Finding Three-Subset Division Property for Ciphers with Complex Linear Layers

Debasmita Chakraborty(✉)

Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
debasmitachakraborty1@gmail.com

Abstract. Conventional bit-based division property (CBDP) and bit-based division property using three subsets (BDPT) introduced by Todo *et al.* at FSE 2016 are the most effective techniques for finding integral characteristics of symmetric ciphers. At ASIACRYPT 2019, Wang *et al.* proposed the idea of modeling the propagation of BDPT, and recently Liu *et al.* described a model set method that characterized the BDPT propagation. However, the linear layers of the block ciphers which are analyzed using the above two methods of BDPT propagation are restricted to simple bit permutation. Thus the feasibility of the MILP method of BDPT propagation to analyze ciphers with complex linear layers is not settled. In this paper, we focus on constructing an automatic search algorithm that can accurately characterize BDPT propagation for ciphers with complex linear layers. We first introduce BDPT propagation rule for the binary diffusion layer and model that propagation in MILP efficiently. The solutions to these inequalities are exact BDPT trails of the binary diffusion layer. Next, we propose a new algorithm that models Key-Xor operation in BDPT based on MILP technique. Based on these ideas, we construct an automatic search algorithm that accurately characterizes the BDPT propagation and we prove the correctness of our search algorithm. We demonstrate our model for the block ciphers with non-binary diffusion layers by decomposing the non-binary linear layer trivially by the COPY and XOR operations. Therefore, we apply our method to search integral distinguishers based on BDPT of SIMON, SIMON(102), PRINCE, MANTIS, PRIDE, and KLEIN block ciphers. For PRINCE and MANTIS, we find $(2 + 2)$ and $(3 + 3)$ round integral distinguishers respectively which are longest to date. We also improve the previous best integral distinguishers of PRIDE and KLEIN. For SIMON, SIMON(102), the integral distinguishers found by our method are consistent with the existing longest distinguishers.

Keywords: BDPT · Complex linear layer · Binary matrix · MILP

1 Introduction

Division Property. At Eurocrypt 2015, Todo [30] introduced Division property which is a novel strategy to discover integral characteristics to search integral distinguishers of block cipher structures (Feistel structure and SPN structure).

T. Isobe and S. Sarkar (Eds.): INDOCRYPT 2022, LNCS 13774, pp. 398–421, 2022.
https://doi.org/10.1007/978-3-031-22912-1_18

Later, Todo and Morii [31] introduced bit-based division property (which is actually called Conventional Bit-based Division Property (CBDP)), which could be treated as an exceptional instance of division property. Actually CBDP classify all vectors $\boldsymbol{u}$ in $\mathbb{F}_2^n$ into two subsets such that the parity of $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}}$ is 0 or *unknown* (where $\boldsymbol{x}^u$ is defined as $\boldsymbol{x}^u := \prod_{i=1}^{n} x_i{}^{u_i}$). Moreover, at CRYPTO 2016, Boura and Canteaut [9] presented a different perspective on the division property, called 'parity set'.

The intricacy of using CBDP was generally equivalent to 2^n for a n-bit primitives. Henceforth, the gigantic intricacy limited the wide uses of CBDP. To tackle the limitation of the tremendous complexity, Xiang *et al.* [34] applied MILP-strategy to look through integral distinguisher dependent on CBDP and they applied this modeling technique to six lightweight block ciphers. By extending and improving this method, the integral attacks have been applied to many ciphers and many better integral distinguisher has been found [18,19,21,23,28,29,37].

Three-Subset Division Property. Although CBDP can find more precise integral distinguishers than other methods, the accuracy is never perfect. To find more accurate distinguishers, the bit-based division property using three subsets (BDPT) was proposed in [31]. BDPT divides all vectors $\boldsymbol{u}$ in $\mathbb{F}_2^n$ into two subsets such that the parity of $\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}}$ is 0, 1 or *unknown*. Essentially, the set *unknown* in CBDP is divided into 1-subset and *unknown* subset in BDPT. As a result, BDPT can find more precise integral characteristics than CBDP. For example, CBDP demonstrated the existence of SIMON32's 14-round integral distinguisher whereas BDPT discovered SIMON32's 15-round integral distinguisher [30].

Despite of its successful combination of the MILP and the CBDP, the MILP modeling technique does not work quite well with the BDPT. As in case of BDPT we have to track the division property propagation of two sets ($\mathbb{K}$ (the *unknown* subset) and $\mathbb{L}$ (the 1-subset)) as well as the influence of the set $\mathbb{L}$ on the set $\mathbb{K}$ should also be traced which makes the procedure of constructing automatic search algorithm based on BDPT complicated.

First, Hu *et al.* [20] proposed variant three subset division property (VTDP) and applied this method to improve some integral distinguishers although it sacrifices quite some accuracy of BDPT. Therefore, Wang *et al.* [32] proposed the idea of modeling the propagation for the BDPT and recently Liu *et al.* [24] proposed a model set method to search integral distinguishers based on BDPT. Both of these methods have been applied to the block ciphers having simple bit permutation as their linear layer.

1.1 Motivation

The idea of modeling BDPT propagation which is described in [32] is that each node on the breadth-first search algorithm is regarded as the starting point of division trails, and the MILP evaluates whether there is a feasible solution from every node. According to their searching algorithm, we can run this algorithm to

any block cipher efficiently only if we can divide the round function into several appropriate parts. Therefore, it is very difficult to model BDPT propagation using this technique for the ciphers with complex linear layers. Next, Liu *et al.* [24] proposed model set method to search BDPT where the authors constructed r different MILP models for r-round block ciphers which is a bit complicated. Moreover, both these methods have been applied to the block ciphers having linear layers as simple bit permutation. Now, the following question arises:
Is MILP method of BDPT propagation efficiently applicable for ciphers with complex linear layers?

1.2 Our Contributions

To address this question, first we propose an idea to find BDPT propagation through the binary (complex) linear layer accurately and then we construct an automatic search algorithm for BDPT in this paper. The details of our technical contributions are listed as follows:

Model the BDPT Propagation of Binary Linear Layer. We give an idea to find exact BDPT propagation through the binary (complex) linear layer which is a new method that helps us to construct MILP model of BDPT propagation through the binary linear layer accurately. We actually find that the rows of the primitive matrix corresponding to the binary mixcolumn matrix can be divided into some cosets with the property that the rows in different cosets have no common nonzero entries in the same column. Using this interesting property, we can easily find accurate BDPT propagation and can give a description of such propagation by smallest number of inequalities.

Construction of Automatic Search Algorithm for BDPT. To search for BDPT, first we construct the MILP models for key-independent components of the round function of block ciphers. When a Key-Xor operation is applied, new vectors generated from the set $\mathbb{L}$ will be added to the set $\mathbb{K}$. Therefore, how to model Key-Xor operation accurately is a complex problem. To solve this problem, we construct a new efficient algorithm that models each Key-Xor operation based on MILP technique. Finally, by selecting appropriate initial BDPT and stopping rules we construct an automatic search algorithm that accurately characterize BDPT propagation using only two MILP models which is much simpler than the algorithm described in [24]. Moreover, we prove the correctness of our search algorithm.

Applications. As for the application of our methodology, first time we apply BDPT on block ciphers with complex linear layers. We apply our automatic search model to search integral distinguishers of PRINCE [8], MANTIS [6], KLEIN [16], PRIDE [4], SIMON [5], and SIMON(102) [22]. The results are shown in Table 1.

At first, we apply our method on PRINCE and MANTIS which have binary linear layer. We find $2 + 2$ round integral distinguisher for PRINCE which is one more round than the previous best integral distinguisher [15] and find $3 + 3$ round integral distinguisher for MANTIS which is also one more round than the previous best integral distinguisher [15] where we denote a are the rounds before the middle layer, and b are the rounds after the middle layer and $a + b$ as total number of rounds.

Table 1. Summarization of integral distinguishers

Cipher	Data	Round	Number of constant bits	Time	References
MANTIS	2^{32}	3+2	16	–	[15]
	$\mathbf{2^{63}}$	**3+3**	**64**	**2 h 8 m**	**Sect.** 5.1
PRINCE	2^{32}	2+1	64	–	[15]
	$\mathbf{2^{63}}$	**2+2**	**64**	**21 h 45 m**	**Sect.** 5.1
PRIDE64*	-	8	–	–	[33]
	$\mathbf{2^{63}}$	**9**	**32**	**2 h 35 m**	**Sect.** 5.2
KLEIN64	2^{32}	5	64	–	[36]
	$\mathbf{2^{62}}$	**6**	**64**	**45 m**	**Sect.** 5.2

* In [33], the authors have only mentioned that PRIDE64 has 8-round integral distinguisher and no other information is available best known to us.

To complete our BDPT analysis on ciphers with complex linear layers, we apply our method to KLEIN and PRIDE which have non-binary linear layers. As there are no known results on them related to CBDP, then we first apply MILP based CBDP on them and find 6-round and 9-round integral distinguishers for KLEIN and PRIDE respectively which are one more rounds to previous best integral distinguishers [33,36]. Therefore, we apply our MILP based BDPT method and the integral distinguishers we find are in accordance with the integral distinguishers we find based on CBDP. Finally, we apply our method to all variants of SIMON, and SIMON(102) block ciphers and the distinguishers we find are in accordance with the previous longest distinguishers [24] but we get these results in better time.

1.3 Organization of the Paper

This paper is organized as follows: In Sect. 2, we briefly recall some background knowledge about the bit-based division property. In Sect. 3, we studies how to model basic operations used in the round function of a block cipher by the MILP technique and introduce exact modelling of complex (binary) linear layer in BDPT. Section 4 studies the initial and stopping rules, and search algorithm. We show some applications of our model in Sect. 5. At last we conclude the paper in Sect. 6.

2 Preliminaries

2.1 Notation

Let $\mathbb{F}_2$ denote the finite field $\{0,1\}$ and $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_2^n$ be an n-bit vector, where a_i denotes the i-th bit of $\boldsymbol{a}$. For n-bit vectors $\boldsymbol{x}$ and $\boldsymbol{u}$, define $\boldsymbol{x}^{\boldsymbol{u}} = \prod_{i=0}^{n-1} x_i^{u_i}$. Then, for any $\boldsymbol{k} \in \mathbb{F}_2^n$ and $\boldsymbol{k'} \in \mathbb{F}_2^n$, define $\boldsymbol{k} \succeq \boldsymbol{k'}$ if $k_i \geq k_i'$ holds for all $i = 0, 1, \ldots, n-1$, and define $\boldsymbol{k} \succ \boldsymbol{k'}$ if $k_i > k_i'$ holds for all $i = 0, 1, \ldots, n-1$. For a subset $\mathcal{I} \subseteq \{0, 1, ..., n-1\}$, $\boldsymbol{u}_{\mathcal{I}}$ denotes an n-dimensional bit vector $(u_0, u_1, \ldots, u_{n-1})$ satisfying $u_i = 1$ if $i \in \mathcal{I}$ and $u_i = 0$ otherwise. We simply write $\mathbb{K} \leftarrow \boldsymbol{k}$ when $\mathbb{K} = \mathbb{K} \cup \{\boldsymbol{k}\}$ and $\mathbb{K} \rightarrow \boldsymbol{k}$ when $\mathbb{K} = \mathbb{K} \setminus \{\boldsymbol{k}\}$. And $|\mathbb{K}|$ denotes the number of elements in the set $\mathbb{K}$. We denote $[n] = \{1, 2, \ldots, n\}$, $\mathbf{1} = 1^n$, and $\mathbf{0} = 0^n$. We denote i-th unit vector in $\mathbb{F}_2^n$ as $\boldsymbol{e}_i$.

2.2 Bit-Based Division Property

Two kinds of bit-based division property (CBDP and BDPT) were introduced by Todo and Morii at FSE 2016 [31]. Their definitions are as follows.

Definition 1 ***(CBDP** [31]**).*** *Let $\mathbb{X}$ be a multiset whose elements take a value of $\mathbb{F}_2^n$. Let $\mathbb{K}$ be a set whose elements take an n-dimensional bit vector. When the multiset $\mathbb{X}$ has the division property $D_{\mathbb{K}}^{1^n}$, it fulfills the following conditions:*

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} \textit{unknown}, & \textit{if there is } \boldsymbol{k} \in \mathbb{K} \textit{ satisfying } \boldsymbol{u} \succeq \boldsymbol{k}, \\ 0 & \textit{otherwise.} \end{cases}$$

Some propagation rules of CBDP are proven in [30,31,34].

Definition 2 ***(BDPT** [31]**).*** *Let $\mathbb{X}$ be a multi-set whose elements take a value of F_2^n. Let $\mathbb{K}$ and $\mathbb{L}$ be two sets whose elements take n-dimensional bit vectors. When the multi-set $\mathbb{X}$ has the division property $D_{\mathbb{K},\mathbb{L}}^{1^n}$, it fulfils the following conditions:*

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}} = \begin{cases} \textit{unknown}, & \textit{if there is} \boldsymbol{k} \in \mathbb{K} \textit{ satisfying } \boldsymbol{u} \succeq \boldsymbol{k}, \\ 1, & \textit{else if there is } \boldsymbol{l} \in \mathbb{L} \textit{ satisfying } \boldsymbol{u} = \boldsymbol{l}, \\ 0, & \textit{otherwise.} \end{cases}$$

If there are $\boldsymbol{k} \in \mathbb{K}$ and $\boldsymbol{k}^{'} \in \mathbb{K}$ satisfying $\boldsymbol{k} \succeq \boldsymbol{k}^{'}$ in the CBDP $D_{\mathbb{K}}^{1^n}$, then $\boldsymbol{k}$ can be removed from $\mathbb{K}$ because the vector $\boldsymbol{k}$ is redundant. This progress is denoted as $\boldsymbol{Reduce0}(\mathbb{K})$. Moreover, if there are $\boldsymbol{l} \in \mathbb{L}$ and $\boldsymbol{k} \in \mathbb{K}$, then the vector $\boldsymbol{l}$ is also redundant if $\boldsymbol{l} \succeq \boldsymbol{k}$. This progress is denoted as $\boldsymbol{Reduce1}(\mathbb{K}, \mathbb{L})$. The redundant vectors in $\mathbb{K}$ and $\mathbb{L}$ will not affect the parity of $\boldsymbol{x}^{\boldsymbol{u}}$ for any $\boldsymbol{u}$.

The propagation rules of $\mathbb{K}$ in CBDP are the same with BDPT. So we only introduce the propagation rules of BDPT which are needed in this paper. For further details, please refer to [31,32].

BDPT Rule 1 (Xor with The Secret Key [31].) *Let $\mathbb{K}$ be the input multiset satisfying $D^{1^n}_{\mathbb{K},\mathbb{L}}$. For the input $\boldsymbol{x} \in \mathbb{X}$, the output $y \in \mathbb{Y}$ is $\boldsymbol{y} = (x_0, \ldots, x_i \oplus r_k, x_{i+1}, \ldots, x_{n-1})$, where r_k is the secret key. Then, the output multiset $\mathbb{Y}$ has $D^{1^n}_{\mathbb{K}',\mathbb{L}'}$, where $\mathbb{K}'$ and $\mathbb{L}'$ are computed as*

$$\begin{cases} \mathbb{L}^{'} \leftarrow \boldsymbol{l} \;\; for\; \boldsymbol{l} \in \mathbb{L}, \\ \mathbb{K}^{'} \leftarrow \boldsymbol{k} \;\; for\; \boldsymbol{k} \in \mathbb{K}, \\ \mathbb{K}^{'} \leftarrow (l_1, l_2, ..., l_i \vee 1, ..., l_n) \;\; for\; \boldsymbol{l} \in \mathbb{L}\; satisfying\; l_i = 0. \end{cases}$$

BDPT Rule 2 (S-box [32].) *For an S-box : $\mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$, let $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ and $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ denote the input and output variables. And every y_i, $i \in \{0, 1, \ldots, n-1\}$ can be expressed as a boolean function of $(x_0, x_1, \ldots, x_{n-1})$. If the input BDPT of S-box is $D^{1^n}_{\mathbb{K},\mathbb{L}=\{\boldsymbol{l}\}}$, then the output BDPT of S-box can be calculated by $D^{1^n}_{\boldsymbol{Reduce0}(\underline{\mathbb{K}}),\boldsymbol{Reduce1}(\underline{\mathbb{K}},\,\underline{\mathbb{L}})}$,*

$$\begin{cases} \underline{\mathbb{K}} = \{\boldsymbol{u'} \in \mathbb{F}_2^n \,|\, for\ any\ \boldsymbol{u} \in \mathbb{K},\ if\ \boldsymbol{y}^{\boldsymbol{u'}}\ contains\ any\ term\ \boldsymbol{x}^{\boldsymbol{v}}\ satisfying\ \boldsymbol{v} \succeq \boldsymbol{u}\} \\ \underline{\mathbb{L}} = \{\boldsymbol{u} \in \mathbb{F}_2^n \,|\, \boldsymbol{y}^{\boldsymbol{u}}\ contains\ the\ term\ \boldsymbol{x}^{\boldsymbol{l}}\} \end{cases}$$

Let $D^{1^n}_{\mathbb{K},\,\mathbb{L}=\{\boldsymbol{l}_0,\ldots,\boldsymbol{l}_{r-1}\}}$ and $D^{1^n}_{\mathbb{K}',\mathbb{L}'}$ be the input and output BDPT of S-box, respectively. We can get the output BDPT $D^{1^n}_{\mathbb{K}',\mathbb{L}'_i}$ from the corresponding input BDPT $D^{1^n}_{\mathbb{K},\,\mathbb{L}=\{\boldsymbol{l}_i\}}$ where $i = 0, 1, \ldots, r-1$. Then,

$$\mathbb{L}' = \{\boldsymbol{l} \,|\, \boldsymbol{l}\ appears\ odd\ times\ in\ sets\ \mathbb{L}'_0, \ldots, \mathbb{L}'_{r-1}\}$$

2.3 The MILP Model for CBDP

At Asiacrypt 2016, Xiang *et al.* [34] applied MILP method to search integral distinguishers based in CBDP, which allowed them to analyze block ciphers with large sizes. Firstly they introduced the concept of CBDP trail as follows:

Definition 3 (CBDP Trail [34]). *Consider the propagation of the division property $\{\boldsymbol{k}\} \equiv \mathbb{K}_0 \xrightarrow{f_1} \mathbb{K}_1 \xrightarrow{f_2} \mathbb{K}_2 \xrightarrow{f_3} ....$ Moreover, for any vector $\boldsymbol{k}_i^* \in \mathbb{K}_i (i \geq 1)$, there must exist an vector $\boldsymbol{k}_{i-1}^* \in \mathbb{K}_{i-1}$ such that $\boldsymbol{k}_{i-1}^*$ can propagate to $\boldsymbol{k}_i^*$ by CBDP propagation rules. Furthermore, for $(\boldsymbol{k}_0^*, \boldsymbol{k}_1^*, ..., \boldsymbol{k}_r^*) \in \mathbb{K}_0 \times \mathbb{K}_1 \times ... \times \mathbb{K}_r$, if $\boldsymbol{k}_{i-1}^*$ can propagate to $\boldsymbol{k}_i^*$ for all $i \in \{1, 2, \ldots, r\}$, we call $(\boldsymbol{k}_0^*, \boldsymbol{k}_1^*, \ldots, \boldsymbol{k}_r^*)$ an r-round CBDP trail.*

With the help of division trail, finding the CBDP is transformed into a problem of finding a division trail ended at a unit vector. For more details please refer to [34].

3 The MILP Model for BDPT

Suppose E_r is a r-round iterated block cipher whose round function f_i for $i \in [r]$ consists of a non-linear layer, linear layer, and Key-Xor operation. Let f_k^i be the Key-Xor operation, and f_e^i be the rest of the operations in the ith round function f_i i.e.

$$f_i = f_k^i \circ f_e^i$$

Let, the input multiset $\mathbb{X}$ to the block cipher E_r has initial BDPT as $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$, and for any $i \in [r]$, we denote the output BDPT as $D^{1^n}_{\mathbb{K}_i,\mathbb{L}_i}$. Now, for the operation f_e^i, we denote the BDPT propagation as

$$f_e^i(\mathbb{K}_{i-1}) = \mathbb{K}^*_{i-1}, \quad f_e^i(\mathbb{L}_{i-1}) = \mathbb{L}^*_{i-1}$$

We can evaluate the BDPT propagation for $\mathbb{K}$ (*unknown* subset) and $\mathbb{L}$ (1 subset) independently as per the BDPT propagation rules for linear and non-linear layers. Now, for the operation f_k^i, according to the BDPT Rule 1 some new vectors which are produced from the vectors in $\mathbb{L}^*_{i-1}$ and those new vectors along with the vectors in $\mathbb{K}^*_{i-1}$ are the vectors in the set $\mathbb{K}_i$, and the set $\mathbb{L}_i$ is same as $\mathbb{L}^*_{i-1}$.

Now, we divide the operation f_k^i into two parts say f_1^i, f_2^i such that f_1^i is the operation where new elements are produced from each elements in $\mathbb{L}^*_{i-1}$ according to BDPT Rule 1, and f_2^i is the operation which includes the new vectors and the vectors from $\mathbb{K}^*_{i-1}$ in $\mathbb{K}_i$ which is as follows:

$$(\mathbb{K}_i, \mathbb{L}_i) = f_k^i(\mathbb{K}^*_{i-1},\ \mathbb{L}^*_{i-1}) = (f_2^i(f_1^i(\mathbb{L}^*_{i-1}),\ \mathbb{K}^*_{i-1}),\ \mathbb{L}^*_{i-1}) \tag{1}$$

Precisely, f_2^i is the union operation i.e. $\mathbb{K}_i = f_1^i(\mathbb{L}^*_{i-1}) \ \cup\ \mathbb{K}^*_{i-1}$.

To model the propagation of BDPT for the operations f_e^i and f_k^i for all $i \in [r]$, we reintroduce a notion named ***BDPT trail***.[1]

Definition 4 (BDPT Trail) *Let $\mathbb{X}$ be the input multiset to the block cipher which has initial BDPT $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\ \mathbb{L}_0=\{\boldsymbol{l}\}}$, and denote the BDPT after r-round propagation through f_e^i, f_k^i for all $i \in [r]$ by $D^{1^n}_{\mathbb{K}_r,\ \mathbb{L}_r}$, where $r \geq 1$. Thus we have the following chain of BDPT propagations:*

$$\begin{tikzcd}
\{\boldsymbol{k}\} \triangleq \mathbb{K}_0 \arrow[r, "f_e^1"] & \mathbb{K}_0^* \arrow[dr, dash] & & \mathbb{K}_1 \arrow[r, "f_e^2"] & \mathbb{K}_1^* \cdots\cdots \mathbb{K}_{r-1} \arrow[r, "f_e^r"] & \mathbb{K}_{r-1}^* \arrow[dr, dash] & & \mathbb{K}_r \\
& & f_k^1 \arrow[ur] \arrow[dr] & & & & f_k^r \arrow[ur] \arrow[dr] & \\
\{\boldsymbol{l}\} \triangleq \mathbb{L}_0 \arrow[r, "f_e^1"] & \mathbb{L}_0^* \arrow[ur, dash] & & \mathbb{L}_1 \arrow[r, "f_e^2"] & \mathbb{L}_1^* \cdots\cdots \mathbb{L}_{r-1} \arrow[r, "f_e^r"] & \mathbb{L}_{r-1}^* \arrow[ur, dash] & & \mathbb{L}_r
\end{tikzcd}$$

where $\mathbb{K}^*_{i-1} = f_e^i(\mathbb{K}_{i-1})$, $\mathbb{L}^*_{i-1} = f_e^i(\mathbb{L}_{i-1})$, and $(\mathbb{K}_i,\ \mathbb{L}_i) = f_k^i(\mathbb{K}^*_{i-1},\ \mathbb{L}^*_{i-1})$ for all $1 \leq i \leq r$. Moreover, for any vector tuple $(\boldsymbol{k_i},\ \boldsymbol{l_i})$, $\boldsymbol{k_i} \in \mathbb{K}_i$, and $\boldsymbol{l_i} \in \mathbb{L}_i$ $(i \in [r])$, there must exist $(\boldsymbol{k^*_{i-1}},\ \boldsymbol{l^*_{i-1}})$, where $\boldsymbol{k^*_{i-1}} \in \mathbb{K}^*_{i-1}$, and $\boldsymbol{l^*_{i-1}} \in \mathbb{L}^*_{i-1}$ such that $\boldsymbol{k^*_{i-1}} \in \mathbb{K}^*_{i-1}$ propagate to $(\boldsymbol{k_i},\ \boldsymbol{l_i})$ by BDPT propagation rule of Key-Xor, and

[1] In [24], the authors have defined ***BDPT trail***. We actually rewrite it according to our notations.

there must exist $(\boldsymbol{k_{i-1}},\ \boldsymbol{l_{i-1}}) \in \mathbb{K}_{i-1} \times \mathbb{L}_{i-1}$ *such that* $\boldsymbol{k_{i-1}}$ *propagate to* $\boldsymbol{k^*_{i-1}}$*, and* $\boldsymbol{l_{i-1}}$ *propagate to* $\boldsymbol{l^*_{i-1}}$ *by BDPT propagation rules of linear and non-linear layers. Furthermore, for* $(\boldsymbol{k}_0,\ \boldsymbol{l}_0), \ldots, (\boldsymbol{k}_r,\ \boldsymbol{l}_r) \in \mathbb{K}_0 \times \mathbb{L}_0 \times \ldots \times \mathbb{K}_r \times \mathbb{L}_r$*, if* $(\boldsymbol{k}_{i-1},\ \boldsymbol{l}_{i-1})$ *can propagate to* $(\boldsymbol{k}_i,\ \boldsymbol{l}_i)$ *for all* $i \in \{1, 2, \ldots, r\}$*, we call*

$$(\boldsymbol{k}_0,\ \boldsymbol{l}_0) \xrightarrow{f_e^1,\, f_k^1} (\boldsymbol{k}_1,\ \boldsymbol{l}_1) \xrightarrow{f_e^2,\, f_k^2} \ldots \xrightarrow{f_e^r,\, f_k^r} (\boldsymbol{k}_r,\ \boldsymbol{l}_r)$$

an r*-round BDPT trail.*

Now, to model BDPT trail, we propose Proposition 1 according to Definition 4.

Proposition 1. *Let the input multiset* $\mathbb{X}$ *has initial BDPT* $D^{1^n}_{\{\boldsymbol{k}\},\,\{\boldsymbol{l}\}}$ *and* $D^{1^n}_{\mathbb{K}_r,\mathbb{L}_r}$ *denote the BDPT of the output multiset after* r*-round propagation. Then, the set of first components of the last vectors of all* r*-round BDPT trails which starts with the vector* $(\boldsymbol{k}, \boldsymbol{l})$ *is equal to the set* $\mathbb{K}_r$ *and the set of second components of the last vectors of all* r*-round BDPT trails which starts with the vector* $(\boldsymbol{k}, \boldsymbol{l})$ *is equal to the set* $\mathbb{L}_r$*.*

Proof of this Proposition 1 directly follows from Definition 4.

3.1 Some Observations on BDPT Propagation Rule for S-box

S-box is an important component of block ciphers. For a lot of block ciphers it is the only non-linear part. Although any Boolean function can be evaluated by using three rules (COPY, XOR, AND), the propagation requires much time and memory complexity when Boolean function is complex. Inspired by the algorithm of calculating CBDP trails of S-box [34], Wang *et al.* proposed a generalized method to calculate BDPT division trails of S-box in [32] and we have mentioned the rule in BDPT Rule 2.

Let, the input BDPT of S-box is $D^{1^n}_{\mathbb{K},\ \mathbb{L}=\{\boldsymbol{l}\}}$, and according to the BDPT Rule 2, we have found the sets $\underline{\mathbb{K}}$, and $\underline{\mathbb{L}}$ from $\mathbb{K}$ and $\mathbb{L}$ respectively as follows:

$$\begin{cases} \underline{\mathbb{K}} = \{\boldsymbol{u'} \in \mathbb{F}_2^n \,|\, \text{for any } \boldsymbol{u} \in \mathbb{K},\ \text{if } \boldsymbol{y}^{\boldsymbol{u'}} \text{ contains any term } \boldsymbol{x}^{\boldsymbol{v}} \text{ satisfying } \boldsymbol{v} \succeq \boldsymbol{u}\} \\ \underline{\mathbb{L}} = \{\boldsymbol{u} \in \mathbb{F}_2^n \,|\, \boldsymbol{y}^{\boldsymbol{u}} \text{ contains the term } \boldsymbol{x}^{l}\} \end{cases} \tag{2}$$

Now, according to the BDPT Rule 2, the output BDPT would be $D^{1^n}_{\mathbb{K}',\ \mathbb{L}'}$ which is as follows:

$$\mathbb{K}' = \boldsymbol{Reduce0}(\underline{\mathbb{K}}),\ \mathbb{L}' = \boldsymbol{Reduce1}(\underline{\mathbb{K}},\ \underline{\mathbb{L}})$$

Therefore, it is obvious that, $\mathbb{K}' \subseteq \underline{\mathbb{K}}$, and $\mathbb{L}' \subseteq \underline{\mathbb{L}}$. Here, we come to two observations as follows:

Observation 1. $\mathbb{L}'$ *does not contain* $\mathbf{1}$ *vector.*

According to the BDPT propagation rule of S-box, as $\mathbb{L}' = \boldsymbol{Reduce1}(\underline{\mathbb{K}},\ \underline{\mathbb{L}})$, and for any $\boldsymbol{u} \in \underline{\mathbb{K}}$, $\mathbf{1} \succeq \boldsymbol{u}$, then $\mathbb{L}'$ does not contain $\mathbf{1}$ vector.

Observation 2. *If* $\mathbb{L} = \{\mathbf{0}\}$*, then* $\mathbb{L}' = \{\mathbf{0}\}$*.*

Whenever, $\mathbb{L} = \{\mathbf{0}\}$, then $\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\mathbf{0}} = 1$ which implies that the input multiset $\mathbb{X}$ contains a constant term. Therefore, for all $\boldsymbol{u} \succ \mathbf{0}$, $\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{u}}$ = unknown. Hence, trivially $\bigoplus_{\boldsymbol{y} \in \mathbb{Y}} \boldsymbol{y}^{\mathbf{0}} = 1$ and $\mathbb{L}' = \{\mathbf{0}\}$ where $\mathbb{Y}$ is the output multiset.

Therefore, given an n-bit S-box and its input BDPT $D^{1^n}_{\mathbb{K}=\{\boldsymbol{k}\},\ \mathbb{L}=\{\boldsymbol{l}\}}$, BDPT Rule 2 returns the output BDPT $D^{1^n}_{\mathbb{K}',\ \mathbb{L}'}$. Thus for any vector $\boldsymbol{k'} \in \mathbb{K}'$, $(\boldsymbol{k}, \boldsymbol{k'})$ is a valid division trail for $\mathbb{K}'$ of the S-box. Similarly, this holds for $\mathbb{L}'$ as well. We know that, the vector $\boldsymbol{l}$ does not affect the propagation of vector $\boldsymbol{k}$ through the S-box, we will obtain a complete list of the division trail for $\mathbb{K}'$ by traversing $\boldsymbol{k} \in \mathbb{F}_2^n$ [34].

Similarly, for a certain input vector $\boldsymbol{l} \in \mathbb{F}_2^n$, we will obtain a certain set of division trails for $\underline{\mathbb{L}}$ using Eq. 2 and then using Observation 1, and Observation 2 we will remove some invalid division trails from $\underline{\mathbb{L}}$ and obtain a set of division trail for $\mathbb{L}'$. Therefore, if we try all the 2^n possible input vector $\boldsymbol{l}$, we will get a complete list of division trails for $\mathbb{L}'$.

In [32], the authors included some invalid BDPT trail for $\mathbb{L}'$ set while obtaining a complete list of division trails for $\mathbb{L}'$. In [24], the authors have removed those invalid BDPT trail from $\mathbb{L}'$ by introducing another algorithm which is actually equivalent to the algorithm of finding BDPT trail of S-box in [32] and by traversing $\boldsymbol{k} \in \mathbb{F}_2^n$. Now, our approach is similar to their idea [24] in a much simplified manner using two observations from BDPT Rule 2 which was introduced in [32].

In the full version of this paper [10], we present the complete lists of all the division trails for $\mathbb{L}$ of PRINCE S-box according to our method which is same if we apply the method the authors described in [24]. Therefore, after getting the BDPT trails for $\mathbb{K}$ and $\mathbb{L}$ of S-box, we construct the linear inequalities using the method described in [34] whose feasible solutions are exactly those BDPT trails which are shown in the full version of this paper [10].

3.2 MILP Model of BDPT for Complex Linear Layer

In this section, we establish the idea to construct MILP model of BDPT for complex linear layer represented by a matrix $M = (m_{i,j})_{s \times s} \in \mathbb{F}_{2^m}^{s \times s}$. Given the irreducible polynomial of the field $\mathbb{F}_2^m$ where the multiplications operate, the representation of the matrix over $\mathbb{F}_2$ is unique, which we call the primitive matrix of M and is denoted by $M' = (m'_{i,j})_{n \times n}$ where $m'_{i,j} \in \mathbb{F}_2$ and $n = m \times s$. Therefore, if each $m_{i,j}$ in M which is a polynomial in the extension field $F_{2^m} \simeq \mathbb{F}[x]/(f)$, where f is the irreducible polynomial over $\mathbb{F}_2$ with degree m, is either 0 or 1 then M is called binary matrix and otherwise M is non-binary matrix.

Therefore, block ciphers with complex linear layer can be partitioned into two parts: (i) Block ciphers with binary linear layer and (ii) Block ciphers with non-binary linear layer, depending on the binary or non-binary matrix as its linear layer. Examples of block ciphers having binary linear layer are MIDORI, SKINNY, CRAFT, PRINCE, MANTIS etc. and AES, LED, KLEIN, PRIDE etc. have non-binary linear layer.

Now, an obvious way to model the BDPT propagation through any complex linear layer i.e. $\boldsymbol{u}_1 \overset{M}{\to} \boldsymbol{v}_1$ in $\mathbb{K}$ subset, and $\boldsymbol{u_2} \overset{M}{\to} \boldsymbol{v}_2$ in $\mathbb{L}$ subset is that one can introduce some auxiliary binary variables and decompose it into the COPY and XOR operations. Therefore, by following the BDPT propagation rule of COPY and XOR, BDPT propagation through linear layer can be modelized. The obvious advantage of this model is that using this technique we can model BDPT propagation of any complex linear layer.

In [21,37], the authors have shown that using this technique one may introduce many invalid division trails in $\mathbb{K}$ subset. Now, here we are going to show that if we use this COPY-XOR technique to handle binary linear layer then many invalid division trails may be added to the $\mathbb{L}$ subset as well which we have shown by giving an example in the full version of this paper [10].

Exact BDPT Modelization for Ciphers Having Binary Linear Layer. Given a binary matrix $M = (m_{i,j})_{s\times s} \in \mathbb{F}_{2^m}^{s\times s}$, and denote $n = m \times s$, we can derive an equivalent matrix working at a bit level which is called primitive matrix $M' = (m'_{i,j})_{n\times n} \in \mathbb{F}_2^{n\times n}$. Now, M' has $n = ms$ number of rows which we denote say $R_0, R_1, \dots, R_{n-1}$, and define a set of all rows $\mathcal{R} = \{R_i \,|\, 0 \le i \le n-1\}$. Therefore, we can construct m disjoint sets $\mathcal{R}_0,\ \mathcal{R}_1, \dots, \mathcal{R}_{m-1}$ in the following way:

$$\mathcal{R}_i = \{R_{mj+i} \,|\, 0 \le j \le s-1\}\ \textit{for all}\ 0 \le i \le m-1 \tag{3}$$

Now, it is obvious that $\sqcup_{i=0}^{m-1}\mathcal{R}_i = \mathcal{R}$, and $\mathcal{R}_i$ contains exactly a number s of rows from M' where $0 \le i \le m-1$. Here we come to an important property that the rows in different sets have no common nonzero entries in the same column, which is the key feature of a binary matrix. Exploiting this property of a binary matrix, the binary linear layer can actually be seen as the application of m many s-bit S-box with algebraic degree 1 in parallel.

Therefore, if $\boldsymbol{x} = (x_0, x_1, \dots, x_{n-1})$, and $\boldsymbol{y} = (y_0, y_1, \dots, y_{n-1})$ are corresponding input and output variables w.r.t the linear layer i.e. $\boldsymbol{y} = M' \cdot \boldsymbol{x}$, then we can write ANF of m many s-bit S-box with algebraic degree 1 as follows:

$$\begin{cases} S_0(\boldsymbol{x_0}) &= (R_0^0 \cdot \boldsymbol{x_0},\ R_m^0 \cdot \boldsymbol{x_0}, \dots, R_{(s-1)m}^0 \cdot \boldsymbol{x_0}) \\ S_1(\boldsymbol{x_1}) &= (R_1^1 \cdot \boldsymbol{x_1},\ R_{m+1}^1 \cdot \boldsymbol{x_1}, \dots, R_{(s-1)m+1}^1 \cdot \boldsymbol{x_1}) \\ \quad\vdots & \\ S_{m-1}(\boldsymbol{x_{m-1}}) &= (R_{m-1}^{m-1} \cdot \boldsymbol{x_{m-1}},\ R_{2m-1}^{m-1} \cdot \boldsymbol{x_{m-1}}, \dots, R_{sm-1}^{m-1} \cdot \boldsymbol{x_{m-1}}) \end{cases}$$

where R^i_{mj+i} is a vector which belongs to the set $\mathbb{F}_2^s$ such that $R^i_{mj+i} = (m'_{mj+i,\,i},\ m'_{mj+i,\,m+i}, \dots, m'_{mj+i,\,(s-1)m+i})$, and $\boldsymbol{x_i} = (x_i,\ x_{m+i}, \dots, x_{(s-1)m+i}) \in \mathbb{F}_2^s$ where $i = 0, 1, \dots, m-1$, and $j = 0, 1, \dots, s-1$.

An Example of Exact BDPT Modelization of Binary Matrix. The MixColumns matrix M of the block cipher MANTIS which is as follows:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \in \mathbb{F}_{2^4}^{4\times 4}$$

Therefore, for this example, $s = 4$, and $m = 4$, and the primitive matrix M' corresponding to the matrix M is a 16×16 matrix where each matrix element is either 0 or 1 i.e. the primitive matrix $M' \in \mathbb{F}_2^{16\times 16}$ is as follows:

$$M' = \begin{pmatrix}
0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1\\
1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1\\
1&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&1&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&1\\
1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0
\end{pmatrix} \in \mathbb{F}_2^{16\times 16}$$

Now, we can easily conclude that applying the matrix M' to a vector $\boldsymbol{x} = (x_0, x_1, \ldots, x_{15}) \in \mathbb{F}_2^{16}$ is actually equivalent to performing the following 4-bit S-box in parallel:

$$S_i(x_i, x_{i+4}, x_{i+8}, x_{i+12}) = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_i \\ x_{i+4} \\ x_{i+8} \\ x_{i+12} \end{pmatrix},\ i \in \{0, 1, 2, 3\}$$

Therefore, we can construct exact BDPT trail for $\mathbb{K}$ and $\mathbb{L}$ for the mixcolumn operation and the linear inequalities whose feasible solutions are exactly those BDPT trail.

Now, the exact BDPT modelization of S-box we have discussed in the previous section. Applying that approach we can get the exact BDPT trail through the binary linear layer and then we can easily represent the BDPT trails of binary linear layer as linear inequalities following the approach mentioned in [34]. Thus, we give a way to generate a set of inequalities that exactly model the valid BDPT propagations through a binary linear layer. For the ciphers with non-binary linear layer, we decompose its linear layer through the COPY and

XOR operation trivially and generate a set of linear inequalities that model the propagations through the linear layer.

3.3 MILP Model of BDPT for Key-XOR

In this section, we explain how to construct MILP model of BDPT for the Key-Xor operation. As per the notation discussed above E_r is the r-round block cipher where we denote f_i is the ith round function and f_k^i is the ith round Key-Xor operation. Moreover, we denote the initial and output BDPT for the Key-Xor operation as $(\mathbb{K}_{i-1}^*,\ \mathbb{L}_{i-1}^*)$, and $(\mathbb{K}_i,\ \mathbb{L}_i)$ respectively. Therefore, as per BDPT Rule 1, we decompose f_k^i into two operations say f_1^i which actually produces some new elements from each elements of $\mathbb{L}_{i-1}^*$ and f_2^i which includes the new vectors and the vectors from $\mathbb{K}_{i-1}^*$ in $\mathbb{K}_i$ which is described in Eqn 1. Hence, we model the operations f_1^i, and f_2^i which jointly present the MILP model for Key-Xor operation.

Table 2. Trails Corresponding to the Function f_1^i

$(l_0,\ l_1,\ l_2,\ l_3)$	$(l_0',\ l_1',\ l_2',\ l_3')$
$(0,\ 0,\ l_2, l_3)$	$(0,\ 1,\ l_2,\ l_3)$, $(1,\ 0,\ l_2,\ l_3)$, $(1,\ 1,\ l_2,\ l_3)$
$(0,\ 1,\ l_2, l_3)$	$(1,\ 1,\ l_2,\ l_3)$,
$(1,\ 0,\ l_2, l_3)$	$(1,\ 1,\ l_2,\ l_3)$,
$(1,\ 1,\ l_2, l_3)$	X

Modeling $\boldsymbol{f_1^i}$. In many ciphers, round key is only XORed with a part of block. Without loss of generality, we assume that the round key is XORed with the left s $(0 \leq s \leq n-1)$ bits. Let, $\mathbb{L}_{i-1}^* \subseteq \mathbb{F}_2^4$ and $s = 2$ i.e. round key is XORed with the leftmost 2 bits. Therefore, according to the BDPT rule 1, f_1^i function creates $\boldsymbol{l'} = (l_0',\ l_1',\ l_2',\ l_3')$ from $\boldsymbol{l} = (l_0,\ l_1,\ l_2,\ l_3)$ where for every vector $\boldsymbol{l} \in \mathbb{L}_{i-1}^*$ satisfying $l_i = 0$, $l_i' = l_i \vee 1$ where $i \in \{0, 1\}$ and $l_j' = l_j$ for all $j = 2, 3$. Therefore, we write the propagation table (Table 2) corresponding to the function f_1^i using which we construct linear inequalities whose feasible solutions are exactly those trails. Now, we are ready to give linear inequalities description of these trails listed in Table 2 as follows:

$$\begin{cases} \qquad\qquad l_j' & \geq\ l_j, \quad \text{for } j = 0, 1 \\ \qquad\qquad l_j' & =\ l_j, \quad \text{for } j = 2, 3 \\ 2\sum_{j=0}^{1} l_j' - \sum_{j=0}^{1} l_j & \geq\ 2 \\ \sum_{j=0}^{3} l_j' - \sum_{j=0}^{3} l_j & \geq\ 1 \end{cases} \tag{4}$$

where $l_0',\ l_1',\ l_2',\ l_3',\ l_0,\ l_1,\ l_2,\ l_3$ are binaries.

Apparently, all feasible solutions of the inequalities in Eq. 4 corresponding to $\boldsymbol{l}$, and $\boldsymbol{l'}$ are exactly the trails of f_1^i function described above in Table 2. Similarly, for a n-bit block cipher where $\mathbb{L}^*_{i-1} \subseteq \mathbb{F}_2^n$, and round key is XORed with the leftmost s $(0 \leq s \leq n-1)$ bits, the linear inequalities we get which describe the trails $\boldsymbol{l} \xrightarrow{f_1^i} \boldsymbol{l'}$ as follows:

$$\begin{cases} l'_j & \geq \; l_j, \quad \text{for } j = 0,\, 1,\, ...,\, s-1 \\ l'_j & = \; l_j, \quad \text{for } j = s,\, s+1 \ldots, n-1 \\ s\sum_{j=0}^{s-1} l'_j - (s-1)\sum_{j=0}^{s-1} l_j & \geq \; s \\ \sum_{j=0}^{n-1} l'_j - \sum_{j=1}^{n} l_j & \geq \; 1 \end{cases} \tag{5}$$

where $l'_0,\, l'_1,\, ...,\, l'_{n-1}, l_0,\, l_1,\, ...,\, l_{n-1}$ are binaries.

Modeling $\boldsymbol{f_2^i}$. After applying f_1^i on each element of the set $\mathbb{L}^*_{i-1}$, we get the set say $\mathbb{L}'_{i-1}$ as follows:

$$\mathbb{L}'_{i-1} = \{\boldsymbol{l'} \in \mathbb{F}_2^n \,|\, f_1^i(\boldsymbol{l}) = \boldsymbol{l'},\; \forall\, \boldsymbol{l} \in \mathbb{L}^*_{i-1}\}$$

Now, from BDPT Rule 1 we know that:

$$f_2^i(\mathbb{K}^*_{i-1},\; \mathbb{L}'_{i-1}) = \mathbb{K}^*_{i-1} \cup \mathbb{L}'_{i-1} = \mathbb{K}_i$$

Therefore, to model f_2^i, we define another function $g : (\mathbb{F}_2^2 \setminus \{(0,0),(1,1)\}) \times \mathbb{K}^*_{i-1} \times \mathbb{L}'_{i-1} \rightarrow \mathbb{K}_i$ such that:

$$g(\lambda_0, \lambda_1, \boldsymbol{k^*}, \boldsymbol{l'}) = (\lambda_0 \wedge k^*_0, \ldots, \lambda_0 \wedge k^*_{n-1}) \oplus (\lambda_1 \wedge l'_0, \ldots, \lambda_1 \wedge l'_{n-1}) \tag{6}$$

where $\lambda = (\lambda_0,\, \lambda_1) \in \mathbb{F}_2^2 \setminus \{(0,0),(1,1)\}$, and $\boldsymbol{k^*} = (k^*_0,\, \ldots, k^*_{n-1})$, and $\boldsymbol{l'} = (l'_0, \ldots, l'_{n-1})$. Therefore, from the definition of g we can easily conclude that $\mathbb{K}_i$ contain all the elements of $\mathbb{L}'_{i-1}$, and $\mathbb{K}^*_{i-1}$. Hence, modeling g is actually equivalent to modeling f_2^i. Now, we are going to construct the linear inequalities whose feasible solutions are exactly the g function trail. In order to do that first we have to construct the linear inequalities which are sufficient to describe the propagation $(a,\, b) \xrightarrow{\wedge} c$ where $a,\, b,\, c \in \mathbb{F}_2$ which is as follows:

$$\begin{cases} a - c & \geq \; 0 \\ b - c & \geq \; 0 \\ a + b - c & \leq \; 1 \end{cases} \tag{7}$$

where a, b, c are binaries. Therefore, using Eq. 6 and Eq. 7 we can easily conclude that the following inequalities are sufficient to describe the propagation of g function i.e. $(\lambda_0,\, \lambda_1,\, \boldsymbol{k}^*,\, \boldsymbol{l}') \xrightarrow{g} \boldsymbol{k}$:

$$\begin{cases} \lambda_0 - p_j & \geq \; 0, \quad \text{for } j = 0,\, 1,\, ...,n-1 \\ k_j^* - p_j & \geq \; 0, \quad \text{for } j = 0,\, 1,\, ...,n-1 \\ \lambda_0 + k_j^* - p_j & \leq \; 1, \quad \text{for } i = 0,\, 1,\, ...,n-1 \\ \lambda_1 - q_j & \geq \; 0, \quad \text{for } i = 0,\, 1,\, ...,n-1 \\ l_j' - q_j & \geq \; 0, \quad \text{for } i = 0,\, 1,\, ...,n-1 \\ \lambda_1 + l_j' - q_j & \leq \; 1, \quad \text{for } i = 0,\, 1,\, ...,n-1 \\ p_j + q_j - k_j & = \; 0, \quad \text{for } j = 0,\, 1,\, ...,n-1 \\ \lambda_0 + \lambda_1 & = \; 1 \end{cases} \tag{8}$$

where $p_0, ..., p_{n-1}, q_0, ...,\, q_{n-1}, l_0', ..., l_{n-1}', k_0, ..., k_{n-1}, k_0^*, ..., k_{n-1}^*, \lambda_0, \lambda_1$ are binaries and $p = (p_0,\, p_1,\, ..., p_{n-1})$, $q = (q_0,\, q_1,\, ..., q_{n-1})$ are auxiliary variables. Hence Eq. 8 and Eq. 5 describe the complete MILP model of the Key-XOR operation w.r.t BDPT.

3.4 MILP Model Construction of *r*-Round Function

For all the functions based on these above mentioned operations, we are finally making a set of linear inequalities depicting one round BDPT propagation. In order to construct an MILP model for r round BDPT propagation we have to iterate this above mentioned procedure r times and finally we conclude upon getting a system of linear inequalities $\mathcal{L}$ which we describe in Algorithm 1.

Algorithm 1 constructs a system of linear inequalities which charecterizes all r-round BDPT trails i.e.

$$(\boldsymbol{k}^0 = \boldsymbol{k},\, \boldsymbol{l}^0 = \boldsymbol{l}) \xrightarrow{f_1} (\boldsymbol{k}^1,\, \boldsymbol{l}^1) \xrightarrow{f_2} \ldots \xrightarrow{f_r} (\boldsymbol{k}^r,\, \boldsymbol{l}^r)$$

Therefore, we have to construct MILP model using $\mathcal{L}$ and appropriate initial and stopping rules and the search algorithm in order to find integral distinguisher.

4 Automatic Search Algorithm for *r*-Round Integral Distinguisher

In this section, we first study the initial BDPT and stopping rule to use when searching for integral distinguisher based on BDPT. From Algorithm 1 we got the linear inequality system $\mathcal{L}$ with the input vector $\boldsymbol{k}$ and $\boldsymbol{l}$. Now, we convert the stopping rule into an objective function and combining $\mathcal{L}$ and objective function, we construct the MILP model $\mathcal{M}_{\mathbb{K},\,\mathbb{L}}$. At last we propose an algorithm to search integral distinguisher based on BDPT given the initial BDPT $D^{1^n}_{\{\boldsymbol{k}\},\,\{\boldsymbol{l}\}}$ for an n-bit block cipher and prove the correctness of the algorithm.

4.1 Initial BDPT

In [31], Todo and Morii set the initial BDPT as ($\mathbb{K} = \{\mathbf{1}\}$, $\mathbb{L} = \{7fffffff\}$) to search the BDPT of SIMON32, where the active bits of the vector $\boldsymbol{l}$ are set as 1 and 0 for constant bits. Hence we do the same. Let the initial input BDPT variables are $\boldsymbol{k}^0 = (k_0^0, k_1^0, ..., k_{n-1}^0)$, and $\boldsymbol{l}^0 = (l_0^0, l_1^0, ..., l_{n-1}^0)$ where n is the block size. The constraints on k_i^0 and l_i^0 are

$$k_i^0 = 1 \quad \text{for } i = 0, 1, ..., n-1$$

$$l_i^0 = \begin{cases} 1, & \text{if } i-th \text{ bit is active} \\ 0, & \text{otherwise} \end{cases}$$

Algorithm 1: Computing A Set of Constraints Characterizing BDPT Propagation

```
Input: The initial input BDPT of an $n$-bit iterated cipher
       $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\}, \mathbb{L}_0=\{\boldsymbol{l}\}}$
       $\mathcal{L}_k(\mathbb{K}_{i-1}, \mathbb{K}^*_{i-1})$: a constraint set of linear inequalities whose
       feasible solutions are all division trails from the set $\mathbb{K}_{i-1}$ to set
       $\mathbb{K}^*_{i-1}$, $\forall\, i \in [r]$.
       $\mathcal{L}_l(\mathbb{L}_{i-1}, \mathbb{L}^*_{i-1})$: a constraint set of linear inequalities whose
       feasible solutions are all division trails from the set $\mathbb{L}_{i-1}$ to set
       $\mathbb{L}^*_{i-1}$, $\forall\, i \in [r]$.
       $New_k(\mathbb{L}^*_{i-1}, \mathbb{L}'_{i-1})$: a constraint set of linear inequalities whose
       feasible solutions are all $f_1^i$ function trails, $\forall\, i \in [r]$.
       $Union_k(\mathbb{L}'_{i-1}, \mathbb{K}^*_{i-1}, \mathbb{K}_i)$: a constraint set of linear inequalities
       whose feasible solutions are all $f_2^i$ function trails, $\forall\, i \in [r]$.
Output: A constraint set of linear inequalities $\mathcal{L}$ describing $r$-round
        BDPT propagation
begin
    $\mathcal{L} = \emptyset$, $\mathcal{C}^i = \mathcal{C}^{i,*} = \emptyset$ where $i = 1, 2, \ldots, r$
    Allocate $n$-bit variables $\boldsymbol{k}^i$, $\boldsymbol{l}^i$ to denote vectors in the set $\mathbb{K}_i$, $\mathbb{L}_i$
    respectively where $i = 0, 1, \ldots, r$
    Allocate $n$-bit variables $\boldsymbol{l}^{i,*}$, $\boldsymbol{p}^i$, and $\boldsymbol{k}^{i,*}$ to denote vectors in the
    set $\mathbb{L}^*_i$, $\mathbb{L}^{'}_i$, and $\mathbb{K}^*_i$ respectively where $i = 0, 1, \ldots, r-1$
    $\mathcal{L} \leftarrow (\boldsymbol{k}^0 = \boldsymbol{k})$
    $\mathcal{L} \leftarrow (\boldsymbol{l}^0 = \boldsymbol{l})$
    for $(i = 1\,; i \le r\,; i++)$ do
        $\mathcal{C}^i \leftarrow \mathcal{L}_l(\mathbb{L}_{i-1}, \mathbb{L}^*_{i-1}) \cup \mathcal{L}_k(\mathbb{K}_{i-1}, \mathbb{K}^*_{i-1})$
        $\mathcal{C}^{i,*} \leftarrow New_k(\mathbb{L}^*_{i-1}, \mathbb{L}'_{i-1})$
        $\mathcal{C}^{i,*} \leftarrow Union_k(\mathbb{L}'_{i-1}, \mathbb{K}^*_{i-1}, \mathbb{K}_i)$
        $\mathcal{L} \leftarrow (\boldsymbol{l}^{i-1,*} = \boldsymbol{l}^i)$
        $\mathcal{L} \leftarrow (\mathcal{C}^i \cup \mathcal{C}^{i,*})$

    end
    return $\mathcal{L}$
end
```

4.2 Stopping Rule

Our automatic search model only focuses on the parity of one output bit. Without loss of generality, we consider the q-th output bit. After r round, the output set has BDPT $D^{1^n}_{\mathbb{K}_r,\mathbb{L}_r}$. Therefore, according to the Proposition 1, we know that the set of the first components of the last vectors of all r-round BDPT trails which start from the vector $(\boldsymbol{k}, \boldsymbol{l})$ is equal to $\mathbb{K}_r$. Hence, to check whether there exist any unit vector in the $\mathbb{K}_r$, the objective function can be set as follows:

$$Obj\ :\ Minimize(k_0^r + k_1^r + \ldots, k_{n-1}^r) \tag{9}$$

Similarly, according to the Proposition 1, the set of the second components of the last vectors of all r-round BDPT trails which start from the vector $(\boldsymbol{k}, \boldsymbol{l})$ is equal to $\mathbb{L}_r$. Thus, we can set the objective function as :

$$Obj\ :\ Minimize(l_0^r + l_1^r + \ldots, l_{n-1}^r) \tag{10}$$

Now, at first we construct the MILP model $\mathcal{M}_{\mathbb{K},\mathbb{L}}$ using the system of linear inequalities $\mathcal{L}$ we get from Algorithm 1 and the objective function defined in Eq. 9. Moreover, we construct another MILP model $\mathcal{M}_{\mathbb{L}}$ as follows:

$$\mathcal{M}_{\mathbb{L}} = ConstructModel(\mathcal{L}^*, Min(l_0^r + \ldots, l_{n-1}^r))$$

where $\mathcal{L}^*$ is the constraint set of linear inequalities whose feasible solutions are all division trails from the set $\mathbb{L}_0$ to $\mathbb{L}_r$.

Stopping Rule for the MILP Model $\mathcal{M}_{\mathbb{K},\mathbb{L}}$. To check whether $\mathbb{K}_r$ contains the unit vector $\boldsymbol{e}_q$ is equivalent to check whether the MILP model $\mathcal{M}_{\mathbb{K},\mathbb{L}}$ has feasible solution satisfying $\boldsymbol{k}^r = \boldsymbol{e}_q$. Therefore, we can set the stopping rule as:

$$k_j^r = \begin{cases} 1 & if\ j = q \\ 0 & otherwise \end{cases} \tag{11}$$

If $\mathcal{M}_{\mathbb{K},\mathbb{L}}$ has such feasible solutions, then the q-th output bit is unknown.

Stopping Rule for the MILP Model $\mathcal{M}_{\mathbb{L}}$. If $\mathbb{K}_r$ does not contain $\boldsymbol{e}_q$, then to check whether $\mathbb{L}_r$ contains $\boldsymbol{e}_q$ is equivalent to check whether the MILP model $\mathcal{M}_{\mathbb{L}}$ has feasible solution satisfying $\boldsymbol{l}^r = \boldsymbol{e}_q$. Therefore, we can set the stopping rule as :

$$l_j^r = \begin{cases} 1 & if\ j = q \\ 0 & otherwise \end{cases} \tag{12}$$

If both $\mathbb{K}_r$ and $\mathbb{L}_r$ do not contain $\boldsymbol{e}_q$, then q-th output bit is balanced. Otherwise, we need to count the number of feasible solutions satisfying $\boldsymbol{l}^r = \boldsymbol{e}_q$ of the model $\mathcal{M}_{\mathbb{L}}$. Therefore, the parity of q-th output bit is 0 or 1 if the number of solutions are even or odd respectively as $\mathbb{K}_r$ does not contain $\boldsymbol{e}_q$.

Algorithm 2: Deciding Parity of q-th Output Bit

```
Input: The $r$-round cipher $E_r$, the initial input BDPT of an $n$-bit
       iterated cipher $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$, the number $q$, and $\mathcal{L}_l(\mathbb{L}_{i-1}, \mathbb{L}_i)$:
       a constraint set of linear inequalities whose feasible solutions are
       all division trails from the set $\mathbb{L}_{i-1}$ to set $\mathbb{L}_i$, $\forall\, i \in [r]$.
Output: The balanced information of the $q$-th output bit based on
        BDPT
begin
    Allocate all the variables denoting the input and output BDPT
    $Obj_1$ = Minimize($k^r_0 + k^r_1 + \ldots, k^r_{n-1}$)
    $Obj_2$ = Minimize($l^r_0 + l^r_1 + \ldots, l^r_{n-1}$)
    Call Algorithm 1 and get a constraint set $\mathcal{L}$ whose feasible solutions
    are $r$-round BDPT trail
    $\mathcal{M}_{\mathbb{K},\mathbb{L}} = ConstructModel(\mathcal{L}, Obj_1)$
    $\mathcal{M}_{\mathbb{K},\mathbb{L}}.AddConstraint(\boldsymbol{k}^r = \boldsymbol{e}_q)$
    if the MILP model $\mathcal{M}_{\mathbb{K},\mathbb{L}}$ has solutions then
        return unknown
    end
    else
        $\mathcal{M}_{\mathbb{L}} = ConstructModel(\bigcup_{i=1}^{r} \mathcal{L}_l(\mathbb{L}_{i-1}, \mathbb{L}_i), Obj_2)$
        $\mathcal{M}_{\mathbb{L}}.AddConstraint(\boldsymbol{l}^0 = \boldsymbol{l})$
        if the MILP model $\mathcal{M}_{\mathbb{L}}$ has no feasible solution satisfying
        $\boldsymbol{l}^r = \boldsymbol{e}_q$ then
            return 0
        end
        else
            $\mathcal{M}_{\mathbb{L}}.AddConstraint(\boldsymbol{l}^r = \boldsymbol{e}^q)$
            Count the number of solutions in $\mathcal{M}_{\mathbb{L}}$
            if Count is even then
                return 0
            end
            else
                return 1
            end
        end
    end
end
```

4.3 Search Algorithm

We present the automatic search algorithm to find integral distinguisher based on BDPT, which decides the parity of the q-th output bit with the given initial BDPT $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$ for an n-bit block cipher. Firstly, we allocate all round variables and auxiliary variables. Therefore, we construct a MILP model $\mathcal{M}_{\mathbb{K},\mathbb{L}}$ that describes all r-round BDPT trails, and another MILP model $\mathcal{M}_{\mathbb{L}}$ that describes all r-round division trails for $\mathbb{L}$. Finally, using appropriate initial and stopping rules, we can obtain the parity of q-th output bit based on BDPT. We illustrate the whole framework in Algorithm 2.

4.4 Correctness of Search Algorithm

Let the initial input division property of an n-bit iterated cipher be $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$, and after r-round propagation, the output BDPT we denote as $D_{\mathbb{K}_r,\mathbb{L}_r}$. It is obvious that if $\boldsymbol{e}_q \in \mathbb{K}_r$, then the parity of q-th bit is *unknown* and if $\boldsymbol{e}_q$ does not belongs to $\mathbb{K}_r$ as well as $\mathbb{L}_r$, then the parity of q-th bit is 0.

Therefore, to prove correctness of Algorithm 2 we have to prove that if the q-th unit vector does not belong to $\mathbb{K}_r$ and belongs to $\mathbb{L}_r$, then the parity of q-th output bit is 0 or 1 provided the number of division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is even or odd respectively. We first prove the following Lemma:

Lemma 1. *Let $\mathbb{X} \subseteq \mathbb{F}_2^n$ has division property $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$ and after r-round propagation, the output set $\mathbb{Y}_r$ has division property $D^{1^n}_{\mathbb{K}_r,\mathbb{L}_r}$. For any $\boldsymbol{l'} \in \mathbb{L}_r$, if the number of division trail in $\mathbb{L}$ from $\boldsymbol{l}$ to $\boldsymbol{l'}$ is even, then there exist at least one j in $[r]$ s.t $\mathbb{L}_j$ contains at least one element $\boldsymbol{u}$ which is produced even number of times from the elements in $\mathbb{L}_{j-1}$.*

The proof is provided in the full version of this paper [10]. Therefore, using Lemma 1 we prove the final result as follows:

Proposition 2. *Let $\mathbb{X} \subseteq \mathbb{F}_2^n$ has division property $D^{1^n}_{\mathbb{K}_0=\{\boldsymbol{k}\},\mathbb{L}_0=\{\boldsymbol{l}\}}$ and after r-round propagation, the output set $\mathbb{Y}$ has division property $D^{1^n}_{\mathbb{K}_r,\mathbb{L}_r}$. If $\boldsymbol{e}_q$ doesn't belongs to the set $\mathbb{K}_r$, where $q \in [n]$, then we have:*

1. *If the number of division trail from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is even in $\mathbb{L}$, then $\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 0$.*
2. *If the number of division trail from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is odd in $\mathbb{L}$, then $\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 1$.*

Proof. Let $S \subseteq (\mathbb{F}_2^n)^{r+1}$ be the set which contains all the division trail in $\mathbb{L}$ from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ and $|S|$ is even. Now, by using Lemma 1, we can easily conclude that there exist at least one $j \in \{2, 3, ..., r\}$ s.t $\mathbb{L}_j$ contains an element $\boldsymbol{u}$ which is produced even number of times from the elements in $\mathbb{L}_{j-1}$. Without loss of generality we choose smallest such j.

According to the BDPT propagation rule of XOR and S-box, we can see that if an element $\boldsymbol{u}$ is produced even number of times in $\mathbb{L}_j$ from $\mathbb{L}_{j-1}$, then the following holds:

$$\bigoplus_{\boldsymbol{y}\in\mathbb{Y}_j} \boldsymbol{y}^{\boldsymbol{u}} = 0$$

where $\mathbb{Y}_j$ is the multiset whose BDPT is $D^{1^n}_{\mathbb{K}_j,\mathbb{L}_j}$ and that implies $\boldsymbol{u}$ shouldn't be in $\mathbb{L}_j$. Hence, all the division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ which contains the vector $\boldsymbol{u}$ are actually redundant and those number of redundant division trails must be even. Therefore, we can remove these redundant division trails from S and we can call the new set as S_1. It is trivial that either $|S_1|$ is even or $|S_1| = 0$.

Case-I. If $|S_1| = 0$, then it implies that all the division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ contains the element $\boldsymbol{u}$. Therefore, as $\boldsymbol{u}$ shouldn't be in $\mathbb{L}_j$, so $\boldsymbol{e}_q$ also shouldn't be in $\mathbb{L}_r$ and it is given that $\boldsymbol{e}_q$ doesn't belongs to $\mathbb{K}_r$ which means

$$\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} \boldsymbol{y}^{\boldsymbol{e}_q} = \bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 0.$$

Case-II. If $|S_1|$ is even, then in a similar way we can find even number of redundant division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ in $\mathbb{L}$ and construct S_2 from S_1 where $|S_2|$ is either even or 0 and so on.

As $|S|$ is finite, then after finitely many p steps, we must get some S_p s.t $|S_p| = 0$. Hence, $\boldsymbol{e}_q$ shouldn't be in $\mathbb{L}_r$ and it is given that $\boldsymbol{e}_q$ doesn't belongs to $\mathbb{K}_r$ which means

$$\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} \boldsymbol{y}^{\boldsymbol{e}_q} = \bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 0$$

which completes the first part of the proof.

Now, it is given that the number of division trail in $\mathbb{L}$ from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is odd. Similarly we can construct a set S' containing all such division trails. Therefore, there may or may not exist $j \in \{2, 3, ..., r\}$ s.t $\mathbb{L}_j$ contains an element $\boldsymbol{u}$ which is produced even number of times from the elements in $\mathbb{L}_{j-1}$.

Case-A. If there doesn't exist any such j, then by BDPT propagation rules, we can easily conclude that no division trail from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is redundant. Therefore, it implies that $\boldsymbol{e}_q$ belongs to $\mathbb{L}_r$ which means $\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 1$.

Case-B. If there exist some j s.t $\mathbb{L}_j$ contains an element $\boldsymbol{u}$ which is produced even number of times from the elements in $\mathbb{L}_{j-1}$, then similarly by the previous argument we can easily conclude that all the division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ which contains $\boldsymbol{u}$ are actually redundant. Therefore, we can remove these redundant division trails from S' and we can call the new set as S'_1. It is obvious that $|S'_1|$ is odd.

Now, continuing like this way, after finitely many steps we arrive at a situation where the number of remaining division trails from $\boldsymbol{l}$ to $\boldsymbol{e}_q$ is odd and no redundant division trails are left which implies e_q belongs to $\mathbb{L}_r$. Therefore, $\bigoplus_{\boldsymbol{y}\in\mathbb{Y}} \boldsymbol{y}^{\boldsymbol{e}_q} = \bigoplus_{\boldsymbol{y}\in\mathbb{Y}} y_q = 1$ which completes the second part of the proof. □

5 Applications to Block Ciphers

In this section, we apply our automatic search algorithm for BDPT to SIMON, SIMON(102), MANTIS, PRINCE, KLEIN, and PRIDE block ciphers. All the experiments are conducted on the platform Intel Core i5-8250U CPU @ 1.60 GHz, 8 G RAM, 64 bit Ubuntu 18.04.5 LTS. The optimizer we used to solve MILP models is Gurobi 9.1.2 [17]. For the integral distinguishers, '?' denotes the bit whose balanced information is unknown, '0' denotes the bit whose sum is zero, ‘1’ denotes the bit whose sum is 1. The detailed integral distinguishers of PRINCE, MANTIS, KLEIN and PRIDE are listed in supporting material of the full version of this paper [10].

5.1 Applications to PRINCE and MANTIS

In this section we present the application of our BDPT model to the cipher PRINCE and MANTIS which have binary matrices to conduct their mixcolumn operations in the round functions. Hence, we apply our method to model binary linear layer in BDPT and construct the MILP model efficiently. Then, choosing appropriate initial BDPT, we find improved integral distinguisher as follows:

Integral Attack on PRINCE. Block ciphers based on the reflection design strategy, introduced by PRINCE [8], are a popular choice for low-latency designs. PRINCE is the 64-bit block cipher which uses 128-bit key. The PRINCE cipher is the substitution-permutation network composed of 12 rounds. The 64-bit state can be organised as the 4×4 array of nibbles. For a complete specification and design rationale of the cipher, a reader is referred to [8].

We will denote the number of rounds of PRINCE as $a + b$ where a are the rounds before the middle layer, and b are the rounds after the middle layer. There are several attacks (Integral attack, higher order differential attack, boomerang attack) on PRINCE [2,25,27]. Now, in [15], the authors applied CBDP on PRINCE and found $2 + 1$ and $1 + 2$ round integral distinguishers which are best integral distinguisher till date.

For PRINCE, we find a $2 + 2$ round integral distinguisher which is one more round than the previous best results [15].

Integral Attack on MANTIS. MANTIS is a tweakable block cipher published at CRYPTO 2016 by Beierle *et al.* [6] and the cipher's structure is similar to PRINCE. This block cipher operate on a 64-bit message block and work with a 64-bit tweak and (64+64) bit key and has a SPN structure. For a more detailed description of the MANTIS family, we refer to the design paper [6].

In the light of cryptanalysis, there are several attacks [7,11,14] on MANTIS. For MANTIS, we find a $3+3$ round integral distinguisher based on BDPT which is one more round than the previous best results [15].

5.2 Applications to KLEIN and PRIDE

To complete our BDPT analysis on ciphers with complex linear layers, we apply our automatic search algorithm for BDPT to block ciphers KLEIN and PRIDE which have non-binary linear layers. In order to handle non-binary linear layers we trivially decompose the linear layers as COPY and XOR operations and construct the MILP model accordingly. Then, choosing appropriate initial BDPT, we find integral distinguisher as follows:

Integral Attack on KLEIN. KLEIN [16] is a family of block ciphers, with a fixed 64-bit block size and variable key length-64, 80 or 96-bits. The structure of KLEIN is a typical Substitution Permutation Network. For more details, please refer to [16].

In the light of cryptanalysis, there are several attacks [1,3,26,36] on the block cipher KLEIN, mostly on KLEIN-64 (key length 64 bits). In [36], the authors have presented a 5-round integral distinguisher using the higher-order integral and the higher-order differential properties which is best integral distinguisher known to us. First we apply MILP based CBDP on KLEIN and find a 6-round integral distinguisher which is one more round than the previous best results [36]. Therefore, we apply the MILP based BDPT on KLEIN and the integral distinguishers we find are in accordance with the distinguishers we find based on CBDP.

Integral Attack on PRIDE. PRIDE is a lightweight block cipher designed by Albrecht et al. [4], appears in CRYPTO 2014. PRIDE is an SPN structure block cipher with 64-bit block cipher and 128-bit key. For more details, please refer to [4]. In the light of cryptanalysis, there are several attacks on PRIDE [12,13,35,38].

First we apply MILP based CBDP on PRIDE and find a 9-round integral distinguisher which is one more round than the previous best results [33]. Therefore, we apply the MILP based BDPT on PRIDE and the integral distinguishers we find are in accordance with the distinguishers we find based on CBDP.

5.3 Applications to SIMON, SIMON (102)

We apply our method to all variants of SIMON [5], and SIMON(102) [22] block ciphers and the distinguishers we find are in accordance with the previous longest distinguishers [24] but we get these results in better time which are shown in the full version of this paper [10].

6 Conclusion and Future Work

In this paper, we provide an idea to model BDPT propagation of ciphers with binary (complex) linear layers and furthermore we construct an efficient automatic search algorithm that accurately characterize BDPT propagation. Based on these, more accurate BDPT for ciphers with binary (complex) linear layers such as PRINCE, MANTIS can be obtained.

For ciphers with non-binary linear layers we trivially decompose the linear layer by COPY-XOR technique which may ignore some balanced property. Therefore, how to model BDPT propagation for ciphers with non-binary linear layers accurately and efficiently is an open problem. Moreover, we construct our model using MILP solver whereas SAT/SMT are also very popular and efficient solvers in this domain. How to implement our model using SAT/SMT solvers or similar ones will be a future work.

Acknowledgement. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Abed, F., Forler, C., List, E., Lucks, S., Wenzel, J.: Biclique cryptanalysis of the PRESENT and LED lightweight ciphers. IACR Cryptology ePrint Archive 2012:591 (2012)
2. Abed, F., List, E., Lucks, S.: On the security of the core of PRINCE against biclique and differential cryptanalysis. IACR Cryptology ePrint Archive, p. 712 (2012)
3. Ahmadian, Z., Salmasizadeh, M., Aref, M.R.: Biclique cryptanalysis of the full-round KLEIN block cipher. IET Inf. Secur. **9**(5), 294–301 (2015)
4. 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
5. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, 7–11 June 2015, pp. 175:1–175:6. ACM (2015)
6. Beierle, C., et al.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 123–153. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_5
7. Beyne, T.: Block cipher invariants as eigenvectors of correlation matrices. J. Cryptol. **33**(3), 1156–1183 (2020)
8. Borghoff, J., et al.: PRINCE - a low-latency block cipher for pervasive computing applications (full version). IACR Cryptology ePrint Archive, p. 529 (2012)
9. Boura, C., Canteaut, A.: Another view of the division property. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 654–682. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_24
10. Chakraborty, D.: Finding three-subset division property for ciphers with complex linear layers (full version). Cryptology ePrint Archive, Paper 2022/1444 (2022). https://eprint.iacr.org/2022/1444
11. Chen, S., Liu, R., Cui, T., Wang, M.: Automatic search method for multiple differentials and its application on MANTIS. Sci. China Inf. Sci. **62**(3), 32111:1–32111:15 (2019)
12. Dai, Y., Chen, S.: Cryptanalysis of full PRIDE block cipher. Sci. China Inf. Sci. **60**(5), 052108:1–052108:12 (2017)
13. Dinur, I.: Cryptanalytic time-memory-data tradeoffs for FX-constructions with applications to PRINCE and PRIDE. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 231–253. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_10
14. Eichlseder, M., Kales, D.: Clustering related-tweak characteristics: application to MANTIS-6. IACR Trans. Symmetric Cryptol. **2018**(2), 111–132 (2018)
15. Eskandari, Z., Kidmose, A.B., Kölbl, S., Tiessen, T.: Finding integral distinguishers with ease. In: Cid, C., Jacobson Jr., M. (eds.) SAC 2018. LNCS, vol. 11349, pp. 115–138. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-10970-7_6

16. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: a new family of lightweight block ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25286-0_1
17. Gurobi Optimization, LLC.: Gurobi Optimizer Reference Manual (2021)
18. Hebborn, P., Lambin, B., Leander, G., Todo, Y.: Lower bounds on the degree of block ciphers. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12491, pp. 537–566. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_18
19. Hebborn, P., Lambin, B., Leander, G., Todo, Y.: Strong and tight security guarantees against integral distinguishers. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021. LNCS, vol. 13090, pp. 362–391. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92062-3_13
20. Hu, K., Wang, M.: Automatic search for a variant of division property using three subsets. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 412–432. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12612-4_21
21. Hu, K., Wang, Q., Wang, M.: Finding bit-based division property for ciphers with complex linear layer. IACR Cryptology ePrint Archive, p. 547 (2020)
22. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 161–185. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_8
23. Lambin, B., Derbez, P., Fouque, P.-A.: Linearly equivalent s-boxes and the division property. Des. Codes Cryptogr. **88**(10), 2207–2231 (2020)
24. Liu, H., Wang, Z., Zhang, L.: A model set method to search integral distinguishers based on division property for block ciphers. Cryptology ePrint Archive, Paper 2022/720 (2022). https://eprint.iacr.org/2022/720
25. Morawiecki, P.: Practical attacks on the round-reduced PRINCE. IET Inf. Secur. **11**(3), 146–151 (2017)
26. Nikolic, I., Wang, L., Shuang, W.: The parallel-cut meet-in-the-middle attack. Cryptogr. Commun. **7**(3), 331–345 (2015)
27. Rasoolzadeh, S., Raddum, H.: Cryptanalysis of PRINCE with minimal data. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 109–126. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_6
28. Sun, L., Wang, W., Wang, M.: Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. IACR Cryptology ePrint Archive, p. 811 (2016)
29. Sun, L., Wang, W., Wang, M.: Automatic search of bit-based division property for ARX ciphers and word-based division property. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 128–157. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_5
30. Todo, Y.: Structural evaluation by generalized integral property. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 287–314. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_12
31. Todo, Y., Morii, M.: Bit-based division property and application to Simon family. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 357–377. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_18
32. Wang, S., Hu, B., Guan, J., Zhang, K., Shi, T.: MILP-aided method of searching division property using three subsets and applications. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11923, pp. 398–427. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34618-8_14

33. Xiang, Z., Zeng, X., Zhang, S.: On the bit-based division property of s-boxes. Sci. China Inf. Sci. **65**(4), 149101 (2021)
34. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 648–678. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_24
35. Yang, Q., et al.: Improved differential analysis of block cipher PRIDE. In: Lopez, J., Wu, Y. (eds.) ISPEC 2015. LNCS, vol. 9065, pp. 209–219. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-17533-1_15
36. Yu, X., Wu, W., Li, Y., Zhang, L.: Cryptanalysis of reduced-round KLEIN block cipher. In: Wu, C.-K., Yung, M., Lin, D. (eds.) Inscrypt 2011. LNCS, vol. 7537, pp. 237–250. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34704-7_18
37. Zhang, W., Rijmen, V.: Division cryptanalysis of block ciphers with a binary diffusion layer. IET Inf. Secur. **13**(2), 87–95 (2019)
38. Zhao, J., Wang, X., Wang, M., Dong, X.: Differential analysis on block cipher PRIDE. IACR Cryptology ePrint Archive 2014:525 (2014)