Linearly equivalent S-boxes and the division property

Baptiste Lambin[1,2] (ID) **· Patrick Derbez[1] · Pierre-Alain Fouque[1]**

Abstract

Division property is a cryptanalysis method that proves to be very efficient on block ciphers. Computer-aided techniques such as MILP have been widely and successfully used to study various cryptanalysis techniques, and it especially led to many new results for the division property. Nonetheless, we claim that the previous techniques do not consider the full search space. We show that even if the previous techniques fail to find a distinguisher based on the division property over a given function, we can potentially find a relevant distinguisher over a linearly equivalent function. We 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. As a result, we exhibit a new distinguisher over 10 rounds of RECTANGLE, while the previous best was over 9 rounds, and rule out such a distinguisher over more than 9 rounds of PRESENT. We also give some insight about the construction of an S-box to strengthen a block cipher against our technique. We prove that using an S-box satisfying a certain criterion is optimal in term of resistance against classical division property. Accordingly, we exhibit stronger variants of RECTANGLE and PRESENT, improving the resistance against division property based distinguishers by 2 rounds.

Keywords Cryptanalysis · Division Property · RECTANGLE

Mathematics Subject Classification 94A60

✉ Baptiste Lambin
  baptiste.lambin@protonmail.com

  Patrick Derbez
  patrick.derbez@irisa.fr

  Pierre-Alain Fouque
  pierre-alain.fouque@univ-rennes1.fr

[1] Univ Rennes, CNRS, IRISA, Rennes, France

[2] Present Address: Horst Görtz Institute for IT-Security, Ruhr-Universität Bochum, Bochum, Germany

1 Introduction

Division property is a distinguishing property which was first presented by Todo at Eurocrypt'15 [15]. This cryptanalysis technique quickly became a hot topic in the community, especially since it led to the first theoretical attack against full MISTY1 [14]. 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 [17]. The idea of the division property is the same as in integral, higher-order differential and cube-attacks, namely, proving that if one encrypts a set of plaintexts with a certain structure, then the resulting set of ciphertexts will have some balanced bits, i.e. bits which sum to zero with probability 1 when going through the whole set of ciphertexts. The main difference between these different techniques comes from how one can prove this property. Division property is a more fine-grained property: it mainly comes down to tracking which monomials may or may not appear in the Algebraic Normal Form (ANF) of the whole block cipher, so that, for a set of ciphertexts $\mathbb{X}$, we can predict with probability 1 the result of $\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{k}}$, where $\boldsymbol{k}$ represents the indicator vector defining the value of the monomial $\boldsymbol{x}^{\boldsymbol{k}} = \prod_i x_i^{k_i}$. The distinguisher begins by generating a set of plaintexts where c bits are fixed to an arbitrary constant, resulting in $n - c$ variables which then take all possible values (thus generating an affine space of dimension $n - c$). We want to know whether a monomial of degree $n - c$, i.e. implying all variables, exists in each coordinate of the ANF of the block cipher. If no such monomials appear in the i-th coordinate, this coordinate is of degree $< n - c$. Consequently, when we sum the i-th bit through the whole corresponding set of ciphertexts, the sum will be zero, as we compute the sum of a function of degree $< n - c$ over a space of dimension $n - c$. Essentially, the division property defines a set $\mathbb{K} \subset \mathbb{F}_2^n$ of monomials which divides $\mathbb{F}_2^n$ into two parts. For one part $\overline{\mathbb{K}} = \{\boldsymbol{k} | \exists \bar{\boldsymbol{k}} \in \mathbb{K} \ s.t. \ \bar{\boldsymbol{k}} \preceq \boldsymbol{k}\}$, for $\preceq$ the usual preceding order, we cannot predict the result of this sum. However, for any $\boldsymbol{k} \in \mathbb{F}_2^n \setminus \overline{\mathbb{K}}$, we know that $\bigoplus_{\boldsymbol{x} \in \mathbb{X}} \boldsymbol{x}^{\boldsymbol{k}} = 0$, i.e. we track which monomials we know to be summing to zero.

Automatic tools Studying the propagation of an initial division property through a block cipher is a challenging task requiring to be computer-aided. At Asiacrypt'16, Xiang et al. [18] 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 result they obtained the best known division property distinguishers on SIMON, SIMECK, PRESENT and RECTANGLE. In [20], 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. [12] 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 the search space considered by previous automatic search tools dedicated to division property is incomplete as they do not exhaust

all the search space. More precisely, propagating an initial division property through a block cipher requires decomposing 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 represented. Indeed, linearly equivalent Sboxes do not change the propagation of differentials, while it is not the case for the division property. For instance, in Sect. 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 preferred 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 significantly higher than one can be thinking and looking efficiently for the best distinguisher boils down to 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 parts 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}, E can be split 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. However, the division property is different from differential and linear cryptanalysis. Hence, one of the main problems 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 distinguisher through $L_{out} \circ E_1 \circ L_{in}$. We focused on linear mappings L_{in}, L_{out} which are block diagonal, of block size m, where m is the size of the S-box. 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. We severely reduce the complexity of the search. Indeed, to search for a distinguisher over r rounds, a naive algorithm would need about $ms^3 2^{2m^2}$ (where s in the number of S-boxes) calls to the MILP solver with a model representing r rounds of the block-cipher. However in our case, we only need about $s^2 2^{2m}$ calls to the solver, on a model representing $r - 2$ rounds which is thus much more efficient to solve. 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. We emphasize that this is an advantage of our algorithm, as it allows us to *prove* that a given cipher is resistant to our technique, as proving negative results is in general harder than findings attacks since we have to check all such attacks.

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. provide new insights into the division property, presenting a new approach to it. In particular they show several interesting results concerning the resistance of S-box-based block ciphers against division property. Here we prove that if an S-box satisfies a specific criterion (which is close to the one in [4]), then this S-box is *optimal* in term of resistance against classical division property (i.e. without our extension technique). We define optimality in the sense that if one uses such a *perfect* S-box and can find a distinguisher on at most $r^\star$ rounds, then using any other S-box will lead to a distinguisher on r rounds with $r^\star \leq r$. To our knowledge, this is the first time that such a result is given for division property, and could be considered as a new criterion for designing S-boxes. Our criterion is the following : if each coordinate of the ANF of an n-bit

S-box contains *all* monomials of degree $n - 1$, then this S-box is optimal. Note that our criterion is equivalent to a very specific structure for the division property propagation table, and this table is a major component in the existing search algorithms [18,20]. Compared to the criterion in [4], we have two major differences. The first is that any S-box satisfying our criterion *does not* satisfy the one in [4]. Indeed, their criterion is that *any* non-trivial linear combination of the coordinates of the ANF must be of degree $n - 1$. In our case, since all monomials of degree $n - 1$ appear in each coordinate, the sum of any two coordinates will be of degree $n - 2$. Nonetheless, the second difference is that our criterion leads to an optimality proof, whereas their criterion is more of an indication that an S-box satisfying it *should* be good enough.

According to this criterion on S-boxes, we try to strengthen both RECTANGLE and PRESENT against our technique. Note that when considering our technique, the criterion mentioned above does not seem to guarantee optimality. However, in regards to our experiments, it still seems to be the best choice. Indeed, to preserve some differential and linear property of the original S-boxes, we chose to only consider S-boxes which are linearly equivalent to the original ones. Unfortunately, for both RECTANGLE and PRESENT, it was not possible to generate a *perfect* S-box from linearly equivalent S-boxes, however, we found many *almost perfect* S-boxes. Trying all of them allowed us to find a linearly equivalent S-box for RECTANGLE such that the best distinguisher is over 7 rounds for classical division property, and 8 rounds when using our technique, while the best distinguisher we found with our technique is over 10 rounds of RECTANGLE when using the original S-box. Doing the same for PRESENT, we found an S-box such that the classical division property could only lead to a distinguisher on up to 6 rounds, and up to 7 rounds with our technique, while the best know division property based distinguisher for PRESENT is over 9 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. These experiments show that our new search process finds distinguishers against one extra round than classical search, highlighting again its interest, and also confirms that our strategy to choose these new S-boxes seems promising, as it improves the resistance of both algorithms by 2 rounds. We made our implementation available at https://github.com/ExtendDivProp/ExtendDivProp.

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 $(x_0, \ldots, x_{n-1})$. We denote $w(\boldsymbol{x})$ the hamming weight of $\boldsymbol{x} \in \mathbb{F}_2^n$. We denote $\boldsymbol{e}_i$ the i-th unit vector, and $\mathbb{E}_m$ denotes the set of all unit vectors of size m, i.e. vectors of hamming weight 1. 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. We denote $\mathcal{P}_m$ the set of all permutations over m elements. We denote $GL_m(\mathbb{F}_2)$ 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 by Todo [15] as a generalization of integral cryptanalysis, and later at FSE'16 [16], 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* [16]) A set $\mathbb{X} \subset \mathbb{F}_2^n$ has the division property $D_{\mathbb{K}}^n$, where $\mathbb{K} \subset \mathbb{F}_2^n$ is a set, if for all $\boldsymbol{u} \in \mathbb{F}_2^n$, we have

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

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 basic operations composing the block cipher.

Definition 2 (*Division trails* [18]) Let f denote the round function of an iterated block cipher. Assume the input set to the block cipher has initial division property $D_{\{k\}}^n$, and denote the division property after propagating through i rounds of the block cipher (i.e. i applications of f) by $D_{\mathbb{K}^i}^n$. Thus, we have the following chain of division property propagations :

$$\{\boldsymbol{k}\} \triangleq \mathbb{K}^0 \xrightarrow{f} \mathbb{K}^1 \xrightarrow{f} \mathbb{K}^2 \xrightarrow{f} \cdots \xrightarrow{f} \mathbb{K}^r.$$

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 $(\boldsymbol{k}^0, \boldsymbol{k}^1, \ldots, \boldsymbol{k}^r) \in \mathbb{K}^0 \times \mathbb{K}^1 \times \cdots \times \mathbb{K}^r$, 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 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 $D_{\{k\}}^n$, we can find whether $D_{\{k\}}^n$ allows to build an integral distinguisher using the following proposition. We recall that a given set $\mathbb{X} \subset \mathbb{F}_2^n$ has an integral property if there exists $0 \leq i \leq n-1$ such that

$$\bigoplus_{x \in \mathbb{X}} x_i = 0.$$

Proposition 1 ([18]) *Assume $\mathbb{X}$ is a set with division property $D_{\mathbb{K}}^n$, then $\mathbb{X}$ does not have any 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 implies that the result of the sum

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

is unknown since $e_i \succeq e_i$ and $e_i \in \mathbb{K}$, i.e. the i-th bit is not balanced. On the other hand, if we suppose that the ith bit is balanced, i.e.

$$\bigoplus_{x \in \mathbb{X}} x^{e_i} = 0,$$

then by definition of the division property $e_i \notin \mathbb{K}$, as it would otherwise mean that the ith bit is in an unknown state, which contradicts the fact that the i-bit is balanced. $\square$

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_k^{128}, where

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

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 means that the resulting set has a division property $D_{\mathbb{K}}^{128}$, 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 [16] 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 [11] proposed to use only the Copy and XOR operations to propagate division property vectors, it has been shown in [20] 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 [20] for the correct way to propagate division property vectors through a given linear layer.

For S-boxes, again using only the basic operations might result in a loss of information. Hence, [18] proposed an algorithm of complexity $\mathcal{O}\left(2^{2m}\right)$ to compute all possible pairs $k \xrightarrow{S} k'$ 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 [16], 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 [10,18,20] or SAT/SMT solvers [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 for MILP. 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 with 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.

2.3.1 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 [18]:

1. each division trail must satisfy all linear inequalities in the linear equality system, i.e. 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, i.e. 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 [18] for basic operations and S-boxes, [20] for linear layers and [10] for ARX block ciphers. Note that this step is not totally free.

For S-boxes, we first 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, [18] 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 of generating 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. [20] showed that the previous method [11] 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` [19], 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.

2.3.2 Searching for a distinguisher

As a result of the previous section, when modelizing an n-bit block cipher over r rounds, we have a set of variables $\{\mathbf{k}_i^j, i \in \{0, \cdots, n-1\}, j \in \{0, \cdots, r\}\}$ such that, for a given solution of the MILP problem, the corresponding values of these variables give a division trail $(\boldsymbol{k}^0, \boldsymbol{k}^1, \ldots, \boldsymbol{k}^r)$ with $\boldsymbol{k}^i = (\mathbf{k}_0^i, \ldots, \mathbf{k}_{n-1}^i)$. 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

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

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, 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 are no more solutions of value 1, we found all unit vectors belonging to $\mathbb{K}^r$, hence we can easily see whether or not there are 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 solutions, 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 this results in integral distinguishers previously unknown. We now explain the main idea behind using L_{out} and L_{in}. For L_{out}, 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 [16] when they introduced the *division property using three subsets*.

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 $D_{\boldsymbol{k}^0}^n$, through all the set, each bit i such that $k_i^0 = 0$ has a constant value, and if $k_i^0 = 1$, the bit i takes all possible values through the set. For example, the following set has division property D_{0011}^4

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

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 that considering $L_{out} \circ E \circ L_{in}$ instead of E is still meaningful. Classically, when an attacker uses 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. 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 asks 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 that is the case, the key guesses are supposed to be correct. 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, 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 omit $\mathcal{L}$ in the last round. Now our goal is to search if E' has an integral distinguisher based on the division property using MILP. Classically, we study the following propagation chain

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

Basically, we model 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 see this through an example.

3.1.1 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 [18] to compute all possible propagations through S_1, S_2 and L. Using this, if we look at the propagation of $x = 0111$ 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

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

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 tables of S_1 and S_2. Vectors of weight 2 are in bold

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 Fig. 1.

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.

We can see that, except when we have either the full zero or the full one vector, if we consider a division property chain $\mathbb{K}^0 \rightarrow \cdots \rightarrow \mathbb{K}^r$ of a block cipher, 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 Fig. 1). Recall that if the set $\mathbb{K}^r$ contains all unit vectors (i.e. of weight 1), 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 Fig. 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, 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 better distinguishers, and thus in the next section, we focus on the search of such distinguishers.

3.2 Searching for extended division property

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 $n = s \cdot m$. 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.

3.2.1 Reducing the search space of L_{in} and L_{out}.

Given 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 get more precise information about the propagation of division property vectors. However, 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. Consequently we want to put an invertible linear map L_{in}^i (resp. L_{out}^i) before (resp. after) each S-box of the first (resp. last) round. By doing so, we will denote by $S_{in}^i = S \circ L_{in}^i$ and $S_{out}^i = L_{out}^i \circ S$ 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_{in}^i and L_{out}^i.

Proposition 2 *Let S be an invertible m-bit S-box and P an m-bit permutation. Let $S_1 = S \circ P$ and $S_2 = P \circ S$, and $\mathbf{k} \xrightarrow{S} \mathbf{k}'$ be any valid division property propagation through S with $\mathbf{k}, \mathbf{k}' \in \mathbb{F}_2^m$. Then both propagations $P^{-1}(\mathbf{k}) \xrightarrow{S_1} \mathbf{k}'$ and $\mathbf{k} \xrightarrow{S_2} P(\mathbf{k}')$ are 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_{in}^i, we do not need to do the search for all $L_{in}^i \circ P$ where P goes through all possible permutations, as we could obtain the same result from the search using L_{in}^i by just permuting the initial division property with P. For example, if we have the set $\mathbb{K}^r$ from a given initial division property $\mathbf{k}$ through $L_{out} \circ E \circ L_{in}$, and we consider $L_{out}' \circ E \circ L_{in}'$ where $L_{in}' = L_{in} \circ (P_{in}^0, \dots, P_{in}^{s-1})$ and $L_{out}' = (P_{out}^0, \dots, P_{out}^{s-1}) \circ L_{out}$, where each P_{in}^i and P_{out}^i is a permutation over m bits, we

directly have that the initial division property $(P_{in}^0, \ldots, P_{in}^{s-1})^{-1}(\boldsymbol{k})$ propagates to the set $(P_{out}^0, \ldots, P_{out}^{s-1})(\mathbb{K}^r)$. In particular, if we have an integral distinguisher for $L_{out} \circ E \circ L_{in}$, so do we for $L'_{out} \circ E \circ L'_{in}$ (and vice-versa if $L_{out} \circ E \circ L_{in}$ does 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 much 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.

3.2.2 Reducing the amount of work for L_{in}

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 our goal is to exhibit a distinguisher on $E \circ L_{in}$, not necessarily the best one. As such, we 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 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 is only one specific S-box S_{in}^i which has an input of weight $m - 1$, while all the others S-boxes S_{in}^j with $j \neq i$ have $1 \ldots 1$ has input. Note that if a set $\mathbb{X}$ has division property $\boldsymbol{k} = 1 \ldots 1$, all bits takes all possibles values through the set, i.e. $\mathbb{X} = \mathbb{F}_2^m$. Hence, since we are considering bijective S-boxes, we have $S_{in}^j(\mathbb{X}) = \mathbb{F}_2^m$ for all $j \neq i$, and thus the resulting division property set is $\mathbb{K} = \{1 \ldots 1\}$.

From the previous remark, 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

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

This leads to the following chain of division property propagation

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

where

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

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 to build all possible $\widetilde{\mathbb{K}}^0$ since 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

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

Hence, instead of trying all possible $L_{in}^i \in \mathbb{L}_{in}$, we 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 propagate each of those vectors through the remaining layers of the cipher, i.e. the following chain of propagation

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

Thus, for each $\tilde{\boldsymbol{k}}^0$, we deduce a set $\mathbb{S}_{\tilde{\boldsymbol{k}}^0}$ of balanced bits using MILP. We then consider 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 $\mathbb{S}_{\widetilde{\mathbb{K}}^0}$, $\widetilde{\mathbb{K}}^0$ 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 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 $f \circ \cdots \circ f \circ \mathcal{L}$. Then, for each set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_S$, we check if each vector of $\widetilde{\mathbb{K}}^0$ lead to the same balanced bits through $f \circ \cdots \circ f \circ \mathcal{L}$. 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.

3.2.3 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 need to try each possible matrix from $\mathbb{L}_{out}$. However, this is actually not necessary. Indeed, recall that there is an integral distinguisher if and only if the last division property set $\mathbb{K}^r$ does not contain 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 $\boldsymbol{k} \xrightarrow{S_{out}} \boldsymbol{e}_i$ where $\mathcal{S}_{out} = (S_{out}^0, \ldots, S_{out}^{s-1})$. In that case, all S-boxes except one have an output division property vector equal to $0 \ldots 0$. Again, since we are using bijective S-boxes, this means that the output set is constant, and thus the input set is also constant, leading to a corresponding input vector $0 \ldots 0$. Hence, $\boldsymbol{k}$ will be of the form

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

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

Consequently, we first 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 $\mathcal{K}_{out}^S$ the set

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

We can write the division property propagation chain

$$\boldsymbol{k^0} \xrightarrow{f} \mathbb{K}^1 \xrightarrow{f} \cdots \mathbb{K}^{r-1} \xrightarrow{S_{out}} \mathbb{K}^r.$$

However, 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 belong to a single S-box block, i.e. $\boldsymbol{k}$ is of the form

$$\overbrace{0\ldots0}^{m} \mid \overbrace{0\ldots0}^{m} \mid \ldots \mid \tilde{\boldsymbol{k}} \mid \ldots \mid \overbrace{0\ldots0}^{m}$$

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

$$\mathbb{K}_i^{r-1} = \{\tilde{\boldsymbol{k}} \text{ s.t. } 0\ldots0 \mid \tilde{\boldsymbol{k}} \mid 0\ldots0 \in \widetilde{\mathbb{K}} \text{ where } \tilde{\boldsymbol{k}} \text{ is on the } i\text{-th S-box}\}.$$

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

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

we know that there exists a linear map $L_{out}^i \in \mathcal{T}_{out}^S[\mathbb{K}_i^{r-1}]$ and a unit vector $\boldsymbol{e} \in \mathbb{F}_2^m$ such that $\mathbb{K}_i^{r-1} \xrightarrow{S_{out}^i} \mathbb{K}'$ where $\boldsymbol{e} \notin \mathbb{K}'$. Hence, the unit vector $0\ldots0 \mid \boldsymbol{e} \mid 0\ldots0 \in \mathbb{F}_2^n$ 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, we can deduce from $\mathcal{T}_{out}^S$ which block L_{out}^i to use such that this results in an integral distinguisher.

3.2.4 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 Fig. 2. We first write $L_{out} \circ E \circ L_{in}$ as

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

and get the following propagation chain

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

where $w(\boldsymbol{k^0}) = n - 1$. According to the two previous sections, 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 an MILP solver to compute all the sets
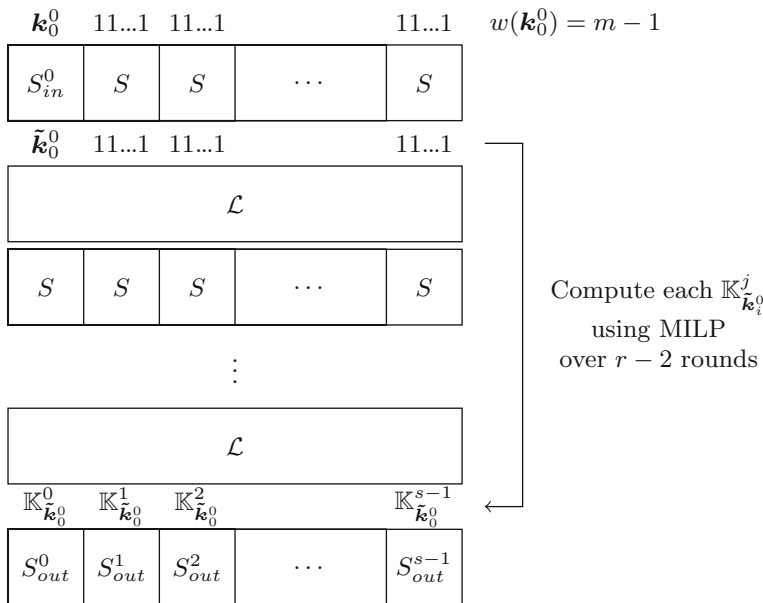
Fig. 2 Overall framework of our search algorithm, where we search for L_{in}^0

$\mathbb{K}_j^{r-1}$, $j \in \{0, \ldots, s-1\}$ through $f \circ \cdots \circ f \circ \mathcal{L}$, where there are $r-2$ applications of f. We denote by $\mathbb{K}_{\tilde{k}_i^0}^j$ these sets to tie them with $\tilde{k}_i^0$.

Next for each set $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_{in}^S$, we compute the following union for each $j \in \{0, \ldots, s-1\}$:

$$\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j = \bigcup_{\tilde{k}_i^0 \in \widetilde{\mathbb{K}}_0} \mathbb{K}_{\tilde{k}_i^0}^j.$$

Now if at least one $\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j$ belongs to $\mathcal{K}_{out}^S$, then we can get a distinguisher. Indeed, $\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j$ is the set of division property vectors that can lead to a unit vectors after the application of S_{out}^j. Thus by definition of $\mathcal{K}_{out}^S$, if $\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j \in \mathcal{K}_{out}^S$ we know that at least one unit vector will not appear after the application of S_{out}^j, i.e. $\mathbb{K}^r$ does not contains all unit vectors. We then put any map from $\mathcal{T}_{out}^S[\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j]$ after the j-th S-box in the last layer, and any map from $\mathcal{T}_{in}^S[\widetilde{\mathbb{K}}^0]$ before the i-th S-box in the first layer, which thus gives us our new distinguisher. Note that we can easily see that

$$\mathcal{T}_{out}^S[\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j] = \bigcap_{\tilde{k}_i^0 \in \widetilde{\mathbb{K}}_0} \mathcal{T}_{out}^S[\mathbb{K}_{\tilde{k}_i^0}^j].$$

Indeed, a linear mapping lead to at least one balanced bit from $\widetilde{\mathbb{K}}^0$ if and only if it lead to at least one balanced bit from each $\tilde{k}_i^0 \in \widetilde{\mathbb{K}}_0$. This will be used later on to even further reduce the work needed with an early-abort strategy. The whole procedure is summarized in Algorithm 1.

Complexity Overall, the number of calls to the MILP solver can be upper bounded as follow. First, we need to compute all $\mathbb{K}_{\tilde{k}_i^0}^j$ for each of the $s(2^m - 2)$ possible $\tilde{k}^0$. Then, each set $\mathbb{K}_{\tilde{k}_i^0}^j$

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

1: Compute $\mathcal{K}_{in}^S, \mathcal{T}_{in}^S, \mathcal{K}_{out}^S$ and $\mathcal{T}_{out}^S$
2: **for** $i = 0 \ldots s - 1$ **do**
3:    **for** each of the $2^m - 2$ vectors $\tilde{k}_i^0$ **do**
4:       Generate a MILP model for $f \circ \cdots \circ f \circ \mathcal{L}$ $r - 2$ *application of f*
5:       Set the initial division property to $1 \ldots \tilde{k}_i^0 \ldots 1$ $\tilde{k}_i^0$ *on the i-th block*
6:       Compute each set $\mathbb{K}_{\tilde{k}_i^0}^j, j \in \{0, \ldots, s - 1\}$ using the MILP model
7:    **end for**

8:    **for** $\widetilde{\mathbb{K}}^0 \in \mathcal{K}_{in}^S$ **do**
9:       $L_{out}^0, \ldots, L_{out}^{s-1} \leftarrow I_m$ I_m : *identity matrix of size m*
10:      $found\_distinguisher \leftarrow false$ *boolean to check if we found a distinguisher*
11:      **for** $j = 0 \ldots s - 1$ **do**
12:         Compute $\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j$
13:         **if** $\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j \in \mathcal{K}_{out}^S$ **then**
14:            $L_{out}^j \leftarrow$ any element from $\mathcal{T}_{out}^S[\mathbb{K}_{\widetilde{\mathbb{K}}^0}^j]$
15:            $found\_distinguisher \leftarrow true$
16:         **end if**
17:      **end for**
18:      **if** $found\_distinguisher$ **then**
19:         $L_{in}^i \leftarrow$ any element in $\mathcal{T}_{in}^S[\widetilde{\mathbb{K}}^0]$
20:         **return** $Diag(I_m, \ldots, L_{in}^i, \ldots, I_m), Diag(L_{out}^0, \ldots, L_{out}^{s-1})$
21:      **end if**
22:   **end for**
23: **end for**

---

can contain at most 2^m vectors, and getting one vector of any of these sets cost one call to the MILP solver. Since there are s of those sets, we need $s2^m$ calls to the MILP solver. Note however that in practice, this is much lower, as we do not need to recover the redundant vectors. This means that for example, the sets $\{0001, 0011\}$ and $\{0001\}$ are considered to be equivalent, as 0011 is redundant in the first set and thus can be removed. If we go through all sets with $m = 4$, the maximum size of any set $\mathbb{K}_{\tilde{k}_i^0}^j$ 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 for a model over $r - 2$ rounds, and the factor 2^m is actually much lower in practice.

This can be compared to the complexity of a naive algorithm. In such an algorithm, one would need to try every possible invertible matrix for each S-box at the first round, so about $s2^{m^2}$ cases (a bit less as there are less than 2^{m^2} invertible matrices). For each of those case, we need to try again every possible matrix for each S-box at the last round, so this add another factor $s2^{m^2}$. This generate $s^2 2^{2m^2}$ models, and then for each of those, we need to check if there is a distinguisher. At most, it costs $n = sm$ calls to the MILP solver, as one call can retrieve one vector of weight 1, and there are n of them. So in total, a naive algorithm would need about $ms^3 2^{2m^2}$ calls to the MILP solver, and each model is over r rounds which is much more expensive to solve.

Moreover for our technique, if we go through each of the $2^m - 2$ vectors $\tilde{k}_i^0$ in a clever 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 $\mathbb{K}^j_{\tilde{k}^0_i}$, we are left with two cases :

- All sets $\mathcal{T}^S_{out}[\mathbb{K}^j_{\tilde{k}^0_i}]$ for all vectors $\tilde{k}^0_i$ of weight $m-1$ are empty, and thus we do not need to go further. Indeed, this means that no linear mapping lead to at least one balanced bit from any initial vector of weight $m-1$. Moreover, for any vector k such that $w(k) < m-1$, we know that there is a vector $\tilde{k}^0_i$ of weight $m-1$ such that $\tilde{k}^0_i \succeq k$. Hence, since there is no balanced bit from all vectors $\tilde{k}^0_i$ of weight $m-1$, then we cannot have any balanced bit from any vector of weight strictly lower than $m-1$ (see [12, 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, 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.

We can even go further by looking at all the possible transitions $k \xrightarrow{\mathcal{S}^i_{in}} \mathbb{K}$ with $w(k) = m-1$ when we go through all linear mappings in $\mathbb{L}_{in}$. Suppose that the two following transitions are possible (and possibly with different linear mappings), $k \xrightarrow{\mathcal{S}^i_{in}} \mathbb{K}$ and $k' \xrightarrow{\mathcal{S}'^i_{in}} \mathbb{K}'$, with $w(k) = w(k') = m-1$. If for all vectors $\widetilde{k'} \in \mathbb{K}'$, there exists a vector $\widetilde{k} \in \mathbb{K}$ such that $\widetilde{k'} \preceq \widetilde{k}$, it is not useful to consider $\mathcal{S}'^i_{in}$. Indeed, in that case, if $\mathcal{S}'^i_{in}$ would lead to a distinguisher, then so would $\mathcal{S}^i_{in}$. Such a transition $k' \xrightarrow{\mathcal{S}'^i_{in}} \mathbb{K}'$ is thus redundant and does not need be examined. We can thus build all possible transitions $k \xrightarrow{\mathcal{S}^i_{in}} \mathbb{K}$ which are not redundant. If there is a vector $\widetilde{k}$ which never belongs to $\mathbb{K}$ among all such non-redundant transitions, we never have to examine the propagation of this vector. This essentially reduces even further the space of all the vectors $\tilde{k}^0_i$ we need to consider. In practice, this allows to significantly reduce the time required to find a distinguisher, or even prove that no such distinguisher exists, and this will be detailed in the next section.

4 Applications

4.1 Division property against 10-round RECTANGLE

RECTANGLE [19] 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 rounds in 25 in both cases. The best known division property based integral distinguisher is from [18] 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

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

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 and ?

denotes a bit in an unknown state.

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

Overall, the time needed to compute all $\mathbb{K}^{j}_{\tilde{k}^{0}_{i}}$ for a given $\tilde{k}^{0}_{i}$ is about 400 seconds in average. The reason this distinguisher exists is that when considering $S' = S \circ L^{0}_{in}$ 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 depend 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 Sect. 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 study how the choice of the S-box affects the resistance of RECTANGLE against division property. We first give some generic insights about the design of an S-box to resist classical division property (i.e. without using our extension technique). Before going further, let us recall how the division property propagation table is built.

Proposition 3 ([18]) *Let S be an n-bit S-box with $y = (y_0, \dots, y_{n-1})$ the ANF of S, where each y_i is a polynomial in the input variables $x = (x_0, \dots, x_{n-1})$ of S. For some $k \in \mathbb{F}^n_2$, let $\mathbb{U}_k = \{\bar{k} \in \mathbb{F}^n_2 | k \preceq \bar{k}\}$ and $F_k = \{x^{\bar{k}} | \bar{k} \in \mathbb{U}_k\}$. Then we have the transition $k \xrightarrow{S} k'$ if and only if $y^{k'}$ contains a monomial in F_k.*

Intuitively, an S-box such that all vectors in the propagation table are of weight 1 should provide a good resistance against division property. This leads us to define a *perfect* S-box, where the choice of the word *perfect* will be justified in Theorem 1.

Definition 4 Let S be an n-bit S-box. We say that S is perfect (w.r.t division property) if its division property propagation table is of the following form :

- $0 \dots 0 \xrightarrow{S} \mathbb{K} = \{0 \dots 0\}$,
- $1 \dots 1 \xrightarrow{S} \mathbb{K} = \{1 \dots 1\}$,
- For any other $k \in \mathbb{F}^n_2$, $k \xrightarrow{S} \mathbb{E}_n$.

Note that, from Proposition 3 this also means that if S is a perfect S-box, for any $k \in \mathbb{F}^n_2 \setminus \{0 \dots 0, 1 \dots 1\}$, the transition $k \xrightarrow{S} k'$ is always valid for any $k' \in \mathbb{F}^n_2 \setminus \{0 \dots 0\}$. However, most vectors will be redundant, i.e. vectors $k', k'' \in \mathbb{F}^n_2$ such that $k \xrightarrow{S} k'$, $k \xrightarrow{S} k''$ and $k' \preceq k''$. Since we do not need to consider redundant vectors in the division property propagation table, we still write $k \xrightarrow{S} \mathbb{E}_n$.

One can wonder whether such S-box exists, and consequently, we give a clear characterization for perfect S-boxes.

Proposition 4 *Let S be an n-bit S-box with $y = (y_0, \ldots, y_{n-1})$ the ANF of S, where each y_i is a polynomial in the input variables $x = (x_0, \ldots, x_{n-1})$ of S. S is perfect if and only if each y_i contains* all *monomials of degree $n - 1$. An example of such an S-box over 4 bits is the following :*

$$S = [1, 4, 3, 5, 13, 7, 12, 10, 8, 0, 11, 15, 6, 14, 9, 2]$$

Proof Let S be an n-bit S-box satisfying the characterization above. Since S is invertible, we know that we already have

$$0 \ldots 0 \xrightarrow{S} \mathbb{K} = \{0 \ldots 0\} \text{ and } 1 \ldots 1 \xrightarrow{S} \mathbb{K} = \{1 \ldots 1\}.$$

We first study the case $k \xrightarrow{S} \mathbb{K}$ where $w(k) = n - 1$. In that case, we have $\mathbb{U}_k = \{k, 1 \ldots 1\}$ and thus $F_k = \{x^k, x_0 \ldots x_{n-1}\}$. Then if $k' \in \mathbb{K}$, from Proposition 3 this means that either x^k or $x_0 \ldots x_{n-1}$ appears in the expression of $y^{k'}$. Especially, for S to be perfect, this needs to hold for every $k' \in \mathbb{E}_n$, thus for every $i \in \{0, \ldots, n - 1\}$, x^k or $x_0 \ldots x_{n-1}$ must appear in the expression of y_i.

However, since S must be invertible, it is well known that each component of its ANF must have a degree at most $n - 1$, hence $x_0 \ldots x_{n-1}$ cannot appear in any y_i. To summarize, to have $k \xrightarrow{S} \mathbb{K} = \mathbb{E}_n$ for every k such that $w(k) = n - 1$, then for every such k, x^k must appear in the expression of each and every y_i, $i \in \{0, \ldots, n - 1\}$, which exactly means that each y_i contains all monomials of degree $n - 1$.

Now for every remaining case, i.e. $1 \leq w(k) \leq n - 2$, $\mathbb{U}_k$ always contains at least one $\bar{k}$ such that $w(\bar{k}) = n - 1$, and thus F_k contains at least one monomial of degree $n - 1$. If we want to have $k \xrightarrow{S} \mathbb{E}_n$, this means that every y_i must contain at least one monomial from F_k. However, each y_i contains all monomials of degree $n - 1$, and F_k contains at least one such monomial, and thus $k \xrightarrow{S} \mathbb{E}_n$ holds, which leads to the fact that S is then a perfect S-box. □

This characterization is very similar to the property that an S-box should verify to have a good resistance against division property given in [4]. However, their representation is a bit different, and we show that choosing a perfect S-box for an SPN block cipher is actually the optimal choice when considering classical division property. First, we need the two following lemmas.

Lemma 1 *Let S be an n-bit S-box, and $k, k' \in \mathbb{F}_2^n$ such that $k \preceq k'$. Let $\tilde{k} \in \mathbb{F}_2^n$ such that $k' \xrightarrow{S} \tilde{k}$. Then we have $k \xrightarrow{S} \tilde{k}$.*

Proof From $k \preceq k'$, we know that $\mathbb{U}_{k'} \subseteq \mathbb{U}_k$, and as such, $F_{k'} \subseteq F_k$. Since $k' \xrightarrow{S} \tilde{k}$, we know that $y^{\tilde{k}}$ contains a monomial in $F_{k'}$. However, since $F_{k'} \subseteq F_k$, we also have that $y^{\tilde{k}}$ contains a monomial in F_k, which exactly means $k \xrightarrow{S} \tilde{k}$. □

Lemma 2 *Let $S^\star$ be an S-box layer such that $S^\star = (S^\star, \ldots, S^\star)$ where $S^\star$ is a perfect S-box. Let S be another S-box layer such that $S = (S, \ldots, S)$ where S is any non-perfect S-box. Let $k, \tilde{k}'$ such that $k \xrightarrow{S} \tilde{k}'$. Then we can always find $\tilde{k} \in \mathbb{F}_2^n$ such that $k \xrightarrow{S^\star} \tilde{k}$ and $\tilde{k} \preceq \tilde{k}'$.*

Proof Denote by k_i the i-th block of k which goes through the i-th S-box, *i.e.* $k_i \xrightarrow{S} \tilde{k}'_i$. Note that $k \preceq k'$ is equivalent to $k_i \preceq k'_i$ for all i. We can build each block $\tilde{k}_i$ of $\tilde{k}$ such that $k \xrightarrow{S^\star} \tilde{k}$ and $\tilde{k} \preceq \tilde{k}'$ as follow :

- If $\widetilde{\boldsymbol{k'}}_i = 0 \ldots 0$, then $\widetilde{\boldsymbol{k}}_i = 0 \ldots 0$,
- If $\widehat{\boldsymbol{k'}}_i = 1 \ldots 1$, then $\widetilde{\boldsymbol{k}}_i = 1 \ldots 1$,
- Otherwise, since $S^\star$ is perfect, we can choose $\widetilde{\boldsymbol{k}}_i = \boldsymbol{e}$, where $\boldsymbol{e}$ is a unit vector such that $\boldsymbol{e} \preceq \widetilde{\boldsymbol{k'}}_i$.

By building $\widetilde{\boldsymbol{k}}_i$ as described, it is clear that for all i we have $\boldsymbol{k}_i \xrightarrow{S^\perp} \widetilde{\boldsymbol{k}}_i$ and $\widetilde{\boldsymbol{k}}_i \preceq \widetilde{\boldsymbol{k'}}_i$. As such, $\boldsymbol{k} \xrightarrow{S^\perp} \widetilde{\boldsymbol{k}}$ and $\widetilde{\boldsymbol{k}} \preceq \widetilde{\boldsymbol{k'}}$. □

We are now ready to prove the following theorem, which shows that using a perfect S-box is the optimal choice for classical division property (i.e. without using our extension technique).

Theorem 1 *For a given n-bit block-cipher where only the S-box remained to be determined (i.e. the linear layer $\mathcal{L}$ is fixed), using an S-box layer $S^\star$ built with a perfect S-box $S^\star$ is optimal in terms of resistance against classical division property. We define optimal in the sense that if we denote by $r^\star$ (resp. r) the smallest number of round such that $\mathbb{K}_{r^\star} = \mathbb{E}_n$ (resp. $\mathbb{K}_r = \mathbb{E}_n$) when using a perfect S-box $S^\star$ (resp. any other S-box S), then we always have $r^\star \leq r$. This basically means that using a perfect S-box always gives the best resistance against classical division property.*

Proof Let the following be a division trail when using a non-perfect S-box :

$$\boldsymbol{k}^0 \xrightarrow{S} \widehat{\boldsymbol{k}}^0 \xrightarrow{\mathcal{L}} \boldsymbol{k}^1 \xrightarrow{S} \cdots \xrightarrow{\mathcal{L}} \boldsymbol{k}^r$$

where $\boldsymbol{k}^r \in \mathbb{E}_n$. We can build the following division trail when using a perfect S-box :

$$\boldsymbol{k}^0 \xrightarrow{S^\perp} \widetilde{\boldsymbol{k}}^0 \xrightarrow{\mathcal{L}} \boldsymbol{k}^1 \xrightarrow{S^\perp} \cdots \xrightarrow{\mathcal{L}} \boldsymbol{k}^r.$$

For this division trail to be valid, we use the two previous lemma :

- Using Lemma 2, since $\boldsymbol{k}^i \xrightarrow{S} \widehat{\boldsymbol{k}}^i$ we can build $\widetilde{\boldsymbol{k}}^i$ such that $\boldsymbol{k}^i \xrightarrow{S^\perp} \widetilde{\boldsymbol{k}}^i$ and $\widetilde{\boldsymbol{k}}^i \preceq \widehat{\boldsymbol{k}}^i$.
- Since $\widetilde{\boldsymbol{k}}^i \preceq \widehat{\boldsymbol{k}}^i$ and $\widehat{\boldsymbol{k}}^i \xrightarrow{\mathcal{L}} \boldsymbol{k}^{i+1}$, using Lemma 1 we know that $\widetilde{\boldsymbol{k}}^i \xrightarrow{\mathcal{L}} \boldsymbol{k}^{i+1}$ is a valid transition.

Hence, for any unit vector $\boldsymbol{e}$, there is a valid division trail over r rounds which ends with $\boldsymbol{e}$ when using a perfect S-box, i.e. we have $\mathbb{K}_r = \mathbb{E}_n$. By definition, $r^\star$ is the smallest number of rounds which should verify this condition, thus $r^\star \leq r$. Moreover, by definition, the best distinguisher we can build when using a perfect S-box (resp. any other S-box) is over $r^\star - 1$ rounds (resp. $r - 1$ rounds). Hence why using a perfect S-box gives the best resistance against division property. □

Thus when considering classical division property, choosing the best S-box in regards to security is pretty clear. Note however that such an S-box has a very peculiar behavior with our technique. Indeed, since *every* monomial of degree $n - 1$ appears in *every* coordinate of the ANF, this property cannot still hold when we consider either $L \circ S$ or $S \circ L$ where L is a linear mapping (different from a permutation). As such, if a perfect S-box is used, when considering our technique, the first and last round will be somewhat weaker, as the S-box will not be perfect for these rounds. However, every other round will still use this perfect S-box, thus in a way, the behavior "in the middle" of the cipher would still be good.

We decided to search for a better S-box to use for RECTANGLE, in the hope that it would lead to a better resistance against our technique. Since the rational behind S-box design highly

depends on potential applications of the resulting block cipher, we restrict 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 and 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 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 pairs (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 $2^{28.6} \times 2^{2 \times 4} = 2^{36.6}$ non trivial operations and took approximately 80h on a Xeon E5-2695 (72 cores). Among all those linearly equivalent S-boxes, none of them were perfect. However we found 56 *almost perfect* S-boxes, i.e. with 13 (instead of 14) transitions $k \to \{1000, 0100, 0010, 0001\}$. Note that many pairs 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_{in}^0 as defined below, others L_{in}^i for $i \neq 0$ are the identity, and each block L_{out}^i of L_{out} are the same

$$L_{in}^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, L_{out}^i = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

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. Moreover, even when using a perfect S-box (such as the one given in Proposition 4), the best distinguisher using the classic search algorithm is also over 7 rounds, which shows that our choice of S-box, even though it is not a perfect one, is optimal with respect to the classic division property.

We believe that this could lead to a new criterion 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, etc.), then look at the linear equivalent S-boxes and take the one with *the best* division property propagation table. According to our experiment, while we do not have the same optimality proof as for classical division property, using an S-box with a division

property propagation table as close as possible to the one of a perfect S-box seems to be the best choice.

About golden S-boxes In 2007, Leander and Poschmann [9] analyzed all 4-bit S-boxes up to linear equivalence and exhibited a set of 16 equivalence classes leading to optimal S-boxes (called *golden S-boxes*) with respect to both differential and linear cryptanalysis. We went through all members of each of these equivalences classes to see if any of them is a perfect S-box. Indeed, recall that division property is not invariant through linear equivalence. As a result, it turns out that there is no S-box among all of these that is a perfect S-box, thus we cannot have an S-box which is optimal for both linear, differential and division property based cryptanalysis . However, four of these classes have some almost perfect S-boxes among them, i.e. with 55 transitions $\boldsymbol{k} \to \boldsymbol{k}'$ with $w(\boldsymbol{k}') = 1$ (instead of the maximum of $4 \times 14 = 56$), namely classes G_0, G_1, G_2 and G_8. We give one example for each of these classes in "Appendix", as well as an example with the maximum number of such transitions for each other class. We also give the number of S-boxes reaching that maximum number of transitions across each class. We want to emphasize that this clearly show that using an S-box satisfying our criterion will not lead to the best resistance against different kinds of attacks, and more research is needed in that area. However as mentioned at the end of the previous paragraph, it turns out that using S-boxes which are very close to our criterion can still reach the same security against division property as an S-box satisfying our criterion.

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 [18] 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 $\mathcal{T}_{out}^S[\mathbb{K}_{\tilde{\boldsymbol{k}}_i^0}^j]$ are empty, meaning that if there is at least one vector of weight 2 or lower in $\widetilde{\mathbb{K}}^0$, then this cannot result 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 $\mathbb{L}_{in}$, 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 search 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, 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 with data complexity 2^{63} and with all 64 bits being balanced 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 $i \neq 0$. The classical search algorithm was only able to find a distinguisher on up to 6 rounds, and again, this choice is optimal for classic division property, as a perfect S-box also gives a classic division property based distinguisher over 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

We studied further the division property and the distinguishers that are built from it. We show 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 show 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 provide an algorithm to find such distinguisher, and successfully apply 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 than the previously known distinguishers. The design of our algorithm also allows us to prove that our technique cannot extend the best distinguisher on PRESENT over one more round. Finally, we give a criterion on S-boxes which allows to prove that if an S-box verifies this criterion, it will provide the best resistance against division property. To our knowledge, this is the first time that such an optimality result is given and formally proven for division property. 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 criterion for S-boxes and further research about this criterion and its interaction with criteria for the design of S-boxes 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 focuses on finding any distinguisher over an SPN block cipher. Thus, how could we find an optimal distinguisher (in terms of data) using this technique, as 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. 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.

Appendix: Almost perfect S-boxes among golden S-boxes

The following table gives an example of an S-box with the maximum number of transitions $k \to k'$ with $w(k') = 1$ among its class and the number of S-boxes reaching that maximum number of transitions, for each golden S-box class. Note that according to Proposition 2, this number of transition is invariant by permutation equivalence, i.e. for a given S-box S with n_1 such transitions, then for any permutations P, P', $S' = P \circ S \circ P'$ also has n_1 such transitions. We can thus reduce the number of member of each class to examine by picking one member S and examining all S-boxes built as $L_2 \circ S \circ L_1$ with $L_1 \in \mathbb{L}_{in}$ and $L_2 \in \mathbb{L}_{out}$, where $\mathbb{L}_{in}$ and $\mathbb{L}_{out}$ are the spaces defined in Sect. 3.2.1. The number of S-boxes given here is thus computed when considering that equivalence relation, but this shows that there are actually a decent amount of choice of S-boxes if one wants to consider other criterion than differential, linear and division property cryptanalysis for choosing an S-box. It is worth noting however that two S-boxes that are permutation equivalent do not necessarily lead to the same result for a given block cipher.

Class	S-box	Number of transitions	Number of S-boxes
G_0	[0, 1, 11, 13, 2, 9, 15, 12, 7, 4, 5, 14, 8, 6, 3, 10]	**55**	1536
G_1	[0, 9, 3, 11, 13, 15, 10, 4, 1, 12, 6, 5, 14, 7, 2, 8]	**55**	1536
G_2	[0, 12, 3, 11, 13, 10, 15, 1, 4, 7, 6, 14, 5, 2, 9, 8]	**55**	1536
G_3	[0, 3, 7, 12, 13, 9, 11, 10, 2, 15, 14, 4, 5, 8, 6, 1]	53	3360
G_4	[0, 3, 7, 5, 11, 15, 2, 12, 13, 6, 14, 9, 10, 1, 8, 4]	53	3360
G_5	[0, 1, 14, 13, 7, 8, 3, 9, 11, 12, 4, 10, 5, 2, 15, 6]	53	3360
G_6	[0, 1, 7, 4, 14, 8, 3, 9, 11, 5, 13, 10, 12, 2, 15, 6]	53	3360
G_7	[0, 3, 1, 10, 14, 12, 11, 15, 7, 9, 8, 5, 6, 4, 2, 13]	53	3360
G_8	[0, 10, 1, 2, 15, 13, 12, 8, 4, 14, 9, 5, 6, 11, 7, 3]	**55**	1536
G_9	[0, 13, 2, 12, 14, 1, 10, 8, 7, 6, 4, 9, 15, 11, 5, 3]	54	1344
G_{10}	[0, 8, 3, 1, 15, 14, 13, 9, 4, 12, 10, 6, 5, 11, 7, 2]	54	1344
G_{11}	[0, 1, 6, 4, 11, 12, 2, 13, 14, 3, 9, 8, 10, 15, 5, 7]	53	3360
G_{12}	[0, 1, 2, 12, 7, 4, 14, 5, 10, 15, 9, 8, 11, 3, 13, 6]	53	3360
G_{13}	[0, 7, 11, 14, 13, 1, 5, 6, 4, 2, 9, 3, 10, 8, 15, 12]	53	3360
G_{14}	[0, 9, 3, 15, 5, 10, 8, 13, 4, 7, 14, 6, 11, 1, 2, 12]	54	1344
G_{15}	[0, 1, 6, 8, 11, 12, 14, 13, 2, 15, 9, 5, 3, 7, 4, 10]	54	1344

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, 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, 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. In: IACR Cryptology ePrint Archive (Accepted at SAC2018), vol. 2018, p. 688 (2018).
8. Gurobi Optimization L.: Gurobi optimizer reference manual. http://www.gurobi.com (2018).
9. Leander G., Poschmann A.: On the classification of 4 bit s-boxes. In: Arithmetic of Finite Fields, First International Workshop, WAIFI 2007, Madrid, Spain, June 21–22, 2007, Proceedings, pp. 159–176 (2007). https://doi.org/10.1007/978-3-540-73074-3_13.
10. Sun L., Wang W., Liu R., Wang M.: MILP-aided bit-based division property for ARX-based Block Cipher. IACR Cryptol. ePrint Arch. **2016**, 1101 (2016).
11. Sun L., Wang W., Wang M.: MILP-aided bit-based division property for primitives with non-bit-permutation linear layers. IACR Cryptol. ePrint Arch. **2016**, 811 (2016).
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 8.0). http://www.sagemath.org (2017).
14. Todo Y.: Integral cryptanalysis on full MISTY1. In: Advances in Cryptology–CRYPTO 2015—35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16–20, 2015, Proceedings, Part I, pp. 413–432 (2015). https://doi.org/10.1007/978-3-662-47989-6_20.
15. 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).
16. 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).
17. 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.
18. 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).
19. Zhang W., Bao Z., Lin D., Rijmen V., Yang B., Verbauwhede I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci. China Inf. Sci. **58**(12), 1–15 (2015).
20. Zhang W., Rijmen V.: Division cryptanalysis of block Ciphers with a binary diffusion layer. IACR Cryptol. ePrint Arch. **2017**, 188 (2017).