{I\in\mathcal{P}(N)} a_I(\prod_{i\in I} x_i),$$

$\mathcal{P}(N)$ is the power set of $N = \{0, 1, \dots, n-1\}$, $a_I \in \mathbb{F}_2$. Note that all vectors in this paper are column vectors if not specified.

3 Searching the Impossible Differentials by Bitwise Characteristic Matrix

3.1 Description of Bitwise Characteristic Matrix

The definition of bitwise characteristic matrix can be obtained from the aspect of boolean function. Let E be an n-bit block cipher, the input and output of one round function are denoted as $x = (x_0, x_1, \dots, x_{n-1})$ and $y = (y_0, y_1, \dots, y_{n-1})$ respectively. The bitwise characteristic matrix is defined as follows:

Definition 1. *For a given block cipher E, the ANF of y_i is*

$$y_i = \bigoplus_{I \in \mathcal{P}(N)} a_I(\prod_{k \in I} x_k).$$

Concerning the correlation between the x_j and y_i, the above ANF can be expanded to

$$y_i = p(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1})x_j \oplus q(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1}),$$

$p(\cdot)$ and $q(\cdot)$ are $(n-1)$-variable boolean functions independent of x_j.

The bitwise characteristic matrix of E is denoted as $\mathcal{M}$, $\mathcal{M}_{ij}$ is defined as:

$$\mathcal{M}_{ij} = \begin{cases} 0, & p(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1}) = 0 \\ 1, & p(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1}) = 1 \\ ?, & p(x_0, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{n-1}) \neq 0, 1 \end{cases}$$

the 0 and 1 of $\mathcal{M}$ are defined over $\mathbb{F}_2$ which are called determined points. $\mathcal{M}_{ij} = 0$ means x_j is independent of y_i; $\mathcal{M}_{ij} = 1$ means when x_j changes, y_i must change; $\mathcal{M}_{ij} =?$ means when x_j changes, we can not tell whether y_i changes. When all x_j and y_i are analyzed according to the above process, the bitwise characteristic matrix $\mathcal{M}$ of E can be obtained.

To explain the operation between the bitwise characteristic matrices, a 4-bit S-box is constructed, and the ANF is as follows

$$\begin{cases} y_0 = x_0, \\ y_1 = x_0 \oplus x_1, \\ y_2 = x_0 \oplus x_1 x_2, \\ y_3 = 1 \oplus x_2 \oplus x_3. \end{cases}$$

The bitwise characteristic matrix of the S-box is:

$$\mathcal{M}_{\mathsf{S}} = \begin{pmatrix} 1\,0\,0\,0 \\ 1\,1\,0\,0 \\ 1\,?\,?\,0 \\ 0\,0\,1\,1 \end{pmatrix}.$$

Let $x, y, z \in \mathbb{F}_2^4$, and $x \xrightarrow{\mathsf{S}} y \xrightarrow{\mathsf{S}} z$. Easy to know the ANF of z is as follows:

$$\begin{cases} z_0 = y_0 = x_0, \\ z_1 = y_0 \oplus y_1 = x_1, \\ z_2 = y_0 \oplus y_1 y_2 = x_0 \oplus (x_0 \oplus x_1)(x_0 \oplus x_1 x_2) = x_0 x_1 \oplus x_1 x_2 \oplus x_0 x_1 x_2, \\ z_3 = 1 \oplus y_2 \oplus y_3 = x_0 \oplus x_2 \oplus x_3 \oplus x_1 x_2. \end{cases}$$

Then the bitwise characteristic matrix of two rounds S-box is:

$$\mathcal{M}_{\mathsf{S}\circ\mathsf{S}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ ? & ? & ? & 0 \\ 1 & ? & ? & 1 \end{pmatrix}.$$

The addition and multiplication between two characteristic matrices is defined as the following tables.

Table 3. Addition

+	0	1	?
0	0	1	?
1	1	0	?
?	?	?	?

Table 4. Multiplication

×	0	1	?
0	0	0	0
1	0	1	?
?	0	?	?

According to the above calculation rules, it is easy to verify that $\mathcal{M}_{\mathsf{S}\circ\mathsf{S}} = \mathcal{M}_{\mathsf{S}}\mathcal{M}_{\mathsf{S}}$. For any two n-variable vectorial boolean functions F_1 and F_2, it can be deduced that

$$\mathcal{M}_{F_2\circ F_1} = \mathcal{M}_{F_2}\mathcal{M}_{F_1},$$

note that the order of matrix multiplication needs to be consistent with the order of function composition.

For the above 4-bit S-box, when some bits of the input difference are fixed to be 0, some bits of the output difference can be linear combinations of the input bits. For example, let the input difference be $\alpha = (\alpha_0, \mathsf{0}, \mathsf{0}, ?)$, the output difference is $\beta = (\alpha_0, \alpha_0, \alpha_0, ?)$. These linearized bits are also called undisturbed points in [20]. The undisturbed points correspond to the determined points in the matrix, which is similar to the idea of Cube attack [6] where the attacker linearizes the nonlinear function by fixing some variables in the boolean function.

For a block cipher, it is difficult to find the undisturbed points, but for the S-box, the undisturbed points can be found easily by enumerating all input differences of the S-box. We find that most 4-bit S-boxes of lightweight block ciphers containing undisturbed points, such as GIFT, PRESENT, PRIDE. And some S-boxes with sizes more than 4 bits also have undisturbed points, such as the 8-bit S-boxes of Skinny-128 and Midori-128, the 5-bit S-box of ASCON.

Since the elements of a bitwise characteristic matrix represent the correlation between the input and output bits, the matrix and the input difference can be multiplied to get the output difference. In order to explain the usage of the bitwise characteristic matrix, this section we construct a simplified 8-bit Feistel block cipher (Fig. 1), and the function F is the 4-bit S-box constructed above.

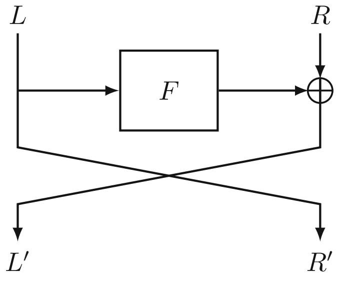

Fig. 1. A toy cipher

$$\mathcal{M}_{en} = \begin{pmatrix} 1&0&0&0&1&0&0&0 \\ 1&1&0&0&0&1&0&0 \\ 1&?&?&0&0&0&1&0 \\ 0&0&1&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \end{pmatrix}, \ \mathcal{M}_{en}^2 = \begin{pmatrix} 1&0&0&0&1&0&0&0 \\ 0&1&0&0&1&1&0&0 \\ ?&?&?&0&1&?&?&0 \\ 1&?&?&1&0&0&1&1 \\ 1&0&0&0&1&0&0&0 \\ 1&1&0&0&0&1&0&0 \\ 1&?&?&0&0&0&1&0 \\ 0&0&1&1&0&0&0&1 \end{pmatrix}.$$

According to the determined points of $\mathcal{M}_{en}^2$, some bits of the output difference can be quickly calculated. For example, let the input difference be $\alpha = (1, 1, 0, 0, 0, 0, 0, 0)$, then the output difference β after two rounds of encryption is

$$\beta = \mathcal{M}_{en}^2 \alpha = \begin{pmatrix} 1&0&0&0&1&0&0&0 \\ 0&1&0&0&1&1&0&0 \\ ?&?&?&0&1&?&?&0 \\ 1&?&?&1&0&0&1&1 \\ 1&0&0&0&1&0&0&0 \\ 1&1&0&0&0&1&0&0 \\ 1&?&?&0&0&0&1&0 \\ 0&0&1&1&0&0&0&1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ ? \\ ? \\ 1 \\ 0 \\ ? \\ 0 \end{pmatrix}.$$

For an SPN cipher, the linear layer can be represented by a $\mathbb{F}_2$ matrix and the S-box layer can be represented by a block diagonal matrix. So the matrix representation of a one round SPN cipher can be denoted as $\mathcal{M}_{P \circ S} = \mathcal{M}_P \mathcal{M}_S$.

3.2 Description of the Contradictions

After defining the bitwise characteristic matrix, we can construct contradictions in the middle state by exploiting the properties of the matrix representation. We introduce two different types of contradiction: direct contradiction and indirect contradiction.

Direct Contradiction: This type of contradiction happens when the middle states after encryption and decryption are obtained by directly multiplying the input and output differences with the corresponding matrix. Let the middle state after r_0-round encryption be α_1 and after r_1-round decryption be β_1, the corresponding matrix representations be $\mathcal{M}_{en}^{r_0}$ and $\mathcal{M}_{de}^{r_1}$, the input difference be α and the corresponding $(r_0 + r_1)$ rounds output difference be β, we have:

$$\alpha_1 = \mathcal{M}_{en}^{r_0} \quad \alpha, \quad \beta_1 = \mathcal{M}_{de}^{r_1} \quad \beta.$$

If specific bits of α_1 must be 1 and the same bits of β_1 must be 0, or vice versa, we can construct the direct contradiction. For convenience, we also denote the direct contradiction as follows:

$$\alpha \xrightarrow{encryption} \alpha_1 \neq \beta_1 \xleftarrow{decryption} \beta.$$

Indirect Contradiction: This type of contradiction happens when the middle states after encryption and decryption are obtained by multiplying the differences with the corresponding matrix and looking up the DDT of S-box before or after the matrix multiplication. If we look up the DDT only once, there are 3 different cases:

$$\begin{aligned} S(\alpha) &\xrightarrow{encryption} \alpha_1 \neq \beta_1 \xleftarrow{decryption} \beta, \\ \alpha &\xrightarrow{encryption} \alpha_1 \neq \beta_1 \xleftarrow{decryption} S^{-1}(\beta), \\ \alpha &\xrightarrow{encryption} \alpha_1 \overset{S}{\not\twoheadrightarrow} \beta_1 \xleftarrow{decryption} \beta. \end{aligned}$$

Take the first case for example. We look up the DDT of a S-box at input layer, we should calculate $\mathcal{M}_{encryption}$ and the input of $\mathcal{M}_{encryption}$ enumerates all possible differences after the S-box. To simplify the enumeration of DDT, we can also choose the undisturbed points and directly sum up the corresponding columns of the bitwise characteristic matrix. Take the S-box of GIFT for example, since the output difference must be $(?, ?, \mathtt{0}, ?)$ when the input difference is $(\mathtt{1}, \mathtt{1}, \mathtt{1}, \mathtt{0})$, we can make the input difference be $(\mathtt{1}, \mathtt{1}, \mathtt{1}, \mathtt{0})$ and sum up the first, second and fourth columns in the r-round matrix representation to get the middle state of $(r + 1)$-round encryption.

By indirect contradiction, we always find new and even longer impossible differentials than direct contradiction as we can utilize all undisturbed points of the S-box. Therefore we can easily deduce that if there is an r-round direct contradiction, there must be an r-round indirect contradiction. But the efficiency of constructing direct contradiction is usually higher than constructing the indirect contraction. Thus we first determine the longest direct contradiction and then search for the longer indirect contradiction. Algorithm 1 shows the processing of searching impossible differentials with the $\mathcal{M}$-method.

In the following section, we will apply our method to several block ciphers and detail the process of finding the impossible differentials of GIFT.

Algorithm 1. Search the impossible differential

```
Input: $\mathcal{M}_{en}$, $\mathcal{M}_{de}$, DDT of the S-box
Output: the longest impossible differential
$r_1 = r_2 = 1$
while There is $(r_1 + r_2)$-round direct contradiction do
    $r_1 = r_1 + 1$
    Calculate $\mathcal{M}_{en}^{r_1}$
    while There is $(r_1 + r_2)$-round direct contradiction do
        $r_2 = r_2 + 1$
        Calculate $\mathcal{M}_{de}^{r_2}$
    end while
end while  //the longest direct contradiction is $r_1 + r_2$
Looking up the DDT, check if there is indirect contradiction
if There is indirect contradiction then
    Output the $(r_1 + r_2 + 1)$-round impossible differential
end if
if There is no indirect contradiction then
    Output the $(r_1 + r_2)$-round impossible differential
end if
```

4 Applications from Cryptanalysis Aspects and Main Results

4.1 GIFT-64 and GIFT-128

GIFT [2] is an SPN lightweight block cipher proposed at CHES 2017. It is composed of 4-bit S-boxes and bit-wiring. The designers of GIFT revisit the design rationale of PRESENT and improve both security and efficiency. According to different block sizes, GIFT can be denoted as GIFT-64 and GIFT-128. Both of them adopt the same 4-bit S-box, the specification of the S-box in hexadecimal notation is given in Table 5.

Table 5. 4-bit S-box of GIFT

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$\mathsf{S}(x)$	1	a	4	c	6	f	3	9	2	d	b	7	5	0	8	e

Firstly, we calculate the bitwise characteristic matrices of the S-box and the results are as follows:

$$\mathcal{M}_{\mathsf{S}} = \begin{pmatrix} ?\ ?\ ?\ ? \\ ?\ ?\ ?\ ? \\ 1\ ?\ ?\ ? \\ 1\ 1\ ?\ ? \end{pmatrix}, \quad \mathcal{M}_{\mathsf{S}^{-1}} = \begin{pmatrix} ?\ 1\ ?\ 1 \\ ?\ 1\ 1\ ? \\ ?\ ?\ ?\ ? \\ ?\ ?\ ?\ ? \end{pmatrix}.$$

Moreover, there are 2 other differentials of an S-box that contain undisturbed points which can not be modelled by bitwise characteristic matrix, they are: $(\mathsf{0},\mathsf{1},\mathsf{1},\mathsf{0}) \xrightarrow{\mathsf{S}} (?,?,\mathsf{1},?)$ and $(\mathsf{1},\mathsf{1},\mathsf{1},\mathsf{0}) \xrightarrow{\mathsf{S}} (?,?,\mathsf{0},?)$.

Then We apply the $\mathcal{M}$-method to evaluate both GIFT-64 and GIFT-128.

For GIFT-64, the designers of GIFT applied the MILP-method and found 6-round impossible differentials for GIFT-64. In our method, we construct indirect contradictions for 6 rounds by looking up the DDT for the first S-box layer, the contradiction is described as follows:

$$S(\alpha) \xrightarrow{PSPS} \alpha_1 \neq \beta_1 \xleftarrow{P^{-1}S^{-1}P^{-1}S^{-1}P^{-1}S^{-1}} \beta.$$

Firstly, we determine the matrix representation of 2-round encryption and 2.5-round decryption i.e. $\mathcal{M}_{S \circ P \circ S \circ P}$ and $\mathcal{M}_{S^{-1} \circ P^{-1} \circ S^{-1} \circ P^{-1} \circ S^{-1}}$, respectively.

The matrix representation of a single S-box layer is a diagonal block matrix with every sub-block on the main diagonal is the bitwise characteristic matrix of a single S-box. Therefore, the matrix representation of a single S-box layer can be denoted as:

$$\mathcal{M}_S = \begin{pmatrix} M & 0 & 0 & 0 \\ 0 & M & 0 & 0 \\ 0 & 0 & M & 0 \\ 0 & 0 & 0 & M \end{pmatrix}, \quad M \triangleq \begin{pmatrix} \mathcal{M}_{\mathsf{S}} & 0 & 0 & 0 \\ 0 & \mathcal{M}_{\mathsf{S}} & 0 & 0 \\ 0 & 0 & \mathcal{M}_{\mathsf{S}} & 0 \\ 0 & 0 & 0 & \mathcal{M}_{\mathsf{S}} \end{pmatrix}.$$

The bit permutation can be easily transferred into a permutation matrix, which has exactly one non-zero entry in each column and each row. So the matrix representation of single round encryption starting at P-layer is $\mathcal{M}_{en} \triangleq \mathcal{M}_{S \circ P} = \mathcal{M}_S \mathcal{M}_P$. Therefore, we can calculate the matrix representation of 2-round encryption i.e. $\mathcal{M}_{en}^2$, and the matrix representation of 3-round decryption i.e. $\mathcal{M}_{de}^3$. The specification of $\mathcal{M}_{en}$, $\mathcal{M}_{en}^2$ and $\mathcal{M}_{de}^3$ are given in Appendix A.

According to $\mathcal{M}_{en}$ and $\mathcal{M}_{en}^2$, there is no 1-entry after 2-round encryption. Since the matrix representations of 3-round encryption and 4-round decryption are all ?-entries, so we can only construct 5-round direct contradiction by $\mathcal{M}_{en}^2$ and $\mathcal{M}_{de}^3$ at most. But we can utilize the undisturbed points which are not contained in the matrix representation to construct an insufficient diffusion state after 2.5-round encryption i.e. $S \circ P \circ S \circ P \circ S(x)$.

Let the input difference only active the first S-box in the first S-box layer. According to the undisturbed points of the S-box, let the non-zero nibble of the input difference be $(\mathsf{1},\mathsf{1},\mathsf{1},\mathsf{0})$ i.e. the e in hexadecimal notation, the corresponding output difference must be $(?,?,\mathsf{0},?)$, which is exactly the non-zero nibble of the input difference of $S \circ P \circ S \circ P(x)$. Hence the output difference of 2.5-round encryption can be obtained by multiply the output difference of the first S-box layer with the $\mathcal{M}_{en}^2$. Since there is only one non-zero column in each C_i $(i = 0,1,2,3)$, when the input difference of $\mathcal{M}_{en}^2$ contains 0-entry, the output difference must have zero nibbles. Take an example, $C_3 \cdot (?,?,?,\mathsf{0}) = (\mathsf{0},\mathsf{0},\mathsf{0},\mathsf{0})$ and $C_2 \cdot (?,?,\mathsf{0},?) = (\mathsf{0},\mathsf{0},\mathsf{0},\mathsf{0})$.

Let the input difference be $\alpha = (\mathsf{e},\mathsf{0},\ldots,\mathsf{0})$ in hexadecimal notation, so the corresponding difference after 2.5-round encryption is $\alpha_1 =$

$\mathcal{M}_{en}^2 \cdot S(\alpha) = (?, 0, ?, ?, ?, 0, ?, ?, ?, 0, ?, ?, ?, 0, ?, ?)$ which means there are 4 nibbles in α_1 that must be zero. Let the output difference β be $(0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, according to $\mathcal{M}_{de}^3$, the corresponding difference after 3-round decryption must be $\beta_1 = \mathcal{M}_{de}^3 \beta = (\beta_{10}, \beta_{11}, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)$ and the nibble $\beta_{10} = (1, ?, ?, ?)$ and $\beta_{11} = (?, 1, ?, ?)$ which means $\beta_1[0] = \beta_1[5] = 1$. Since the $\beta_1[5] = 1$ and $\alpha_1[5] = 0$ are the same bit in the same sate, we construct a 6-round impossible differential for GIFT-64 as follows:

$$(\mathtt{e}, 0, \ldots, 0) \overset{6R}{\nrightarrow} (0, 0, 0, 0, 4, 0, \ldots, 0).$$

Moreover, since the matrix representation can reveal the linear correlations between every bit, we can construct impossible differentials activating more S-boxes. Besides looking up the DDT of the first S-box layer, we can also look up the DDT of the last S-box layer, which can provide more impossible differentials by utilizing more undisturbed points. The contradictions can be denoted as:

$$S(\alpha) \xrightarrow{PSPS} \alpha_1 \neq \beta_1 \xleftarrow{P^{-1}S^{-1}P^{-1}S^{-1}P^{-1}} S^{-1}(\beta).$$

In Appendix B, we present the difference propagation of one impossible differential which activates 8 S-boxes in the first layer and 8 S-boxes in the last layer. The 6-round impossible differential in hexadecimal notation is

$$(\mathtt{e}, \mathtt{c}, 0, 0, \mathtt{e}, \mathtt{c}, 0, 0, \mathtt{e}, \mathtt{c}, 0, 0, \mathtt{e}, \mathtt{c}, 0, 0) \overset{6R}{\nrightarrow} (9, 9, 9, 9, 4, 6, 6, 6, 0, 0, 0, 0, 0, 0, 0, 0).$$

For GIFT-128, the designers only claim that GIFT-128 can achieve full diffusion after 4 rounds. According to the full diffusion state, GIFT-128 has no 8-round truncated impossible differentials. But the impossible differentials which consider the information of the S-box are missing in the document. At ASIACRYPT 2020, the SAT-method only considered GIFT-64, and there is no impossible differential cryptanalysis of GIFT-128 in other public documents.

Utilizing the indirect contradiction, we construct 7-round impossible differentials for GIFT-128. The contradiction can be denoted as:

$$\alpha \xrightarrow{SPSPSP} \alpha_1 \overset{S}{\nrightarrow} \beta_1 \xleftarrow{P^{-1}S^{-1}P^{-1}S^{-1}P^{-1}S^{-1}} \beta,$$

hence we need to calculate the matrix representation of 3-round encryption and 3-round decryption denoted as $\mathcal{M}_{en}^3$ and $\mathcal{M}_{de}^3$ respectively. And $\mathcal{M}_{en}^3 \cdot \alpha = \alpha_1$, $\mathcal{M}_{de}^3 \cdot \beta = \beta_1$.

Firstly, we calculate the matrix representation of 3-round encryption and the result is as follows:

$$\mathcal{M}_{en}^3 = \begin{pmatrix} D_0 & D_0 & D_1 & D_1 \\ D_1 & D_1 & D_0 & D_0 \\ D_0 & D_0 & D_1 & D_1 \\ D_1 & D_1 & D_0 & D_0 \end{pmatrix},$$

D_0 and D_1 represent two different 32×32 matrices, which can be denoted as two 8×8 block matrices:

$$D_0 = \left(\begin{array}{cccc|cccc} B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ \hline B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \\ B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 & B_0 \end{array}\right), D_1 = \left(\begin{array}{cccc|cccc} B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ \hline B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \\ B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 & B_1 \end{array}\right).$$

The definitions of B_0 and B_1 are as follows:

$$B_0 = \begin{pmatrix} 0\,0\,0\,0 \\ ?\,?\,?\,? \\ 0\,0\,0\,0 \\ ?\,?\,?\,? \end{pmatrix}, \quad B_1 = \begin{pmatrix} ?\,?\,?\,? \\ 0\,0\,0\,0 \\ ?\,?\,?\,? \\ 0\,0\,0\,0 \end{pmatrix}.$$

When we focus on the difference propagation through B_1, the output difference can be denoted as $(?, \mathtt{0}, ?, \mathtt{0})$ which might be one of $\{\mathtt{0}, \mathtt{2}, \mathtt{8}, \mathtt{a}\}$ in hexadecimal notation. According to the DDT of S-box, differences in $\{\mathtt{0}, \mathtt{2}, \mathtt{8}, \mathtt{a}\}$ can **never** propagate to $\{\mathtt{2}, \mathtt{4}, \mathtt{8}, \mathtt{c}\}$. Hence we can deduce two bit-level truncated impossible differentials for B_1, which are $(?, \mathtt{0}, ?, \mathtt{0}) \overset{\mathsf{S}}{\nrightarrow} (\mathtt{1}, ?, \mathtt{0}, \mathtt{0})$ and $(?, \mathtt{0}, ?, \mathtt{0}) \overset{\mathsf{S}}{\nrightarrow} (?, \mathtt{1}, \mathtt{0}, \mathtt{0})$. From the matrix representation of 3-round encryption, it is clear that if the input difference $\alpha = e_0 \triangleq (\mathtt{1}, \mathtt{0}, \ldots, \mathtt{0})$, every nibble of the corresponding output difference is one column of B_0 or B_1, take a example, the 15th nibble of α_1 in binary notation is $\alpha_1[60 \cdots 63] = (?, \mathtt{0}, ?, \mathtt{0})$.

Secondly, we calculate the matrix representation of 3-round decryption $\mathcal{M}_{de}^3$. Since the 128×128 matrix $\mathcal{M}_{de}^3$ is too large to present even by block matrix representation, we only depict 4 columns of the matrix. Let the output difference after 7-round encryption active the first S-box in the last layer, so we only need to present the first 4 columns of $\mathcal{M}_{de}^3$.

$$\begin{aligned} Row[0 \cdots 31] &= (B_6, B_4, B_5, B_7, B_6, B_4, B_5, B_7), \\ Row[32 \cdots 63] &= (B_6, B_4, B_5, B_7, B_3, B_4, B_5, B_2), \\ Row[64 \cdots 95] &= (B_6, B_4, B_5, B_7, B_6, B_4, B_5, B_7), \\ Row[96 \cdots 127] &= (B_6, B_4, B_5, B_7, B_6, B_4, B_5, B_7). \end{aligned}$$

The definitions of B_i $(i = 2, 3, 4, 5, 6, 7)$ are as follows:

$$B_2 = \begin{pmatrix} ?\,1\,1\,? \\ ?\,?\,?\,? \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{pmatrix}, B_3 = \begin{pmatrix} 0\,0\,0\,0 \\ ?\,1\,1\,? \\ ?\,?\,?\,? \\ 0\,0\,0\,0 \end{pmatrix}, B_4 = \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ ?\,?\,?\,? \\ ?\,?\,?\,? \end{pmatrix},$$

$$B_5 = \begin{pmatrix} ?\ ?\ ?\ ? \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ ?\ ?\ ?\ ? \end{pmatrix}, B_6 = \begin{pmatrix} 0\ 0\ 0\ 0 \\ ?\ ?\ ?\ ? \\ ?\ ?\ ?\ ? \\ 0\ 0\ 0\ 0 \end{pmatrix}, B_7 = \begin{pmatrix} ?\ ?\ ?\ ? \\ ?\ ?\ ?\ ? \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \end{pmatrix}.$$

When the output difference $\beta = e_1 \triangleq (0, 1, 0, \ldots, 0)$ or $\beta = e_2 \triangleq (0, 0, 1, \ldots, 0)$, the corresponding difference $\beta_1[60 \cdots 63] = (1, ?, 0, 0)$.

Let $\alpha = e_0$ and $\beta = e_1$, $\alpha_1[60 \cdots 63] = (?, 0, ?, 0)$ is exactly the input difference of the 15th S-box in the fourth round and $\beta_1[60 \cdots 63] = (1, ?, 0, 0)$ is the output difference of the same S-box, therefore $\alpha_1 \overset{S}{\nrightarrow} \beta_1$ and $(\alpha, \beta) \triangleq (e_0, e_1)$ is an impossible differential.

According to the matrix representation of 3-round encryption, the 3-round GIFT-128 dose not achieve full diffusion and the first 16 nibbles cannot influence the 61st bit and the 63rd bit of α_1. Therefore, the input difference α can at most activate 16 S-boxes. And by looking up the DDT of the S-box in the last round, we can investigate more linear properties which make the output difference can at most activate 8 S-boxes and we present one of them in Appendix C.

4.2 Other Block Ciphers

By Algorithm 1, we also make applications to many other block ciphers. Due to the limitation of the page size, we only present the new impossible differentials found by $\mathcal{M}$-method.

For PRIDE, an 64-bit block cipher proposed at CRYPTO 2014, we find the first 5-round impossible differentials and there are only indirect contradictions for 5-round PRIDE, one of which is as follows:

$$S(\alpha) \xrightarrow{PSPS} \alpha_1 \neq \beta_1 \xleftarrow{P^{-1}S^{-1}P^{-1}} S^{-1}(\beta).$$

One of the impossible differentials is as follows:

$$(0, 0, 8, 0, 0, 1, 0, 0, 8, 0, 8, 0, 7, 0, 0, 0) \overset{5R}{\nrightarrow} (0, 0, 0, \beta_0, 0, 0, 0, \beta_1, 0, 0, \beta_2, 0, 0, 0, \beta_3, 0).$$

$\beta_i \in \mathbb{F}_{2^4}^* \ (i = 0, 1, 2, 3)$, therefore the input difference activates 5 S-boxes and the output difference activates 4 S-boxes.

For Pyjamask, one of the 2nd round candidates of the NIST lightweight cryptography project, the block size has two different versions i.e. 96-bit and 128-bit. As Pyjamask adopts complex binary matrices to be the linear component and LS-design, it can achieve full diffusion in 2 rounds which means there is no 4-round truncated impossible differentials. For Pyjamask-96, taking into consideration the information of the S-box, we construct 4-round impossible differentials by indirect contradiction. And our impossible differentials surpass the previous results which cover only 3 rounds. The contradiction for Pyjamask-96 is as follows:

$$S(\alpha) \xrightarrow{PSP} \alpha_1 \neq \beta_1 \xleftarrow{S^{-1}P^{-1}} S^{-1}(\beta).$$

One of the impossible differentials in Octal notation is as follows:

$$(6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) \overset{4R}{\nrightarrow}$$
$$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,\beta_0,0,2,0,0,0,\beta_1,0,0,0),$$

$\beta_i \in \mathbb{F}_{2^3}^*$ $(i = 0,1)$, therefore the input difference activates 1 S-boxes and the output difference activates 3 S-boxes. For Pyjamask-128, we can only find 3-round impossible differentials by direct contradictions and we construct an impossible differential which activate 13 input S-boxes and 21 output S-boxes. And the impossible differential is as follows:

$$(9,0,0,9,\alpha_0,0,0,\alpha_1,0,\alpha_2,2,\alpha_3,0,0,0,0,2,0,1,0,2,\alpha_4,0,0,0,0,0,\alpha_5,0,\alpha_6,0,0)$$
$$\overset{4R}{\nrightarrow} (\beta_0,0,\beta_1,\beta_2,\beta_3,\beta_4,0,\beta_5,\beta_6,0,\beta_7,\beta_8,0,0,0,0,\beta_9,0,0,\beta_{10},\beta_{11},0,0,\beta_{12},\beta_{13},$$
$$\beta_{14},\beta_{15},\beta_{16},\beta_{17},\beta_{18},0,\beta_{19}) \quad \alpha_i,\beta_i \in \mathbb{F}_{2^4}^*.$$

For ASCON, one of the finalists of the NIST lightweight cryptography project, the block size is 320 bits and the S-box size is 5 bits, we construct new 5-round impossible differentials by indirect contradictions. One of the contradictions is as follows:

$$S(\alpha) \xrightarrow{PSPSP} \alpha_1 \neq \beta_1 \xleftarrow{S^{-1}P^{-1}S^{-1}} \beta.$$

And one of the 5-round impossible differentials in hexadecimal notation is as follows:

$$(0,0,0,0,0,0,0,0,0,10,0,0,4,0,0,0,0,7,0,0,0,\alpha_0,\mathtt{f},0,0,0,0,0,0,0,0,\mathtt{c},0,0,$$
$$13,0,0,\mathtt{c},0,0,0,8,0,0,\mathtt{c},0,0,0,0,0,0,\mathtt{c},0,0,0,0,0,0,0,13,0,0,\alpha_1,\mathtt{1c}) \overset{5R}{\nrightarrow} (0,0,$$
$$0,0,0,0,0,0,0,0,0,0,0,0,0,\beta_0,0,\cdots,0) \quad \alpha_i,\beta_i \in \mathbb{F}_{2^5}^*.$$

For RECTANGLE, a 64-bit lightweight block cipher, we construct new 8-round impossible differentials by indirect contradictions. One of the contradictions is as follows:

$$S(\alpha) \xrightarrow{PSPSPSP} \alpha_1 \neq \beta_1 \xleftarrow{S^{-1}P^{-1}S^{-1}P^{-1}S^{-1}P^{-1}} S^{-1}(\beta).$$

And one of the 8-round impossible differentials in hexadecimal notation is as follows:

$$(0,0,0,0,5,0,0,\mathtt{c},0,0,0,0,0,0,0,0) \overset{8R}{\nrightarrow} (0,0,0,0,0,0,0,0,0,0,0,2,6,0,0,0).$$

For PRESENT, a 64-bit block cipher proposed at CHES 2007, we construct new 6-round impossible differentials by indirect contradictions. One of the impossible differentials is as follows:

$$(9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9) \overset{6R}{\nrightarrow} (5,5,5,5,5,1,5,5,5,5,5,5,5,5,5,5).$$

5 Conclusion

In this paper, we defined the bitwise characteristic matrix and applied it to search for impossible differentials. By iterating the matrix to represent r-round block ciphers, we improve the efficiency of searching. Moreover, the $\mathcal{M}$-method can easily model block ciphers with block sizes more than 256 bits and reveal the positions of the contradictions. And the matrix multiplication defined in this paper can function with low time and memory complexity. As a result, we find new impossible differentials for some block ciphers including the 7-round impossible differentials for GIFT-128, the 5-round impossible differentials for PRIDE, and the 4-round impossible differentials for Pyjamask-96.

Although $\mathcal{M}$-method has some advantages, there are still some limitations which are also the targets of our future works. The first one is to make our method cover more cryptanalysis techniques such as linear cryptanalysis and more block cipher structures such as ARX. The second one is to make our method containing more details of the block ciphers including the key schedule. The last but not least is to apply our method to optimize the key recovery phases.

A The Matrix Representations of GIFT-64

Because of the page size, we can only represent each bitwise matrix as a block matrix and the dimension of each sub-block is equal to the size of the S-box, and the sub-block 0 in the matrix denotes a 4×4 matrix with all 16 entries are 0, the sub-block ? denotes a sub-block with all entries are ?.

$$\mathcal{M}_{en} = \left(\begin{array}{cccc:cccc:cccc:cccc}
A_1 & A_2 & A_3 & A_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_1 & A_2 & A_3 & A_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1 & A_2 & A_3 & A_0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_1 & A_2 & A_3 & A_0 \\
\hdashline
A_2 & A_3 & A_0 & A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_2 & A_3 & A_0 & A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 & A_3 & A_0 & A_1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_2 & A_3 & A_0 & A_1 \\
\hdashline
A_3 & A_0 & A_1 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_3 & A_0 & A_1 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_3 & A_0 & A_1 & A_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_3 & A_0 & A_1 & A_2 \\
\hdashline
A_0 & A_1 & A_2 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & A_0 & A_1 & A_2 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_0 & A_1 & A_2 & A_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_0 & A_1 & A_2 & A_3
\end{array}\right),$$

$$\mathcal{M}_{en}^{2} = \left(\begin{array}{cccc|cccc|cccc|cccc}
C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 \\
C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 \\
C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 \\
C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 \\
\hline
C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 \\
C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 \\
C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 \\
C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 \\
\hline
C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 \\
C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 \\
C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 \\
C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 \\
\hline
C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 \\
C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 \\
C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 \\
C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3 & C_0 & C_1 & C_2 & C_3
\end{array}\right),$$

$$\mathcal{M}_{de}^{3} = \left(\begin{array}{cccc|cccc|cccc|cccc}
R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? \\
R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? \\
\hline
? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 \\
? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 \\
\hline
? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? \\
? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? \\
\hline
? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ? & ? & R_0 & ? & ? \\
? & R_0 & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & R_1 & ? & ?
\end{array}\right).$$

$$A_0 \triangleq \begin{pmatrix} ? & 0 & 0 & 0 \\ ? & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, A_1 \triangleq \begin{pmatrix} 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \end{pmatrix}, A_2 \triangleq \begin{pmatrix} 0 & 0 & ? & 0 \\ 0 & 0 & ? & 0 \\ 0 & 0 & ? & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, A_3 \triangleq \begin{pmatrix} 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \end{pmatrix}, R_1 \triangleq \begin{pmatrix} ? & ? & ? & ? \\ ? & 1 & 1 & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix},$$

$$C_0 \triangleq \begin{pmatrix} ? & 0 & 0 & 0 \\ ? & 0 & 0 & 0 \\ ? & 0 & 0 & 0 \\ ? & 0 & 0 & 0 \end{pmatrix}, C_1 \triangleq \begin{pmatrix} 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & ? & 0 & 0 \end{pmatrix}, C_2 \triangleq \begin{pmatrix} 0 & 0 & ? & 0 \\ 0 & 0 & ? & 0 \\ 0 & 0 & ? & 0 \\ 0 & 0 & ? & 0 \end{pmatrix}, C_3 \triangleq \begin{pmatrix} 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \\ 0 & 0 & 0 & ? \end{pmatrix}, R_0 \triangleq \begin{pmatrix} ? & 1 & 1 & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{pmatrix}.$$

B New 6-Round Impossible Differential for GIFT-64

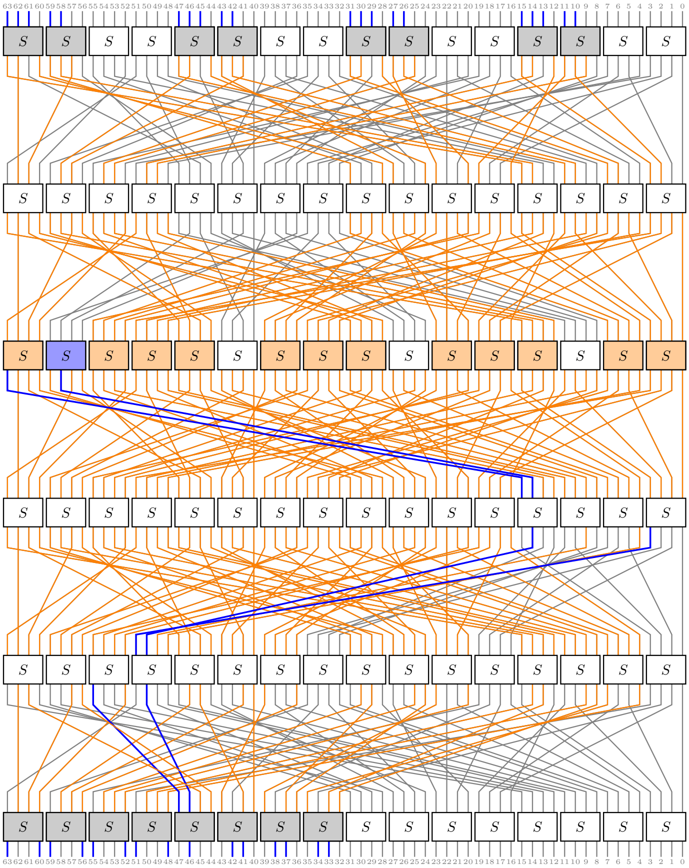

Fig. 2. 6-round impossible differential for GIFT-64

$$(\mathtt{e},\mathtt{c},\mathtt{0},\mathtt{0},\mathtt{e},\mathtt{c},\mathtt{0},\mathtt{0},\mathtt{e},\mathtt{c},\mathtt{0},\mathtt{0},\mathtt{e},\mathtt{c},\mathtt{0},\mathtt{0}) \overset{6R}{\nrightarrow} (\mathtt{9},\mathtt{9},\mathtt{9},\mathtt{9},\mathtt{4},\mathtt{6},\mathtt{6},\mathtt{6},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0},\mathtt{0})$$

The blue lines mean the bit must be 1 according to DDT of the S-box, the orange lines mean the value of the bit cannot be determined. And the following figures adopt the same notation.

C 7-Round Truncated Impossible Differential for GIFT-128

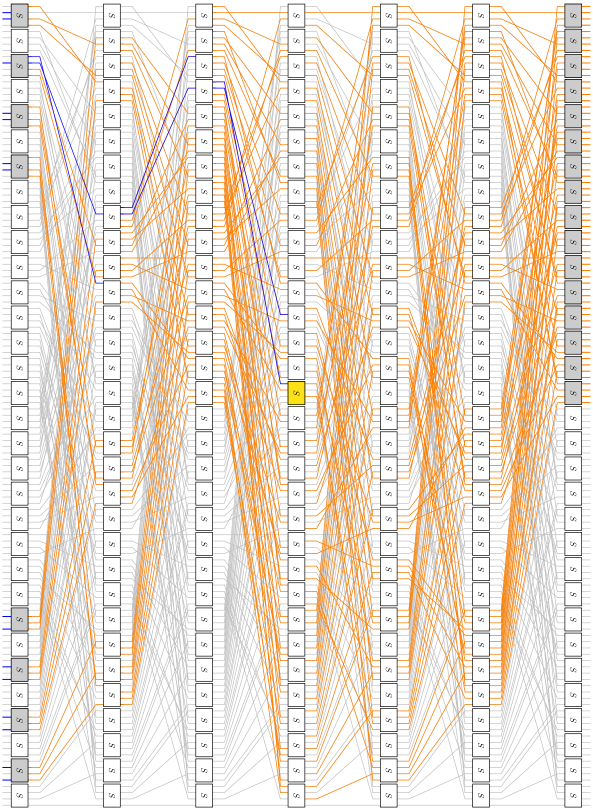

Fig. 3. 7-round truncated impossible differential

$$(\alpha[0\cdots 64]||0_{64}) \overset{7R}{\nrightarrow} (6,0,4,0,6,0,6,\underbrace{0,\ldots,0}_{17},5,0,5,0,5,0,5,0)$$

In Fig. 3, the left side denotes the 128-bit output difference and the right side denotes the 128-bit input difference. The difference propagation in the yellow S-box is a contradiction and the specification is given in Fig. 4.

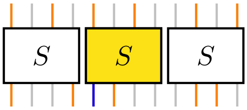

Fig. 4. Specification of the contradiction

References

1. Albrecht, M.R., Driessen, B., Kavun, E.B., Leander, G., Paar, C., Yalçın, T.: Block ciphers – focus on the linear layer (feat. PRIDE). In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 57–76. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_4
2. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: Gift: a small present. In: Fischer, W., Homma, N. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2017, pp. 321–345. Springer Cham (2017)
3. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_2
4. Bogdanov, A., et al.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_31
5. Cui, T., Jia, K., Fu, K., Chen, S., Wang, M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptol. ePrint Arch. p. 689 (2016). http://eprint.iacr.org/2016/689
6. Dinur, I., Shamir, A.: Cube attacks on tweakable black box polynomials. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 278–299. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_16
7. Dobraunig, C., Eichlseder, M., Mendel, F., Schlffer, M.: Ascon - submission to the CASEAR competition (2016)
8. Dolmatov, V.: GOST R 34.12-2015: Block cipher “kuznyechik”. In: RFC, pp. 1–14 (2016)
9. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON family of lightweight hash functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_13
10. Hu, X., Li, Y., Jiao, L., Tian, S., Wang, M.: Mind the propagation of states. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, pp. 415–445. Springer, Cham (2020)
11. Kim, J., Hong, S., Sung, J., Lee, S., Lim, J., Sung, S.: Impossible differential cryptanalysis for block cipher structures. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 82–96. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24582-7_6
12. Knudsen, L.: Deal - a 128-bit block cipher. In: NIST AES Proposal (1998)
13. Liu, Y., Weiming T., Shen, Z.: H.L.L.W.: Impossible differential cryptanalysis of lightweight block cipher pyjamask. Appl. Res. Comput. (2021). https://doi.org/10.19734/j.issn.1001-3695.2021.03.0063
14. Luo, Y., Wu, Z., Lai, X., Gong, G.: A unified method for finding impossible differentials of block cipher structures. IACR Cryptol. ePrint Arch. p. 627 (2009). http://eprint.iacr.org/2009/627
15. Sasaki, Yu., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017. LNCS, vol. 10212, pp. 185–215. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_7
16. Shen, X., Li, R., Sun, B., Cheng, L., Li, C., Liao, M.: Dual relationship between impossible differentials and zero correlation linear hulls of simon-like ciphers. In: Liu, J.K., Samarati, P. (eds.) Information Security Practice and Experience, pp. 237–255. Springer International Publishing, Cham (2017)

17. Shen, X., Liu, G., Sun, B., Li, C.: Impossible differentials of SPN ciphers. In: Chen, K., Lin, D., Yung, M. (eds.) Information Security and Cryptology, pp. 47–63. Springer, Cham (2017)
18. Sun, B., Liu, M., Guo, J., Rijmen, V., Li, R.: Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 196–213. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_8
19. Sun, B., et al.: Links among impossible differential, integral and zero correlation linear cryptanalysis. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 95–115. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_5
20. Tezcan, C.: Improbable differential attacks on PRESENT using undisturbed bits. J. Comput. Appl. Math. **259**, 503–511 (2014)
21. Wu, S., Wang, M.: Automatic search of truncated impossible differentials for word-oriented block ciphers. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 283–302. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34931-7_17