IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2021, No. 1, pp. 411–442. DOI:10.46586/tosc.v2021.i1.411-442

On the Relationships between Different Methods for Degree Evaluation

Siwei Chen, Zejun Xiang ✉, Xiangyong Zeng and Shasha Zhang

Faculty of Mathematics and Statistics, Hubei Key Laboratory of Applied Mathematics,
Hubei University, Wuhan, China,
chensiwei@stu.hubu.edu.cn,{xiangzejun,xzeng}@hubu.edu.cn,amushasha@163.com

Abstract. In this paper, we compare several non-tight degree evaluation methods i.e., Boura and Canteaut's formula, Carlet's formula as well as Liu's *numeric mapping* and division property proposed by Todo, and hope to find the best one from these methods for practical applications. Specifically, for the substitution-permutation-network (SPN) ciphers, we first deeply explore the relationships between division property of an Sbox and its algebraic properties (e.g., the algebraic degree of its inverse). Based on these findings, we can prove theoretically that division property is never worse than Boura and Canteaut's and Carlet's formulas, and we also experimentally verified that the division property can indeed give a better bound than the latter two methods. In addition, for the nonlinear feedback shift registers (NFSR) based ciphers, according to the propagation of division property and the core idea of numeric mapping, we give a strict proof that the estimated degree using division property is never greater than that of numeric mapping. Moreover, our experimental results on Trivium and Kreyvium indicate the division property actually derives a much better bound than the numeric mapping. To the best of our knowledge, this is the first time to give a formal discussion on the relationships between division property and other degree evaluation methods, and we present the first theoretical proof and give the experimental verification to illustrate that division property is the optimal one among these methods in terms of the accuracy of the upper bounds on algebraic degree.

Keywords: Degree Evaluation · Division Property · Numeric Mapping · SPN · NFSR

1 Introduction

The outputs (e.g., keystreams, ciphertexts or message digests) of symmetric ciphers can be regarded as Boolean functions over public variables (e.g., plaintext bits or IV bits) and/or secret variables (e.g., key bits). Distinguishing attacks or key-recovery attacks can be achieved if the targeting cipher exhibits low algebraic degrees, such as integral attacks [KW02], higher-order differential attacks [Knu94, Lai94], cube attacks [DS09] and some algebraic attacks [Cou03, CM03]. Thus, the algebraic degree of a cipher is one of the necessary criterions for security analysis, and it is always of great significance to get a tighter bound.

For the degree evaluation of block ciphers, the first improvement of the trivial bound $(\deg(G \circ F) \leq \deg(G) \cdot \deg(F))$ of the composition $G \circ F$ was proposed by Canteaut and Videau [CV02]. Later, Boura *et al.* [BCC11] gave a new bound on the degree of iterated SPN block ciphers with nonlinear layer composed of parallel bijective Sboxes. As an application, they found a zero-sum partition of size 2^{1590} for the full Keccak-f [BDP09] permutation. Afterwards, Duan *et al.* [DL11] improved the bound for the inverse Keccak-f permutation focusing on the inverse Sbox based on [BCC11], which

Received: 2020-09-01 Revised: 2020-11-23 Accepted: 2021-01-23 Published: 2021-03-19

lowered the size of the full-round zero-sum partition of Keccak-f permutation from 2^{1590} to 2^{1579}. Almost at the same time, Boura and Canteaut [BC13] also studied the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$ and proposed a tighter bound than [BCC11]. Recently, Carlet [Car20] obtained a set of formulas to estimate the upper bound on the degree of composite functions by studying the graph indicators of vectorial Boolean functions, one of which is most applicable for the degree evaluation of SPN ciphers. At CRYPTO 2017, Liu [Liu17] presented a general framework to iteratively estimate the algebraic degree for NFSR-based ciphers, by exploiting the technique called *numeric mapping*. It was the first formalized and systematic method for finding upper bounds on the algebraic degree of NFSR-based stream ciphers. Based on the framework, a concrete and efficient algorithm to find an upper bound on the algebraic degree for Trivium-like ciphers was proposed. This algorithm has linear time complexity and needs a negligible amount of memory. Due to the high efficiency of the algorithm, a large set of cubes with large size can be exhausted. As an illustration, [Liu17] obtained the best distinguishing attacks on all Trivium-like ciphers at that time.

Besides aforementioned methods, division property is also an effective technique to estimate the upper bound. Division property [Tod15b], proposed by Todo at EUROCRYPT 2015, is a generalized integral property which is aimed to construct longer integral distinguishers of block ciphers. Since its proposal, there were a lot researches focusing on this topic [Tod15a, BC16, XZBL16, SWW17, TIHM17, SHZ⁺17]. Later, Todo and Morri [TM16] introduced the bit-based division property to achieve a more accurate structural evaluation. Therefore, the experimental 15-round integral distinguisher of Simon-32 [WLV⁺14] can be verified using three-subset bit-based division property (3SBDP). Then the Mixed Integer Linear Programming (MILP), which has been widely used in cryptanalysis [MWGP11, SHS⁺13, SHW⁺14, CJF⁺16], was first adopted by Xiang *et al.* [XZBL16] to automatically search integral distinguishers based on bit-based division property. The MILP-aided technique was extended to cube attacks by Todo *et al.* [TIHM17], which is the first application of division property to steam ciphers. Then Wang *et al.* [WHT⁺18] introduced the flag technique and term enumeration to describe the propagation of division property more accurately and decrease the time complexity of recovering superpolys. At ASIACRYPT 2019, Wang *et al.* [WHG⁺19a] proposed the MILP method based on 3SBDP for searching integral distinguishers. As applications to several lightweight block ciphers, more balanced bits or longer integral distinguishers can be found compared with [XZBL16]. Recently, applying 3SBDP to cube attacks [WHG⁺19b, HLM⁺20] were presented, where the exact superpolys can be recovered. Based on [TM16, XZBL16, TIHM17], the MILP-aided division property could be used to effectively estimate the upper bound on the degree of not only block ciphers [BKL⁺17] but also stream ciphers [WHT⁺18]. Moreover, it breaks the structural limitation of ciphers and only needs to construct a corresponding MILP model which can be solved by off-and-shelf solvers like Gurobi[1]. In the following content, the (bit-based) division property generally represents the two-subset (bit-based) division property if there is no special statement.

In the applications of Boura and Canteaut's [BC13] and Carlet's [Car20] methods to SPN ciphers, the only parameter involved in their formulas is the algebraic degree of Sboxes used in the ciphers. Thus the advantages of the two methods are high efficiency and less manual work. However, both of them ignored the influence of linear layers between two rounds, and this may result in a less accurate upper bound, especially for weak linear layers. As for stream ciphers, Liu's numeric mapping [Liu17] is an efficient technique, especially for searching cubes with large size. It is required to analyze not only the ANF of the update function used in a particular cipher but also the algebraic expression of the state computed backward for several rounds. Moreover, the numeric mapping method frequently utilizes the trivial bound on the degree of the product of two functions i.e.,

[1] https://www.gurobi.com/

$\deg(g \cdot f) \leq \deg(g) + \deg(f)$. Therefore, the precision will be inevitably lost in a long iteration. In [TIHM17], some experimental results indicate that using division property can derive longer zero-sum distinguishers than [Liu17] for Trivium and Kreyvium. However, the reason why division property is superior to numeric mapping is still unclear until now.

1.1 Our Contributions

Which is the best one among these methods in terms of the accuracy of degree evaluations? In order to seek this answer, in this paper, we attempt to establish the relationships between division property and other methods.

We first present an iterative method based on division property to get upper bounds on the degree of SPN ciphers in this paper. To be specific, for the composite function $G \circ F$, if we obtain an integer d as the minimal weight of the input division property of function F such that the weight of the corresponding output division property is always greater than $\deg(G)$, then we can regard $d - 1$ as the upper bound on $\deg(G \circ F)$. The framework corresponding to this method is described in Algorithm 1. In addition, we introduce the concept of *word-based division trail* and according to the propagation of word-based division property of a public Sbox, which is discussed in [Tod15a], we deeply explore some relationships between the word-based division trail of an Sbox and its algebraic properties as indicated in Lemma 1 and 2. Moreover, these observations are closely related to Boura and Canteaut's and Carlet's formulas. Based on these observations, we theoretically prove from the point of view of word-based division property that our new method is never worse than Boura and Canteaut's and Carlet's formulas. Furthermore, we used small-PRESENT as a toy example to make this conclusion seem more intuitive. Note that bit-based division property is more accurate than word-based division property, thus our new method when implemented by bit-based division property can further improve the upper bounds. Based on this fact, *we conclude that bit-based division property will never be worse than both of the two formulas for degree evaluations of SPN ciphers.* As an illustration, we apply bit-based division property, Boura and Canteaut's and Carlet's formulas, to estimate degrees of Keccak and KNOT ciphers. The best bounds are obtained actually by division property in these experiments and these results provide strong evidences for our final conclusion.

For the degree evaluation of NFSR-based ciphers, Liu's numeric mapping [Liu17] can derive an upper bound quite efficiently. As an application, an algorithm was proposed in [Liu17] to estimate the algebraic degree of Trivium-like ciphers. For a more intuitive comparison with bit-based division property, we formalized Liu's algorithm in this paper. In order to establish a relationship between numeric mapping and division property, we present some observations on the degree evaluation by division property. Based on these observations and the core idea of numeric mapping, we provide comparisons of the two methods on not only generalized stream ciphers but also the particular Trivium-like ciphers, and strictly prove *the division property is never worse than the numeric mapping for degree evaluations of NFSR-based ciphers.* In addition, we introduce a divide-and-conquer approach and a new notion called *maximal polynomial* to improve the efficiency of the MILP-aided degree evaluation based on division property, which is described in Algorithm 2. As an application, we apply Algorithm 2 to Trivium and Kreyvium. The comparison of our experimental results with [Liu17] shows that the gap between the estimated degree derived by division property and numeric mapping becomes more and more significant with the round increasing.

Very recently, two works about computing the *exact* algebraic degree by Hu *et al.* [HSWW20] and Hebborn *et al.* [HLLT20] appear, both of which have been accepted to ASIACRYPT 2020. The methods used in [HLLT20] and [HSWW20] are 3SBDP and *monomial prediction*, respectively. Generally, when computing the exact degree, 3SBDP (monomial prediction) requires to enumerate all the division trails (*monomial trails*). But their applications in practice may be limited, e.g., for block ciphers with large block size or stream ciphers with complex update functions. In these cases, evaluating the exact algebraic degree will be quite difficult and it would be better to use some simple but non-tight methods to evaluate the upper bound. Our paper exactly devotes attention to the question that which non-tight method is the most accurate.

1.2 Organization

The rest of this paper is organized as follows: In Sect.2, we revisit division property and some previous work on degree evaluations. In Sect.3, we give the comparison between division property and Boura and Canteaut's formula as well as Carlet's formula for degree evaluations of SPN ciphers. The comparison of degree evaluations of NFSR-based ciphers between division property and numeric mapping is given in Sect.4. We conclude our paper and discuss in Sect.5.

2 Preliminaries

We first introduce some notations used throughout this paper. Let $\mathbb{F}_2$ denote the finite field with two elements (0 and 1) and $\boldsymbol{a} \in \mathbb{F}_2^n$ be an n-bit vector where a_i denotes the ith bit of $\boldsymbol{a}$. A unit vector where the ith element is 1 and the others are 0 is denoted by $\boldsymbol{e}_i$. Especially, a vector whose all elements are 0 (or 1) is denoted by $\mathbf{0}$ (or $\mathbf{1}$). The Hamming weight of $\boldsymbol{a} \in \mathbb{F}_2^n$ is denoted by $wt(\boldsymbol{a}) = \#\{i : a_i = 1, 1 \leq i \leq n\}$. Denote $\vec{\boldsymbol{a}}$ an element in $(\mathbb{F}_2^n)^m$ where the ith element of $\vec{\boldsymbol{a}}$, denoted by $\boldsymbol{a}_i$, belongs to $\mathbb{F}_2^n$. Let $\boldsymbol{k}$ and $\boldsymbol{k}^*$ be two vectors in $\mathbb{F}_2^n$, define $\boldsymbol{k} \succeq \boldsymbol{k}^*$ if $k_i \geq k_i^*$ holds for all $i \in \{1, 2, ..., n\}$, otherwise we write $\boldsymbol{k} \not\succeq \boldsymbol{k}^*$.

Bit Product Function $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ and $\pi_{\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}})$ [Tod15b]. For any $\boldsymbol{u} \in \mathbb{F}_2^n$, let $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. For any $\boldsymbol{x} \in \mathbb{F}_2^n$, $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ is defined as

$$\pi_{\boldsymbol{u}}(\boldsymbol{x}) := \prod_{i=1}^{n} x_i^{u_i}.$$

For any $\vec{\boldsymbol{u}} \in (\mathbb{F}_2^n)^m$, let $\pi_{\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}})$ be a function from $(\mathbb{F}_2^n)^m$ to $\mathbb{F}_2$. For any $\vec{\boldsymbol{x}} \in (\mathbb{F}_2^n)^m$, $\pi_{\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}})$ is defined as

$$\pi_{\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}}) := \prod_{i=1}^{m} \pi_{\boldsymbol{u}_i}(\boldsymbol{x}_i).$$

Algebraic Normal Form. For any Boolean function f on n variables, it can be uniquely represented by its Algebraic Normal Form (ANF) as

$$f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^f \left(\prod_{i=1}^{n} x_i^{u_i} \right) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^f \pi_{\boldsymbol{u}}(\boldsymbol{x}),$$

where $a_{\boldsymbol{u}}^f \in \mathbb{F}_2$ is a constant depending on f and $\boldsymbol{u}$. The algebraic degree of f, denoted by $\deg(f)$, is defined as $\max\{wt(\boldsymbol{u}) | \boldsymbol{u} \in \mathbb{F}_2^n, a_{\boldsymbol{u}}^f = 1\}$. Let $\boldsymbol{u}$ and $\boldsymbol{u}^*$ be two vectors in $\mathbb{F}_2^n$, define $\pi_{\boldsymbol{u}}(\boldsymbol{x}) \succ \pi_{\boldsymbol{u}^*}(\boldsymbol{x})$ if $\boldsymbol{u} \succeq \boldsymbol{u}^* \neq \mathbf{0}$ holds and there exists $i \in \{1, ..., n\}$ such that $u_i > u_i^*$, otherwise we write $\pi_{\boldsymbol{u}}(\boldsymbol{x}) \not\succ \pi_{\boldsymbol{u}^*}(\boldsymbol{x})$.

Algebraic Degree of Vectorial Boolean Function. For a vectorial Boolean function F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. Denote the algebraic degree of F by $\deg(F)$, which is defined as:

$$\deg(F) = \max_{i \in \{1,2,...,m\}} \deg(F_i),$$

where F_i denotes the ith coordinate of F.

2.1 Division Property

Division property [Tod15b] is a generalized integral property, which was proposed by Todo at EUROCRYPT 2015. It has shown its great power [Tod15a] since its proposal as it is a more accurate description of integral property. At FSE 2016, Todo and Morii [TM16] extended this technique to two-subset and three-subset bit-based division property, both of which are much more accurate since bit is the smallest unit in cryptography. We now briefly revisit division property and related theories as follows.

Definition 1 (Division Property [Tod15b])**.** Let $\mathbb{X}$ be a multiset whose elements take values from $\mathbb{F}_2^n$ and k takes a value between 0 and n. When the multiset $\mathbb{X}$ has division property $\mathcal{D}_k^n$, it fulls the following conditions:

$$\bigoplus_{\boldsymbol{x}\in\mathbb{X}} \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \begin{cases} \text{unknown} & \text{if } wt(\boldsymbol{u}) \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 2 (Vectorial Division Property [Tod15b])**.** Let $\mathbb{X}$ be a multiset whose elements take values from $(\mathbb{F}_2^n)^m$, and $\boldsymbol{k}$ is an m-dimensional vector where k_i denotes the ith element of $\boldsymbol{k}$ and $0 \leq k_i \leq n$ for all $i \in \{1, 2, ..., m\}$. When the multiset $\mathbb{X}$ has division property $\mathcal{D}_{\boldsymbol{k}}^{n,m}$, it fulls the following conditions:

$$\bigoplus_{\vec{\boldsymbol{x}}\in\mathbb{X}} \pi_{\vec{\boldsymbol{u}}}(\vec{\boldsymbol{x}}) = \begin{cases} \text{unknown} & \text{if } wt(\boldsymbol{u}_i) \geq k_i \text{ for any } i \in \{1, ..., m\}, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3 (Bit-based Division Property [TM16])**.** Let $\mathbb{X}$ be a multiset whose elements take values from $\mathbb{F}_2^n$ and $\mathbb{K}$ denote a set of n-dimensional bit vectors whose elements take the value 0 or 1. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{K}}^{1,n}$, it fulls the following conditions:

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

Proposition 1 (Propagation Characteristic of Sbox [Tod15b])**.** *Let S be a function (Sbox) from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ with degree d. Assuming that an input multiset $\mathbb{X}$ has division property $\mathcal{D}_k^n$, then the output multiset $\mathbb{Y}$ has division property $\mathcal{D}_{\lceil \frac{k}{d} \rceil}^n$. In addition, if the Sbox is a permutation, the output multiset $\mathbb{Y}$ has division property $\mathcal{D}_n^n$ when the input multiset has division property $\mathcal{D}_n^n$.*

Proposition 1 is applicable for the case where the only available information of an Sbox is its algebraic degree. When the Sbox is a public function, which means the ANF of the Sbox is known, Todo *et al.* [Tod15a] used the 7-bit Sbox of MISTY [Mat97] as an example to illustrate how to accurately describe the propagation characteristic. Based on [Tod15a] and with the help of the following definition in [BC13], we formalize the propagation rule of division property of a public Sbox as Property 1 for brevity.

Definition 4 ([BC13])**.** Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. For any integer k, $0 \leq k \leq m$, $\delta_k(F)$ denotes the maximal degree of the product of any k (or fewer) coordinates of F:

$$\delta_k(F) = \max_{K\subset\{1,2,...,m\},|K|\leq k} \deg\left(\prod_{i\in K} F_i\right).$$

In particular, $\delta_0(F) = 0$ and $\delta_1(F) = \deg(F)$.

Property 1. Let S be an $n \times n$ public Sbox. Assuming that an input multiset $\mathbb{X}$ has division property $\mathcal{D}_k^n$, then the output multiset $\mathbb{Y}$ has division property $\mathcal{D}_{k'}^n$ where

$$k' = \min_{0\leq i\leq n} \{i|\delta_i(S) \geq k\}.$$

2.2 The MILP Technique Used in Division Property

Mixed Integer Linear Programming. MILP was first introduced by Mouha *et al.* [MWGP11] to evaluate the number of differentially and linearly active Sboxes of AES and Enocoro-128v2. Since then, MILP method has been widely applied to cryptography [SHS$^+$13, SHW$^+$14, CJF$^+$16, XZBL16, SWW17, ST17]. The propagations of cryptographic properties (such as differential characteristics, linear approximations or division property) are converted into a system of linear inequalities in MILP-aided cryptanalysis, and these linear inequalities are sent to an MILP solver with an appropriate objective function. In general, an MILP model $\mathcal{M}$ is composed of variables $\mathcal{M}.var$, constraints $\mathcal{M}.con$ and an objective function $\mathcal{M}.obj$. If $\mathcal{M}$ is *feasible*, then an optimized solution, denoted by $OBJ(\mathcal{M})$, will be returned. In addition, if $\mathcal{M}$ has no objective function, the MILP solver only evaluate the feasibility of $\mathcal{M}$.

Searching Integral Distinguishers Based on MILP-aided Division Property. The application of bit-based division property is limited because of its high time and memory complexity. At ASIACRYPT 2016, Xiang *et al.* [XZBL16] applied MILP to division property to overcome this drawback.

Definition 5 (Division trail [XZBL16])**.** Consider the propagation of division property $\{\mathbf{k}\} \overset{def}{=} \mathbb{K}_0 \to \mathbb{K}_1 \to \cdots \to \mathbb{K}_r$ where $\mathcal{D}^{1,n}_{\mathbb{K}_i}$ is the division property after i rounds. Moreover, for any vector $\boldsymbol{k}^*_{i+1} \in \mathbb{K}_{i+1}$, there must exist a vector $\boldsymbol{k}^*_i \in \mathbb{K}_i$ such that $\boldsymbol{k}^*_i$ can propagate to $\boldsymbol{k}^*_{i+1}$ by the propagation rules of division property. Furthermore, for $(\boldsymbol{k}_0, \boldsymbol{k}_1, ..., \boldsymbol{k}_r) \in (\mathbb{K}_0 \times \mathbb{K}_1 \times \cdots \times \mathbb{K}_r)$, if $\boldsymbol{k}_i$ can propagate to $\boldsymbol{k}_{i+1}$ for all $i \in \{0, 1, ..., r-1\}$, we call $(\boldsymbol{k}_0 \to \boldsymbol{k}_1 \to \cdots \to \boldsymbol{k}_r)$ an r-round division trail.

When we consider an r-round iterated cipher E and give the input division property $\mathcal{D}^{1,n}_{\boldsymbol{k}_0}$, if there is no division trail $\boldsymbol{k}_0 \xrightarrow{E} \boldsymbol{e}_i$, the ith bit of r-round output is balanced. The details of converting propagations of division property into linear inequalities in an MILP model can be referred to [XZBL16, SWW20]. With the MILP-aided technique to search division trails, we can easily verify whether each bit of r-round output is balanced or not. Later, Todo *et al.* [TIHM17] extended the MILP-aided division property to cube attacks at CRYPTO 2017, which can recover the superpolys of cubes beyond practical size.

The Flag Technique. At CRYPTO 2018, Wang *et al.* [WHT$^+$18] proposed an improved division property based cube attack. The flag technique was introduced to enhance the precision of propagations of division property. Specifically, for all variables in the MILP $v \in \mathcal{M}$, they added a flag variable $v.F \in \{0_c, 1_c, \delta\}$, where 0_c or 1_c means the state bit is constant 0 or 1, δ means this state bit is a variable. Moreover, they redefined the constraints for several basic operations i.e., XOR, COPY, AND, which can be used to construct a more accurate MILP model to describe the propagations of division property. As a result, they achieved improved cube attacks on Trivium, Kreyvium, Ascon and Grain-128a.

2.3 The Known Methods for Degree Evaluation

In this subsection, we briefly revisit several known methods for estimating the upper bounds on the degree of ciphers.

Two Formulas for SPN Ciphers. There are several formulas in [CV02, BCC11, BC13] and [Car20] for the degree evaluation of SPN ciphers. The following two formulas, proposed in [BC13] and [Car20], can derive more tighter bounds than [CV02, BCC11].

Theorem 1 ([BC13]). *Let F be a permutation from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ corresponding to the concatenation of s bijective Sboxes, $S_1, ..., S_s$, defined over $\mathbb{F}_2^{n_0}$. Then, for any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, we have*

$$\deg(G \circ F) \leq n - \frac{n - \deg(G)}{\gamma}, \tag{1}$$

where

$$\gamma = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \max_{1 \leq j \leq s} \delta_i(S_j)}.$$

Note that $\deg(G \circ F)$ must be an integer, thus (1) is equivalent to

$$\deg(G \circ F) \leq \left\lfloor n - \frac{n - \deg(G)}{\gamma} \right\rfloor.$$

Besides, the authors in [BC13] proposed another bound $\deg(G \circ F) < n - \left\lfloor \frac{n-1-\deg(G)}{\deg(F^{-1})} \right\rfloor$ to show the influence of $\deg(F^{-1})$ on $\deg(G \circ F)$, which is derived from (1). Recently, Carlet proposed a new bound (see (2)) in [Car20] by exploiting the graph indicators of vectorial Boolean functions. In the earlier version of [Car20], the author claimed that (2) improves by one unit the bound $\deg(G \circ F) < n - \left\lfloor \frac{n-1-\deg(G)}{\deg(F^{-1})} \right\rfloor$, but later was fixed as the two bounds are actually fully equivalent. Both of the two bounds are related to the degree of F^{-1}, but they are constructed from different aspects. Thus, in order to facilitate the following discussion and be consistent with (1), we will focus on (2) instead of $\deg(G \circ F) < n - \left\lfloor \frac{n-1-\deg(G)}{\deg(F^{-1})} \right\rfloor$ and present the bound in Theorem 2. Then we will compare division property with (1) and (2) respectively in Sect.3.

Theorem 2 ([Car20]). *Let F be a permutation of $\mathbb{F}_2^n$ and let G be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. Then we have*

$$\deg(G \circ F) \leq n - \left\lceil \frac{n - \deg(G)}{\deg(F^{-1})} \right\rceil. \tag{2}$$

Numeric Mapping for NFSR-Based Ciphers. Let $f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^m} a_{\boldsymbol{u}}^f \prod_{i=1}^m x_i^{u_i}$ be a Boolean function on m variables. Denoted by $\mathbb{B}_m$ and $\mathbb{Z}$ the set of all m-variable Boolean functions and the integer ring. Moreover, $\mathbb{Z}^m$ denotes the set of all m-dimensional vectors whose elements belong to $\mathbb{Z}$. The *numeric mapping* [Liu17], denoted by $\texttt{DEG}$, is defined as

$$\begin{aligned} \texttt{DEG} : \mathbb{B}_m \times \mathbb{Z}^m &\to \mathbb{Z}, \\ (f, D) &\mapsto \max_{a_{\boldsymbol{u}}^f \neq 0} \{\sum_{i=1}^m u_i d_i\}, \end{aligned}$$

where $D = (d_1, d_2, ..., d_m)$, and $a_{\boldsymbol{u}}^f$'s are coefficients of the ANF of f as defined previously. Let $g_1, g_2, ..., g_m$ be Boolean functions on n variables, $G = (g_1, g_2, ..., g_m)$ and $\texttt{deg}(G) = (\deg(g_1), \deg(g_2), ..., \deg(g_m))$. The numeric degree of the composite function $h = f \circ G$ is defined as $\texttt{DEG}(f, \texttt{deg}(G))$, denoted by $\texttt{DEG}(h)$ for short. We call $\texttt{DEG}(f, D)$ a super numeric degree of h if $d_i \geq \deg(g_i)$ for all $1 \leq i \leq m$. We can check that the algebraic degree of h is always less than or equal to the numeric degree of h, i.e.,

$$\deg(h) = \deg(f(g_1, g_2, ..., g_m)) \leq \texttt{DEG}(h) = \max_{a_{\boldsymbol{u}}^f \neq 0} \{\sum_{i=1}^m u_i \cdot \deg(g_i)\}.$$

The numeric mapping can be generally applied to the algebraic degree evaluation of NFSR-based ciphers. In fact, the core idea of numeric mapping is to estimate the degree of the monomial by computing the sum of degrees of all the variables contained in this monomial. In order to obtain a tighter bound on the algebraic degree for a particular cipher as Liu did in [Liu17] for Trivium and Kreyvium, we can iteratively compute the algebraic expression backward for several rounds, and use the degree of the previous states to estimate the degree of the current state.

Division Property. Another effective and accepted application of division property is the degree evaluation. It is a reverse use of the division property for searching balanced bits. Assuming that the MILP model $\mathcal{M}$ in which the propagation rules of the division property for a given block cipher with n-bit block size are described, and two n-bit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ denote MILP variables corresponding to the input and output division property, respectively. Moreover we constrain $\sum_{i=1}^{n} y_i = 1$ and maximize $\sum_{i=1}^{n} x_i$ by MILP. Suppose the optimized solution is d of an r-round cipher, then it indicates that the algebraic degree of the r-round cipher is upper bounded by d. Furthermore, if we focus on a specific bit in the r-round output, e.g., the jth bit, we constrain $\boldsymbol{y} = \boldsymbol{e}_j$ and maximize $\sum_{i=1}^{n} x_i$. The optimized solution being d means the algebraic degree of the jth bit of the r-round output is at most d. Similarly, for a given stream cipher with n-bit internal state and m-bit IV, let $\boldsymbol{iv}$, $\boldsymbol{u}^{(r)}$ and ks denote MILP variables corresponding to division property of IV, the r-round internal state and the r-round keystream bit, respectively. We constrain $\sum_{i=1}^{n} s_i^{(r)} = 0$, $ks = 1$ and maximize $\sum_{i=1}^{m} iv_i$. In addition, we need to set the division property of the initial states except IVs to 0. If the optimized solution returned by solvers is d, then we ensure that the upper bound on algebraic degree of r-round keystream bit defined over IV is d. These applications have been discussed in [BKL+17] and [WHT+18], respectively.

3 Degree Evaluation of SPN Ciphers

In this section, we establish relationships between word-based division property and Boura *et al.*'s and Carlet's formulas. Then we use word-based division property as a link to illustrate that bit-based division property can always derive a bound never worse than both of the two formulas. Finally, we show applications to Keccak and KNOT to provide evidences for our conclusion.

3.1 An Iterative Method for Degree Evaluation

Proposition 2. *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Denote W_w^F the minimal weight of the corresponding output division property when the weight of the input division property of F is equal to w. For any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, we have*

$$\deg(G \circ F) \leq \min_{0 \leq w \leq n} \{w - 1 | W_w^F > \deg(G)\}.$$

Proof. Let $\mathbb{W}$ be the set $\{w - 1 | W_w^F > \deg(G), 0 \leq w \leq n\}$. Suppose that the input multiset $\mathbb{X}$ of F has division property $\mathcal{D}_k^n$ and the corresponding output multiset $\mathbb{Y}$ has division property $\mathcal{D}_{k'}^n$, where $k \in \{0, 1, ..., n\}$. Then we have

$$\bigoplus_{\boldsymbol{x} \in \mathbb{X}} (G \circ F)(\boldsymbol{x}) = \bigoplus_{\boldsymbol{x} \in \mathbb{X}} G(F(\boldsymbol{x})) = \bigoplus_{\boldsymbol{y} \in \mathbb{Y}} G(\boldsymbol{y}). \tag{3}$$

Assuming that $k - 1 \in \mathbb{W}$, thus k' is great than $\deg(G)$ according to the definition of $\mathbb{W}$, which follows that (3) is always equal to 0 due to Proposition 1. This implies that the output multiset of $G \circ F$ has a balanced property. In other words, the algebraic degree of $G \circ F$ is bounded by $k-1$. Therefore, for an arbitrary $\mathbb{X}$ with division property $\mathcal{D}_k^n$, $\deg(G \circ F) \leq k-1$ always holds if $k-1 \in \mathbb{W}$, thus $\deg(G \circ F) \leq \min \mathbb{W}$. □

Based on Proposition 2, we introduce Algorithm 1 to evaluate the upper bound on $\deg(G \circ F)$. Let $\mathcal{M}_w^F$ be an MILP model, which describes the propagation from the input division property of F to the output division property of F. Moreover, its objective is to get the minimal weight of the output division property of F, and it contains an extra constraint that the weight of the input division property of F is fixed to w. Denoted by $OBJ(\mathcal{M}_w^F)$ the optimized solution of $\mathcal{M}_w^F$, thus W_w^F is equal to $OBJ(\mathcal{M}_w^F)$. In Algorithm 1, *in* and *out* respectively denote the weight of the input division property of F and the minimal weight of the corresponding output division property. Note that we can initialize *in* to be $\deg(G)$, since the weight of the corresponding output division property of F is clearly less than or equal to that of the input due to Proposition 1.

Algorithm 1 Calculate the upper bound on the degree of $G \circ F$ based on Proposition 2

```
Input: The function $F$ and the degree of $G$.
Output: The upper bound on $\deg(G \circ F)$.
$in \leftarrow \deg(G)$;
Construct the model $\mathcal{M}_{in}^F$;
$out \leftarrow OBJ(\mathcal{M}_{in}^F)$;
while $out \leq \deg(G)$ do
    $in \leftarrow in + 1$;
    Construct the model $\mathcal{M}_{in}^F$;
    $out \leftarrow OBJ(\mathcal{M}_{in}^F)$;
end while
return $in - 1$.
```

Additional, our target is to get the minimal *in* such that the corresponding *out* is always greater than $\deg(G)$, thus if a current objective value being equal to $\deg(G)$ occurs in the searching process of optimized solutions of Gurobi, we can always terminate the current searching process in order to save time. Note that this can be easily achieved using the `terminate`() function of Gurobi. In our experiments, this approach can provide a great improvement on the efficiency of Algorithm 1.

3.2 Comparisons between Division Property and Two Formulas

In this subsection, we will clarify the relationships between word-based division property, bit-based division property and other two formulas for the degree evaluation of SPN ciphers.

Definition 6 (Word-based Division Trail)**.** Let S be an $n_0 \times n_0$ Sbox. Assuming that the input multiset $\mathbb{X}$ and the corresponding output multiset $\mathbb{Y}$ of S respectively have division property $\mathcal{D}_k^{n_0}$ and $\mathcal{D}_{k'}^{n_0}$, where $k, k' \in \{0, 1, ..., n_0\}$, then we call $(k \longmapsto k')$ a word-based division trail of S. Similarly, let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ composed of s Sboxes ($S_1, ..., S_s$ defined over $\mathbb{F}_2^{n_0}$), assuming that the input multiset $\mathbb{X}$ and the output multiset $\mathbb{Y}$ of F respectively have division property $\mathcal{D}_{\boldsymbol{k}}^{n_0,s}$ and $\mathcal{D}_{\boldsymbol{k'}}^{n_0,s}$ where $\boldsymbol{k} = (k_1, ..., k_s), \boldsymbol{k'} = (k'_1, ..., k'_s)$ and $k_j, k'_j \in \{0, 1, ..., n_0\}$, we call $(\boldsymbol{k} \longmapsto \boldsymbol{k'})$ a word-based division trail of F.

Lemma 1. *Let S be an $n_0 \times n_0$ bijective Sbox. Assuming that $(k \longmapsto k')$ is an arbitrary word-based division trail of S, then we have*

$$k' \geq n_0 - (n_0 - k) \cdot \eta \tag{4}$$

where $\eta = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i(S)}$ and δ_i is the same as in Definition 4.

Proof. It is obvious that if $k = 0$, then $k' = 0$. Apparently, the conclusion holds since $\eta \geq \frac{n_0-1}{n_0-\delta_1(S)} = \frac{n_0-1}{n_0-\deg(S)} > 1$, where $\deg(S)$ denotes the algebraic degree of S. Similarly, if $k = n_0$ then $k' = n_0$, thus (4) holds when k equals to 0 and n_0, we will prove that (4) holds when $0 < k < n_0$. Because S is bijective, which indicates that $\delta_0(S) = 0$ and $\delta_{n_0}(S) = n_0$. Thus $0 \notin \{i|\delta_i(S) \geq k\}$ when $1 \leq k \leq n_0 - 1$. In this case, for any $x \in \{i|\delta_i(S) \geq k\} \setminus \{n_0\}$, we have

$$n_0 - (n_0 - k) \cdot \eta \leq n_0 - (n_0 - \delta_x(S)) \cdot \eta \leq n_0 - (n_0 - \delta_x(S)) \cdot \frac{n_0 - x}{n_0 - \delta_x(S)} = x.$$

Therefore, combining with the previous two cases $k = 0$ and $k = n_0$, for any k satisfying $0 \leq k \leq n_0$, we have $x \geq n_0 - (n_0 - k) \cdot \eta$ for all $x \in \{i|\delta_i(S) \geq k\}$, and due to Property 1

$$k' \geq n_0 - (n_0 - k) \cdot \eta$$

holds. □

According to the Lemma 1, we give a comparison between word-based division property and Boura and Canteaut's formula on degree evaluations as in the following proposition.

Proposition 3. *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$, which is the concatenation of s bijective Sboxes ($S_1, ..., S_s$ defined over $\mathbb{F}_2^{n_0}$, $sn_0 = n$). Denote W_w^F the minimal weight of the corresponding output division property when the weight of the input division property of F is equal to w. For any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, we have*

$$\deg(G \circ F) \leq \min_{0 \leq w \leq n} \{w - 1 | W_w^F > \deg(G)\} \leq \left\lfloor n - \frac{n - \deg(G)}{\gamma} \right\rfloor,$$

where γ is the same as in Theorem 1.

Proof. It is equivalent to prove that $\left\lfloor n - \frac{n-\deg(G)}{\gamma} \right\rfloor$ always belongs to $\{w - 1|W_w^F > \deg(G), 0 \leq w \leq n\}$. Let $(\boldsymbol{k} \longmapsto \boldsymbol{k'})$ be an arbitrary word-based division trail of F, where $\boldsymbol{k} = (k_1, ..., k_s)$, $\boldsymbol{k'} = (k'_1, ..., k'_s)$ and $k_j, k'_j \in \{0, 1, ..., n_0\}$ for $j \in \{1, 2, ..., s\}$. Assuming that the weight of the input division property of F is equal to $\left\lfloor n - \frac{n-\deg(G)}{\gamma} \right\rfloor + 1$, thus

$$\sum_{j=1}^{s} k_j = \left\lfloor n - \frac{n - \deg(G)}{\gamma} \right\rfloor + 1 > n - \frac{n - \deg(G)}{\gamma}.$$

Note that $(k_j \longmapsto k'_j)$ is the word-based division trail of jth Sbox S_j. Thus according to Lemma 1, we have

$$\sum_{j=1}^{s} k'_j \geq \sum_{j=1}^{s} [n_0 - (n_0 - k_j) \cdot \eta_j]$$

where $\eta_j = \max_{1 \leq i \leq n_0 - 1} \frac{n_0 - i}{n_0 - \delta_i(S_j)}$. In addition, it is clear that $\eta_j \leq \gamma$ always holds for any $j \in \{1, 2, ..., s\}$. Thus we have

$$\begin{aligned}
\sum_{j=1}^{s} k'_j &\geq \sum_{j=1}^{s} [n_0 - (n_0 - k_j) \cdot \eta_j] \\
&\geq \sum_{j=1}^{s} [n_0 - (n_0 - k_j) \cdot \gamma] \\
&\geq n - (n - \sum_{j=1}^{s} k_j) \cdot \gamma \\
&> n - [n - (n - \frac{n - \deg(G)}{\gamma})] \cdot \gamma = \deg(G).
\end{aligned}$$

It indicates that for any word-based division trail of F, if the weight of input division property of F is fixed to $\left\lfloor n - \frac{n-\deg(G)}{\gamma} \right\rfloor + 1$, then the weight of the corresponding output division property is always greater than $\deg(G)$. In other words, $W_w^F > \deg(G)$ always holds if $w = \left\lfloor n - \frac{n-deg(G)}{\gamma} \right\rfloor + 1$, i.e., $\left\lfloor n - \frac{n-\deg(G)}{\gamma} \right\rfloor \in \{w-1|W_w^F > \deg(G)\}$. Thus

$$\min_{0 \le w \le n} \{w-1|W_w^F > \deg(G)\} \le \left\lfloor n - \frac{n-\deg(G)}{\gamma} \right\rfloor$$

holds. □

For an r-round SPN cipher $E = (R_L \circ R_N)^r$, where R_L and R_N are the linear layer and non-linear layer of the round function. Let E_i denote the i-round reduced cipher. Boura and Conteaut's formula first decomposes $E_1 = R_L \circ R_N \triangleq G \circ F$, and computes its degree according to Theorem 1. Then the degree of E_i $(i = 2, ..., r)$ is iteratively computed by decomposing E_i as $(E_{i-1} \circ R_L) \circ R_N \triangleq G \circ F$. In addition, there are four models based on division property to estimate the degree of E. We list them as follows:

Model 1. Let the function F in Proposition 2 always equal to R_N and the function G is varying, i.e. $F \triangleq R_N$ and $G \triangleq E_{i-1} \circ R_L$, and the MILP model in Algorithm 1 is constructed based on the word-based division property (WDP).

Model 2. Let G be an identical function and F equal to E in Proposition 2, i.e. $G \triangleq I$ and $F \triangleq E$. We still use WDP to construct the MILP model in Algorithm 1.

Model 3. Let G be an identical function and F equal to E in Proposition 2, i.e. $G \triangleq I$ and $F \triangleq E$. But different from **Model 2**, we now use the bit-based division property (BDP) to construct the MILP model in Algorithm 1.

Model 4. The traditional degree evaluation method based on BDP, which has been discussed in Sect.2.

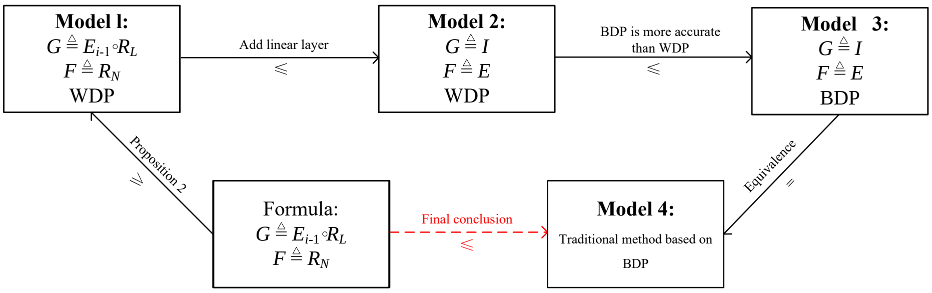

Figure 1: The relationship of these four models and Boura and Canteaut's formula. In this diagram, $\mathbf{A} \le \mathbf{B}$ means the value of the upper bound derived by $\mathbf{B}$ is less than or equal to that of $\mathbf{B}$.

The relationship of these four models and Boura and Canteaut's formula is illustrated as in Figure 1. First, Proposition 3 indicates that the value of the upper bound obtained by **Model 1** is less than or equal to that of Boura and Canteaut's formula. Next, **Model 2** is more precise than **Model 1**, since **Model 2** takes the influence of linear layers on the division property propagation into consideration. In addition, **Model 3** is more precise than **Model 2**, because **Model 3** is based on BDP and BDP is more accurate than WDP. Note that **Model 4** returns the maximal weight of the input division property of E when the weight of the output division property is fixed to 1. Assuming that the result obtained by **Model 4** is d, in other words, if we set the weight of the input division property of E to $d+1$, then the minimal weight of the corresponding output division property is exactly equal to 2. Thus **Model 3** is actually equivalent to **Model 4**. Finally, we deduce that the value of the upper bound obtained by **Model 4** is less than or equal to that of Boura and Canteaut's formula.

As a conclusion, *the bit-based division property will never be worse than Boura and Canteaut's formula for degree evaluations of SPN ciphers.* We can also obtian a similar conclusion for Carlet's formula, which will be discussed as follows.

Theorem 3 ([BC13]). *Let F be a permutation on $\mathbb{F}_2^n$. Then, for any integers k and l, $\delta_l(F^{-1}) < n-k$ if and only if $\delta_k(F) < n-l$.*

The contraposition of Theorem 3 can greatly help us to link the word-based division trail of an Sbox with the algebraic degree of its inverse as indicated in the following lemma.

Lemma 2. *Let S be an $n_0 \times n_0$ bijective Sbox. Assuming that $(k \longmapsto k')$ is a word-based division trail of S, then we have*

$$k' \geq n_0 - (n_0 - k) \cdot \deg(S^{-1}),$$

where S^{-1} is the inverse of S.

Proof. Due to Property 1, we have

$$\delta_{k'}(S) \geq k = n_0 - (n_0 - k). \tag{5}$$

Note that S is bijective, thus it is a permutation. According to the contraposition of Theorem 3 and (5), we can deduce

$$\delta_{n_0-k}(S^{-1}) \geq n_0 - k'.$$

Then, based on the trivial bound $\delta_{n_0-k}(S^{-1}) \leq (n_0-k)\cdot\delta_1(S^{-1})$, we have

$$n_0 - k' \leq (n_0 - k)\cdot \delta_1(S^{-1}) = (n_0-k)\cdot\deg(S^{-1}).$$

□

Proposition 4. *Let F be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$, which is the concatenation of s bijective Sboxes ($S_1, ..., S_s$ defined over $\mathbb{F}_2^{n_0}$, $sn_0 = n$). Denote W_w^F the minimal weight of the corresponding output division property when the weight of the input division property of F is equal to w. For any function G from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$, we have*

$$\deg(G \circ F) \leq \min_{0 \leq w \leq n} \{w - 1 | W_w^F > \deg(G)\} \leq n - \left\lceil \frac{n - \deg(G)}{\deg(F^{-1})} \right\rceil,$$

where F^{-1} denotes the inverse of F.

Proof. We reuse some notations in the proof of Proposition 3. Similarly, we only need to prove that $n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil$ belongs to the set $\{w-1 | W_w^F > \deg(G), 0 \leq w \leq n\}$. Assuming that the weight of the input division property of F is equal to $n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil + 1$, thus we have

$$\sum_{j=1}^{s} k_j = n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil + 1 > n - \frac{n-\deg(G)}{\deg(F^{-1})}.$$

According to Lemma 2, the weight of the corresponding output division property can be calculated as

$$\begin{aligned}\sum_{j=1}^{s} k'_j &\geq \sum_{j=1}^{s}[n_0 - (n_0 - k_j)\cdot \deg(S_j^{-1})] \\ &\geq \sum_{j=1}^{s}[n_0 - (n_0 - k_j)\cdot \deg(F^{-1})] \qquad (6) \\ &\geq n - (n - \sum_{j=1}^{s} k_j)\cdot \deg(F^{-1}) \\ &> n - [n - (n - \frac{n-\deg(G)}{\deg(F^{-1})})]\cdot \deg(F^{-1}) = \deg(G).\end{aligned}$$

The inequality (6) comes from the fact that $\deg(F^{-1}) \geq \deg(S_j^{-1})$ for any $j \in \{1, 2, ..., s\}$. It implies that the weight of the corresponding output division property is always greater than $\deg(G)$ if the weight of the input division property of F is equal to $n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil + 1$. Thus $W_w^F > \deg(G)$ holds for $w = n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil + 1$, i.e., $n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil \in \{w - 1 | W_w^F > \deg(G)\}$. Thus, we have

$$\min_{0\leq w\leq n}\{w-1|W_w^F > \deg(G)\} \leq n - \left\lceil \frac{n-\deg(G)}{\deg(F^{-1})} \right\rceil .$$

□

Similarly, *we can conclude that bit-based division property will never be worse than Carlet's formula for degree evalautions of SPN ciphers.*

Example 1. Small-PRESENT is a simplified version of PRESENT block cipher, and its round function is shown as in Figure 2.

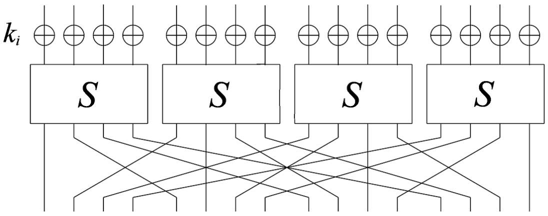

Figure 2: One round SP structure of small-PRESENT

When implementing Algorithm 1 to calculate the bound, we directly construct the 4-round MILP model. With the help of solvers, we obtain the estimated degree as well as a word-based division trail as follows.

$$\begin{aligned}&\text{Round 1: } (4,4,3,4) \xrightarrow{\mathcal{S}} (4,4,1,4) \xrightarrow{\mathcal{P}} (3,3,3,4)\\ &\text{Round 2: } (3,3,3,4) \xrightarrow{\mathcal{S}} (1,1,1,4) \xrightarrow{\mathcal{P}} (1,1,3,2)\\ &\text{Round 3: } (1,1,3,2) \xrightarrow{\mathcal{S}} (1,1,1,1) \xrightarrow{\mathcal{P}} (1,0,3,0)\\ &\text{Round 4: } (1,0,3,0) \xrightarrow{\mathcal{S}} (1,0,1,0) \xrightarrow{\mathcal{P}} (1,0,1,0)\end{aligned}$$

A four-dimensional vector, whose elements belong to $\{0, 1, 2, 3, 4\}$, denotes the word-based division property of the state, and the substitution layer and the permutation layer are denoted by $\mathcal{S}$ and $\mathcal{P}$, respectively. According to Proposition 2, the algebraic degree of 4-round small-PRESENT is upper bounded by $15 - 1 = 14$.

In addition, [BC13] and [Car20] give the same iterative formula to compute the upper bound as

$$\deg(G \circ F) \leq \min\{3 \cdot \deg(G), \left\lfloor 16 - \frac{16 - \deg(G)}{3} \right\rfloor\}.$$

In this case, the function F always represents the nonlinear layer, i.e. $F \triangleq \mathcal{S}$, and in order to avoid confusion, the function $G \triangleq (\mathcal{P} \circ \mathcal{S})^{i-1} \circ \mathcal{P}$ is represented by G_i in the i-round iterative computation. The computation is listed as follows, where d_i denotes the upper bound of i-round small-PRESENT.

$$\begin{aligned}
&\text{Round 1:} && \deg(G_1) = 1, \deg(F) = 3, d_1 = 3\\
&\text{Round 2:} && \deg(G_2) = 3, \deg(F) = 3, d_2 = 9\\
&\text{Round 3:} && \deg(G_3) = 9, \deg(F) = 3, d_3 = 13\\
&\text{Round 4:} && \deg(G_4) = 13, \deg(F) = 3, d_4 = 15
\end{aligned}$$

The 4-round degree is finally obtained as 15 by the two formulas, which is one more greater than the bound estimated by division property. The reason why this gap exists is that the former method does not consider the influence of $\mathcal{P}$. If we set the weight of the input division property to 15 and implement Algorithm 1 with an ideal linear layer, which means that any input division property can propagate to any output division property with the only restriction that they have the same weight. In this case, we enumerate all the word-based division trails of the 4-round cipher as

$$15 \xrightarrow{\mathcal{S}} 13 \xrightarrow{\mathcal{S}} \begin{cases} 7 \xrightarrow{\mathcal{S}} \begin{cases} 3 \xrightarrow{\mathcal{S}} 1 \text{ and } 2,\\ 4 \xrightarrow{\mathcal{S}} 2 \text{ and } 3,\\ 5 \xrightarrow{\mathcal{S}} 2, 3 \text{ and } 4,\\ 6 \xrightarrow{\mathcal{S}} 2, 3, 4 \text{ and } 5, \end{cases}\\ 10 \xrightarrow{\mathcal{S}} \begin{cases} 4 \xrightarrow{\mathcal{S}} 2 \text{ and } 3,\\ 6 \xrightarrow{\mathcal{S}} 2, 3, 4 \text{ and } 5,\\ 7 \xrightarrow{\mathcal{S}} 3, 4, 5 \text{ and } 6,\\ 9 \xrightarrow{\mathcal{S}} 3, 4, 5, 6 \text{ and } 7. \end{cases} \end{cases}$$

There exists a trail $15 \xrightarrow{\mathcal{S}} 13 \xrightarrow{\mathcal{S}} 7 \xrightarrow{\mathcal{S}} 3 \xrightarrow{\mathcal{S}} 1$, which indicates the estimated degree of 4-round small-PRESENT is lower bounded by 15. Combining with Proposition 3 and 4, the estimated degree of the 4-round cipher is 15. However, this trail $15 \xrightarrow{\mathcal{S}} 13 \xrightarrow{\mathcal{S}} 7 \xrightarrow{\mathcal{S}} 3 \xrightarrow{\mathcal{S}} 1$ is infeasible when taking $\mathcal{P}$ into consideration.

3.3 Experiments on Keccak and KNOT

In this subsection, we use the bit-based division property and two formulas in [BC13] and [Car20] to evaluate the algebraic degree of two SPN ciphers: Keccak [BDP09] and KNOT [ZDY+19]. Moreover, we list the comparisons to support our conclusions of Sect.3.2.

Degree Evaluation of Keccak. The Keccak sponge family, designed by Bertoni *et al.* [BDP09] in 2007, was selected as SHA-3 cryptographic hash function in 2012. The core component of Keccak sponge family is the Keccak-f permutation, which is a 1600-bit SPN permutation with 24 rounds. One can refer to [BDP09] for more details. The best known bound on the algebraic degree of Keccak-f was given in [BCC11], which led to a full-round zero-sum distinguisher. Later, focusing on the inverse Sbox, Duan *et al.* [DL11] improved the bound for the inverse Keccak-f, which decreased the size of the full-round zero-sum partition from 2^{1590} to 2^{1579} compared with [BCC11].

Algebraic degrees of the Keccak Sbox and its inverse are 2 and 3, respectively. Therefore, for the forward degree evaluation, the upper bounds on $\deg(G \circ F)$ can be obtained according to [BC13] and [Car20] as:

$$\deg(G \circ F) \leq \left\lfloor 1600 - \frac{1600 - \deg(G)}{3} \right\rfloor \text{ and } \deg(G \circ F) \leq 1600 - \left\lceil \frac{1600 - \deg(G)}{3} \right\rceil .$$

It is clear that $\left\lfloor 1600 - \frac{1600-\deg(G)}{3} \right\rfloor$ is identical to $1600 - \left\lceil \frac{1600-\deg(G)}{3} \right\rceil$. Similarly, for the backward degree evaluation of Keccak-f, the corresponding upper bounds on $\deg(G \circ F)$ are $\left\lfloor 1600 - \frac{1600-\deg(G)}{2} \right\rfloor$ and $1600 - \left\lceil \frac{1600-\deg(G)}{2} \right\rceil$.

Note that all the three linear transformations in Keccak-f are implemented on lanes, thus the whole linear layer is *translation-invariant* in the direction of each lane, which indicates that all the 64 bits within a lane have the same algebraic degree. When we use division property to estimate algebraic degrees, this provides us a tip that we need to focus only one bit in each lane of the output. This tip can largely reduce a lot of redundant computations. As a result, the shortest rounds where the degree is bounded by 1599 for forward and backward Keccak-f are both extended from 16 to 17. The comparisons are shown in Table 1.

Degree Evaluation of KNOT. KNOT [ZDY+19] is a family of authenticated encryption schemes and hash functions, which is one of the Round 2 candidates of NIST Lightweight Cryptography Standardization process[2]. KNOT contains two specific algorithms: KNOT-AEAD and KNOT-Hash. Both of them use the same underlying primitive: KNOT permutation, which is an SPN structure permutation with 256-, 384-, 512-bit internal state.

Note that the algebraic degrees of KNOT's Sbox and its inverse are both 3. Thus in the degree evaluations of forward and backward KNOT-n ($n = 256, 384, 512$), [BC13] and [Car20] provide formulas as $\left\lfloor n - \frac{n-\deg(G)}{3} \right\rfloor$ and $n - \left\lceil \frac{n-\deg(G)}{3} \right\rceil$ to calculate $\deg(G \circ F)$. Similar to Keccak-f, the linear layer in KNOT is *translation-invariant* in the direction of the word, thus we can utilize this property to save redundant computations. As a result, division property extends the shortest rounds where the degree is bounded by $n - 1$ from 9 to 13, 10 to 14, 11 to 15 for KNOT-256, 384, 512 respectively compared with [BC13] and [Car20], which are shown in Table 1.

4 Degree Evaluation of NFSR-based Ciphers

As an important cryptographic component, NSFR can be used to construct not only stream ciphers but also block ciphers. In this section, we mainly focus on the degree evaluation of stream ciphers based on NFSR.

[2] https://csrc.nist.gov/Projects/Lightweight-Cryptography

Table 1: The upper bounds on degree of the permutations of Keccak, KNOT and their inverses.

Permutation	Direction	#Round	#Bound†	Ref.
Keccak-f	Forward	16	1599	[BCC11][BC13][Car20]
			1598	Sect.3.3
		17	**1599**	Sect.3.3
	Backward	13	1599	[BCC11]
			1595	[DL11][BC13][Car20]
			1591	Sect.3.3
		16	1599	[DL11][BC13][Car20]
			1598	Sect.3.3
		17	**1599**	Sect.3.3
KNOT-256	Forward	9	255	[BC13][Car20]
			229	Sect.3.3
		13	**255**	Sect.3.3
	Backward	9	255	[BC13][Car20]
			222	Sect.3.3
		13	**255**	Sect.3.3
KNOT-384	Forward	10	383	[BC13][Car20]
			358	Sect.3.3
		14	**383**	Sect.3.3
	Backward	10	383	[BC13][Car20]
			361	Sect.3.3
		14	**383**	Sect.3.3
KNOT-512	Forward	11	511	[BC13][Car20]
			485	Sect.3.3
		15	**511**	Sect.3.3
	Backward	11	511	[BC13][Car20]
			479	Sect.3.3
		15	**511**	Sect.3.3

† More specific bounds can be referred to the full version at https://eprint.iacr.org/2021/175.pdf.

4.1 Comparison between Division Property and Numeric Mapping

For a generalized stream cipher based on an n-bit NFSR, assuming that the register at clock t is updated as

$$s_i^{(t)} = \begin{cases} s_{i+1}^{(t-1)} & \text{if } 1 \leq i < n, \\ g(\boldsymbol{s}^{(t-1)}) & \text{if } i = n, \end{cases}$$

where $\boldsymbol{s}^{(t)} = (s_1^{(t)}, s_2^{(t)}, ..., s_n^{(t)})$ denotes the internal state at clock t and g denotes the update function. The output key stream bit at clock t, denoted by $KS^{(t)}$, can be represented as $KS^{(t)} = f(\boldsymbol{s}^{(t)})$, where f is the output function. Assuming that the ANFs of the update function and output function are

$$g(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^g \pi_{\boldsymbol{u}}(\boldsymbol{x}) \text{ and } f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^f \pi_{\boldsymbol{u}}(\boldsymbol{x}).$$

When evaluating the algebraic degree of the output $KS^{(t)}$, the numeric mapping first utilizes trivial bound to iteratively compute degrees of the state $\boldsymbol{s}^{(t)}$ as

$$\begin{aligned}
\deg(s_n^{(1)}) &= \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\deg(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(0)}))\} \leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(0)})\} = d(s_n^{(1)}) \\
\deg(s_n^{(2)}) &= \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\deg(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(1)}))\} \leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(1)})\} = d(s_n^{(2)}) \\
&\vdots \\
\deg(s_n^{(t)}) &= \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\deg(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t-1)}))\} \leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(t-1)})\} = d(s_n^{(t)})
\end{aligned}$$

where $d(s_i^{(j)})$ denotes the degree of the state $s_i^{(j)}$ estimated by numeric mapping. Note that $d(s_i^{(0)})$ is initialized to the exact algebraic degree of the state $s_i^{(0)}$. Thus the degree of the output bit $KS^{(t)}$, denoted by $d(KS^{(t)})$, can be similarly calculated using numeric mapping as

$$\deg(KS^{(t)}) = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^f\neq 0}\{\deg(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)}))\} \leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^f\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(t)})\} = d(KS^{(t)}).$$

When applying bit-based division property to estimate the degree of $KS^{(t)}$, we convert the division property propagations from the initial state to the t-clock state into a system of inequalities of an MILP model, and constrain the division property of $s_i^{(t)}$ (for all $1 \leq i \leq n$) to 0 and the division property of $KS^{(t)}$ to 1. Then the maximal weight of division property of the initial state, which will be taken as the degree of $KS^{(t)}$, can be computed by solvers. This is a whole process but we can also interpret this process in an iterative way in order to compare it with numeric mapping intuitively. Before presenting a high-level comparison, we first introduce some important conclusions.

Proposition 5. *For the aforementioned generalized stream cipher, let h be an arbitrary function on the internal state at clock t as $h(\boldsymbol{s}^{(t)}) = \bigoplus_{\boldsymbol{u}\in\mathbb{F}_2^n} a_{\boldsymbol{u}}^h \pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)})$. Assuming that $\hat{d}(h)$ and $\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)}))$ are the degrees of h and $\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)})$ estimated by division property, then we have $\hat{d}(h) = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^h\neq 0}\{\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)}))\}$.*

Proof. Obviously, the conclusion holds when $h(\boldsymbol{s}^{(t)})$ contains only one monomial. Besides, it is equivalent to prove that it still holds for the case when $h(\boldsymbol{s}^{(t)})$ contains two monomials. Suppose that $h(\boldsymbol{s}^{(t)}) = \pi_{\boldsymbol{u_1}}(\boldsymbol{s}^{(t)}) \oplus \pi_{\boldsymbol{u_2}}(\boldsymbol{s}^{(t)})$, where $\boldsymbol{u_1} \neq \boldsymbol{u_2}$, $\boldsymbol{u_1} \neq \boldsymbol{0}$ and $\boldsymbol{u_2} \neq \boldsymbol{0}$. On the one hand, we know that $\hat{d}(h)$ is the maximal weight of division property of the initial state when division property of h is 1. On the other hand, according to the propagation rule of division property on XOR operation, there are two cases for division property of $(\pi_{\boldsymbol{u_1}}(\boldsymbol{s}^{(t)}), \pi_{\boldsymbol{u_2}}(\boldsymbol{s}^{(t)}))$: $(1, 0)$ and $(0, 1)$. The maximal weight of division property of the initial state are $\hat{d}(\pi_{\boldsymbol{u_1}}(\boldsymbol{s}^{(t)}))$ and $\hat{d}(\pi_{\boldsymbol{u_2}}(\boldsymbol{s}^{(t)}))$ for $(1, 0)$ and $(0, 1)$ respectively. Thus $\hat{d}(h)$ is naturally equal to the greater one among $\hat{d}(\pi_{\boldsymbol{u_1}}(\boldsymbol{s^{(t)}}))$ and $\hat{d}(\pi_{\boldsymbol{u_2}}(\boldsymbol{s^{(t)}}))$, i.e. $\hat{d}(h) = \max\{\hat{d}(\pi_{\boldsymbol{u_1}}(\boldsymbol{s^{(t)}})), \hat{d}(\pi_{\boldsymbol{u_2}}(\boldsymbol{s^{(t)}}))\}$. □

Proposition 6. *For the aforementioned generalized stream cipher, let h be a function on the internal state at clock t as $h(\boldsymbol{s}^{(t)}) = \prod_{i\in I, I\subseteq\{1,2,\ldots,n\}} s_i^{(t)}$. Assuming that $\hat{d}(h)$ and $\hat{d}(s_i^{(t)})$ are the degrees of h and $s_i^{(t)}$ estimated by division property, then we have $\hat{d}(h) \leq \sum_{i\in I} \hat{d}(s_i^{(t)})$.*

Before proving Proposition 6, we first introduce a helpful lemma.

Lemma 3. *Assuming that $\boldsymbol{k} \xrightarrow{f} \boldsymbol{l}$ is a valid division trail. Then for any pair $(\boldsymbol{l}_1, \boldsymbol{l}_2)$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l}_2 = \boldsymbol{l}$, there must exist a pair $(\boldsymbol{k}_1, \boldsymbol{k}_2)$ satisfying $\boldsymbol{k}_1 \vee \boldsymbol{k}_2 = \boldsymbol{k}$ such that $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ are two valid division trails, where f denotes a Boolean function and $\vee$ denotes the bitwise OR operation.*

Proof. With a slight abuse of notation, we will use $\vee$ in the following to denote the bitwise OR operation of both two vectors and two bits. Since f is a combination of the bitwise XOR, AND and COPY operations, we give the proof for each operation respectively.

1. Let f be the XOR operation as $(*, ..., *, x, y) \xrightarrow{f} (*, ..., *, x \oplus y)$. Assuming that $\boldsymbol{k} = (*, ..., *, k_1, k_2)$, $\boldsymbol{l} = (*, ..., *, l)$ and $\boldsymbol{k} \xrightarrow{f} \boldsymbol{l}$ is a valid division trail, where $k_1, k_2, l \in \{0, 1\}$. Then we have the relation $k_1 + k_2 = l$ from the propagation rule of division property on XOR operation. First, if $k_1 = k_2 = 0$, then $l = 0$. In this case, the conclusion holds apparently. If $k_1 = 1$ and $k_2 = 0$, then we have $l = 1$, and there are three pairs $(\boldsymbol{l}_1, \boldsymbol{l_2})$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l}_2 = \boldsymbol{l}$. When $\boldsymbol{l}_1 = (*, ..., *, 1)$ and $\boldsymbol{l}_2 = (*, ..., *, 0)$, let $\boldsymbol{k}_1 = (*, ..., *, 1, 0)$ and $\boldsymbol{k}_2 = (*, ..., *, 0, 0)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. When $\boldsymbol{l}_1 = (*, ..., *, 0)$ and $\boldsymbol{l}_2 = (*, ..., *, 1)$, let $\boldsymbol{k}_1 = (*, ..., *, 0, 0)$ and $\boldsymbol{k}_2 = (*, ..., *, 1, 0)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. When $\boldsymbol{l}_1 = (*, ..., *, 1)$ and $\boldsymbol{l}_2 = (*, ..., *, 1)$, let $\boldsymbol{k}_1 = (*, ..., *, 1, 0)$ and $\boldsymbol{k}_2 = (*, ..., *, 1, 0)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. In addition, if $k_1 = 0$ and $k_2 = 1$, which is similar to the case $k_1 = 1$ and $k_2 = 0$, we can always find a pair $(\boldsymbol{k}_1, \boldsymbol{k_2})$ satisfying $\boldsymbol{k}_1 \vee \boldsymbol{k}_2 = \boldsymbol{l}$ such that $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ are two valid division trails for any pair $(\boldsymbol{l}_1, \boldsymbol{l}_2)$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l_2} = \boldsymbol{l}$. Thus the conclusion holds for XOR operation.

2. Let f be the AND operation as $(*, ..., *, x, y) \xrightarrow{f} (*, ..., *, x\&y)$. Assuming that $\boldsymbol{k} = (*, ..., *, k_1, k_2)$, $\boldsymbol{l} = (*, ..., *, l)$ and $\boldsymbol{k} \xrightarrow{f} \boldsymbol{l}$ is a valid division trail, where $k_1, k_2, l \in \{0, 1\}$. Then we have the relation $k_1 \vee k_2 = l$ from the propagation rule of division property on AND operation. First, if $k_1 = k_2 = 0$, then $l = 0$. In this case, the conclusion holds apparently. If $k_1 = 1, k_2 = 0$ or $k_1 = 0, k_2 = 1$, we have $l = 1$, this is completely similar to the proof when f represents the XOR operation. Thus we only focus on the case that $k_1 = k_2 = 1$. Because $l = 1$, there are three pairs $(\boldsymbol{l}_1, \boldsymbol{l_2})$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l}_2 = \boldsymbol{l}$. When $\boldsymbol{l}_1 = (*, ..., *, 1)$ and $\boldsymbol{l}_2 = (*, ..., *, 0)$, let $\boldsymbol{k}_1 = (*, ..., *, 1, 1)$ and $\boldsymbol{k}_2 = (*, ..., *, 0, 0)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. When $\boldsymbol{l}_1 = (*, ..., *, 0)$ and $\boldsymbol{l}_2 = (*, ..., *, 1)$, let $\boldsymbol{k}_1 = (*, ..., *, 0, 0)$ and $\boldsymbol{k}_2 = (*, ..., *, 1, 1)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. When $\boldsymbol{l}_1 = (*, ..., *, 1)$ and $\boldsymbol{l}_2 = (*, ..., *, 1)$, let $\boldsymbol{k}_1 = (*, ..., *, 1, 1)$ and $\boldsymbol{k}_2 = (*, ..., *, 1, 1)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. Thus the conclusion holds for AND operation.

3. Let f be the COPY operation as $(*, ..., *, x) \xrightarrow{f} (*, ..., *, x, x)$. Assuming that $\boldsymbol{k} = (*, ..., *, k)$, $\boldsymbol{l} = (*, ..., *, l_1, l_2)$ and $\boldsymbol{k} \xrightarrow{f} \boldsymbol{l}$ is a valid division trail, where $k, l_1, l_2 \in \{0, 1\}$. Then we have the relation $l_1 + l_2 = k$ from the propagation rule of division property on COPY operation. First, if $k = 0$, then $l_1 = l_2 = 0$. In this case, the conclusion holds apparently. For the case of k being 1, if $l_1 = 1$ and $l_2 = 0$, then there are three pairs $(\boldsymbol{l}_1, \boldsymbol{l_2})$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l}_2 = \boldsymbol{l}$. When $\boldsymbol{l}_1 = (*, ..., *, 1, 0)$ and $\boldsymbol{l}_2 = (*, ..., *, 0, 0)$, let $\boldsymbol{k}_1 = (*, ..., *, 1)$ and $\boldsymbol{k}_2 = (*, ..., *, 0)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. When $\boldsymbol{l}_1 = (*, ..., *, 0, 0)$ and $\boldsymbol{l}_2 = (*, ..., *, 1, 0)$, let $\boldsymbol{k}_1 = (*, ..., *, 0)$ and $\boldsymbol{k}_2 = (*, ..., *, 1)$, then $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid

division trails. When $\boldsymbol{l}_1 = (*, ..., *, 1, 0)$ and $\boldsymbol{l}_2 = (*, ..., *, 1, 0)$, let $\boldsymbol{k}_1 = (*, ..., *, 1)$ and $\boldsymbol{k}_2 = (*, ..., *, 1)$, $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ can be two valid division trails. In addition, if $l_1 = 0$ and $l_2 = 1$, which is similar to the case $l_1 = 1$ and $l_2 = 0$, we can always find a pair $(\boldsymbol{k}_1, \boldsymbol{k_2})$ satisfying $\boldsymbol{k}_1 \vee \boldsymbol{k}_2 = \boldsymbol{l}$ such that $\boldsymbol{k}_1 \xrightarrow{f} \boldsymbol{l}_1$ and $\boldsymbol{k}_2 \xrightarrow{f} \boldsymbol{l}_2$ are two valid division trails for any pair $(\boldsymbol{l}_1, \boldsymbol{l}_2)$ satisfying $\boldsymbol{l}_1 \vee \boldsymbol{l_2} = \boldsymbol{l}$. Thus the conclusion holds for COPY operation.

In summary, the conclusion holds. □

Now we are ready to prove Proposition 6.

Proof. It is clearly true for $|I| = 1$. Thus it is significant to prove that it also holds for $|I| = 2$. Now we will prove it by contradiction. Assume $I = \{p, q\} \subseteq \{1, 2, ..., n\}$, $h(\boldsymbol{s}^{(t)}) = s_p^{(t)} \cdot s_q^{(t)}$ and $\hat{d}(h) > \hat{d}(s_p^{(t)}) + \hat{d}(s_q^{(t)})$. According to the propagation rules of division property, when the division property of $h(\boldsymbol{s}^t)$ is 1, there are three cases for the division property of $(s_p^{(t)}, s_q^{(t)})$: (1, 0), (0, 1), (1, 1). Denoted by $\hat{d}$ the maximal weight of the initial division property corresponding to the third case, thus $\hat{d}(h) = \max\{\hat{d}(s_p^{(t)}), \hat{d}(s_q^{(t)}), \hat{d}\}$.

Assume $\boldsymbol{k} \xrightarrow{t-round} \boldsymbol{l}$ be a division trail, where $wt(\boldsymbol{k}) = \hat{d}$ and $\boldsymbol{l} = \boldsymbol{e}_p \vee \boldsymbol{e}_q$. Then according to Lemma 3, there exists a pair denoted by $(\boldsymbol{k}_p, \boldsymbol{k}_q)$ satisfying $\boldsymbol{k}_p \vee \boldsymbol{k}_q = \boldsymbol{k}$ such that $\boldsymbol{k}_p \xrightarrow{t-round} \boldsymbol{e}_p$ and $\boldsymbol{k}_q \xrightarrow{t-round} \boldsymbol{e}_q$ are two valid division trails. Note that $wt(\boldsymbol{k}_p) + wt(\boldsymbol{k}_q) \geq wt(\boldsymbol{k})$ clearly holds and we have $\hat{d} > \hat{d}(s_p^{(t)}) + \hat{d}(s_q^{(t)})$ from the assumption $\hat{d}(h) > \hat{d}(s_p^{(t)}) + \hat{d}(s_q^{(t)})$, thus at least one of the two inequalities $wt(\boldsymbol{k}_p) > \hat{d}(s_p^{(t)})$ and $wt(\boldsymbol{k}_q) > \hat{d}(s_q^{(t)})$ holds. This contradicts with the fact that there exists no initial division property with weight greater than $\hat{d}(s_p^{(t)})$ or $\hat{d}(s_q^{(t)})$ that can propagate to $\boldsymbol{e}_p$ or $\boldsymbol{e}_q$, since $\hat{d}(s_p^{(t)})$ and $\hat{d}(s_q^{(t)})$ are the estimated degrees of $s_p^{(t)}$ and $s_q^{(t)}$ by division property. Thus the assumption $\hat{d}(h) > \hat{d}(s_p^{(t)}) + \hat{d}(s_q^{(t)})$ is negative, and our conclusion holds. □

Proposition 6 indicates a "trivial" bound for the degree of the product of several functions estimated by division property. Based on this property, we now can interpret the division property method in an iterative way and easily compare it with the numeric mapping, and finally obtain the following conclusion.

Proposition 7. *For the aforementioned generalized stream cipher, denoted by $d(s_i^{(t)})$ and $\hat{d}(s_i^{(t)})$ the degree of the ith state bit estimated by numeric mapping and division property at clock t, respectively. Then for any $1 \leq i \leq n$, we have $\hat{d}(s_i^{(t)}) \leq d(s_i^{(t)})$. Moreover, denoted by $d(KS^{(t)})$ and $\hat{d}(KS^{(t)})$ the degree of the output bit estimated by numeric mapping and division property at clock t, then we have $\hat{d}(KS^{(t)}) \leq d(KS^{(t)})$.*

Proof. We only need to prove that $\hat{d}(s_n^{(t)}) \leq d(s_n^{(t)})$ holds since $s_n^{(t)}$ is the updated state bit at clock t. Note that when $t = 0$, both of $\hat{d}(s_i^{(t)})$ and $d(s_i^{(t)})$ are initialized to the exact algebraic degree of the state bit $s_i^{(0)}$. Thus for all $1 \leq i \leq n$, $\hat{d}(s_i^{(0)}) = d(s_i^{(0)})$. For $t > 0$, we have

$$\deg(s_n^{(1)}) \leq \hat{d}(s_n^{(1)}) = \hat{d}(g(\boldsymbol{s}^{(0)})) = \max_{\boldsymbol{u} \in \mathbb{F}_2^n, a_{\boldsymbol{u}}^g \neq 0} \{\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(0)}))\}$$

due to Proposition 5, where $\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(0)}))$ denotes the computed degree of the monomial $\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(0)})$ by division property. According to Proposition 6, we know

$$\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(0)})) = \hat{d}(\prod_{i=1}^{n} (s_i^{(0)})^{u_i}) \leq \sum_{i=1}^{n} u_i \cdot \hat{d}(s_i^{(0)}).$$

Thus we have

$$\hat{d}(s_n^{(1)}) \leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot \hat{d}(s_i^{(0)})\} = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(0)})\} = d(s_n^{(1)}).$$

Similarly,

$$\begin{aligned}
\hat{d}(s_n^{(2)}) = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(1)}))\} &\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot \hat{d}(s_i^{(1)})\} \\
&\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(1)})\} = d(s_n^{(2)}) \\
&\vdots \\
\hat{d}(s_n^{(t)}) = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t-1)}))\} &\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot \hat{d}(s_i^{(t-1)})\} \\
&\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^g\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(t-1)})\} = d(s_n^{(t)}).
\end{aligned}$$

From the above iteration, it is clear that $\hat{d}(s_i^{(t)}) \leq d(s_i^{(t)})$ always holds for all $1 \leq i \leq n$ at any clock t. Thus we have

$$\begin{aligned}
\hat{d}(KS^{(t)}) = \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^f\neq 0}\{\hat{d}(\pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t)}))\} &\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^f\neq 0}\{\sum_{i=1}^{n} u_i \cdot \hat{d}(s_i^{(t)})\} \\
&\leq \max_{\boldsymbol{u}\in\mathbb{F}_2^n, a_{\boldsymbol{u}}^f\neq 0}\{\sum_{i=1}^{n} u_i \cdot d(s_i^{(t)})\} = d(KS^{(t)}).
\end{aligned}$$

□

Proposition 7 indicates that the division property is never worse than the numeric mapping for degree evaluations of NFSR-based stream ciphers. In a particular stream cipher, there may be more than one registers and update functions, and the output function may be more complex, but these factors will not affect our final conclusion. In this case, the numeric mapping may compute a tighter bound by exploiting the algebraic properties of the update functions and output function. Liu specially introduced an algorithm to estimate the algebraic degree for Trivium-like ciphers in [Liu17], and we will give a more detailed comparison on degree evaluations of Trivium-like ciphers in the next subsections.

4.2 Applications of Numeric Mapping to Trivium-like Ciphers

In order to compare the applications of division property and numeric mapping to Trivium-like ciphers intuitively, we first introduce some special notations in this subsection.

4.2.1 Trivium-like Stream Ciphers

Let X, Y and Z be three feedback shift registers with size n_X, n_Y and n_Z respectively. Denoted by $\boldsymbol{x}^{(t)}$, $\boldsymbol{y}^{(t)}$ and $\boldsymbol{z}^{(t)}$ their corresponding states at clock t,

$$\begin{aligned}
\boldsymbol{x}^{(t)} &= (x_1^{(t)}, x_2^{(t)}, ..., x_{n_X}^{(t)}), \\
\boldsymbol{y}^{(t)} &= (y_1^{(t)}, y_2^{(t)}, ..., y_{n_Y}^{(t)}), \\
\boldsymbol{z}^{(t)} &= (z_1^{(t)}, z_2^{(t)}, ..., z_{n_Z}^{(t)}),
\end{aligned}$$

and the states are updated as follows,

$$x_i^{(t)} = \begin{cases} x_{i+1}^{(t-1)} & \text{if } 1 \le i \le n_X - 1, \\ z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)} + \ell_X(\boldsymbol{s}^{(t-1)}) & \text{otherwise,} \end{cases}$$

$$y_i^{(t)} = \begin{cases} y_{i+1}^{(t-1)} & \text{if } 1 \le i \le n_Y - 1, \\ x_{r_X}^{(t-1)} \cdot x_{r_X+1}^{(t-1)} + \ell_Y(\boldsymbol{s}^{(t-1)}) & \text{otherwise,} \end{cases}$$

$$z_i^{(t)} = \begin{cases} z_{i+1}^{(t-1)} & \text{if } 1 \le i \le n_Z - 1, \\ y_{r_Y}^{(t-1)} \cdot y_{r_Y+1}^{(t-1)} + \ell_Z(\boldsymbol{s}^{(t-1)}) & \text{otherwise,} \end{cases}$$

where $1 \le r_\lambda < n_\lambda$ and ℓ_λ is a linear function for $\lambda \in \{X, Y, Z\}$. The internal state at clock t denoted by $\boldsymbol{s}^{(t)}$ is composed of the three registers, i.e.,

$$\boldsymbol{s}^{(t)} = (s_1^{(t)}, s_2^{(t)}, ..., s_n^{(t)}) = (x_1^{(t)}, ..., x_{n_X}^{(t)}, y_1^{(t)}, ..., y_{n_Y}^{(t)}, z_1^{(t)}, ..., z_{n_Z}^{(t)})$$

where n denotes the size of internal states, i.e. $n = n_X + n_Y + n_Z$. Let f be the output function, after an initialization of N rounds, the cipher generates a keystream bit $KS^{(t)}$ by $f(\boldsymbol{s}^{(t)})$ for each $t \ge N$.

Trivium-like stream ciphers can be represented as above roughly, additional details depend on the specific cipher. Trivium exactly falls into this kind of ciphers. Kreyvium is a variant of Trivium with 128-bit security. Moreover, two extra registers K^* and IV^* without updating but shifting, which only involve the key bits and IV bits respectively, are used in Kreyvium to provide single bit to each of ℓ_X and ℓ_Y. Trivium uses an 80-bit IV and key, while Kreyvium uses a 128-bit IV and key. Both ciphers have 1152-round initialization. One can refer to [CP08, CCF+16] for more details of this two ciphers.

4.2.2 Formalizing the Applications of Numeric Mapping to Trivium-like Ciphers

In [Liu17], an efficient algorithm is proposed to estimate degrees of updated states and the output keystream bit for Trivium-like ciphers. It takes advantage of the property that the only nonlinear term of the update functions is the product of two neighboring state bits. Thus, it can obtain a more accurate degree based on numeric mapping by iteratively computing the algebraic expression backward for one round. We give a formalized description of this algorithm for intuition as follows. Denote $d(x_i^{(t)}), d(y_i^{(t)})$ and $d(z_i^{(t)})$ the degrees of the states $x_i^{(t)}, y_i^{(t)}$ and $z_i^{(t)}$ computed by numeric mapping at clock t, respectively. The degree of the internal state at clock t $(0 \le t \le N)$ is represented as

$$\begin{aligned} D^{(t)} &= (d(s_1^{(t)}), d(s_2^{(t)}), ..., d(s_n^{(t)})) \\ &= (d(x_1^{(t)}), ..., d(x_{n_X}^{(t)}), d(y_1^{(t)}), ..., d(y_{n_Y}^{(t)}), d(z_1^{(t)}), ..., d(z_{n_Z}^{(t)})) \end{aligned}$$

where $n = n_X + n_Y + n_Z$. Particularly, $D^{(0)}$ denotes the degree of the initial state and is equal to the exact algebraic degree. The degree of the linear function $\ell_\lambda(\boldsymbol{s}^{(t)})$ for $\lambda \in \{X, Y, Z\}$, denoted by $\texttt{DEG}(\ell_\lambda, D^{(t)})$, is computed as $\texttt{DEG}(\ell_\lambda, D^{(t)}) = \max_{1 \le i \le n}\{a_{\boldsymbol{e}_i}^{\ell_\lambda} \cdot d(s_i^{(t)})\}$ (see in Sect.2). If $D^{(j)}$ for $j \le t-1$ is known, then we can compute the degree of the updated state of the register X due to numeric mapping as follows:

$$d(x_{n_X}^{(t)}) = \begin{cases} \max\{d(z_{n_Z+t_1}^{(0)}) + d(z_{r_Z}^{(t-1)}), \texttt{DEG}(\ell_X, D^{(t-1)})\} & \text{if } t_1 \le 0, \\ \max\{d_1, d_2, d_3, \texttt{DEG}(\ell_X, D^{(t-1)})\} & \text{otherwises.} \end{cases}$$

where

$$\begin{aligned}
t_1 &= t + r_Z - n_Z - 1,\\
d_1 &= \min\{d(z_{n_Z}^{(t_1)}) + d(y_{r_Y+1}^{(t_1)}), d(z_{n_Z}^{(t_1+1)}) + d(y_{r_Y}^{(t_1-1)}), d(y_{r_Y}^{(t_1-1)}) + d(y_{r_Y}^{(t_1)}) + d(y_{r_Y+1}^{(t_1)})\},\\
d_2 &= \mathtt{DEG}(\ell_Z, D^{(t_1)}) + d(z_{n_Z}^{(t_1)}),\\
d_3 &= \mathtt{DEG}(\ell_Z, D^{(t_1-1)}) + d(z_{n_Z}^{(t_1+1)}).
\end{aligned}$$

The processes to compute degrees of the updated states of registers Y and Z are similar to the above. Finally, the degree of the key stream $KS^{(t)}$ is easy to be computed as $d(KS^{(t)}) = \mathtt{DEG}(f, D^t)$ after we get $D^{(t)}$.

4.3 A Detailed Comparison on the Degree Evaluation of Trivium-like Ciphers between Division Property and Numeric Mapping

When applying division property to the degree evaluation of Trivium-like ciphers, we denote $\hat{d}(x_i^{(t)}), \hat{d}(y_i^{(t)})$ and $\hat{d}(z_i^{(t)})$ the estimated degrees of the states $x_i^{(t)}, y_i^{(t)}$ and $z_i^{(t)}$ at clock t, respectively. The degree of the internal state at clock t $(0 \le t \le N)$ is denoted as

$$\begin{aligned}
\hat{D}^{(t)} &= (\hat{d}(s_1^{(t)}), \hat{d}(s_2^{(t)}), ..., \hat{d}(s_n^{(t)}))\\
&= (\hat{d}(x_1^{(t)}), ..., \hat{d}(x_{n_X}^{(t)}), \hat{d}(y_1^{(t)}), ..., \hat{d}(y_{n_Y}^{(t)}), \hat{d}(z_1^{(t)}), ..., \hat{d}(z_{n_Z}^{(t)})).
\end{aligned}$$

Especially, $\hat{D}^{(0)}$ denotes the degree of the initial state and is equal to the exact algebraic degree. According to Proposition 5, the estimated degree of the linear function $\ell_\lambda(\boldsymbol{s}^{(t)})$ for $\lambda \in \{X, Y, Z\}$, denoted by $\hat{d}(\ell_\lambda(\boldsymbol{s}^{(t)}))$, can be represented as

$$\hat{d}(\ell_\lambda(\boldsymbol{s}^{(t)})) = \max_{1\le i\le n}\{a_{\boldsymbol{e}_i}^{\ell_\lambda} \cdot \hat{d}(s_i^{(t)})\}$$

Proposition 8. *For Trivium-like ciphers, denoted by $\hat{d}(s_i^{(t)})$ and $d(s_i^{(t)})$ the degree of the ith state bit $s_i^{(t)}$ estimated by division property and numeric mapping at clock t respectively, where $d(s_i^{(t)})$ is computed as in the last subsection. Then we have $\hat{d}(s_i^{(t)}) \le d(s_i^{(t)})$ for all $1 \le i \le n$ and $t \ge 0$. Moreover, denoted by $\hat{d}(KS^{(t)})$ and $d(KS^{(t)})$ the degree of the output bit estimated by division property and numeric mapping at clock t, then we have $\hat{d}(KS^{(t)}) \le d(KS^{(t)})$.*

Proof. We know both $D^{(0)}$ and $\hat{D}^{(0)}$ are initialized by the exact algebraic degree, thus this conclusion apparently holds for $t = 0$. In fact, we only need to pay attention to the updated state bits instead of every state bit. Here we just illustrate the details of the proof for register X due to the similarity of these three registers. Assuming that $\hat{d}(s_i^{(j)}) \le d(s_i^{(j)})$ always holds for all $1 \le i \le n$ and $1 \le j \le t-1$, then we need to prove $\hat{d}(x_{n_X}^{(t)}) \le d(x_{n_X}^{(t)})$ holds.

When estimating the value of $\hat{d}(x_{n_X}^{(t)})$, the division property of state $x_{n_X}^{(t)}$ is 1. Thus, according to the update function and Proposition 5, we have

$$\hat{d}(x_{n_X}^{(t)}) = \max\{\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}), \hat{d}(\ell_X(\boldsymbol{s}^{(t-1)}))\}$$

where $\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)})$ denotes the degree of the monomial $z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)})$ by division property. It is clear that $\hat{d}(\ell_X(\boldsymbol{s}^{(t-1)})) \le \mathtt{DEG}(\ell_X, D^{(t-1)})$ holds because of the assumption that $\hat{d}(s_i^{(j)}) \le d(s_i^{(j)})$ for any $1 \le i \le n$ and $1 \le j \le t-1$. Thus we have

$$\hat{d}(x_{n_X}^{(t)}) \le \max\{\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}), \mathtt{DEG}(\ell_X, D^{(t-1)})\}.$$

Now we focus on the degree of the monomial $z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}$. Let $t_1 = t + r_Z - n_Z - 1$, similar to the numeric mapping, there are two cases as the following discussion.

1. $t_1 \leq 0$. In this case, the state $z_{r_Z}^{(t-1)}$ can be computed as

$$z_{r_Z}^{(t-1)} = z_{r_Z+1}^{(t-2)} = \cdots = z_{r_Z+m}^{(t-1-m)} = \cdots = z_{r_Z+t-1}^{(0)} = z_{n_Z+t_1}^{(0)}$$

due to the update function. Note that $z_{n_Z+t_1}^{(0)}$ is the initial state because $n_Z + t_1 \leq n_Z$. Thus we can rewrite $z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}$ as $z_{n_Z+t_1}^{(0)} \cdot z_{r_Z+1}^{(t-1)}$ and according to Proposition 6 we have $\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}) \leq \hat{d}(z_{n_Z+t_1}^{(0)}) + \hat{d}(z_{r_Z+1}^{(t-1)})$. Thus,

$$\begin{aligned}\hat{d}(x_{n_X}^{(t)}) &\leq \max\{\hat{d}(z_{n_Z+t_1}^{(0)}) + \hat{d}(z_{r_Z+1}^{(t-1)}), \mathtt{DEG}(\ell_X, D^{(t-1)})\} \\ &\leq \max\{d(z_{n_Z+t_1}^{(0)}) + d(z_{r_Z+1}^{(t-1)}), \mathtt{DEG}(\ell_X, D^{(t-1)})\} = d(x_{n_X}^{(t)}).\end{aligned}$$

2. $t_1 \geq 1$. In this case, both of $z_{r_Z}^{(t-1)}$ and $z_{r_Z+1}^{(t-1)}$ are generated by state bits of other registers. From the update function of Z, we have

$$\begin{aligned}z_{r_Z}^{(t-1)} &= z_{n_Z}^{(t-1-n_Z+r_Z)} = z_{n_Z}^{(t_1)} = y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} + \ell_Z(\boldsymbol{s}^{(t_1-1)}), \\ z_{r_Z+1}^{(t-1)} &= z_{n_Z}^{(t-n_Z+r_Z)} = z_{n_Z}^{(t_1+1)} = y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)} + \ell_Z(\boldsymbol{s}^{(t_1)}).\end{aligned}$$

Thus,

$$\begin{aligned}&z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)} \\ &= (y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} + \ell_Z(\boldsymbol{s}^{(t_1-1)})) \cdot z_{r_Z+1}^{(t-1)} \\ &= y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} \cdot z_{r_Z+1}^{(t-1)} + \ell_Z(\boldsymbol{s}^{(t_1-1)}) \cdot z_{r_Z+1}^{(t-1)} \\ &= y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} \cdot (y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)} + \ell_Z(\boldsymbol{s}^{(t_1)})) + \ell_Z(\boldsymbol{s}^{(t_1-1)}) \cdot z_{r_Z+1}^{(t-1)} \\ &= y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} \cdot y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)} + y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} \cdot \ell_Z(\boldsymbol{s}^{(t_1)}) + \ell_Z(\boldsymbol{s}^{(t_1-1)}) \cdot z_{r_Z+1}^{(t-1)}.\end{aligned}$$

Note that $y_{r_Y+1}^{(t_1-1)} = y_{r_Y}^{(t_1)}$, therefore we have

$$z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)} = \ y_{r_Y}^{(t_1-1)} \cdot y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)} + y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)} \cdot \ell_Z(\boldsymbol{s}^{(t_1)}) + \ell_Z(\boldsymbol{s}^{(t_1-1)}) \cdot z_{n_Z}^{(t_1+1)}.$$

We replace the three parts in the above equation by M_1, M_2 and M_3 for short, i.e., $z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)} = \ M_1 + M_2 + M_3$. Naturally, we have

$$\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}) = \max\{\hat{d}(M_1), \hat{d}(M_2), \hat{d}(M_3)\}$$

from Proposition 5. For the first part M_1, according to Proposition 6, we have

$$\begin{aligned}\hat{d}(M_1) &\leq \hat{d}(y_{r_Y}^{(t_1-1)}) + \hat{d}(y_{r_Y}^{(t_1)}) + \hat{d}(y_{r_Y+1}^{(t_1)}), \\ \hat{d}(M_1) &\leq \hat{d}(y_{r_Y}^{(t_1-1)}) + \hat{d}(y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)}), \\ \hat{d}(M_1) &\leq \hat{d}(y_{r_Y}^{(t_1-1)} \cdot y_{r_Y}^{(t_1)}) + \hat{d}(y_{r_Y+1}^{(t_1)}).\end{aligned}$$

Note that $y_{r_Y}^{(t_1-1)} \cdot y_{r_Y}^{(t_1)}$ and $y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)}$ are monomials contained in $z_{n_Z}^{(t_1)}$ and $z_{n_Z}^{(t_1+1)}$, respectively. Therefore, $\hat{d}(y_{r_Y}^{(t_1-1)} \cdot y_{r_Y}^{(t_1)}) \leq \hat{d}(z_{n_Z}^{(t_1)})$ and $\hat{d}(y_{r_Y}^{(t_1)} \cdot y_{r_Y+1}^{(t_1)}) \leq \hat{d}(z_{n_Z}^{(t_1+1)})$ hold from Proposition 5. We can furthermore deduce that

$$\begin{aligned}\hat{d}(M_1) \leq \min\{&\hat{d}(z_{n_Z}^{(t_1)}) + \hat{d}(y_{r_Y+1}^{(t_1)}), \hat{d}(z_{n_Z}^{(t_1+1)}) + \hat{d}(y_{r_Y}^{(t_1-1)}), \\ &\hat{d}(y_{r_Y}^{(t_1-1)}) + \hat{d}(y_{r_Y}^{(t_1)}) + \hat{d}(y_{r_Y+1}^{(t_1)})\} \\ \leq \min\{&d(z_{n_Z}^{(t_1)}) + d(y_{r_Y+1}^{(t_1)}), d(z_{n_Z}^{(t_1+1)}) + d(y_{r_Y}^{(t_1-1)}), \\ &d(y_{r_Y}^{(t_1-1)}) + d(y_{r_Y}^{(t_1)}) + d(y_{r_Y+1}^{(t_1)})\} = d_1,\end{aligned}$$

where d_1 is the estimated degree by numeric mapping as shown in the last subsection. Similarly, for the second and third parts, M_2 and M_3, we have

$$\begin{aligned}\hat{d}(M_2) \leq \hat{d}(y_{r_Y}^{(t_1-1)} \cdot y_{r_Y+1}^{(t_1-1)}) + \hat{d}(\ell_Z(\boldsymbol{s}^{(t_1)}) &\leq \hat{d}(z_{n_Z}^{t_1}) + \hat{d}(\ell_Z(\boldsymbol{s}^{(t_1)})) \\ &\leq d(z_{n_Z}^{t_1}) + \mathtt{DEG}(\ell_Z, D^{(t_1)}) = d_2,\end{aligned}$$

and

$$\hat{d}(M_3) \leq \hat{d}(z_{n_Z}^{(t_1+1)}) + \hat{d}(\ell_Z(\boldsymbol{s}^{(t_1-1)})) \leq d(z_{n_Z}^{(t_1+1)}) + \mathtt{DEG}(\ell_Z, D^{(t_1-1)}) = d_3.$$

Thus, $\hat{d}(z_{r_Z}^{(t-1)} \cdot z_{r_Z+1}^{(t-1)}) \leq \max\{d_1, d_2, d_3\}$ holds. Finally, for the degree of $x_{n_X}^{(t)}$, we have

$$\hat{d}(x_{n_X}^{(t)}) \leq \max\{d_1, d_2, d_3, \mathtt{DEG}(\ell_X, D^{(t-1)})\} = d(x_{n_X}^{(t)}).$$

In conclusion, $\hat{d}(x_{n_X}^{(t)}) \leq d(x_{n_X}^{(t)})$ always holds for any $t \geq 0$. Additional, $\hat{d}(KS^{(t)}) \leq d(KS^{(t)})$ naturally holds because $\hat{d}(s_i^{(t)}) \leq d(s_i^{(t)})$ holds for all $1 \leq i \leq n$ at clock t. □

4.4 Improving the Efficiency of the MILP-aided Degree Evaluation

In this subsection, we introduce a divide-and-conquer strategy and the *maximal polynomial* technique to improve the efficiency of the MILP-aided degree evaluation.

Definition 7 (Maximal Polynomial and Maximal Term)**.** Given a polynomial on n variables as

$$f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{F}_2^n} a_{\boldsymbol{u}}^f \pi_{\boldsymbol{u}}(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \mathbb{M}, \mathbb{M} \subseteq \mathbb{F}_2^n} \pi_{\boldsymbol{u}}(\boldsymbol{x})$$

where the set $\mathbb{M}$ contains all $\boldsymbol{u}$ in $\mathbb{F}_2^n$ such that $a_{\boldsymbol{u}}^f = 1$. For any $\boldsymbol{u} \in \mathbb{M}$, if there is no $\boldsymbol{u'} \in \mathbb{M}$ such that $\pi_{\boldsymbol{u'}}(\boldsymbol{x}) \succ \pi_{\boldsymbol{u}}(\boldsymbol{x})$, then we call $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ is a *maximal term* of $f(\boldsymbol{x})$. Moreover, we define another polynomial as

$$\overline{f}(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u} \in \overline{\mathbb{M}}} \pi_{\boldsymbol{u}}(\boldsymbol{x})$$

where $\overline{\mathbb{M}} = \{\boldsymbol{u} | \boldsymbol{u} \in \mathbb{M}$ and $\pi_{\boldsymbol{u}}(\boldsymbol{x})$ is a *maximal term of* $f(\boldsymbol{x})\} \subseteq \mathbb{M}$. Then we call $\overline{f}(\boldsymbol{x})$ is the *maximal polynomial* of $f(\boldsymbol{x})$.

According to Proposition 5 and 6, it is not difficult to deduce the following property.

Property 2 (Degree Equivalence)**.** Assuming that $\overline{f}$ is the maximal polynomial of a polynomial f. Then the degree of f estimated by division property is equal to that of $\overline{f}$.

When we use division property to estimate the degree of a composite function, Property 2 can help us partly save time. For example, assuming that a polynomial $p(q_0, q_1, q_2, q_3) = q_0q_1q_3 + q_0q_2 + q_1q_3 + q_2q_3 + q_1 + q_2$, where q_i is a function on m variables. Thus we have $\mathbb{M}$ ={(1,1,0,1), (1,0,1,0), (0,1,0,1), (0,0,1,1), (0,1,0,0), (0,0,1,0)}, which means we have to construct five MILP models to estimate the degree of p. However, according to Property 2, it is equivalent to estimate the degree of p's maximal polynomial $\overline{p}(q_0, q_1, q_2, q_3) = q_0q_1q_3 + q_0q_2 + q_2q_3$. Note that $\overline{\mathbb{M}} = \{(1, 1, 0, 1), (1, 0, 1, 0), (0, 0, 1, 1)\}$, thus estimating the degree of $\overline{p}$ instead of p can decrease the number of MILP models from five to three.

Before proposing our improved MILP-aided degree evaluation, we first introduce some notations. For an NFSR-based stream cipher with n-bit internal state, g and f denote the update function and output function, and $\boldsymbol{s}^{(t)}$ denotes the internal state at clock t. Denoted by $\mathcal{M}^d_{\boldsymbol{u}}(t)$ an MILP model constructed using flag technique [WHT+18] where d is a positive integer and $\boldsymbol{u} \in \mathbb{F}_2^n$. It covers t rounds and maximizes $\sum_{i=1}^{m} iv_i$, where $\boldsymbol{iv} = (iv_1, ..., iv_m)$ represents the division property of IV. Moreover, we constrain $\sum_{i=1}^{m} iv_i > d$ and $\boldsymbol{u} \succeq \boldsymbol{u}^{(t)}$ in $\mathcal{M}^d_{\boldsymbol{u}}(t)$ where $\boldsymbol{u}^{(t)}$ represents the division property of $\boldsymbol{s}^{(t)}$. Denoted by $OBJ(\mathcal{M}^d_{\boldsymbol{u}}(t))$ the optimized solution of $\mathcal{M}^d_{\boldsymbol{u}}(t)$ obtained by MILP solvers. If $\mathcal{M}^d_{\boldsymbol{u}}(t)$ is infeasible, we assign 0 to $OBJ(\mathcal{M}^d_{\boldsymbol{u}}(t))$. In addition, we define $f_i(\boldsymbol{x}) = f(\underbrace{g \circ g \circ \cdots \circ g}_{i}(\boldsymbol{x}))$.

Now we introduce a divide-and-conquer strategy based on maximal polynomial technique to further speed up the MILP-aided degree evaluation, which is described as follows:

1. For an r-round initialization, split r to the former part t_1 and the latter part t_2 as $r = t_1 + t_2$, and compute the maximal polynomial of $f_{t_2}(\boldsymbol{s}^{(t_1)})$ as $\overline{f_{t_2}}(\boldsymbol{s}^{(t_1)}) = \bigoplus_{\boldsymbol{u}\in\overline{\mathbb{M}}} \pi_{\boldsymbol{u}}(\boldsymbol{s}^{(t_1)})$. In addition, initialize d to be 0.
2. Choose a $\boldsymbol{u}$ from $\overline{\mathbb{M}}$ and construct model $\mathcal{M}^d_{\boldsymbol{u}}(t_1)$. If $OBJ(\mathcal{M}^d_{\boldsymbol{u}}(t_1))$ is greater than d then we update d by $OBJ(\mathcal{M}^d_{\boldsymbol{u}}(t_1))$. Remove $\boldsymbol{u}$ from $\overline{\mathbb{M}}$.
3. Repeat step 2 until $\overline{\mathbb{M}}$ is empty. Then we regard d as the upper bound on the algebraic degree of $f_r(\mathrm{IV})$.

The scale of MILP model is decreased from r to $t_1 = r - t_2$ by the divide-and-conquer strategy, meanwhile the number of MILP model is pruned from $|\overline{\mathbb{M}}|$ to $|\mathbb{M}|$ by the maximal polynomial technique. In our experiments, this strategy can greatly improve the efficiency of the MILP-aided degree evaluation, especially for large rounds. For example, in a laptop with 8GB RAM and i7-8550U CPU, the traditional method based on division property cannot return a result in 2.5 hours for 788-round Trivium. However, it takes no more than 20 minutes to return the degree using the divide-and-conquer strategy. The framework of estimating the degree for a stream cipher up to r rounds is described in Algorithm 2. Note that the number of models is $|\overline{\mathbb{M}}|$ in step 2, which is entirely depends on the selection of t_2. We will illustrate how to achieve a trade-off on determining t_2 in the next subsection.

4.5 Experiments on Trivium and Kreyvium

In this subsection, we apply Algorithm 2 to evaluate the upper bounds on the degree of Trivium and Kreyvium, and compare the results with [Liu17] to illustrate the advantage of degree evaluations using division property. We will omit the details of the two ciphers, and one can refer to [CP08, CCF+16] for more details.

First of all, we need to choose an appropriate split on the round. In our experiments, we calculate the precise ANFs of $f_r(\boldsymbol{s}^{(0)})$ when $r \leq 250$ both for Trivium and Kreyvium. Table 2 lists the number of monomials for several rounds, where $|\mathbb{M}|$ and $|\overline{\mathbb{M}}|$ respectively denote the number of monomials of $f_r(\boldsymbol{s}^{(0)})$ and $\overline{f_r}(\boldsymbol{s}^{(0)})$ for $215 \leq r \leq 235$. Comparing the number of monomials for different rounds, we set $t_2 = 225$ in the divide-and-conquer strategy for Trivium and Kreyvium. Hence, we can use Algorithm 2 to estimate the algebraic degree of the two ciphers, where the degree of rounds 1-225 are obtained by precisely computing corresponding ANFs and remaining rounds are obtained by division property.

Table 2: The number of monomials in the output bit and its maximal polynomial of Trivium and Kreyvium with round from 215 to 235.

#Round		215	216-218	219-221	222-224	**225**	226	227-228	229	230-234	235
Trivium	$\lvert\mathbb{M}\rvert$	192	195	195	196	**197**	233	294	315	345	348
	$\lvert\overline{\mathbb{M}}\rvert$	182	183	183	185	**185**	220	278	299	328	331
Kreyvium	$\lvert\mathbb{M}\rvert$	277	282	283	285	**287**	342	436	468	516	520
	$\lvert\overline{\mathbb{M}}\rvert$	225	227	227	229	**231**	277	353	374	403	407

Algorithm 2 Estimate the upper bounds on degree of a stream cipher up to r rounds

```
procedure FastDegEvaForNFSR(round $r$, round $t_2$)
    $\mathbb{D} \leftarrow$ an empty list; /* This list is used to store degrees */
    $\overline{\mathbb{M}} \leftarrow$ an empty set;
    $t_1 \leftarrow n - t_2$;
    $(\mathbb{D}, \overline{\mathbb{M}}) \leftarrow$ CalExaDegAndMaxPoly($\mathbb{D}, t_2$);
    for $i$ from 1 to $t_1$ do
        /* Calculate degrees of the remianing rounds by division property */
        $d \leftarrow 0$;
        while $\overline{\mathbb{M}} \neq \emptyset$ do
            Choose an $\boldsymbol{u}$ from $\overline{\mathbb{M}}$ and construct model $\mathcal{M}_{\boldsymbol{u}}^{d}(i)$;
            if $OBJ(\mathcal{M}_{\boldsymbol{u}}^{d}(i)) > d$ then
                $d \leftarrow OBJ(\mathcal{M}_{\boldsymbol{u}}^{d}(i))$;
            end if
            Remove $\boldsymbol{u}$ from $\overline{\mathbb{M}}$;
        end while
        $\mathbb{D}$.append($d$);
    end for
    return $\mathbb{D}$;
end procedure
procedure CalExaDegAndMaxPoly(list $\mathbb{D}$, round $t_2$)
    /* Calculate the exact algebraic degree of the cipher up to $t_2$ and $\overline{\mathbb{M}}$ of $\overline{f_{t_2}}(\boldsymbol{s}^{(0)})$ */
    for $i$ from 1 to $t_2$ do
        Calculate the ANF of $f_i$(IV);
        $\mathbb{D}$.append(deg($f_i$(IV)));
    end for
    Calculate the maximal polynomial $\overline{f_{t_2}}(\boldsymbol{s}^{(0)})$ and the corresponding set $\overline{\mathbb{M}}$;
    return $(\mathbb{D}, \overline{\mathbb{M}})$
end procedure
```

By calling Algorithm 2, the results show that the longest round, where the bound cannot reach the full degree, is 839 for Trivium and 897 for Kreyvium. Whereas, the corresponding rounds obtained in [Liu17] are 793 and 862, meanwhile the estimated degrees of 793-round Trivium and 862-round Kreyvium using division property are bounded by 65 and 108, respectively. In addition, our results are consistent with the longest zero-sum distinguishers where cubes contain full IV bits in [TIHM17]. Hence, compared with [Liu17], division property can get more accurate bounds. Moreover, this gap becomes more and more distinct with the round increasing. The comparisons of the estimated degrees[3] of the two ciphers using division property and numeric mapping are illustrated in Figure 3.

5 Conclusion and Discussion

There are several researches on algebraic degree evaluations of symmetric ciphers. Specifically, Boura and Canteaut [BC13] and Carlet [Car20] proposed formulas to calculate the upper bounds on the degree of SPN ciphers. Liu [Liu17] proposed *numeric mapping* technique to estimate the algebraic degree of NFSR-based ciphers. Besides, division property can also be utilized to estimate the upper bounds on the degree of all kinds of symmetric ciphers. However, there is no related work to illustrate the relationships of these methods. In this paper, we focused on the relationships between division property based degree evaluation and other methods and concluded for the first time that division property is actually the optimal one among these methods in terms of the accuracy.

[3] More specific degrees can be referred to the full version at https://eprint.iacr.org/2021/175.pdf.

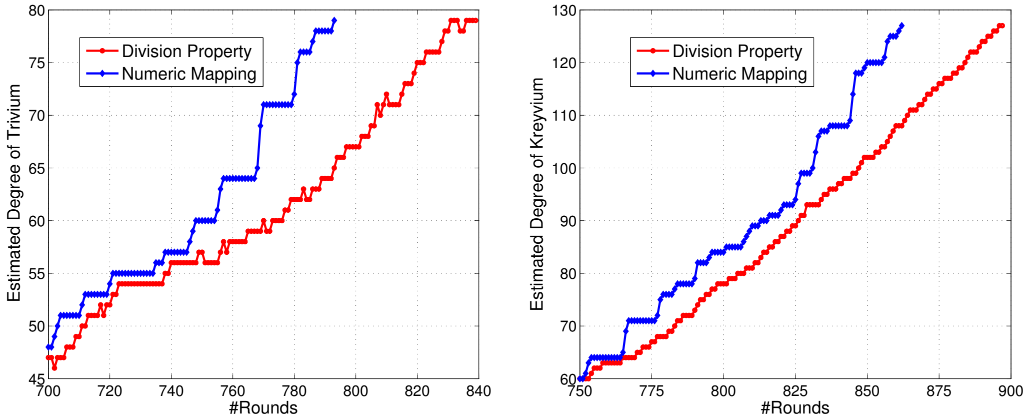

Figure 3: Estimated degrees of Trivium and Kreyvium derived by division property and numeric mapping.

Besides, we need state that 3SBDP as well as monomial prediction can be used to compute the exact degree of ciphers. In order to avoid enumerating all division trails, the authors in [HLLT20] provide an idea to explore the exact algebraic degree by evaluating both the lower and upper bounds on the degree, i.e., the exact degree is determined if the two bounds are equal. Meanwhile, they gave the concept of *inconsistent sub-trails* and proposed the *trail extension* technique to avoid inconsistent sub-trails to improve the searching efficiency. Thanks to this idea, the exact algebraic degree evaluation can be achieved using 3SBDP for some block ciphers (PRESENT, GIFT, SKINNY-64 and AES in [HLLT20]). In addition, the authors in [HSWW20] used the H-representation of convex hull to describe the monomial trail propagation of the update function of Trivium cipher instead of modeling the specific AND, COPY and XOR operations. This tip was used to improve the searching efficiency and they could obtain the exact degree of Trivium up to 834 rounds. It is worth noticing that the gap between the upper bounds estimated by two-subset bit-based division property and the exact degrees given by [HSWW20] is no more than 1 and the upper bounds are actually equal to the exact degrees for most cases. Thus, if we only require an overview of the degree with limited effort, two-subset division property would be a better choice than Boura and Canteaut's formula, Carlet's formula and the numeric mapping method. However, 3SBDP or monomial prediction combined with a more carefully analysis like in [HLLT20] or [HSWW20] would be preferable when we want to explore the exact algebraic degree.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No.61802119) and the Application Foundation Frontier Project of Wuhan Science and Technology Bureau (Grant No.2020010601012189).

References

[BC13] Christina Boura and Anne Canteaut. On the influence of the algebraic degree of F^{-1} on the algebraic degree of $G \circ F$. *IEEE Trans. Inf. Theory*, 59(1):691–702, 2013.

[BC16] Christina Boura and Anne Canteaut. Another view of the division property. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 654–682. Springer, 2016.

[BCC11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and *Luffa*. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.

[BDP09] G.M. Bertoni, Joan Daemen, and Michael Peeters. Keccak sponge function family main document. *Submission to NIST (Round 2)*, 3, 01 2009.

[BKL+17] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2017.

[Car20] Claude Carlet. Graph indicators of vectorial functions and bounds on the algebraic degree of composite functions. *IEEE Trans. Inf. Theory*, 66(12):7702–7716, 2020.

[CCF+16] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrède Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 313–333. Springer, 2016.

[CJF+16] Tingting Cui, Keting Jia, Kai Fu, Shiyao Chen, and Meiqin Wang. New automatic search tool for impossible differentials and zero-correlation linear approximations. *IACR Cryptology ePrint Archive*, 2016:689, 2016.

[CM03] Nicolas Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2003.

[Cou03] Nicolas Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer, 2003.

[CP08] Christophe De Cannière and Bart Preneel. Trivium. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 244–266. Springer, 2008.

[CV02] Anne Canteaut and Marion Videau. Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 518–533. Springer, 2002.

[DL11] Ming Duan and Xuejia Lai. Improved zero-sum distinguisher for full round Keccak-f permutation. *IACR Cryptology ePrint Archive*, 2011:23, 2011.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 278–299, 2009.

[HLLT20] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Lower bounds on the degree of block ciphers. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 537–566. Springer, 2020.

[HLM$^+$20] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128AEAD. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 466–495. Springer, 2020.

[HSWW20] Kai Hu, Siwei Sun, Meiqin Wang, and Qingju Wang. An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 446–476. Springer, 2020.

[Knu94] Lars R. Knudsen. Truncated and higher order differentials. In Bart Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.

[KW02] Lars R. Knudsen and David A. Wagner. Integral cryptanalysis. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer, 2002.

[Lai94] Xuejia Lai. Higher order derivatives and differential cryptanalysis. In *Symposium on Communication, Coding and Cryptography, in Honor of J. L. Massey on the Occasion of His 60'th Birthday*. 1994.

[Liu17] Meicheng Liu. Degree evaluation of NFSR-based cryptosystems. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 227–249. Springer, 2017.

[Mat97] Mitsuru Matsui. New block encryption algorithm MISTY. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 1997.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

[SHS$^+$13] Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, and Peng Wang. Automatic security evaluation of block ciphers with s-bp structures against related-key differential attacks. In Dongdai Lin, Shouhuai Xu, and Moti Yung, editors, *Information Security and Cryptology - 9th International Conference, Inscrypt 2013, Guangzhou, China, November 27-30, 2013, Revised Selected Papers*, volume 8567 of *Lecture Notes in Computer Science*, pages 39–51. Springer, 2013.

[SHW$^+$14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 158–178. Springer, 2014.

[SHZ$^+$17] Bing Sun, Xin Hai, Wenyu Zhang, Lei Cheng, and Zhichao Yang. New observation on division property. *SCIENCE CHINA Information Sciences*, 60(9):98102, 2017.

[ST17] Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 185–215, 2017.

[SWW17] Ling Sun, Wei Wang, and Meiqin Wang. Automatic search of bit-based division property for ARX ciphers and word-based division property. In Tsuyoshi Takagi and Thomas Peyrin, editors, *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*, volume 10624 of *Lecture Notes in Computer Science*, pages 128–157. Springer, 2017.

[SWW20] Ling Sun, Wei Wang, and Meiqin Wang. Milp-aided bit-based division property for primitives with non-bit-permutation linear layers. *IET Inf. Secur.*, 14(1):12–20, 2020.

[TIHM17] Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 250–279, 2017.

[TM16] Yosuke Todo and Masakatu Morii. Bit-based division property and application to simon family. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 357–377. Springer, 2016.

[Tod15a] Yosuke Todo. Integral cryptanalysis on full MISTY1. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 413–432. Springer, 2015.

[Tod15b] Yosuke Todo. Structural evaluation by generalized integral property. In Elisabeth Oswald and Marc Fischlin, editors, *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*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314. Springer, 2015.

[WHG$^+$19a] SenPeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. MILP-aided method of searching division property using three subsets and applications. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 398–427. Springer, 2019.

[WHG$^+$19b] SenPeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. A practical method to recover exact superpoly in cube attack. *IACR Cryptol. ePrint Arch.*, 2019:259, 2019.

[WHT$^+$18] Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier. Improved division property based cube attacks exploiting algebraic properties of superpoly. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 275–305. Springer, 2018.

[WLV$^+$14] Qingju Wang, Zhiqiang Liu, Kerem Varici, Yu Sasaki, Vincent Rijmen, and Yosuke Todo. Cryptanalysis of reduced-round SIMON32 and SIMON48. In Willi Meier and Debdeep Mukhopadhyay, editors, *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 2014.

[XZBL16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pages 648–678, 2016.

[ZDY$^+$19] Wentao Zhang, Tianyou Ding, Bohan Yang, Zhenzhen Bao, Zejun Xiang, Fulei Ji, and Xuefeng Zhao. KNOT: Algorithm Specifications and Supporting Document. *Submission to NIST (Round 2)*, 2019. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/knot-spec-round.pdf.