Linearly equivalent S-boxes and the Division Property

Patrick Derbez[1⋆], Pierre-Alain Fouque[1⋆⋆], Baptiste Lambin[1⋆⋆⋆]

Univ Rennes, CNRS, IRISA
baptiste.lambin,patrick.derbez@irisa.fr
pierre-alain.fouque@univ-rennes1.fr

Abstract. Division property is a new cryptanalysis method introduced by Todo at Eurocrypt'15 that proves to be very efficient on block ciphers and stream ciphers. It can be viewed as a generalization or a more precise version of integral cryptanalysis, that allows to take into account bit properties. However, it is very cumbersome to study the propagation of a given division property through the layers of a block cipher. Fortunately, computer-aided techniques can be used to this end and many new results have been found. Nonetheless, we claim that the previous techniques do not consider the full search space. Indeed, we show that even if the previous techniques fail to find a distinguisher based on the division property over a given function E, we can find a distinguisher over a linearly equivalent function, i.e. over $L_{out} \circ E \circ L_{in}$ with L_{out} and L_{in} being linear mappings, and such distinguisher is still relevant. We first show that the representation of the block cipher heavily influences the propagation of the division property, and exploiting this, we give an algorithm to efficiently search for such linear mappings L_{out} and L_{in}. As a result, we are able to exhibit a new distinguisher over 10 rounds of RECTANGLE, while the previous best known distinguisher was over 9 rounds. Our algorithm also allows us to rule out such distinguisher over strictly more than 9 rounds of PRESENT, which is the highest known number of rounds on which we can build an integral distinguisher. Finally, we also give some insight about the construction of an S-box to strengthen a block cipher against our technique. As such, we exhibit some variant of RECTANGLE and PRESENT, on which we improve the maximum number of rounds we can distinguish by two.

Keywords: Cryptanalysis · Division Property · RECTANGLE

⋆ Patrick Derbez was supported by the French Agence Nationale de la Recherche through the CryptAudit project under Contract ANR-17-CE39-0003.

⋆⋆ Pierre-Alain was supported by the French Agence Nationale de la Recherche through the BRUTUS project under Contract ANR-14-CE28-0015.

⋆⋆⋆ Baptiste Lambin was supported by the Direction Générale de l'Armement (Pôle de Recherche CYBER).

1 Introduction

Division property is a distinguishing property which was first presented by Todo at Eurocrypt'15 [14]. This cryptanalysis technique quickly became an hot topic in the community, especially since it lead to the first theoretical attack against full MISTY1. This property can be seen as a generalization of integral and higher-order differential distinguishers. At Crypto'16, Boura et al. [4] provided a simpler formulation of the division property, especially for the construction of the division trails of S-boxes. Recently, division property was used to improve cube attacks and allowed to improve the best known results against several stream ciphers including ACORN, Trivium, Grain-128a and Kreyvium [16].

Automatic tools. Studying the propagation of an initial division property through a block cipher is a challenging task which requires to be computer-aided. At Asiacrypt'16, Xiang et al. [17] showed how to model the division property propagation of the three basic operations **copy**, **AND** and **XOR**, as well as the propagation through an S-box, by a system of linear inequalities. Hence they built MILP models for several block ciphers which they efficiently solved using a third-party MILP solver. As a results they obtained the best known division property distinguishers on SIMON, SIMECK, PRESENT and RECTANGLE. In [19], Zhang et al. gave a new way to model the propagation of division property through linear diffusion layers by the smallest amount of inequalities which are generated from linear combinations of row vectors of the diffusion matrix. Using this new description, they found the best known distinguishers for both PRINCE and MIDORI. Finally, at Asiacrypt'17, Sun et al. [11] presented two new automatic search tools: one dedicated to ARX ciphers based on a SAT solver and one dedicated to word-based division property based on SMT (Satisfiability Modulo Theories) solver. Those tools are much faster than previous MILP-based works and were able to study primitives with large internal state such as CLEFIA, WHIRLPOOL and RIJNDAEL.

Our contributions. In this paper we show that previous automatic search tools dedicated to division property are incomplete in the sense they do not exhaust all the search space. More precisely, propagating an initial division property through a block cipher requires to decompose the block cipher into small components such as **AND**, **XOR**, S-boxes, ... for which we can compute division property propagation. However, contrary to differential and linear cryptanalysis, the result highly depends on how the block cipher is split. Indeed, linearly equivalent Sboxes do not change the propagation of differentials, while it is not the case for the division property. For instance, in Section 3.1 we give two S-boxes S_1 and S_2 such that $S_2 = S_1 \circ L$, where L is linear, such that propagating division property through L then S_1 leads to a completely different result than propagating it directly through S_2. Hence, given an S-box based block cipher, it is not clear which representation should be studied since replacing any internal S-box with a linearly equivalent one could possibly lead to a different result. The main issue is that the number of distinguishers is much more higher than

one can be thinking and looking efficiently for the best distinguisher boils down efficiently finding the best decomposition.

In this paper we solved a sub-case of this problem. Mounting an attack against a block cipher E most often requires to split E in three as $E = E_2 \circ E_1 \circ E_0$ and to find a distinguisher on E_1. Usually, E_0, E_1 and E_2 are round-reduced versions of E. However it is not the only way to split E and, for any linear operations L_{in} and L_{out} we can split E as:

$$E = (E_2 \circ L_{out}^{-1}) \circ (L_{out} \circ E_1 \circ L_{in}) \circ (L_{in}^{-1} \circ E_0).$$

This kind of carving was for instance used in [6] by Derbez et al. to provide several new meet-in-the-middle attacks against AES. Hence, one of the main problem we solved in this paper is to answer the question of how to find L_{in} and L_{out} such that there exists a division property through $L_{out} \circ E_1 \circ L_{in}$. Note that finding a distinguisher over $L_{out} \circ E_1 \circ L_{in}$ still allow to mount a key-recovery attack. Indeed, the attacker starts with a set $\mathbb{X}$ respecting a given initial division property (according to the distinguisher over $L_{out} \circ E_1 \circ L_{in}$) and compute $\mathbb{X}' = E_0^{-1} \circ L_{in}(\mathbb{X})$ by guessing part of the key. He then ask for the encryption of $\mathbb{X}'$ through E to get a set of ciphertexts $\mathbb{Y}$, compute $\mathbb{Y}' = L_{out} \circ E_2^{-1}(\mathbb{Y})$ using some other key guesses and check whether $\mathbb{Y}'$ has some balanced bits (according to the distinguisher over $L_{out} \circ E_1 \circ L_{in}$). If so, then the key guesses are supposed to be correct.

In a nutshell, we first show how to highly reduce the number of candidates for both L_{in} and L_{out}, and then present how to efficiently check the remaining candidates without performing a complete search on each of them. As a result we improve the best known division property distinguisher against RECTANGLE by one round and show that the previous best known distinguisher against PRESENT cannot be improved with this technique.

The second result presented in this paper concerns the design of S-boxes that would offer maximal resistance against division property. In [4], Boura et al. provided new insights into the division property, presenting a new approach to it. In particular they shown several interesting results concerning resistance of S-box-based block ciphers against division property. They also gave the intuition that an m-bit S-box S such that $\lambda \cdot S(x)$ has degree $m-1$ for all non-zero $\lambda \in \mathbb{F}_2^m$ should offer maximal resistance against division property. However, one may want to design an S-box that does not satisfy this degree condition, in particular for lightweight primitives. We describe the methodology we followed to strengthen both RECTANGLE and PRESENT against division property. Our first idea was to search for S-boxes leading to *perfect* division property propagation tables, i.e. 14 transitions $k \to \{1000, 0100, 0010, 0001\}$. Unfortunately, for both RECTANGLE and PRESENT, it was not possible to generate such a table from linearly equivalent S-boxes. However we found many *almost perfect* tables i.e. with 13 transitions $k \to \{1000, 0100, 0010, 0001\}$ instead of 14. Trying all of them allowed us to find a linearly equivalent S-box for RECTANGLE such that there is no division property against 9 rounds (while with the original S-box we found a division property against 10 rounds) as well as a linearly equivalent

S-box for PRESENT such that there is no division property against 8 rounds. Furthermore, on non-table-based implementations, the extra cost of the new S-boxes is only 2 extra XORs per S-box for PRESENT and 5 extra XORs per S-box for RECTANGLE. Finally, for both RECTANGLE and PRESENT, our new search process find distinguishers against one extra round than classical search, highlighting again its interest.

2 Background

2.1 Notations

We will use the following notations in the paper.

- We denote $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_2^n$ an n-bit vector over $\mathbb{F}_2$, where x_0 is the least significant bit. We will often write $x_0 x_1 \ldots x_{n-1}$ instead of $(x_0, \ldots, x_{n-1})$.
- $\boldsymbol{e}_i$ denotes the i-th unit vector, and $\mathbb{E}_m$ denotes the set of all unit vectors of size m.
- $w(\boldsymbol{x})$ denotes the hamming weight of $\boldsymbol{x} \in \mathbb{F}_2^n$.
- For $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{F}_2^n$, we denote by $\boldsymbol{x}^{\boldsymbol{u}}$ the bit product

$$\boldsymbol{x}^{\boldsymbol{u}} = \prod_{i=0}^{n-1} x_i^{u_i}.$$

- For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_2^n$, we define $\boldsymbol{x} \succeq \boldsymbol{y}$ if $x_i \geq y_i$ for all i, where x_i and y_i are considered as integers.
- $\mathcal{P}_m$ the set of all permutations over m elements.
- $GL_m(\mathbb{F}_2)$ the set of all invertible matrices of size $m \times m$ over $\mathbb{F}_2$.

2.2 Division Property and Division Trails

The division property was introduced at EUROCRYPT'15 [14] by Todo as a generalization of integral cryptanalysis, and later at FSE'16 [15], Todo and Morii defined a more refined version of it, called bit-based division property. Here, we only consider the bit-based division property, and will often refer to it directly as *division property*. As it is not relevant for this paper, we refer the reader to the original articles for further details about the differences.

Definition 1 (Bit-based Division Property [15]). *A set $\mathbb{X} \subset \mathbb{F}_2^n$ has the division property $D_{\mathbb{K}}^n$, where $\mathbb{K} \subset \mathbb{F}_2^n$, if for all $\boldsymbol{u} \in \mathbb{F}_2^n$, we have*

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

Note that if there are some vectors $\boldsymbol{k}, \boldsymbol{k'} \in \mathbb{K}$ such that $\boldsymbol{k} \succeq \boldsymbol{k'}$, then $\boldsymbol{k}$ can be removed from $\mathbb{K}$ because it is redundant.

A common way to study division property for a block cipher is to study the *division trails* of this cipher, which show the propagation of the division property through the operations composing the block cipher.

Definition 2 (Division Trails [17]). *Let f denote the round function of an iterated block cipher. Assume the input set to the block cipher has initial division property $D^n_{\{\boldsymbol{k}\}}$, and denote the division property after propagating through i rounds of the block cipher (i.e. i applications of f) by $D^n_{\mathbb{K}_i}$. Thus, we have the following chain of division property propagations :*

$$\{\boldsymbol{k}\} \stackrel{\Delta}{=} \mathbb{K}^0 \xrightarrow{f} \mathbb{K}^1 \xrightarrow{f} \mathbb{K}^2 \xrightarrow{f} \dots \xrightarrow{f} \mathbb{K}^r.$$

Moreover, for any vector $\boldsymbol{k}^i$ in $\mathbb{K}^i (i \geq 1)$, there must exist a vector $\boldsymbol{k}^{i-1}$ in $\mathbb{K}^{i-1}$ such that $\boldsymbol{k}^{i-1}$ can propagate to $\boldsymbol{k}^i$ by the division property propagation rules. Furthermore, for $(\boldsymbol{k}^0, \boldsymbol{k}^1, \dots, \boldsymbol{k}^r) \in \mathbb{K}^0 \times \mathbb{K}^1 \times \dots \times \mathbb{K}^r$, if $\boldsymbol{k}^{i-1}$ can propagate to $\boldsymbol{k}^i$ for all $i \in \{1, 2, \dots, r\}$, we call $(\boldsymbol{k}^0, \boldsymbol{k}^1, \dots, \boldsymbol{k}^r)$ an r-round division trail.

In the rest of the paper, we will denote $\boldsymbol{k} \xrightarrow{f} \boldsymbol{k'}$ if the vector $\boldsymbol{k} \in \mathbb{F}_2^n$ can propagate to a vector $\boldsymbol{k'} \in \mathbb{F}_2^n$ through the function f. In the same way, $\boldsymbol{k} \xrightarrow{f} \mathbb{K}$ denotes that for all $\boldsymbol{k'} \in \mathbb{K}$, we have $\boldsymbol{k} \xrightarrow{f} \boldsymbol{k'}$.

Given the set $\mathbb{K}^r$ resulting of the propagation of an initial division property $D^n_{\{\boldsymbol{k}\}}$, we can find whether $D^n_{\{\boldsymbol{k}\}}$ allows to build an integral distinguisher using the following proposition.

Proposition 1 ([17]). *Assume $\mathbb{X}$ is a set with division property $D^n_{\mathbb{K}}$, then $\mathbb{X}$ does not have integral property if and only if $\mathbb{K}$ contains all the n unit vectors. As a result, if $\boldsymbol{e}_i \notin \mathbb{K}$, then the i-th bit is balanced.*

Proof. Suppose that the vector $\boldsymbol{e}_i$ belongs to $\mathbb{K}$. Then according to the definition of the division property, this would mean that the result of the sum

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{e_i}} = \bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x_i}$$

is unknown since $\boldsymbol{e}_i \succeq \boldsymbol{e}_i$ and $\boldsymbol{e}_i \in \mathbb{K}$, i.e. the i-th bit is not balanced. On the other hand, if we suppose that the i-th bit is balanced, then we cannot have $\boldsymbol{e}_i \in \mathbb{K}$ as it would mean that the i-th bit is in an unknown state, which contradicts the definition of the division property.

For example, we can make a parallel with the well known Square attack on AES [5]. In this attack, the set of plaintexts has one byte taking all possible values while the others are constant. In term of division property, this would translate to the set of plaintexts having a division property $D^{128}_{\boldsymbol{k}}$, where

$$\boldsymbol{k} = \underbrace{11111111}_{8 \text{ bits}} 0 \dots 0.$$

Then, it is shown in [5] that after 3 rounds of AES, such a set of plaintexts has all its bits balanced. According to Proposition 1, this would mean that the resulting set has a division property $D^{128}_{\mathbb{K}}$, where $\mathbb{K}$ does not contain any unit vector.

Hence, to study whether we can build an integral distinguisher over a block cipher from a given initial division property $\mathbb{K}^0$, we need to propagate $\mathbb{K}^0$ through the different operations of the block cipher. Fortunately, propagation rules were defined in [15] for most basic operations in a block cipher, namely Copy, AND and XOR. However, for SPN block ciphers, there are two main components that, while they can be described using only these operations, should have their own way to propagate the division property vectors. These components are linear layers and S-boxes. For linear layers, while [10] proposed to use only the Copy and XOR operations to propagate division property vectors, it has been shown in [19] that this is actually not the right way to propagate through linear layers, as it looses some information and is not able to recover all possible integral distinguishers. We thus refer the reader to [19] for the correct way to propagate division property vectors through a given linear layer.

For S-boxes, again using only the basic operations might results in a loss of information. Hence, [17] proposed an algorithm of complexity $\mathcal{O}\left(2^{2m}\right)$ to compute all possible pairs $\boldsymbol{k} \xrightarrow{S} \boldsymbol{k'}$ for a given m-bit S-box S.

2.3 Searching for division property based integral distinguishers

While Todo and Morii proposed a way to search for integral distinguishers based on the division property [15], its complexity is quite hard to estimate, and the authors gave an upper bound of 2^n, where n is the block size of the block cipher. In practice, they said that their algorithm is not suitable for block ciphers with block size over 32 bits, and thus especially for standard block size of 64 and 128 bits. However, a lot of work has been done towards efficiently searching such distinguishers, based on either MILP [9, 17, 19] or SAT/SMT solver [7, 12]. We refer the reader to these papers for further details about the modeling, and will only give a brief description of the idea behind it when considering MILP modeling. Note that using SAT/SMT solvers is very similar to using MILP, and mostly differs in efficiency when considering different primitives. For example searching division property based integral distinguishers on ARX ciphers seems to be easier when using SAT solvers.

First we briefly recall what is MILP.

Definition 3. *An MILP problem is formulated as follows. Given a matrix $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, find a vector $x \in \mathbb{Z}^k \times \mathbb{R}^{n-k}$ with $Ax \leq b$ which minimize (or maximize) the value of*

$$f(x) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n.$$

Here, f is called the objective function of the MILP problem.

Modelizing division property propagation with MILP. The idea of using MILP to search for integral distinguishers is first to modelize the set of all possible division trails by an MILP problem. That is, building a set of linear inequalities such that [17] :

1. each division trail must satisfy all linear inequalities in the linear equality system. That is, each division trail corresponds to a feasible solution of the linear inequality system ;
2. each feasible solution of the linear inequality system corresponds to a division trail. That is, the set of all feasible solutions of the linear inequality system does not contain any impossible division trail.

We can thus build an MILP model satisfying the previous conditions using [17] for basic operations and S-boxes, [19] for linear layers and [9] for ARX block ciphers. Note that this step is not totally free.

For S-boxes, we first need to compute the set of all possible propagations through a given m-bit S-box, which has complexity $\mathcal{O}\left(2^{2m}\right)$. Then, we need to compute a set of linear inequalities which represents these possible propagations, according to the two previous rules. To do so, [17] proposed to first use the function `inequality_generator()` from the Sagemath [13] software to get such a set of inequalities, and then use a greedy algorithm to reduce their number. While this works for small S-boxes (e.g. 4-bit S-boxes), this approach fails when considering bigger S-boxes (e.g. 8-bit S-boxes) as the complexity to generate the initial set of inequalities is too high. However, Abdelkhalek et al. showed a new method in [1] to tackle this problem, and thus proposed a way to modelize 8-bit S-boxes in MILP. Note that while this allows us to modelize 8-bit S-boxes, it often leads to a lot of inequalities, thus the resulting model can be quite huge and this can result in a high solving time.

For linear layers, Zhang et al. showed [19] showed that the previous method [10] proposed to modelize linear layers does not actually fulfill the above rules, as it introduces some impossible propagations, resulting in some integral distinguishers being omitted. Hence, they proposed a new way to modelize such layers, and proved that their way was optimal, i.e. removing any one inequality will result in some fraudulent propagations. To modelize a given linear layer L, the number of inequalities generated is given by $n(2^s - 1)$, where s is the size of the smallest square matrix M such that M is the representation of L over the field $\mathbb{F}_{2^n}$ and M is *binary*. For example, the matrix used in `SKINNY64` [2] is a binary matrix of size 4 over $\mathbb{F}_{2^4}$, thus needs $4(2^4 - 1) = 60$ inequalities. However, if we take the matrix used in AES, which is described as a *non-binary* matrix of size 4 over $\mathbb{F}_{2^8}$, the amount of inequalities is much higher. Indeed, since the multiplication over $\mathbb{F}_{2^8}$ corresponds to a linear operation over $\mathbb{F}_2^8$, the matrix used in AES can be represented as a matrix of size 32 over $\mathbb{F}_2$, which is obviously binary. This is the smallest way to represent this operation with a binary matrix, and thus, it would need $2^{32} - 1$ inequalities to modelize only *one* propagation through this linear layer, which would result in a very huge model which cannot be solved in practical time. Hence, not all linear layers can be modelized in an exact way, and complex linear layers may lead to a model which is much harder to solve. Note however that if the linear layer is only a permutation, such as in `PRESENT` [3] or `RECTANGLE` [18], then the above formula does not apply, as we can just reorder the different variables, and thus we can always modelize such kind of linear layers.

Searching for a distinguisher. As a result, we have a set of variables $\{\mathbf{k}_i^j, i \in \{0,\dots,n-1\}, j \in \{0,\dots,r\}\}$ such that, for a given solution of the MILP problem, the corresponding values of these variables gives a division trail $(\boldsymbol{k}^0, \boldsymbol{k}^1, \dots, \boldsymbol{k}^r)$ with $\boldsymbol{k}^i = (\mathbf{k}_0^i, \dots, \mathbf{k}_{n-1}^i)$. In particular, this allows us to see whether each unit vector belongs to $\mathbb{K}^r$. Indeed, once we have the MILP model for r rounds of a given block cipher, we can set the objective function to $\mathbf{k}_0^r + \cdots + \mathbf{k}_{n-1}^r$. Then we set the initial division property using equality constraints, i.e. if the initial division property is $\boldsymbol{a} \in \mathbb{F}_2^n$, we add the constraints

$$\forall i \in \{0,\dots,n-1\}, \mathbf{k}_i^0 = a_i,$$

and then ask the solver (e.g. Gurobi [8]) to solve this problem by minimizing the value of the objective function. If the solver finds a solution of value 1, this means that there is a vector $\boldsymbol{k}^r$ of weight 1 (i.e. a unit vector) that belongs to $\mathbb{K}^r$. We can then add a linear constraint to remove this vector $\boldsymbol{k}^r$ from the set of solutions, and solve the problem again. Once there is no more solution of value 1, we know that we found all unit vectors belonging to $\mathbb{K}^r$, hence we can easily see whether or not there is some balanced bits using Proposition 1. Note that we do not need to stop after finding all solutions of value 1. Indeed, we can keep going until the problem does not have any remaining solution, and we will thus have computed the whole $\mathbb{K}^r$ set. This will be useful later in the paper, and will be accompanied with a bit more details.

3 Extended Division Property Using Linear Mappings

3.1 First observations

Several integral distinguishers were found using the previously described method. However, we claim that this method does not actually search through the whole space of all possible integral distinguishers based on the division property. Indeed, we show that for a given block cipher E, we can instead consider $L_{out} \circ E \circ L_{in}$, where both L_{out} and L_{in} are linear mappings, and that this results in integral distinguishers previously unknown. We now explain the main idea behind using L_{out} and L_{in}. For L_{out}, the idea is to see that while all bits could be unbalanced after E, it might occur however that a linear combination of some bits is balanced. This was already mentioned by Todo and Morii in [15] when they introduced the *division property using three subsets*. However, such kind of division property has yet to have an efficient search algorithm.

For L_{in}, the idea is very close. The initial division property $\boldsymbol{k}^0$ basically sets some constant bits. That is, if the set $\mathbb{X}$ has division property $D^n_{\boldsymbol{k}^0}$, then through all the set, each bit i such that $\boldsymbol{k}_i^0 = 0$ has a constant value, and if $\boldsymbol{k}_i^0 = 1$, then the bit i takes all possible values through the set. For example, the following set has division property D^4_{0011}

$$\mathbb{X} = \{0100, 0101, 0110, 0111\}.$$

Hence, the idea behind L_{in} is to get a set such that a linear combination of some bits is constant, while those bits are not necessarily constant.

Finally, we can see than considering $L_{out} \circ E \circ L_{in}$ instead of E is still meaningful. Classically, when an attacker use a distinguisher to mount an attack, he basically splits the cipher E into $E = E_2 \circ E_1 \circ E_0$, where he has a distinguisher over E_1. In that case, E_1 can be seen as a reduced version of E, containing only a certain number of rounds of E. However, we could also rewrite E as

$$E = (E_2 \circ L_{out}^{-1}) \circ (L_{out} \circ E_1 \circ L_{in}) \circ (L_{in}^{-1} \circ E_0).$$

In that case, the attacker would search a distinguisher over $L_{out} \circ E_1 \circ L_{in}$, and could still use it to mount an attack. Note that this idea was already successfully used in the past, for example in [6].

So considering $E' = L_{out} \circ E \circ L_{in}$ instead of E could lead to some new integral distinguishers. In the following, we will consider that E is an SPN block cipher, i.e. the round function of E is $f = \mathcal{L} \circ \mathcal{S}$, where $\mathcal{L}$ is linear and $\mathcal{S}$ is the parallel application of an S-box S over the state. Note that we can omit $\mathcal{L}$ in the last round. Now suppose that we want to search if E' has an integral distinguisher based on the division property using MILP. Using the classic way to do this, we would modelize the following propagation chain

$$\mathbb{K}_0 \xrightarrow{L_{in}} \widehat{\mathbb{K}}^0 \xrightarrow{\mathcal{S}} \widetilde{\mathbb{K}}^0 \xrightarrow{\mathcal{L}} \mathbb{K}^1 \xrightarrow{\mathcal{S}} \ldots \xrightarrow{\mathcal{S}} \widehat{\mathbb{K}}^r \xrightarrow{L_{out}} \mathbb{K}^r.$$

Basically, we modelize independently the propagation through the linear layers and the S-box layers, especially for L_{in} and the first S-box layer, and for L_{out} and the last S-box layer. However, this might actually not be the best way to modelize this, and we will see this through an example.

Merging linear mappings and S-boxes Let S_1 and S_2 be two S-boxes over $\mathbb{F}_2^4$ such that

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_1(x)$	12	13	11	9	6	0	5	10	3	2	8	4	15	7	14	1
$S_2(x)$	12	11	14	15	1	7	13	9	10	0	2	4	3	8	5	6

where S_2 is obtained as $S_2 = S_1 \circ L$ with

$$L = \begin{bmatrix} 0\,0\,1\,1 \\ 1\,1\,1\,1 \\ 0\,1\,1\,1 \\ 0\,1\,1\,0 \end{bmatrix}.$$

We can use the algorithm from [17] to compute all possible propagations through S_1, S_2 and L. Using this, if we look at the propagation of $\boldsymbol{x} = 0111$ through L and S_1 independently, we have the following trail

$$0111 \xrightarrow{L} \{1101, 1011\} \xrightarrow{S_1} \{0100, 0010, 0001\} = \mathbb{K}.$$

However, if we now consider L and S_1 together, i.e. by looking at the propagation of 0111 through $S_2 = S_1 \circ L$, then we have the trail

$$0111 \xrightarrow{S_2} \{1100, 1001, 0110, 0011\} = \mathbb{K}'.$$

As we can see, the resulting division property set is completely different, yet comes from the same initial division property, and goes through the same function. Moreover, this is not just a local change, and not only $\mathbb{K}'$ is a set which was not reachable through only S_1, but the whole propagation tables of S_1 and S_2 are different, as we can see in Figure 1.

Propagations of S_1	
0000	0000
1000	1000, 0100, 0010, 0001
0100	1000, 0100, 0010, 0001
0010	1000, 0100, 0010, 0001
0001	1000, 0100, 0010, 0001
1100	1000, 0100, 0010, 0001
1010	1000, 0100, 0010, 0001
1001	0100, 0010, 0001
0110	0100, 0010, 0001
0101	0100, 0010, 0001
0011	0100, 0010, 0001
1110	0100, 0001
1101	0100, 0010, 0001
1011	0100, 0010, 0001
0111	0100, 0010, 0001
1111	1111

Propagations of S_2	
0000	0000
1000	1000, 0100, 0010, 0001
0100	0100, 0010, 0001
0010	1000, 0100, 0010, 0001
0001	1000, 0100, 0010, 0001
1100	0100, 0010, 0001
1010	1000, 0010, **0101**
1001	1000, 0100, 0010, 0001
0110	0100, 0010, 0001
0101	0100, 0001, **1010**
0011	0010, **1100**, **1001**, **0101**
1110	0010, **1100**, **0101**
1101	0100, 0001, **1010**
1011	**1100**, **1010**, **1001**, **0110**, **0011**
0111	**1100**, **1001**, **0110**, **0011**
1111	1111

Fig. 1: Propagation table of S_1 and S_2. Vectors of weight 2 are in bold.

This clearly shows that considering both the S-box and the linear mapping *together* gives way more information about the propagation of the division property. Note that we give this example by putting a linear mapping at the input of the S-box, but similar observations can be made when considering S and $L \circ S$ for some S-box S and linear mapping L.

Moreover, not only this gives more information about the propagation, but this could, and will, actually help us to find new distinguishers when considering $L_{out} \circ E \circ L_{in}$ instead of E. Before giving an intuition about why this is the case, we first give the following proposition.

Proposition 2. *Let S be an invertible function over $\mathbb{F}_2^n$ of degree d, and $\boldsymbol{k} \xrightarrow{S} \mathbb{K}$ be a division property propagation through S. Then :*

1. *If* $\boldsymbol{k} = 00\ldots0$, *then* $\mathbb{K} = \{00\ldots0\}$.
2. *If* $\boldsymbol{k} = 11\ldots1$, *then* $\mathbb{K} = \{11\ldots1\}$.
3. *Otherwise, then* $\forall \boldsymbol{k}' \in \mathbb{K}, w(\boldsymbol{k}') \leq w(\boldsymbol{k})$.

Proof. 1. If a set $\mathbb{X}$ has division property $\boldsymbol{k} = 00\ldots0$, then this means that all bits are constant through the set, i.e. $\mathbb{X}$ contains a single vector. Hence, $S(\mathbb{X})$ also only contains one vector, thus all bits are still constant and the resulting division property set is $\mathbb{K} = \{00\ldots0\}$.
2. If a set $\mathbb{X}$ has division property $\boldsymbol{k} = 11\ldots1$, then this means that all bits takes all possibles values through the set, i.e. $\mathbb{X} = \mathbb{F}_2^n$. Hence, since S is a bijection, $S(\mathbb{X}) = \mathbb{F}_2^n$, and thus the resulting division property set is $\mathbb{K} = \{11\ldots1\}$.
3. Otherwise, then from [14] :
   - If $w(\boldsymbol{k}) = d$, then we directly have $w(\boldsymbol{k}') \leq w(\boldsymbol{k})$ for any $\boldsymbol{k}' \in \mathbb{K}$.
   - Otherwise, then $w(\boldsymbol{k}') \leq \left\lceil \frac{w(\boldsymbol{k})}{d} \right\rceil \leq w(\boldsymbol{k})$.

We can see that, except when we have either the full zero or the full one vector, if we consider a division property chain $\mathbb{K}^0 \to \cdots \to \mathbb{K}^r$ of a block cipher, then the weight of the vectors in each $\mathbb{K}^i$ can only decrease (or remain constant, but in practice, this is rarely the case, see Figure 1). Recall that if the set $\mathbb{K}^r$ contains all unit vectors (i.e. of weight 1), then no integral distinguisher can be built from it. Thus, intuitively, if we want to find an integral distinguisher, we would like to have vectors of relatively high weight in each set $\mathbb{K}^i$ as long as possible.

Now consider a block cipher E such that the first layer of S-boxes contains only S_1 as defined previously. Then from the propagation table in Figure 1, we can see that the output of each S-box will always be of weight 1 (except for 0000 and 1111). So after the first round, if the weight at the input of any S-box is different from 0 and 4, then we will already only have vectors of weight 1 at the output of the S-box. However, if we now consider $E \circ \mathcal{M}$, where $\mathcal{M} = (M, \ldots, M)$ apply the linear mapping M on all S-box's input before the first round, then this is the same as considering the first layer of S-boxes to be built as $(S_2, \ldots, S_2)$. This time, if one carefully chooses the input division property of the S-box, he can now only have vectors of weight 2, which *could* result in a better propagation through the remaining layers of the cipher.

Clearly, considering $L_{out} \circ E \circ L_{in}$ instead of E, and considering the propagation of the division property vectors through $M \circ S$ (or $S \circ M$) as a whole instead of independently through M and S, could result in some better distinguishers than previously known, and thus in the next section, we focus on the search of such distinguishers.

3.2 Searching for Extended Division Property

First, let us recall what we are looking for. In this paper, we will only consider SPN block ciphers, i.e. the round function is $f = \mathcal{L} \circ \mathcal{S}$, where $\mathcal{L}$ is linear and $\mathcal{S}$ is built as the concatenation of s S-boxes of size m applied in parallel on the state,

hence the block cipher has block size $n = s.m$. Moreover, we will consider that all S-boxes are the same. This is to get an easier analysis, but we can extend this with different S-boxes.

Reducing the search space of L_{in} and L_{out}. Given such a block cipher E which does not have any integral distinguisher based on the division property, we want to find two linear mappings L_{in} and L_{out} such that $L_{out} \circ E \circ L_{in}$ has an integral distinguisher based on the division property which is supported by the previous observations. Moreover, we also would like to exploit the fact that we have a more precise propagation when considering the propagation of division property vectors through $S \circ M$ as a single function, instead of independently through M then S. Note that, theoretically, we could consider the whole round function of the block cipher as a single function (or even the whole block cipher), and thus getting more precise informations about the propagation of division property vectors. However, recall that computing the propagation table of division vectors needs $\mathcal{O}(2^{2n})$ operations, where n is the size of the function. Hence for classical block ciphers with 64 or 128 bits block size, this is clearly impractical.

This also means that we cannot choose any L_{in} and L_{out}, as we want to somehow merge L_{in} with the first S-box layer and, respectively, merge L_{out} with the last S-box layer. Hence, we will focus our search on linear maps L_{in} and L_{out} which are block diagonal, of block size m. Basically, this means that we want to put an invertible linear map L^i_{in} (resp. L^i_{out}) before (resp. after) each S-box of the first (resp. last) round. By doing so, we will denote by $S^i_{in} = S \circ L^i_{in}$ and $S^i_{out} = L^i_{out} \circ S$ the modified S-boxes.

First, we give the following proposition to show that we do not need to consider every possible choice for each block L^i_{in} and L^i_{out}.

Proposition 3. *Let S be an invertible m-bit S-box, L an invertible m-bit linear map and P an m-bit permutation. Let $S_1 = S \circ P$ and $S_2 = P \circ S$, and $\boldsymbol{k} \xrightarrow{S} \boldsymbol{k}'$ be any valid division property propagation through S with $\boldsymbol{k}, \boldsymbol{k}' \in \mathbb{F}_2^m$. Then :*

- *The propagation $P(\boldsymbol{k}) \xrightarrow{S_1} \boldsymbol{k}'$ is always valid.*
- *The propagation $\boldsymbol{k} \xrightarrow{S_2} P(\boldsymbol{k}')$ is always valid.*

Proof. This directly comes from the fact that S_1 is obtained by just permuting the input variables of S, and respectively S_2 is obtained by permuting the output bits of S.

Hence, if we search an integral distinguisher for any given block L^i_{in}, we do not need to do the search for all $L^i_{in} \circ P$ where P goes through all possible permutations, as we could obtain the same result from the search using L^i_{in} by just permuting the initial division property with P. This means, for example, that if we have the set $\mathbb{K}^r$ from a given initial division property $\boldsymbol{k}$ through $L_{out} \circ E \circ L_{in}$, and we consider $L'_{out} \circ E \circ L'_{in}$ where $L'_{in} = L_{in} \circ (P^0_{in}, \ldots, P^{s-1}_{in})$ and $L'_{out} = (P^0_{out}, \ldots, P^{s-1}_{out}) \circ L_{out}$, where each P^i_{in} and P^i_{out} is a permutation over

m bits, then we directly have that the initial division property $(P_{in}^0, \ldots, P_{in}^{s-1})(\boldsymbol{k})$ propagates to the set $(P_{out}^0, \ldots, P_{out}^{s-1})(\mathbb{K}^r)$. In particular, if we have an integral distinguisher for $L_{out} \circ E \circ L_{in}$, then so do we for $L'_{out} \circ E \circ L'_{in}$ (and vice-versa if $L_{out} \circ E \circ L_{in}$ do not have any integral distinguisher).

This allows us to restrict the search space for each block L_{in}^i to a set $\mathbb{L}_{in}$ containing a representative of each equivalence class

$$\mathcal{E}_{in}(L) = \{L' \in GL_m(\mathbb{F}_2) \mid \exists P \in \mathcal{P}_m \ s.t. \ L' = L \circ P\},$$

and in the same way, to restrict the search space of each L_{out}^i to a set $\mathbb{L}_{out}$ containing a representative of each equivalence class

$$\mathcal{E}_{out}(L) = \{L' \in GL_m(\mathbb{F}_2) \mid \exists P \in \mathcal{P}_m \ s.t. \ L' = P \circ L\}.$$

The size of these spaces $\mathbb{L}_{in}$ and $\mathbb{L}_{out}$ can be obtained by

$$\frac{\prod_{i=0}^{m-1} 2^m - 2^i}{m!},$$

as it is the total number of invertible matrices of size m divided by the number of permutations over m elements. Note that this is way lower than the total number of matrices of size $m \times m$ over $\mathbb{F}_2$ which is 2^{m^2}, and for example if $m = 4$, then there are only 840 matrices to consider.

Reducing the amount of work for L_{in} Just for now, let us focus on finding a distinguisher over $E \circ L_{in}$. We will see later that we can use the idea of this section together with the next section to search for a distinguisher over $L_{out} \circ E \circ L_{in}$. Note that we just want to exhibit *a* distinguisher on $E \circ L_{in}$, not necessarily the best one. As such, we will focus on finding a distinguisher requiring 2^{n-1} data, i.e. the initial division property will be $\mathbb{K}^0 = \boldsymbol{k}^0$ with $w(\boldsymbol{k}^0) = n - 1$. By doing so, we can focus our search on only one modified S-box S_{in}^i and set the others to S. Indeed, if $w(\boldsymbol{k}^0) = n - 1$, there will be only one specific S-box S_{in}^i which will have an input of weight $m - 1$, while all the others S-boxes $S_{in}^j, j \neq i$ will have $1 \ldots 1$ has input. Hence, according to Proposition 2, the vector $1 \ldots 1$ will always be propagated to $1 \ldots 1$ through all $S_{in}^j, j \neq i$, no matter what L_{in}^i is.

From the previous remark, we know that we only need to look at each matrix from $\mathbb{L}_{in}$. However, we can reduce even further the amount of propagation we need to compute. Since the input of the S-box S_{in}^i is $\boldsymbol{k}_i^0$ with $w(\boldsymbol{k}_i^0) = m - 1$, we know that this can only result in at most $2^m - 2$ possible vectors (by excluding the full-zero and full-one vectors) after the application of S_{in}^i. Thus, to search for a distinguisher over $E' = E \circ L_{in}$ with E containing r rounds with round function $f = \mathcal{L} \circ \mathcal{S}$, we first decompose E' as

$$E' = f \circ f \circ \cdots \circ f \circ \mathcal{L} \circ \mathcal{S}_{in}, \text{ with } \mathcal{S}_{in} = (S, \ldots, S_{in}^i, \ldots, S).$$

This leads to the following chain of division property propagation

$$\boldsymbol{k}^0 \xrightarrow{\mathcal{S}_{in}} \widetilde{\mathbb{K}}^0 \xrightarrow{\mathcal{L}} \mathbb{K}^1 \xrightarrow{f} \ldots \xrightarrow{f} \mathbb{K}^r$$

where

$$\boldsymbol{k}^0 = \overbrace{1\dots1}^{m}|\overbrace{1\dots1}^{m}|\dots|\boldsymbol{k}_i^0|\dots|\overbrace{1\dots1}^{m}.$$

We first define the set $\mathcal{K}_{in}^S$ as

$$\mathcal{K}_{in}^S := \{\mathbb{K} \mid \exists L_{in} \in \mathbb{L}_{in}, \boldsymbol{k} \in \mathbb{F}_2^m \text{ s.t. } \boldsymbol{k} \xrightarrow{S_{in}^i} \mathbb{K} \text{ and } w(\boldsymbol{k}) = m-1\}.$$

Computing $\mathcal{K}_{in}^S$ allows us to build all possible $\widetilde{\mathbb{K}}^0$ since we know there exists a set $\mathbb{K} \in \mathcal{K}_{in}^S$ such that every vector $\tilde{\boldsymbol{k}}^0$ of $\widetilde{\mathbb{K}}^0$ is of the form

$$\tilde{\boldsymbol{k}}^0 = \overbrace{1\dots1}^{m}|\overbrace{1\dots1}^{m}|\dots|\tilde{\boldsymbol{k}}_i^0|\dots|\overbrace{1\dots1}^{m}, \text{ with } \tilde{\boldsymbol{k}}_i^0 \in \mathbb{K}.$$

Hence, instead of trying all possible $L_{in}^i \in \mathbb{L}_{in}$, we can skip the first propagation through $\mathcal{S}_{in}$ and directly consider that the propagation starts at $\widetilde{\mathbb{K}}^0$.

We now need to test each set in $\mathcal{K}_{in}^S$. Recall that $\widetilde{\mathbb{K}}^0$ can only be built from 2^m-2 vectors $\tilde{\boldsymbol{k}}^0$. We will propagate each of those vectors through the remaining layers of the cipher, i.e. the following chain of propagation

$$\tilde{\boldsymbol{k}}^0 \xrightarrow{\mathcal{L}} \mathbb{K}^1 \xrightarrow{f_1} \dots \xrightarrow{f_{r-1}} \mathbb{K}^r.$$

Thus, for each $\tilde{\boldsymbol{k}}^0$, we can deduce a set $\mathbb{S}_{\tilde{\boldsymbol{k}}^0}$ of balanced bits using MILP. We then go through each set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_{in}^S$, and compute

$$\mathbb{S}_{\widetilde{\mathbb{K}}^0} = \bigcap_{\tilde{\boldsymbol{k}}^0 \in \widetilde{\mathbb{K}}^0} \mathbb{S}_{\tilde{\boldsymbol{k}}^0}.$$

If there is one non-empty $\mathbb{S}_{\widetilde{\mathbb{K}}^0}$, then we know that $\widetilde{\mathbb{K}}^0$ will lead to a set of balanced bits, given by $\mathbb{S}_{\widetilde{\mathbb{K}}^0}$.

Finally, using a precomputed table $\mathcal{T}_{in}^S$ defined as

$$\mathcal{T}_{in}^S[\mathbb{K}] := \{(L_{in}, \boldsymbol{k}) \in \mathbb{L}_{in} \times \mathbb{F}_2^m \mid \boldsymbol{k} \xrightarrow{S_{in}^i} \mathbb{K} \text{ and } w(\boldsymbol{k}) = m-1\},$$

we are able to deduce a linear map $L_{in}^i \in \mathbb{L}_{in}$ and a vector $\boldsymbol{k}^0$ such that we get an integral distinguisher over $E \circ L_{in}$ starting from the initial division property $\boldsymbol{k}^0$.

In summary, we first propagate each of the 2^m-2 vectors through $f \circ \dots \circ f \circ \mathcal{L}$. Then according to this, for each set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_S$, we check if each vectors of $\widetilde{\mathbb{K}}^0$ lead to the same balanced bits through $f \circ \dots \circ f \circ \mathcal{L}$. If so, then using $\mathcal{T}_{in}^S$ we can easily deduce a linear map L_{in} and an initial division property which results in an integral distinguisher.

Reducing the amount of work for L_{out}. Again, we first only consider $L_{out} \circ E$, and will see in the next part how to combine this with the previous section to get a distinguisher over $L_{out} \circ E \circ L_{in}$. For L_{out}, if we search naively, we would need

to try each possible matrix from $\mathbb{L}_{out}$. However, this is actually not necessary. Indeed, recall that there is an integral distinguisher i.i.f the last division property set $\mathbb{K}^r$ does not contains all unit vectors, and thus we only need to check if each unit vector belongs to $\mathbb{K}^r$. Now consider a division property vector $\boldsymbol{k}$ which is sent to such a unit vector $\boldsymbol{e}_i$ through the last (modified) S-box layer. That is, we have $\boldsymbol{k} \xrightarrow{\mathcal{S}_{out}} \boldsymbol{e}_i$ where $\mathcal{S}_{out} = (S^0_{out}, \ldots, S^{s-1}_{out})$. In that case, then all S-boxes except one have a output division property vector equal to $0 \ldots 0$. From Proposition 2, we know that this means that the corresponding input vector is also $0 \ldots 0$. Hence, $\boldsymbol{k}$ will be of the form

$$\overbrace{0 \ldots 0}^{m} | \overbrace{0 \ldots 0}^{m} | \ldots | \tilde{\boldsymbol{k}} | \ldots | \overbrace{0 \ldots 0}^{m}$$

where $\tilde{\boldsymbol{k}}$ is a non-zero vector of $\mathbb{F}_2^m$.

According to this, we can do the following. First, we compute, for each $L_{out} \in \mathbb{L}_{out}$, all possible sets $\mathbb{K}$ such that $\mathbb{K} \xrightarrow{S_{out}} \mathbb{K}'$, with $S_{out} = L_{out} \circ S$ and $\mathbb{K}'$ does not contain all unit vectors over m bits. According to those notations, denote by $\mathcal{K}^S_{out}$ the set

$$\mathcal{K}^S_{out} = \{\mathbb{K} \mid \exists L_{out} \in \mathbb{L}_{out} \text{ and } \mathbb{K}' \text{ s.t. } \mathbb{K} \xrightarrow{S_{out}} \mathbb{K}'\}.$$

We can write the division property propagation chain

$$\boldsymbol{k^0} \xrightarrow{f} \mathbb{K}^1 \xrightarrow{f} \ldots \mathbb{K}^{r-1} \xrightarrow{\mathcal{S}_{out}} \mathbb{K}^r.$$

The thing is, we do not know which L_{out} to use, and thus cannot propagate through S_{out}. But instead, we compute a subset $\widetilde{\mathbb{K}}$ of $\mathbb{K}^{r-1}$ such that for every vector $\boldsymbol{k}$ of $\widetilde{\mathbb{K}}$, the non-zeros elements of $\boldsymbol{k}$ all belongs to a single S-box block, i.e. is of the form

$$\overbrace{0 \ldots 0}^{m} | \overbrace{0 \ldots 0}^{m} | \ldots | \tilde{\boldsymbol{k}} | \ldots | \overbrace{0 \ldots 0}^{m}$$

with $\tilde{\boldsymbol{k}}$ a non-zero vector of $\mathbb{F}_2^m$. Thus, if there is a propagation $\boldsymbol{k} \xrightarrow{\mathcal{S}_{out}} \boldsymbol{e}$ where $\boldsymbol{e}$ is a unit vector, the we must have $\boldsymbol{k} \in \widetilde{\mathbb{K}}$. Now from $\widetilde{\mathbb{K}}$, build the following sets for each $i \in \{0, \ldots, s-1\}$

$$\mathbb{K}^{r-1}_i = \{\tilde{\boldsymbol{k}} \text{ s.t. } 0 \ldots 0 | \tilde{\boldsymbol{k}} | 0 \ldots 0 \in \widetilde{\mathbb{K}} \text{ where } \tilde{\boldsymbol{k}} \text{ is on the } i\text{-th S-box}\}.$$

These sets K^{r-1}_i will allow us to see if we can get a distinguisher. Indeed, if for at least one $i \in \{0, \ldots, s-1\}$ we have $\mathbb{K}^{r-1}_i \in \mathcal{K}^S_{out}$, then we know that we can get a distinguisher over $L_{out} \circ E$. Then, if we use a precomputed table $\mathcal{T}^S_{out}$ defined as

$$\mathcal{T}^S_{out}[\mathbb{K}] = \{L_{out} \in \mathbb{L}_{out} \mid \exists \mathbb{K}' \text{ s.t. } \mathbb{K} \xrightarrow{S_{out}} \mathbb{K}' \text{ and } \mathbb{E}_m \not\subset \mathbb{K}'\},$$

we know that there exists a linear map $L^i_{out} \in \mathcal{T}^S_{out}[\mathbb{K}^{r-1}_i]$ and a unit vector $\boldsymbol{e} \in \mathbb{F}_2^m$ such that $\mathbb{K}^{r-1}_i \xrightarrow{S^i_{out}} \mathbb{K}'$ where $\boldsymbol{e} \notin \mathbb{K}'$. Hence, we know that the unit

vector $0\dots0|\boldsymbol{e}|0\dots0 \in \mathbb{F}_2^n$ will not belong to $\mathbb{K}^r$, which means that we have at least one balanced bit. In summary, to search for each block L_{out}^i, we just need to compute all sets $\mathbb{K}_i^{r-1}$ and check if at least one of them belongs to $\mathcal{K}_{out}^S$. If so, then we can deduce from $\mathcal{T}_{out}^S$ which block L_{out}^i to use such that this results in an integral distinguisher.

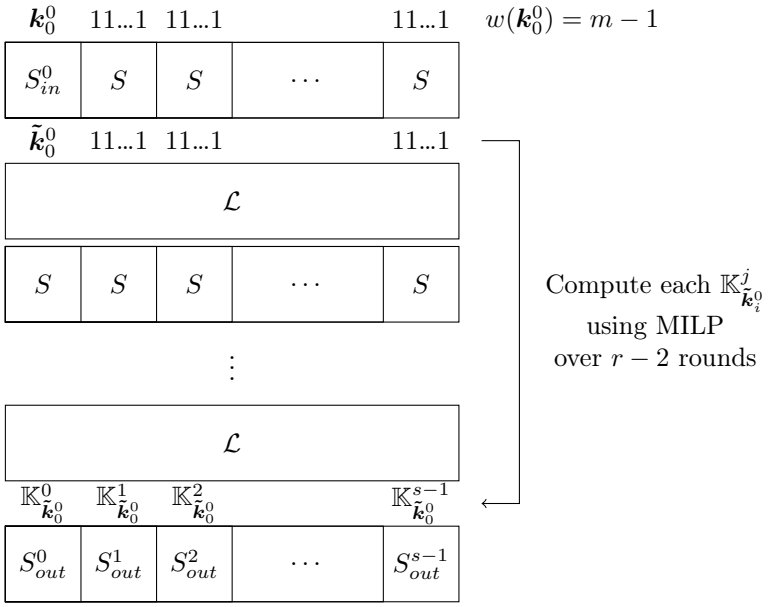

Fig. 2: Overall framework of our search algorithm, where we search for L_{in}^0

Putting everything together. We can now combine the two previous sections to search for a distinguisher over $L_{out} \circ E \circ L_{in}$. The overall idea is given in Figure 2. We first write $L_{out} \circ E \circ L_{in}$ as

$$\mathcal{S}_{out} \circ \underbrace{f \circ \cdots \circ f}_{r-2 \text{ rounds}} \circ \mathcal{L} \circ \mathcal{S}_{in},$$

and get the following propagation chain

$$\boldsymbol{k}^0 \xrightarrow{\mathcal{S}_{in}} \widetilde{\mathbb{K}}^0 \xrightarrow{\mathcal{L}} \mathbb{K}^1 \xrightarrow{f} \dots \xrightarrow{f} \mathbb{K}^{r-1} \xrightarrow{\mathcal{S}_{out}} \mathbb{K}^r,$$

where $w(\boldsymbol{k}^0) = n-1$. According to the previous part, we first start by computing $\mathcal{K}_{in}^S, \mathcal{T}_{in}^S, \mathcal{K}_{out}^S$ and $\mathcal{T}_{out}^S$. Then, for each S-box block i of the first layer, and for each of the $2^m - 2$ initial division property vectors $\tilde{\boldsymbol{k}}_i^0$, we use MILP to compute all the sets $\mathbb{K}_j^{r-1}, j \in \{0, \dots, s-1\}$ through $f \circ \cdots \circ f \circ \mathcal{L}$, where there are $r-2$

applications of f. We denote by $\mathbb{K}^j_{\tilde{\boldsymbol{k}}^0_i}$ these sets to tie them with $\tilde{\boldsymbol{k}}^0_i$, and $\mathbb{S}^j_{\tilde{\boldsymbol{k}}^0_i}$ the resulting (possibly empty) set of balanced bits. Then for each set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}^S_{in}$, we compute the following intersections for each $j \in \{0, \ldots, s-1\}$:

$$\mathbb{S}^j_{\widetilde{\mathbb{K}}^0} = \bigcap_{\tilde{\boldsymbol{k}}^0_i \in \widetilde{\mathbb{K}}_0} \mathbb{S}^j_{\tilde{\boldsymbol{k}}^0_i},$$
$$\mathbb{L}^j_{\widetilde{\mathbb{K}}^0} = \bigcap_{\tilde{\boldsymbol{k}}^0_i \in \widetilde{\mathbb{K}}_0} \mathcal{T}^S_{out}[\mathbb{K}^j_{\tilde{\boldsymbol{k}}^0_i}].$$

Now, if there is at least one j such that both $\mathbb{S}^j_{\widetilde{\mathbb{K}}^0}$ and $\mathbb{L}^j_{\widetilde{\mathbb{K}}^0}$ are non-empty, then we got a distinguisher. Indeed, by putting any map from $\mathbb{L}^j_{\widetilde{\mathbb{K}}^0}$ after the j-th S-box in the last layer, and any map from $\mathcal{T}^S_{in}[\widetilde{\mathbb{K}}^0]$ before the i-th S-box in the first layer, we know that all bits in $\mathbb{S}^j_{\widetilde{\mathbb{K}}^0}$ will be balanced. The whole procedure is summarized in Algorithm 1.

Overall, the number of call to the MILP model can be upper bounded as follow. First, we need to compute all $\mathbb{K}^j_{\tilde{\boldsymbol{k}}^0_i}$ for each of the $s(2^m-2)$ possible $\tilde{\boldsymbol{k}}^0$. Then, each set $\mathbb{K}^j_{\tilde{\boldsymbol{k}}^0_i}$ can contain at most 2^m vectors, and getting one vector of any of these sets cost one call to the MILP model. Since there are s of those sets, we need $s2^m$ calls to the MILP model. Note however that in practice, this is much lower, as we don't need to recover the redundant vectors. This means that for example, the sets $\{0001, 0011\}$ and $\{0001\}$ are considered the same, as 0011 is redundant in the first set and thus can be removed. If we go through all sets with $m = 4$, then the maximum size of any set $\mathbb{K}^j_{\tilde{\boldsymbol{k}}^0_i}$ is 6, and there are only 167 possible sets (compared to, in theory, a maximum size of 16, and 2^{16} possible sets). In total, we need at most $s^2(2^m-2)2^m$ calls to the MILP solver, and the factor 2^m is actually much lower in practice. Moreover, if we go through each of the 2^m-2 vectors $\tilde{\boldsymbol{k}}^0_i$ in a smart way, we can often reduce further the number of calls to the MILP solver. Indeed, if we first go through all vectors of weight $m-1$ and compute all corresponding $\mathbb{S}^j_{\tilde{\boldsymbol{k}}^0_i}$, we are left with two cases :

- All $\mathbb{S}^j_{\tilde{\boldsymbol{k}}^0_i}$ are empty for all vectors $\tilde{\boldsymbol{k}}^0_i$ of weight $m-1$, and thus we do not need to go further. Indeed, for any vector $\boldsymbol{k}$ such that $w(\boldsymbol{k}) < m-1$, we know that there is a vector $\tilde{\boldsymbol{k}}^0_i$ of weight $m-1$ such that $\tilde{\boldsymbol{k}}^0_i \succeq \boldsymbol{k}$. Hence, since there is no balanced bit from all vectors $\tilde{\boldsymbol{k}}^0_i$ of weight $m-1$, then we cannot have any balanced bit from any vector of weight strictly lower than $m-1$ (see [11] Proposition 2).
- Otherwise, we first check if there is any set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_{in}$ built only from vectors of weight $m-1$. If so, then we apply Algorithm 1 from line 11 to line 22 to check if we can find a distinguisher. If no distinguisher exists, or if none of the set of $\mathcal{K}_{in}$ are built only from vectors of weight $m-1$, then we go through all vectors of weight $m-2$ and do the same procedure and so on.

In practice, this allows us to significantly reduce the time needed to find a distinguisher, or even prove than none exists, and this will be detailed in the next section.

Algorithm 1 Searching L_{in} and L_{out}

Compute $\mathcal{K}_{in}^S, \mathcal{T}_{in}^S, \mathcal{K}_{out}^S$ and $\mathcal{T}_{out}^S$
for $i = 1 \dots s$ **do**
  for each of the $2^m - 2$ vectors $\tilde{\boldsymbol{k}}_i^0$ **do**
    Generate a MILP model for $f \circ \cdots \circ f \circ \mathcal{L}$ *$r-2$ application of f*
    Set the initial division property to $1 \dots \tilde{\boldsymbol{k}}_i^0 \dots 1$ *$\tilde{\boldsymbol{k}}_i^0$ on the i-th block*
    Compute each set $\mathbb{K}^j_{\tilde{\boldsymbol{k}}_i^0}, j \in \{0, \dots, s-1\}$ using the MILP model
    Deduce each set $\mathbb{S}^j_{\tilde{\boldsymbol{k}}_i^0}$ according to $\mathcal{K}_{out}$
  end for
  for $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_{in}$ **do**
    $L_{out}^0, \dots, L_{out}^{s-1} \leftarrow I_m$ *I_m : identity matrix of size m*
    $\mathbb{S} \leftarrow \varnothing$
    for $j = 1 \dots s$ **do**
      Compute $\mathbb{S}^j_{\widetilde{\mathbb{K}}^0}$ and $\mathbb{L}^j_{\widetilde{\mathbb{K}}^0}$
      if $\mathbb{S}^j_{\widetilde{\mathbb{K}}^0} \neq \varnothing$ and $\mathbb{L}^j_{\widetilde{\mathbb{K}}^0} \neq \varnothing$ **then**
        $L_{out}^j \leftarrow$ any element from $\mathbb{L}^j_{\widetilde{\mathbb{K}}^0}$
        $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{S}^j_{\widetilde{\mathbb{K}}^0}$
      end if
    end for
    if $\mathbb{S} \neq \varnothing$ **then** *We have at least one balanced bit*
      $L_{in}^i \leftarrow$ any element in $\mathcal{T}_{in}^S[\widetilde{\mathbb{K}}^0]$
      return $Diag(I_m, \dots, L_{in}^i, \dots, I_m), Diag(L_{out}^0, \dots, L_{out}^{s-1})$
    end if
  end for
end for

4 Application to RECTANGLE and PRESENT

4.1 Division Property against 10-round RECTANGLE

RECTANGLE [18] is a lightweight block cipher designed for fast implementation using bit-slice techniques. It is a 64-bit block cipher, using 4-bit S-boxes and a permutation as the linear layer. There are 80-bit and 128-bit key sizes, and the total number of round in 25 in both cases. The best known division property

based integral distinguisher is from [17] over 9 rounds, using 2^{60} data and resulting in 16 balanced bits. By applying the previous algorithm, we were able to find a distinguisher over 10 rounds, using 2^{63} data and resulting in 1 balanced bit. The distinguisher is built on $L_{out} \circ E \circ L_{in}$, where the block 0 of L_{in} is

$$L_{in}^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and L_{out} is the identity. This results in the following distinguisher, where :

- `c` denotes a constant bit,
- `a` denotes a bit taking all possible values through the set,
- `b` denotes a balanced bit,
- `?` denotes a bit in an unknown state.

$$\text{Input : } \begin{pmatrix} \texttt{aaaaaaaaaaaaaaaa} \\ \texttt{aaaaaaaaaaaaaaaa} \\ \texttt{aaaaaaaaaaaaaaa}\mathbf{\texttt{c}} \\ \texttt{aaaaaaaaaaaaaaaa} \end{pmatrix} \rightarrow \text{Output : } \begin{pmatrix} \texttt{??????????}\mathbf{\texttt{b}}\texttt{?????} \\ \texttt{????????????????} \\ \texttt{????????????????} \\ \texttt{????????????????} \end{pmatrix}$$

Overall, the time needed to compute all $\mathbb{K}^j_{\tilde{\boldsymbol{k}}_i^0}$ for a given $\tilde{\boldsymbol{k}}_i^0$ is about 400 seconds in average. The reason this distinguisher exists is that when considering $S' = S \circ L_{in}^0$ where S is the S-box of `RECTANGLE`, the transition $1101 \xrightarrow{S'} \{0101, 1110\}$ is now possible, while the set $\{0101, 1110\}$ was not reachable from the original S-box S. Note that this distinguisher does not depends on the key size, and thus is applicable to both the 80-bit and the 128-bit key variants.

4.2 Strengthening RECTANGLE

According to our observations in Section 3.1, it is natural to think that the resistance of an S-box-based cipher against division property is highly related to the number of weight 1 vectors in the division property propagation table of the S-box. As such we wondered how the choice of the S-box affects the resistance of `RECTANGLE` against division property. Since the rational behind S-box design highly depends on potential applications of the resulting block cipher, we restricted the search space to S-boxes linearly equivalent to the original `RECTANGLE` S-box. Indeed, linearly equivalent S-boxes have similar structures regarding differential and linear properties. Given two m-bit S-boxes S and S' such that $S' = B \circ S \circ A$, if there is a differential $(\Delta_i, \Delta_o) \in \mathbb{F}_2^{2m}$ such that $S(x) \oplus S(x \oplus \Delta_i) = \Delta_o$ holds with probability p, then since A ans B are linear and invertible, there is a differential $(\Delta_i', \Delta_o') = (A^{-1}.\Delta_i, B.\Delta_o)$ of the same probability for S'. Hence the DDT is essentially the same, and we may expect that it should not drastically change the resistance against differential attacks compared to using the original S-box, and the same kind of observations can be made for linear attacks.

For 4-bit S-boxes, as there are about $2^{14.3}$ invertible matrices of size 4, the main issue we are facing is the high complexity of trying all the $2^{28.6}$ candidates for (A, B). Indeed, many hours are required to search for a division property distinguisher, making the whole search infeasible. Hence, we propose to use several heuristics to select which couples (A, B) to try.

Selecting good S-boxes. Our first idea was to compute the division property propagations tables of all candidates (A, B). This required to perform $2^{28.6} \times 2^{2\times4} = 2^{36.6}$ non trivial operations and took approximately 80h on a Xeon E5-2695 (72 cores). Among all those propagation tables, none of them was *perfect* : i.e. having 14 transitions $k \to \{1000, 0100, 0010, 0001\}$. However we found 56 almost *perfect* tables: i.e. having 13 transitions $k \to \{1000, 0100, 0010, 0001\}$. Note that many couples lead to the same table but the division property only depends on the table. Hence it is enough to try only one representative per table. Since some implementations of block ciphers do not use a table to store the S-box, we believe it makes sense to select the representative which would add less extra XORs. Hence, for each of the 56 tables we selected the couple (A, B) with the lower XOR count and ran our new automated search tool.

As a result, we found that by using

$$A = \begin{pmatrix} 1&1&0&0\\ 1&0&1&0\\ 1&0&0&1\\ 0&1&0&0 \end{pmatrix} \text{ and } B = \begin{pmatrix} 1&0&1&0\\ 0&1&1&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix}$$

which results in only 5 XOR, and replacing all S-boxes of `RECTANGLE` by $S' = B \circ S \circ A$ where S is the original S-box of `RECTANGLE`, then even when using our technique, there is no distinguisher over 9 rounds of this variant of `RECTANGLE`. We were however able to find a distinguisher over 8 rounds of this variant, using our technique where L_{in} is built with

$$L^0_{in} = \begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&1&1\\ 0&0&1&0 \end{pmatrix}$$

and all others L^i_{in} for $i \neq 0$ are the identity, and each block L^i_{out} of L_{out} is

$$L^i_{out} = \begin{pmatrix} 1&1&0&0\\ 1&0&0&0\\ 0&0&1&0\\ 0&0&0&1 \end{pmatrix}.$$

This results in a distinguisher of data complexity 2^{63} resulting in 14 balanced bits. Note that the classic search algorithm for division property distinguishers lead to no distinguisher even over 8 rounds, which shows again that our extension technique can find new distinguishers.

We believe that this could lead to a new criteria when designing S-boxes, as for the case of RECTANGLE, it improves the resistance against division property based distinguishers by 2 rounds. We would thus first build the S-box according to classical criteria (differential and linear resistance, ...), then look at the linear equivalent S-boxes and take the one with "the best" division property propagation table. We do not have an exact definition of what would be "the best" propagation table, but maximizing the number of vectors of weight 1 over all the output sets seems to be an interesting choice.

4.3 Division Property against PRESENT

PRESENT [3] is a 64-bit lightweight block cipher, using either 80 bits or 128 bits keys, with a round function very similar to RECTANGLE and using 4-bit S-boxes. The best known division property based integral distinguisher is from [17] over 9 rounds, requiring 2^{60} data and resulting in 1 balanced bit. We applied our previous algorithm to this block cipher, and were actually able to show that our technique cannot lead to a distinguisher over 10 rounds of PRESENT. Indeed, as mentioned at the end of the previous section, if we go through all vectors $\tilde{\boldsymbol{k}}_i^0$ of weight 2, then all of the resulting sets $\mathbb{S}^j_{\tilde{\boldsymbol{k}}_i^0}$ are empty, meaning that if there is at least one vector of weight 2 or lower in $\widetilde{\mathbb{K}}^0$, then this cannot results in some balanced bits after 10 rounds. Moreover, if we go through all linear mappings $L \in \mathbb{L}_{in}$ and compute all possibles propagations $\boldsymbol{k} \xrightarrow{S'} \mathbb{K}$ where $w(\boldsymbol{k}) = 3$ and $S' = S \circ L$ with S the S-box of PRESENT, then $\mathbb{K}$ will *always* contains at least one vector of weight 2, or at least one vector of weight 1. Hence, no matter which linear map we take from $\mathbb{L}_{in}$, we know that after the first S-box layer, there will always be a vector of weight either 1 or 2, which lead to a set $\mathbb{K}^{10}$ containing all unit vectors, and thus no distinguisher over 10 rounds can be built using our technique.

4.4 Strengthening PRESENT

As for RECTANGLE, we searched for another S-box to use which is linear equivalent to the S-box of PRESENT such that it would improve the resistance against division property based distinguishers. By using $S' = B \circ S \circ A$ with

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } B = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

instead of S for all S-boxes of PRESENT, then we do not have any division property based distinguisher over 8 rounds of this variant of PRESENT even when using our extension technique. However, we found a distinguisher over 7 rounds using our technique, with L_{out} being the identity and L_{in} being built

with

$$L_{in}^0 = \begin{pmatrix} 1\,1\,0\,0 \\ 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}$$

and L_{in}^i as the identity for $i \neq 0$. This results in a distinguisher of data complexity 2^{63} with all 64 bits being balanced. The classical search algorithm was only able to find a distinguisher on up to 6 rounds. This again highlights that our extension technique allows to find better distinguishers than the classical search.

Note that, for non table-based implementation, the new S-box we propose only requires two extra XORs compared to the original S-box of PRESENT.

5 Conclusion

In this paper, we studied further the division property and the distinguishers that are built from it. We showed that while the previous search methods were able to efficiently find some integral distinguishers based on the division property, the search space explored by these methods does not actually cover all possibilities. As such, we showed that for r rounds of a block cipher E, considering $E' = L_{out} \circ E \circ L_{in}$ instead of E, where L_{out} and L_{in} are block diagonal linear maps, can lead to some integral distinguisher over E', while E does not have any. We provided an algorithm to find such distinguisher, and successfully applied it to the block cipher RECTANGLE, on which we found an integral distinguisher over 10 rounds, requiring 2^{63} data and leading to 1 balanced bit. This is one more round that the previously known distinguishers. The design of our algorithm also allowed us to prove that our technique cannot extend the best distinguisher on PRESENT over one more round. Finally, according to our observations, we were able to exhibit some variants of RECTANGLE and PRESENT which have a better resistance against integral distinguisher based on the division property. Namely the maximum number of round on which we could find an integral distinguisher over our variant of RECTANGLE and PRESENT is 2 rounds lower than when using the original S-box. This might give a new design criteria for S-boxes and further research about this will be needed.

We believe that overall, this technique could open up a lot of questions and possibilities. Indeed, we basically decomposed a block cipher E as

$$E = (E_2 \circ L_{out}^{-1}) \circ (L_{out} \circ E_1 \circ L_{in}) \circ (L_{in}^{-1} \circ E_0),$$

and merged L_{in} and L_{out} with the first S-box layer. But could we use the same technique at a lower level, i.e. decomposing the round function as $f = \mathcal{L} \circ L^{-1} \circ L \circ \mathcal{S}$, merging L with $\mathcal{S}$ for example ? In a more general view, the question is : what is the best representation of a block cipher to propagate the division property ?

Also, our algorithm focus on finding any distinguisher over an SPN block cipher. Thus, how could we find an optimal distinguisher (in term of data) using this technique ? Applying our algorithm when more than one S-box has

an input division property which differs from $1 \dots 1$ seems quite hard in term of complexity, thus we may need either to improve our algorithm, or to find a new one. The same issue comes up when considering 8-bit S-boxes, as we need more calls to the solver, and the resulting MILP models are way more complicated, and thus takes a longer time to be solved. Finally, could this also apply to other constructions such as Feistel block ciphers or permutation based block ciphers ? Indeed, our algorithm is efficient because we can basically only study the propagation from after the first S-box layer to before the last S-box layer. We made our implementation available at `https://github.com/ExtendDivProp/ExtendDivProp`.

References

1. Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Trans. Symmetric Cryptol. **2017**(4), 99–129 (2017)
2. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. pp. 123–153 (2016)
3. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings. pp. 450–466 (2007)
4. Boura, C., Canteaut, A.: Another view of the division property. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. pp. 654–682 (2016)
5. Daemen, J., Rijmen, V.: Aes proposal: Rijndael (1999)
6. Derbez, P., Fouque, P.: Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES. In: Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. pp. 541–560 (2013)
7. Eskandari, Z., Kidmose, A.B., Kölbl, S., Tiessen, T.: Finding integral distinguishers with ease. IACR Cryptology ePrint Archive (accepted at SAC2018) **2018**, 688 (2018)
8. Gurobi Optimization, L.: Gurobi optimizer reference manual (2018), `http://www.gurobi.com`
9. Sun, L., Wang, W., Liu, R., Wang, M.: Milp-aided bit-based division property for arx-based block cipher. IACR Cryptology ePrint Archive **2016**, 1101 (2016)
10. Sun, L., Wang, W., Wang, M.: Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. IACR Cryptology ePrint Archive **2016**, 811 (2016)
11. Sun, L., Wang, W., Wang, M.: Automatic search of bit-based division property for ARX ciphers and word-based division property. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications

of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. pp. 128–157 (2017)
12. Sun, L., Wang, W., Wang, M.: Automatic search of bit-based division property for ARX ciphers and word-based division property. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. pp. 128–157 (2017)
13. The Sage Developers: SageMath, the Sage Mathematics Software System (Version x.y.z) (YYYY), http://www.sagemath.org
14. Todo, Y.: Structural evaluation by generalized integral property. In: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. pp. 287–314 (2015)
15. Todo, Y., Morii, M.: Bit-based division property and application to simon family. In: Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. pp. 357–377 (2016)
16. Wang, Q., Hao, Y., Todo, Y., Li, C., Isobe, T., Meier, W.: Improved division property based cube attacks exploiting algebraic properties of superpoly. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I. pp. 275–305 (2018). https://doi.org/10.1007/978-3-319-96884-1_10, https://doi.org/10.1007/978-3-319-96884-1_10
17. 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: Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. pp. 648–678 (2016)
18. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. SCIENCE CHINA Information Sciences **58**(12), 1–15 (2015)
19. Zhang, W., Rijmen, V.: Division cryptanalysis of block ciphers with a binary diffusion layer. IACR Cryptology ePrint Archive **2017**, 188 (2017)