Coefficient Grouping for Complex Affine Layers

Fukang Liu[1], Lorenzo Grassi[2], Clémence Bouvier[3,4], Willi Meier[5], Takanori Isobe[6,7]

[1] Tokyo Institute of Technology, Tokyo, Japan,
liu.f.ad@m.titech.ac.jp
[2] Ruhr University Bochum, Bochum, Germany
Lorenzo.Grassi@ruhr-uni-bochum.de
[3] Sorbonne University, Paris, France
[4] Inria, Paris, France
clemence.bouvier@inria.fr
[5] FHNW, Windisch, Switzerland
willi.meier@fhnw.ch
[6] University of Hyogo, Hyogo, Japan
[7] NICT, Tokyo, Japan
takanori.isobe@ai.u-hyogo.ac.jp

Abstract. Designing symmetric-key primitives for applications in Fully Homomorphic Encryption (FHE) has become important to address the issue of the ciphertext expansion. In such a context, cryptographic primitives with a low-AND-depth decryption circuit are desired. Consequently, quadratic nonlinear functions are commonly used in these primitives, including the well-known χ function over $\mathbb{F}_2^n$ and the power map over a large finite field $\mathbb{F}_{p^n}$. In this work, we study the growth of the algebraic degree for an SPN cipher over $\mathbb{F}_{2^n}^m$, whose S-box is defined as the combination of a power map $x \mapsto x^{2^d+1}$ and an $\mathbb{F}_2$-linearized affine polynomial $x \mapsto c_0 + \sum_{i=1}^{w} c_i x^{2^{h_i}}$ where $c_1, \ldots, c_w \neq 0$. Specifically, motivated by the fact that the original coefficient grouping technique published at EUROCRYPT 2023 becomes less efficient for $w > 1$, we develop a variant technique that can efficiently work for arbitrary w. With this new technique to study the upper bound of the algebraic degree, we answer the following questions from a theoretic perspective:

1. can the algebraic degree increase exponentially when $w = 1$?
2. what is the influence of w, d and $(h_1, \ldots, h_w)$ on the growth of the algebraic degree?

Based on this, we show (i) how to efficiently find $(h_1, \ldots, h_w)$ to achieve the exponential growth of the algebraic degree and (ii) how to efficiently compute the upper bound of the algebraic degree for arbitrary $(h_1, \ldots, h_w)$. Therefore, we expect that these results can further advance the understanding of the design and analysis of such primitives.

Keywords: Degree evaluation – Coefficient grouping technique – Finite fields

1 Introduction

Since the birth of the first Fully Homomorphic Encryption (FHE) scheme by Gentry [24], various improved schemes [8, 10, 11, 12, 13, 26, 38] have been proposed to push the FHE technique closer to practice. Especially, the transciphering framework [38] is an important technique to address the issue of computational overload and the ciphertext expansion, which suggests to use a symmetric-key primitive to encrypt the data on the client side and then to perform the homomorphic decryption and additional homomorphic operations on the server side. It is soon observed that standard symmetric-key primitives may be not the best in such a framework due to the high AND-depth, e.g., the widely-deployed and well-studied encryption algorithm AES is inefficient [25]. Hence, this opens a new direction to design FHE-friendly symmetric-key primitives and many proposals have emerged, currently including Kreyvium [9], LowMC [3], FiLIP [37], RASTA [19], DASTA [28], MASTA [27], PASTA [20], HERA [13] Chaghri [4], and Elisabeth [15].

As one may expect, these new designs differ from the traditional ones in several aspects. As a result, new dedicated attacks (especially based on algebraic techniques) have been recently developed for breaking these primitives. Especially, it is found that for some proposals, there are undesirable or fatal weaknesses in the used components, which can even allow a complete break of the cipher [1, 5, 17, 18, 22, 23, 30, 31, 32, 33, 34, 35]. Despite these successful attacks, it is still challenging to further understand their security after they are patched to resist against the corresponding attacks.

1.1 Our target: the studied constructions

In this work, we consider SPN ciphers over $\mathbb{F}_{2^n}^m$ for $n \geq 3$ and $m \geq 1$ as illustrated in Fig. 1, where

- the invertible S-box is defined as the composition of a power map $S(x) = x^{2^d+1}$ for $d \geq 1$ and of an $\mathbb{F}_2$-linearized affine polynomial $B(x) = c_0 + \sum_{i=1}^{w} c_i x^{2^{h_i}}$ where $c_1, \ldots, c_w \neq 0$. In the following, we call $B(x)$ the affine layer and w **the density of** $B(x)$;
- the linear transformation is defined via the multiplication with an arbitrary invertible matrix $M \in \mathbb{F}_{2^n}^{m \times m}$.

Examples include MiMC [2] and Chaghri. MiMC is a special case of this construction for which $m = 1, B(x) = x$ and M is the identity matrix of size 1×1, i.e. $M = (1)$. For the original parameter of Chaghri, we have $m = 3, B(x) = c_0 + c_1 x^{2^{h_1}}$. Just to clarify that the designers of Chaghri have revised their design by choosing $w = 3$. For convenience, in the following, Chaghri refers to the original design Chaghri where $w = 1$. If we instead consider $S(x) = x^{2^n-2}$, then the MPC-friendly cipher Rain [21] and the well-known block cipher AES also follow this SPN structure.

Although M can be any invertible matrix, to keep the number of secure rounds as small as possible, it is better to choose an MDS matrix. The reason

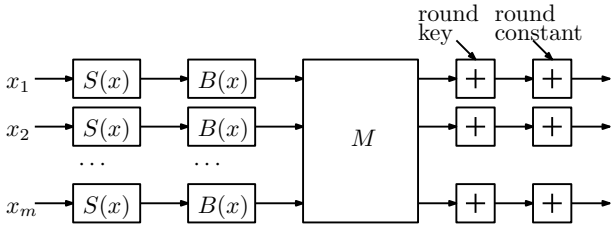

Fig. 1: The round function of SPN ciphers over $\mathbb{F}_{2^n}^m$

to only consider this nonlinear power map $S(x) = x^{2^d+1}$ is that its algebraic degree is the lowest, i.e. 2, and that it has good cryptographic properties [39], i.e. resistance against differential and linear attacks.

1.2 Related works

Higher-order differential (HD) attacks [29] form a prominent class of attacks exploiting the low algebraic degree of a nonlinear transformation. Given a function F of degree $\deg(F)$ over $\mathbb{F}_2^n$, the attack exploits the fact that $\bigoplus_{x\in\mathcal{V}} F(x) = 0$ for each affine subspace $\mathcal{V} \subseteq \mathbb{F}_2^n$ whose dimension $\dim(\mathcal{V})$ satisfies $\dim(V) \geq \deg(F) + 1$. This version of the attack has been extended to a large finite field of any characteristics at CRYPTO 2020 [5], and it was refined for the extension field $\mathbb{F}_{2^n}$ at ASIACRYPT 2020 [23] via the link between $\mathbb{F}_{2^n}$ and $\mathbb{F}_2^n$. The two techniques are successfully applied to MiMC-based constructions. E.g., for MiMC over $\mathbb{F}_{2^n}$ (an iterated Even-Mansour cipher instantiated via $x \mapsto x^3$), it is found that the algebraic degree increases linearly rather than exponentially.

Since evaluating the degree is essential to mount the HD attack, several follow-up works have appeared. In [14], the authors also studied the algebraic degree for SPN ciphers over $\mathbb{F}_{2^n}^m$ where a general nonlinear map $x \mapsto \sum_{i=0}^{j} c_i' x^i$ over $\mathbb{F}_{2^n}$ and a binary matrix over $\mathbb{F}_2^{nm\times nm}$ are used as the nonlinear and linear components, respectively. (This binary matrix can be decomposed as the combination of m $\mathbb{F}_2$-linearized polynomials $(B_1, \ldots, B_m)$ over $\mathbb{F}_{2^n}$ and a matrix over $\mathbb{F}_{2^n}^{m\times m}$.) In [7], MiMC over $\mathbb{F}_{2^n}$ with $S(x) = x^j$ was studied. In both works, the underlying idea to bound the algebraic degree is similar to [23] where the upper bound of the algebraic degree is computed as $\lfloor \log_2(j^r + 1) \rfloor$.

However, the techniques in [7, 14, 23] are basically inefficient when targeting the construction proposed in Fig. 1 if $(2^d + 1) \cdot 2^{\max\{h_1,\ldots,h_w\}}$ (equivalently, $\max\{d, d \cdot \max\{h_1, \ldots, h_w\}\}$) is large and n is small. As a concrete example, the bounds obtained with these techniques do *not* violate the designers' claim for Chaghri that the algebraic degree increases exponentially. This fact has been recently disproved via the new "coefficient grouping technique" [30]. Compared with the techniques proposed in [7, 14, 23], the coefficient grouping technique allows to efficiently handle the case when $((2^d + 1) \cdot 2^{\max\{h_1,\ldots,h_w\}})^r > 2^n$ for the case $w = 1$. With this new technique, it is shown that the algebraic degree of Chaghri indeed increases linearly [30].

As a parallel work to the coefficient grouping technique, the division property [41] has been also extended to $\mathbb{F}_{2^n}$ and successfully applied to MiMC-based constructions [16]. However, this method completely relies on a blackbox solver and it has the feature to model the heavy monomial transitions round by round as in the division property [41]. Hence, its performance for complex affine layers is questionable. More importantly, with this technique, we seem to have little insight into the main factors to influence the growth of the algebraic degree.

1.3 Our contributions

Problems to be addressed. One notable feature of the coefficient grouping technique for the case $w = 1$ is that the problem to upper bound the algebraic degree can be reduced to a well-structured optimization problem. However, while it works quite efficiently [36] and accurately for the case $w = 1$, its performance and accuracy become low for $w > 1$, though it has been used to patch Chaghri for $w > 1$. For these reasons, we are interested to answer the following problems:

Problem 1: For $w = 1$, can we prove that the algebraic degree will *never* increase exponentially for any d, even for the first few rounds? Note that with the techniques in [7, 14, 23], we cannot conclude whether the algebraic degree will increase exponentially at the first few rounds if d is large.

Problem 2: Chaghri has been revised by replacing $w = 1$ with $w = 3$. However, we are still lacking theories regarding the influence of w on the growth of the algebraic degree. Can we build such a theory?

Problem 3: Are there equivalent $(h_1, \ldots, h_w)$ from the point of view of the growth of the algebraic degree?

Problem 4: It has been demonstrated in [30] that using $B(x)$ with $w > 1$ allows to achieve the exponential growth of the algebraic degree. However, it is an open problem to efficiently find concrete parameter sets of $(h_1, \ldots, h_w)$ for reaching this goal. We emphasize that the method in [30] is inefficient and the accuracy is sacrificed for the efficiency.

Problem 5: Can we efficiently provide meaningful upper bounds of the growth of the degree for a cipher that uses arbitrary $B(x)$? Note that no result about upper bounds when $w > 1$ has been proposed in [30].

Our results. To address the above problems, we first deeply analyse the set of all monomials that will possibly appear in the polynomial representation of the internal state after any rounds. Our analysis shows that such a set is well-structured, which directly allows us to systematically study the above problems. Based on our theoretical analysis, we obtain the following results:

Answer 1: When $w = 1$, for any (n, d), the algebraic degree will *never* increase exponentially after the 1st round and it is always below the quadratic growth.

Answer 2: Necessary conditions on w and the concrete parameter $(h_1, \ldots, h_w)$ are derived to ensure the exponential growth of the algebraic degree. Especially, we found that for $w = 2, 3, 4$, the algebraic degree can *never* increase exponentially at the 4th, 7th and 10th round for any $(h_1, \ldots, h_w)$, respectively.

Answer 3: With our new technique, some equivalent $(h_1, \ldots, h_w)$ are derived. With the equivalent relations, we only need to consider about $\frac{1}{w} \cdot \binom{n}{w-1}$ possible candidates of $(h_1, \ldots, h_w)$ when aiming to find one that can achieve the exponential growth of the algebraic degree.

Answer 4: We build a very light model to check whether the exponential growth can be achieved for a concrete $(h_1, \ldots, h_w)$. It is found that this problem can be reduced to the problem whether it is possible to select 2^r different numbers from $r + 1$ sets under some special conditions.

Answer 5: For arbitrary $B(x)$, we show how to reduce the problem to upper bound the algebraic degree to a well-structured optimization problem. However, the well-structured problem is constructed at the sacrifice of using relaxed constraints. Hence, the upper bound may be not tight, but it is still useful for many cases as shown in our practical tests.

Comparison with previous works. Similar to the analysis proposed in [7,14,23,30], our work exploits the link between the degree of the functions over $\mathbb{F}_{2^n}$ and over $\mathbb{F}_2^n$, making it different from e.g. the well-known technique proposed in [6]. Still, our analysis differs from [7,14,23], since such methods become completely inefficient when $\max\{d, d \cdot \max\{h_1, \ldots, h_w\}\}$ is large.

Compared with the division property [16,41], our technique does not require to model the heavy propagation of monomials round by round, i.e. we can directly identify the relations between the polynomial representation of the input and polynomial representation of the output after any rounds, where the propagation of monomials in the middle rounds are simply captured by some conditions.

Similar to [30], our strategy aims to capture the properties of the polynomial representation for determining the growth of the algebraic degree. However, while the original coefficient grouping technique and the technique proposed in this paper almost overlap for the case $w = 1$, the former one becomes completely inefficient for $w > 1$, as we are going to discuss in details in the following. As a concrete example, in [30], authors discuss a possible way to choose the affine layer in Chaghri for guaranteeing the exponential growth of the degree. In this paper, we show that the suggested choice is *not* optimal, in the sense that the fastest growth of the algebraic degree may *not* be achieved. This is related to the fact that the technique proposed in [30] requires several approximation for being efficient in practice, with a consequent lost in precision. By exploiting our new technique proposed in this paper, we show how to choose the affine layer for ensuring the fastest growth of the algebraic degree.

Outline of this paper. In Sect. 2, we briefly recall the higher-order differential attacks over $\mathbb{F}_{2^n}$ and the original coefficient grouping technique [30]. In Sect.

3, we build the most important theory for complex affine layers, which is the basis of all the results in this paper. In Sect. 4, we aim to answer Problem 1, 2 and 3. In Sect. 5 and 6, we try to solve Problem 4 and 5 in the univariate setting respectively, and will report some practical applications and experiments[8]. Finally, we mention some problems worth further study in Sect. 7.

2 Preliminaries

2.1 Notation

In this paper, variables and/or parameters are denoted via lower letters and/or Greek letters, while functions and/or arrays are denoted via capital letters. We use the calligraphic notation for denoting sets. Given two integers a and b, we use the notation $a|b$ to denote the fact that a divides b, while we denote the rest of the division as $a\%b := a \bmod b$. $[a, b]$ is a set of integers i satisfying $a \leq i \leq b$. $|\mathcal{S}|$ denotes the size of the set $\mathcal{S}$. $\mathrm{Hw}(a)$ is the hamming weight of $a \in \mathbb{N}$, i.e. the number of 1 in its binary representation.

Furthermore, similar to [30], we introduce the function $M_n : \mathbb{N} \to \mathbb{N}$ defined as follows:

$$M_n(x) := \begin{cases} 2^n - 1 & \text{if } 2^n - 1 | x \text{ and } x \geq 2^n - 1, \\ x\%(2^n - 1) & \text{otherwise.} \end{cases}$$

The symbol $\binom{a}{\leq b}$ where $a \geq b$ is defined as:

$$\binom{a}{\leq b} := \begin{cases} \binom{a}{0} + \cdots + \binom{a}{b} & \text{if } b \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

By definition of $M_n(x)$, we get the following lemma:

Lemma 1. *For each $x_1, x_2, i, x \in \mathbb{N}$, we have:*

1. $M_n(x_1 + x_2) = M_n(M_n(x_1) + M_n(x_2))$;
2. $M_n(2^i) = 2^{i\%n}$;
3. $M_n(2^i \cdot x) = M_n(2^{i\%n} \cdot M_n(x))$.

The definition of $M_n(x)$ is mainly to capture the following facts:

$$\begin{cases} x^{p^n} = x \ \forall x \in \mathbb{F}_{p^n}, \\ x^{p^n - 1} = 1 \ \forall x \in \mathbb{F}_{p^n} \text{ and } x \neq 0, \\ (x + y)^{p^i} = x^{p^i} + y^{p^i} \ \forall x, y \in \mathbb{F}_{p^n}, i \in \mathbb{N}. \end{cases}$$

[8] The code is available at `https://github.com/LFKOKAMI/degreeEva`.

2.2 Algebraic Degree for Polynomials over $\mathbb{F}_{2^n}^m$

To mount the HD attack, the first step is to find the upper bound of the algebraic degree of the polynomial representation of the output in terms of the input. In order to do this, in the following we heavily exploit the link between the algebraic degree of the functions over $\mathbb{F}_{2^n}$ and over $\mathbb{F}_2^n$. We recall that even if such a link is well established, it has not been widely used for attacking ciphers until recently [7, 14, 23, 30].

Given a polynomial $F \in \mathbb{F}_{2^n}[x_1, \ldots, x_t]$, in this paper we call it (i) a *univariate polynomial* if $t = 1$, and (ii) *multivariate polynomial* if $t \geq 2$. As is well-known, e.g. [7, 14, 23, 30], the algebraic degree of polynomials over $\mathbb{F}_{2^n}$ is defined as follows:

- for a *univariate* polynomial $F = \sum_{i=0}^{2^n-1} \mu_i x^i \in \mathbb{F}_{2^n}[x]$, its algebraic degree δ_F is defined as

  $$\delta_F = \max\{\mathrm{Hw}(i) \mid i \in [0, 2^n - 1] \text{ and } \mu_i \neq 0\}.$$

  This definition is mainly based on how to convert a function over $\mathbb{F}_{2^n}$ into a vectorial function over $\mathbb{F}_2^n$, e.g., as shown in Chapter 6 in [40].
- for a *multivariate* polynomial

  $$F = \sum_{i_1=0}^{2^n-1} \sum_{i_2=0}^{2^n-1} \cdots \sum_{i_t=0}^{2^n-1} \mu_{i_1,i_2,\ldots,i_t} x_1^{i_1} \cdots x_t^{i_t} \in \mathbb{F}_{2^n}[x_1, \ldots, x_t],$$

  its algebraic degree δ_F is defined as

  $$\delta_F = \max \left\{ \sum_{j=1}^{t} \mathrm{Hw}(i_j) \mid i_j \in [0, 2^n - 1] \text{ and } \mu_{i_1,i_2,\ldots,i_t} \neq 0 \right\}.$$

To mount the HD attack on ciphers over $\mathbb{F}_{2^n}$, it is paramount to know the growth of δ_F. E.g., if δ_F is smaller than $t \cdot n$ when considering polynomials in t variables, it is possible to mount a HD attack with time and data complexity 2^{δ_F+1}.

2.3 The Coefficient Grouping Technique

For our studied SPN ciphers, the coefficient grouping technique can be classified into two settings: the univariate setting and the multivariate setting. This technique is very efficient for $w = 1$, i.e., $B(x) = c_0 + c_1 x^{2^{h_1}}$, as stated below.

The univariate setting. In this case, the attacker considers $(x_1, x_2, \ldots, x_m)$ of the form $x_i = \lambda_{1,i} \cdot z + \lambda_{2,i}$ for each $i \in \{1, 2, \ldots, m\}$, where $\lambda_{1,i}, \lambda_{2,i} \in \mathbb{F}_{2^n}$ are randomly chosen constants and $z \in \mathbb{F}_{2^n}$ is the variable. Computing the upper bound for the algebraic degree after r rounds reduces to solving the following optimization problem, which we denote by **OP-1**:

$$\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=0}^{n-1} 2^i a_i\right)\right),$$

$$\text{subject to } 0 \le a_i \le \nu_{r,i} \text{ for } i \in [0, n-1],$$

where the parameter $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0}) \in \mathbb{N}^n$ is computed with the following recursive relation:

$$\begin{cases} \nu_{0,0} = 1, \nu_{0,i} = 0 & \text{for } i \in [1, n-1], \\ \nu_{j,i} = \nu_{j-1,(i-d-h_1)\%n} + \nu_{j-1,(i-h_1)\%n} & \text{for } j \in [1, r], i \in [0, n-1]. \end{cases} \tag{1}$$

The multivariate setting. In this case, the attacker considers such $(x_1, x_2, \ldots, x_m)$ that $x_i = \sum_{j=1}^{t} \lambda_{j,i} z_j$ for each $i \in \{1, 2, \ldots, m\}$, where $\lambda_{j,i} \in \mathbb{F}_{2^n}$ are randomly chosen constants and $(z_1, \ldots, z_t) \in \mathbb{F}_{2^n}^t$ are t variables. Computing the upper bound after r rounds reduces to solving the following problem denoted by **OP-t**:

$$\text{maximize} \sum_{j=1}^{t} \text{Hw}\left(M_n\left(\sum_{i=0}^{n-1} 2^i a_{j,i}\right)\right),$$

$$\text{subject to } 0 \le \sum_{j=1}^{t} a_{j,i} \le \nu_{r,i} \text{ for } i \in [0, n-1],$$

where the parameter $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0})$ is still computed with the recursive relation specified in Equation 1. Note that **OP-t** ($t \ge 1$) can always be solved in time $\mathcal{O}(n)$ via the reduction algorithm in [36].

Weaknesses and open problems. When $w = 1$, there is always only one parameter vector ν_r and it can be computed in time $\mathcal{O}(n)$ with Equation 1. This implies that we only need to solve the optimization problem once. However, for $w > 1$, after r rounds, there will be a set of parameter vectors denoted by $\mathcal{P}_r$, i.e. $\nu_r \in \mathcal{P}_r$ where $\mathcal{P}_0 = \{(0, \ldots, 0, 1)\}$. Moreover, $\mathcal{P}_r$ ($r \ge 1$) is updated as follows:

$$\begin{aligned} \mathcal{P}_r = \{(e_{n-1}, \ldots, e_0) \in \mathbb{N}^n \mid \forall \nu_{r-1}, \nu'_{r-1} \in \mathcal{P}_{r-1}, \forall j \in [1, w], \forall i \in [0, n-1], \\ e_i = \nu_{r-1,(i-d-h_j)\%n} + \nu'_{r-1,(i-h_j)\%n}\}. \end{aligned} \tag{2}$$

We point out that some redundant elements can be removed when constructing $\mathcal{P}_r$ via Equation 2 (note that $\nu''_r = (v''_{r,n-1}, \ldots, v''_{r,0})$ is *redundant* if $\exists \nu_r \in \mathcal{P}_r$ such that $\nu_{r,i} \ge \nu''_{r,i}$ holds for $\forall i \in [0, n-1]$). Still, this does not have an impact on the size of $\mathcal{P}_r$, since the number of redundant elements is – in general – relatively small compared to its size.

In order to get an accurate upper bound for the case $w > 1$:

- in the worst case, one has to solve all the corresponding optimization problems by enumerating all $\nu_r \in \mathcal{P}_r$;
- in the best case, when one vector for which the solution of the corresponding optimization problem is $\min\{2^r, t \cdot n\}$ is found, one can stop enumerating the remaining vectors, since the highest upper bound $\min\{2^r, t \cdot n\}$ has been already reached.

The best case has been used to find proper $B(x)$ that can realize the (almost) exponential growth of the algebraic degree, as shown in [30].

Limitations when $w > 1$ and our solutions. Roughly speaking, the time and memory complexity to construct $\mathcal{P}_r$ can be estimated as $\mathcal{O}(w^{2^{r-1}})$. It is easy to observe that the inefficiency comes from that $\mathcal{P}_r$ has to be computed round by round, which is crucial for determining the upper bound after r rounds. E.g., in [30], to find proper $B(x)$ achieving the (almost) exponential growth of the algebraic degree, it is suggested to store at most 2^{13} desired elements for each $\mathcal{P}_r$ and hence the time complexity becomes $2^{26} \times r$. However, too much accuracy is sacrificed because many possible ν_r are ignored.

In the following, we present an analogous approach exploited by the coefficient grouping technique for the case $w = 1$, which will allow to avoid the problem to construct $\mathcal{P}_r$ round by round because we can directly identify many crucial properties of $\mathcal{P}_r$ without explicitly computing it. This will significantly improve the efficiency and accuracy for $w > 1$.

3 On the Polynomial Representation for Complex $B(x)$

In this section, we aim to build a theory to capture the features of the polynomial representation of the analyzed cipher. We start by working in the univariate setting defined as before.

This will give us a necessary condition for the growth of the degree. Indeed, we recall that, if the degree does not increase exponentially in such setting in which the inputs $(x_1, \ldots, x_m)$ depend on a single variable, then it cannot increase exponentially in general.

Initial remarks. We start by pointing out some preliminary observations.

Remark 1. We recall that our theory covers also the case where $S(x) = x^{2^{d_1}+2^{d_2}}$ for $(d_2 > d_1)$. Indeed, it is sufficient to notice that

$$S(x) = x^{2^{d_1}+2^{d_2}} = (x^{2^{d_2-d_1}+1})^{2^{d_1}} = B' \circ S'(x)$$

where $B'(x) = x^{2^{d_1}}$ and $S'(x) = x^{2^{d_2-d_1}+1}$.

Remark 2. Since we are only interested in an upper bound of the growth of the degree, our goal is to predict which monomials do *not* appear in the polynomial representation of the analyzed cipher. Equivalently, we are not interested in the details of the coefficients of the monomials that appear. For this reason, from now on, we set them to 1. We emphasize that the same approach has been used in almost all the previous papers that cover the same topic, including [6, 7, 14, 16, 23, 30, 41].

Based on this, since the invertible matrix M only changes the coefficients of the monomials in the polynomial representation, we can omit its influence on the value of the coefficients of the monomial, exactly as in [30]. Note that a bad choice of M may affect the tightness of our upper bound and tracing its

influence requires non-trivial work, i.e. how to detect the cancellations in the monomials for our technique is quite challenging. This is an interesting open problem and we emphasize that the cipher is defined over $\mathbb{F}_{2^n}$, which makes detecting cancellations much more difficult, i.e. the coefficient of a monomial can take 2^n different possible values.

Tracing the evolution of polynomials. In the univariate setting, each input state word x_i ($i = 1, \ldots, m$) can be written as a certain polynomial $x_i = P_0(x) = u_1 \cdot x + u_0$, where we fix $u_0 = u_1 = 1$ due to the argument just given. (Note that each internal state word will have the same polynomial representation if we treat the coefficients of all possible monomials as 1, as already shown in the coefficient grouping technique [30].) In the following, we denote the polynomial representation of each state word after r rounds by $P_r(x)$ and the polynomial representation after $S(x)$ in the r-th round by $P_r^S(x)$.

Based on these considerations, we simply have that

$$P_r(x) = (B \circ S)^r(P_0(x)), \qquad \text{and} \qquad P_r^S(x) = S(P_{r-1}(x)).$$

For the particular case $r = 1$:

$$P_1^S(x) = (x+1)^{2^d}(x+1) = x^{2^d} + x^{2^d+1} + x + 1,$$
$$P_1(x) = 1 + \sum_{i=1}^{w} \left(P_1^S(x)\right)^{2^{h_i}} = 1 + \sum_{i=1}^{w} x^{2^{d+h_i}} + x^{2^{d+h_i}+2^{h_i}} + x^{2^{h_i}}.$$

It follows that only the monomials

$$\left\{x^{2^d}, x^{2^d+1}, x, x^0\right\}$$

can possibly appear in $P_1^S(x)$, and that only the monomials

$$\left\{x^{2^{d+h_i}}, x^{2^{d+h_i}+2^{h_i}}, x^{2^{h_i}}, x^0 \mid 1 \leq i \leq w\right\}$$

can possibly appear in $P_1(x)$.

Working iteratively, our goal is to find similar sets of possible monomials that can appear in $P_r^S(x)$. Denote such a set by $\mathcal{M}_r^S$. Given this set, the upper bound – denoted by δ_r – on the algebraic degree in the univariate setting after r rounds can be simply obtained as follows:

$$\delta_r = \max\left\{\mathrm{Hw}(M_n(e)) \mid x^e \in \mathcal{M}_r^S\right\}.$$

Since each monomial is uniquely identified by its degree, we can simply introduce the set $\mathcal{W}_r^S$ for the exponents, that is,

$$\mathcal{W}_r^S = \left\{e \in \mathbb{N} \mid x^e \in \mathcal{M}_r^S\right\}, \quad \delta_r = \max\left\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^S\right\}.$$

One more time, we recall that some of the monomials present in the previous set $\mathcal{M}_r^S$ do not necessarily appear for some particular values of the key and/or

of the linear layer M. At the same time, if a monomial that does *not* appear in such set, then it does *not* appear independently of the values of the key and/or of the linear layer M. This is sufficient for setting up an upper bound. Obviously, such bound is tight if the cancellation due to the key and/or of the linear layer M is almost null.

Given such definitions, we have the following results:

- case $r = 0$:
$$\mathcal{W}_0 = \{0, 1\};$$
- case $r = 1$:
$$\begin{aligned}\mathcal{W}_1^S &= \left\{2^d, 2^d + 1, 1, 0\right\} = \left\{a_{1,1}2^d + a_{1,2} \mid 0 \leq a_{1,1}, a_{1,2} \leq 1\right\}\\ \mathcal{W}_1 &= \left\{2^{d+h_i}, 2^{h_i} + 2^{d+h_i}, 2^{h_i}, 0 \mid 1 \leq i \leq w\right\}\\ &= \left\{a_{1,1}2^{d+h_i} + a_{1,2}2^{h_i} \mid 0 \leq a_{1,1}, a_{1,2} \leq 1, 1 \leq i \leq w\right\},\end{aligned}$$
- case $r = 2$:
$$\begin{aligned}\mathcal{W}_2^S = \big\{&a_{2,1}2^{2d+h_{i_0}} + a_{2,2}2^{d+h_{i_0}} + a_{2,3}2^{d+h_{i_1}} + a_{2,4}2^{h_{i_1}}\\ &\mid\ 0 \leq a_{2,j} \leq 1, 1 \leq i_0, i_1 \leq w, 1 \leq j \leq 4\big\},\\ \mathcal{W}_2 = \big\{&a_{2,1}2^{2d+h_{i_0}+h_{i_2}} + a_{2,2}2^{d+h_{i_0}+h_{i_2}} + a_{2,3}2^{d+h_{i_1}+h_{i_2}} + a_{2,4}2^{h_{i_1}+h_{i_2}}\\ &\mid\ 0 \leq a_{1,j} \leq 1, 1 \leq i_0, i_1, i_2 \leq w, 1 \leq j \leq 4\big\},\end{aligned}$$

and so on. The case $r = 2$ may be not obvious, especially for $\mathcal{W}_2^S$. This can be explained using the following remark:

Remark 3. We have $y^{2^d+1} = y^{2^d} \cdot y$ where y is a polynomial whose monomials can always be represented as $x^{a_{1,1}2^{d+h_i}+a_{1,2}2^{h_i}}$. Hence, for the left part in $y^{2^d} \cdot y$, i.e. y^{2^d}, we can choose any possible monomial $x^{a_{1,1}2^{d+h_{i_0}}+a_{1,2}2^{h_{i_0}}}$ for y, while for the right part in $y^{2^d} \cdot y$, we can also independently choose any possible monomial $x^{a'_{1,1}2^{d+h_{i_1}}+a'_{1,2}2^{h_{i_1}}}$ for y. Hence, we have $x^{a_{1,1}2^{2d+h_{i_0}}+a_{1,2}2^{d+h_{i_0}}+a'_{1,1}2^{d+h_{i_1}}+a'_{1,2}2^{h_{i_1}}}$ is a possible monomial in $y^{2^d+1} = y^{2^d} \cdot y$.

Hence, we can deduce the following theorem.

Theorem 1 *For each $r \geq 1$, let $\mathcal{V}_{r,w}$ be the set defined as*

$$\mathcal{V}_{r,w} = \left\{ e \in \mathbb{N} \mid e = \sum_{i=1}^{w} b_i h_i, \sum_{i=1}^{w} b_i = r - 1, b_i \geq 0 \right\}, \tag{3}$$

which represents all possible values by summing up $r - 1$ elements from the set $\{h_1, \ldots, h_w\}$. The set $\mathcal{W}_r^S$ can then be described as follows:

$$\mathcal{W}_r^S = \left\{ \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{r,v} 2^{(r-i)d+f_v}, v = j + \binom{r}{\leq i-1}, 0 \leq a_{r,v} \leq 1, f_v \in \mathcal{V}_{r,w} \right\},$$

where

$$f_{\binom{r}{\leq i}+\ell} = f_{\binom{r}{\leq i}-\binom{r-1}{i}+\ell} \text{ for } 0 \leq i \leq r-1, 1 \leq \ell \leq \tbinom{r-1}{i}. \tag{4}$$

Proof. We prove it by induction. Based on the above deduction, this has been proved for $r \in \{1, 2\}$. For example, $\mathcal{W}_2^S$ can be rewritten as the form in this theorem:

$$\mathcal{W}_2^S = \{a_{2,1}2^{2d+f_1} + a_{2,2}2^{d+f_2} + a_{2,3}2^{d+f_3} + a_{2,4}2^{f_4}, 0 \le a_{2,1}, a_{2,2}, a_{2,3}, a_{2,4} \le 1\},$$
$$f_2 = f_1, \quad f_4 = f_3,$$
$$f_1, f_2, f_3, f_4 \in \mathcal{V}_{2,w} = \left\{ e \in \mathbb{N} \mid e = \sum_{i=1}^{w} b_i h_i, \sum_{i=1}^{w} b_i = 1, b_i \ge 0 \right\} = \{h_i \mid 1 \le i \le w\}.$$

Assuming the theorem holds for $r = q$, we now prove that it also holds for $r = q + 1$. According to the assumption, we have

$$\mathcal{W}_q^S = \left\{ \sum_{i=0}^{q} \sum_{j=1}^{\binom{q}{i}} a_{q,v} 2^{(q-i)d+f_v}, v = j + \binom{q}{\le i-1}, 0 \le a_{q,v} \le 1, f_v \in \mathcal{V}_{q,w} \right\},$$

where

$$f_{\binom{q}{\le i}+\ell} = f_{\binom{q}{\le i}-\binom{q-1}{i}+\ell} \text{ for } 0 \le i \le q-1, 1 \le \ell \le \binom{q-1}{i}.$$

Then, after applying the affine transformation $B(x)$, the $\{h_u\}_{1 \le u \le w}$ appear in the power of 2, as described below:

$$\mathcal{W}_q = \left\{ \sum_{i=0}^{q} \sum_{j=1}^{\binom{q}{i}} a_{q,v} 2^{(q-i)d+f_v+h_u}, v = j + \binom{q}{\le i-1}, 0 \le a_{q,v} \le 1, f_v \in \mathcal{V}_{q,w}, 1 \le u \le w \right\},$$

Consider the set

$$\{f_v + h_u \mid f_v \in \mathcal{V}_{q,w}, 1 \le u \le w\}.$$

Then we have

$$\{f_v + h_u \mid f_v \in \mathcal{V}_{q,w}, 1 \le u \le w\} = \mathcal{V}_{q+1,w} = \left\{ e \mid e = \sum_{i=1}^{w} b_i h_i, \sum_{i=1}^{w} b_i = q, b_i \ge 0 \right\},$$

by definition of $\mathcal{V}_{q,w}$. Hence, $\mathcal{W}_q$ can be rewritten as

$$\mathcal{W}_q = \left\{ \sum_{i=0}^{q} \sum_{j=1}^{\binom{q}{i}} a_{q,v} 2^{(q-i)d+f_v}, v = j + \binom{q}{\le i-1}, 0 \le a_{q,v} \le 1, f_v \in \mathcal{V}_{q+1,w} \right\}$$

and there is still

$$f_{\binom{q}{\le i}+\ell} = f_{\binom{q}{\le i}-\binom{q-1}{i}+\ell} \text{ for } 0 \le i \le q-1, 1 \le \ell \le \binom{q-1}{i}.$$

Based on this new representation of $\mathcal{W}_q$, and using the idea of Remark 3, by introducing another 2^q variables $(f'_1, \ldots, f'_{2^q})$ satisfying:

$$f'_{\binom{q}{\leq i}+\ell} = f'_{\binom{q}{\leq i}-\binom{q-1}{i}+\ell} \text{ for } 0 \leq i \leq q-1, 1 \leq \ell \leq \binom{q-1}{i},$$

we can write $\mathcal{W}^S_{q+1}$ as follows:

$$\begin{aligned}
\mathcal{W}^S_{q+1} = \Bigg\{ & \sum_{i=0}^{q} \sum_{j=1}^{\binom{q}{i}} a_{q,v} 2^{(q-i+1)d+f_v} + \sum_{i=0}^{q} \sum_{j=1}^{\binom{q}{i}} a'_{q,v} 2^{(q-i)d+f'_v}, \\
& 0 \leq a_{q,v}, a'_{q,v} \leq 1, v = j + \binom{q}{\leq i-1}, f_v, f'_v \in \mathcal{V}_{q+1,w} \Bigg\} \\
= \Bigg\{ & a_{q+1,1} 2^{q+1d+g_1} + \sum_{i=1}^{q} \sum_{j=1}^{\binom{q}{i-1}+\binom{q}{i}=\binom{q+1}{i}} a_{q+1,v} 2^{(q+1-i)d+g_v} + a_{q+1,2^{q+1}} 2^{g_{2^{q+1}}}, \\
& 0 \leq a_{q+1,v} \leq 1, v = j + \binom{q+1}{\leq i-1}, g_v \in \mathcal{V}_{q+1,w} \Bigg\} \\
= \Bigg\{ & \sum_{i=0}^{q+1} \sum_{j=1}^{\binom{q+1}{i}} a_{q+1,v} 2^{(q+1-i)d+g_v}, 0 \leq a_{q+1,v} \leq 1, v = j + \binom{q+1}{\leq i-1}, g_v \in \mathcal{V}_{q+1,w} \Bigg\}
\end{aligned}$$

In the deduction, we replace $a_{q,1}$ and $a'_{q,2^r}$ with $a_{q+1,1}$ and $a_{q+1,2^{q+1}}$, respectively, i.e. $g_1 = f_1$ and $g_{2^{q+1}} = f'_{2^q}$.

Moreover, for $i \in [0, q-1]$, we make such changes:

$$\begin{aligned}
& \left(a_{q,\binom{q}{\leq i}+1}, \ldots, a_{q,\binom{q}{\leq i}+\binom{q-1}{i}}, a_{q,\binom{q}{\leq i}+\binom{q-1}{i}+1}, \ldots, a_{q,\binom{q}{\leq i}+\binom{q-1}{i}+\binom{q-1}{i+1}} \right) \\
= & \left(a_{q+1,\binom{q+1}{\leq i}+1}, \ldots, a_{q+1,\binom{q+1}{\leq i}+\binom{q-1}{i}}, a_{q+1,\binom{q+1}{\leq i}+\binom{q}{i}+1}, \ldots, a_{q+1,\binom{q+1}{\leq i}+\binom{q}{i}+\binom{q-1}{i+1}} \right),
\end{aligned}$$

In addition, we make $a_{q+1,\binom{q+1}{\leq 1}} = a'_1$. For $i \in [1, q-1]$, we make such changes:

$$\begin{aligned}
& \left(a'_{q,\binom{q}{\leq i-1}+1}, \ldots, a'_{q,\binom{q}{\leq i-1}+\binom{q-1}{i-1}}, a'_{q,\binom{q}{\leq i-1}+\binom{q-1}{i-1}+1}, \ldots, a'_{q,\binom{q}{\leq i-1}+\binom{q-1}{i-1}+\binom{q-1}{i}} \right) \\
= & \left(a_{q+1,\binom{q+1}{\leq i}+\binom{q-1}{i}+1}, \ldots, a_{q+1,\binom{q+1}{\leq i}+\binom{q}{i}}, a_{q+1,\binom{q+1}{\leq i}+\binom{q}{i}+\binom{q-1}{i+1}+1}, \ldots, a_{q+1,\binom{q+1}{\leq i+1}} \right).
\end{aligned}$$

By this construction, we have

$$g_{\binom{q+1}{\leq i}+j} = f_{\binom{q}{\leq i}+j} \text{ for } 0 \leq i \leq q-1,\ 1 \leq j \leq \binom{q-1}{i},$$

$$g_{q+2} = f'_1, g_{\binom{q+1}{\leq i}+\binom{q-1}{i}+j} = f'_{\binom{q}{\leq i-1}+j} \text{ for } 1 \leq i \leq q-1,\ 1 \leq j \leq \binom{q-1}{i-1},$$

$$g_{\binom{q+1}{\le i}+\binom{q}{i}+j} = f_{\binom{q}{\le i}+\binom{q-1}{i}+j} \text{ for } 0 \le i \le q-2,\ 1 \le j \le \binom{q-1}{i+1},$$
$$g_{\binom{q+1}{\le i}+\binom{q}{i}+\binom{q-1}{i+1}+j} = f'_{\binom{q}{\le i-1}+\binom{q-1}{i-1}+j} \text{ for } 0 \le i \le q-1,\ 1 \le j \le \binom{q-1}{i},$$

For better understanding, Fig. 2 illustrates these changes of variables.

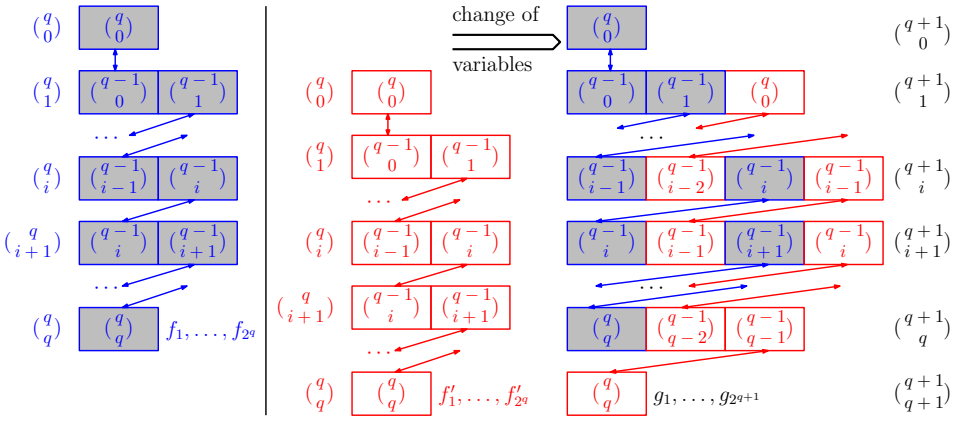

Fig. 2: The change of variables, where arrows represent the identical relation

Therefore, for $1 \le i \le q-1$ and $1 \le \ell \le \binom{q-1}{i}$, we have

$$g_{\binom{q+1}{\le i}+\ell} = f_{\binom{q}{\le i}+\ell} = f_{\binom{q}{\le i}-\binom{q-1}{i}+\ell} = f_{\binom{q}{\le i-1}+\binom{q-1}{i-1}+\ell} = g_{\binom{q+1}{\le i}-\binom{q}{i}+\ell}.$$

For $2 \le i \le q-1$ and $\binom{q-1}{i}+1 \le \ell \le \binom{q-1}{i}+\binom{q-1}{i-1} = \binom{q}{i}$, we have

$$g_{\binom{q+1}{\le i}+\ell} = f'_{\binom{q}{\le i-1}-\binom{q-1}{i}+\ell} = f'_{\binom{q}{\le i-2}+\binom{q-1}{i-2}-\binom{q-1}{i}+\ell} = g_{\binom{q+1}{\le i}-\binom{q}{i}+\ell}.$$

Moreover, we have $g_2 = f_2 = f_1 = g_1$, $g_{2q+2} = f'_2 = f'_1 = g_{q+2}$ and $g_{2^{q+1}} = f'_{2^q} = f_{2^q-1} = g_{2^{q+1}-1}$. Therefore, for $0 \le i \le q$ and $1 \le \ell \le \binom{q}{i}$, we have $g_{\binom{q+1}{\le i}+\ell} = g_{\binom{q+1}{\le i}-\binom{q}{i}+\ell}$. □

Based on Theorem 1, we can give a well-structured description of the monomials that will possibly appear after r rounds. Especially, from its current form, we immediately observe that the exponent of each possible monomial is a linear combination of $\sum_{i=0}^{r}\binom{r}{i} = 2^r$ numbers $(2^{j_1}, \ldots, 2^{j_{2^r}})$ satisfying certain properties. In the following, we show how to use this well-structured description to study several problems relevant to the growth of the algebraic degree.

4 Implications on the Growth of the Algebraic Degree

As next step, we exploit the result proposed in Theorem 1 to study the growth of the algebraic degree for complex affine layers in a systematic way. In the following, we will separately study

- the growth of the algebraic degree for the special case $w = 1$;
- necessary conditions to achieve the exponential growth;
- equivalent affine layers.

4.1 Special Case: $w = 1$

Let us start by pointing out the connection between Theorem 1 and the results proposed via the coefficient grouping technique developed for the case $w = 1$. In this case,

$$\mathcal{W}_r^S = \left\{ \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{r,v} 2^{(r-i)d+f_v}, v = j + \binom{r}{\leq i-1}, 0 \leq a_{r,v} \leq 1, f_v \in \mathcal{V}_{r,1} \right\},$$

where

$$\mathcal{V}_{r,1} = \{(r-1)h_1\}.$$

By definition, this means that $f_v = (r-1)h_1$ for each $v \in [1, 2^r]$. Hence, we have

$$\mathcal{W}_r^S = \left\{ \sum_{i=0}^{r} a_i 2^{(r-i)d+(r-1)h_1}, 0 \leq a_i \leq \binom{r}{i} \right\},$$

which implies

$$\mathcal{W}_r = \left\{ \sum_{i=0}^{r} a_i 2^{(r-i)d+rh_1}, 0 \leq a_i \leq \binom{r}{i} \right\} \text{ when } w = 1.$$

According to the definition of δ_r, it follows that

$$\delta_r = \max\left\{ \mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^S \right\} = \max\{ \mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r \}.$$

In order to efficiently compute δ_r, we suggest to iteratively construct a vector $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0}) \in \mathbb{N}^n$ via Algorithm 1. By the construction of this vector ν_r, we can equivalently describe $\{M_n(e) \mid e \in \mathcal{W}_r\}$ as follows:

$$\{M_n(e) \mid e \in \mathcal{W}_r\} = \left\{ M_n(e) \mid e = \sum_{i=0}^{n-1} 2^i a_i, a_i \in [0, \nu_{r,i}] \right\}$$

In this way, computing δ_r for the case $w = 1$ is converted into solving **OP-1** parameterized by the constructed vector $(\nu_{r,n-1}, \ldots, \nu_{r,0})$.

Algorithm 1 Constructing $(\nu_{r,n-1},\ldots,\nu_{r,0}) \in \mathbb{N}^n$ when $w=1$

procedure CONVERSION(ν_r, r, n)
    initialize $(\nu_{r,n-1},\ldots,\nu_{r,0})$ as all 0
    for all $i \in [0,r]$ **do**
        $j = ((r-i)\times d + r \times h_1)\%n$
        $\nu_{r,j} = \nu_{r,j} + \binom{r}{i}$

By induction, it is possible to prove that the vector ν_r constructed via Algorithm 1 is indeed the same as that computed with the recursive relation specified in Equation 1. The main difference is that the recursive relation in Equation 1 computes the vector ν_r round by round, while we no more compute it in such a round-by-round way. If interpreting this from a mathematic perspective, we now deal with the modular addition in the exponent at the last round, while it is handled round by round in the recursive relation when tracing the polynomial representation. This reveals that the original coefficient grouping technique [30] developed for the case $w=1$ can be fully interpreted via this new representation of $\mathcal{W}_r$.

4.2 (At Most) Quadratic Growth for $w=1$

Meanwhile, our new representation allows us to show that *the algebraic degree in the univariate setting will never increase exponentially after the 1st round if* $w=1$. In particular, we can prove that it can increase at most quadratically.

Specifically, for $0 \le a_i \le \nu_{r,i}$, based on the inequality $\mathrm{Hw}(M_n(a+b)) \le \mathrm{Hw}(M_n(a)) + \mathrm{Hw}(M_n(b))$ [36], the following inequality always holds:

$$\mathrm{Hw}\left(M_n\left(\sum_{i=0}^{n-1} 2^i a_i\right)\right) \le \sum_{i=0}^{n-1} \mathrm{Hw}\left(M_n(2^i a_i)\right) \le \sum_{i=0}^{n-1} \mathrm{Hw}(a_i) \le \sum_{i=0}^{n-1} \lfloor \log_2(\nu_{r,i}+1) \rfloor.$$

Denote the nonzero elements in ν_r by $(\nu_{r,i_1},\ldots,\nu_{r,i_\tau})$. With Algorithm 1 to construct such ν_r, for each $1 \le j \le \tau$, we have

$$\nu_{r,i_j} = \binom{r}{\ell_{i_j,1}} + \ldots + \binom{r}{\ell_{i_j,k_j}} \text{ where } k_j \ge 1. \tag{5}$$

In this way, we have $k_1 + \ldots + k_\tau = r+1$ and

$$\{\ell_{i_1,1},\ldots,\ell_{i_1,k_1},\ldots,\ell_{i_\tau,1},\ldots,\ell_{i_\tau,k_\tau}\} = \{0,\ldots,r\}. \tag{6}$$

Since $k_j \ge 1$ for $j \in [1,\tau]$, we have

$$\delta_r \le \sum_{i=0}^{n-1} \lfloor \log_2(\nu_{r,i}+1) \rfloor = \sum_{j=1}^{\tau} \lfloor \log_2(\nu_{r,i_j}+1) \rfloor \le \sum_{j=1}^{\tau} \log_2(\nu_{r,i_j}+1) \le \sum_{j=1}^{\tau} \log_2(\nu_{r,i_j}+k_j).$$

Since $\log_2(a+b) \leq \log_2 a + \log_2 b$ for each $a, b \geq 2$, due to Equation 5 and Equation 6, we further have

$$\delta_r \leq \sum_{j=1}^{\tau} \log_2 \left(\nu_{r,i_j} + k_j\right) = \sum_{j=1}^{\tau} \log_2 \left(\sum_{u=1}^{k_j} \left(\binom{r}{\ell_{i_j,u}} + 1 \right) \right) \leq \sum_{j=0}^{r} \log_2 \left(\binom{r}{j} + 1 \right),$$

which implies $\delta_1 \leq 2$ and $\delta_2 \leq 3$. Since $\binom{r}{i} + 1 \leq 2^{r-1}$ if $r \geq 3$, for $r \geq 3$, we have

$$\delta_r \leq 2 + \sum_{j=1}^{r-1} \log_2 \left(\binom{r}{j} + 1 \right) \leq 2 + (r-1) \cdot \log_2 2^{r-1} = r^2 - 2r + 3.$$

As can be noted, these bounds are not tight. Still, *since* $r^2 - 2r + 3 < 2^r$ *for each* $r \geq 2$, it follows that the algebraic degree will never increase exponentially after the 1st round, and that it is always below the quadratic growth $r^2 - 2r + 3$ for any d when $w = 1$. Only for completeness, we emphasize that the accurate bound for the case $w = 1$ can be computed via the $\mathcal{O}(n)$ reduction algorithm in [36]. Compared with the results in [7, 14, 23], we point out that:

- the upper bounds proposed in such papers is exponential for the first r_0 rounds where $\lfloor \log_2(2^d+1)^{r_0}+1 \rfloor > 2^{r_0}$ (roughly speaking, $d \cdot r_0 > 2^{r_0}$), while here we proved that it is always at most quadratic for any rounds and any d.
- although the upper bounds $\approx d \cdot r$ proposed in such papers become linear in the remaining rounds, it is still possible that our quadratic bound is better for some d in some of these rounds, e.g. this is obvious for large d.

4.3 Necessary Conditions for the Exponential Growth for $w > 1$

In the above, we have proved that the algebraic degree will never increase exponentially when $w = 1$. It is then natural to ask whether similar conclusions can be derived for complex affine layers, i.e., the case $w > 1$. Here, we answer this question.

Observations on $\mathcal{W}_r^S$. First, focus on the new representation of $\mathcal{W}_r^S$, as shown below:

$$\mathcal{W}_r^S = \left\{ \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{r,v} 2^{(r-i)d+f_v}, v = j + \binom{r}{\leq i-1}, 0 \leq a_{r,v} \leq 1, f_v \in \mathcal{V}_{r,w} \right\},$$

Observe that each element in $e \in \mathcal{W}_r^S$ is a linear combination of 2^r numbers specified as:

$$\underbrace{2^{rd+f_1}}_{i=0}, \underbrace{2^{(r-1)d+f_{1+1}}, \ldots, 2^{(r-1)d+f_{1+\binom{r}{1}}}}_{i=1}, \ldots,$$

$$\underbrace{2^{(r-i)d+f_{\binom{r}{\leq i-1}}+1}, \ldots, 2^{(r-i)d+f_{\binom{r}{\leq i-1}}+\binom{r}{i}}}_{i}, \ \ldots, \ \underbrace{2^{f_{2^r}}}_{i=r}.$$

For each valid assignment to $(f_1, \ldots, f_{2^r})$, we can obtain a subset $\mathcal{W}_r^{S,f} \subseteq \mathcal{W}_r^S$ defined as:

$$\mathcal{W}_r^{S,f} = \left\{ \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{r,v} 2^{(r-i)d+f_v}, v = j + \binom{r}{\leq i-1}, 0 \leq a_{r,v} \leq 1 \right\}.$$

For this subset $\mathcal{W}_r^{S,f}$, computing $\max\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^{S,f}\}$ is equivalent to solving **OP-1** parameterized by the vector $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0})$, which is computed according to Algorithm 2. The underlying reason is that with the vector ν_r constructed with Algorithm 2, we have $\{M_n(e) \mid e \in \mathcal{W}_r^{S,f}\} = \{M_n(e) \mid e = \sum_{i=0}^{n-1} 2^i a_i, \ 0 \leq a_i \leq \nu_{r,i} \ \text{ for } \ 0 \leq i \leq n-1\}$.

Algorithm 2 Constructing $(\nu_{r,n-1}, \ldots, \nu_{r,0}) \in \mathbb{N}^n$ for the subset $\mathcal{W}_r^{S,f}$

```
1: procedure CONVERSION_SUBSET(ν_r, r, n)
2:     initialize (ν_{r,n-1}, ..., ν_{r,0}) as all 0
3:     v = 1
4:     for all i ∈ [0, r] do
5:         for all j ∈ [1, C(r,i)] do
6:             u = ((r − i) × d + f_v)%n
7:             ν_{r,u} = ν_{r,u} + 1
8:             v = v + 1
```

Based on this, we can derive the following theorem.

Theorem 2 *For each valid assignment to $(f_1, \ldots, f_{2^r})$, the corresponding vector $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0})$ computed with Algorithm 2 satisfies the following property:*

$$\sum_{i=0}^{n-1} \nu_{r,i} = 2^r.$$

Especially, if there exists (i_1, v_1) and (i_2, v_2) where $i_1, i_2 \in [0, r]$ and $v_1, v_2 \in [1, 2^r]$ such that $(r - i_1) \times d + f_{v_1} = (r - i_2) \times d + f_{v_2}$, there are at most $2^r - 1$ nonzero elements in the vector ν_r.

Proof. The first sum property directly follows from Algorithm 2 because there are 2^r iterations and in each iteration, one $\nu_{r,u}$ will be increased by 1. The second part can be observed from the formula to compute the index u, i.e. $u = ((r - i) \times d + f_v)\%n$. Specifically, if there exists such a pair (i_1, v_1) and (i_2, v_2), the same index u will appear at least twice and hence there are at most $2^r - 1$ different elements in ν_r that will be updated. □

Necessary condition for exponential growth. Based on this, we can deduce a necessary condition for ensuring the exponential growth.

Theorem 3 *To ensure that the sharp exponential growth can occur for the first r rounds, i.e. the algebraic degree 2^r can be reached after r rounds, one necessary condition is that there exists a valid assignment to $(f_1, \ldots, f_{2^r})$ such that the following 2^r elements are all different:*

$$\underbrace{(rd + f_1)\%n}_{i=0}, \underbrace{((r-1)d + f_{1+1})\%n, \ldots, ((r-1)d + f_{1+\binom{r}{1}})\%n}_{i=1}, \ \ldots,$$
$$\underbrace{((r-i)d + f_{\binom{r}{\leq i-1}+1})\%n, \ldots, ((r-i)d + f_{\binom{r}{\leq i-1}+\binom{r}{i}})\%n}_{i}, \ \ldots, \ \underbrace{f_{2^r}\%n}_{i=r}.$$

Proof. For each valid assignment to $(f_1, \ldots, f_{2^r})$ and the corresponding subset $\mathcal{W}_r^{S,f}$, we always have

$$\max\left\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^{S,f}\right\} \leq \sum_{i=0}^{n-1} \lfloor \log_2(\nu_{r,i} + 1) \rfloor.$$

If there does not exist an assignment to make the above 2^r numbers different, according to Theorem 2, there will be at most $2^r - 1$ nonzero elements in ν_r and we denote them by $(\nu_{r,j_1}, \ldots, \nu_{r,j_\tau})$ where $\tau \leq 2^r - 1$ and $\sum_{i=1}^{\tau} \nu_{r,j_i} = 2^r$. Due to $\lfloor \log_2(a+1) \rfloor \leq a$ for $a \geq 1$ and $\lfloor \log_2(a+1) \rfloor \leq a - 1$ for $a \geq 2$, we have $\sum_{i=0}^{n-1} \lfloor \log_2(\nu_{r,i} + 1) \rfloor = \sum_{i=1}^{\tau} \lfloor \log_2(\nu_{r,j_i} + 1) \rfloor \leq \sum_{i=1}^{\tau} \nu_{r,j_i} - 1 = 2^r - 1$, which implies $\delta_r \leq 2^r - 1$. If there exists such an assignment, we trivially have $\delta_r = 2^r$. □

Corollary 1 *For a given $(h_1, \ldots, h_w)$, a necessary condition to ensure that the exponential growth can occur is*

$$\left|\mathcal{V}_{r,w}^R\right| \geq \binom{r}{\lceil \frac{r}{2} \rceil}$$

where

$$\mathcal{V}_{r,w}^R = \{e\%n \mid e \in \mathcal{V}_{r,w}\} = \left\{ e\%n \mid e = \sum_{i=1}^{w} b_i h_i, \sum_{i=1}^{w} b_i = r - 1, b_i \geq 0 \right\}.$$

Proof. To ensure the necessary condition in Theorem 3, then at least the following $\binom{r}{i}$ numbers must be different from each other for $i = \lceil \frac{r}{2} \rceil$:

$$\left((r-i)d + f_{\binom{r}{\leq i-1}+1}\right)\%n, \ \ldots, \ \left((r-i)d + f_{\binom{r}{\leq i-1}+\binom{r}{i}}\right)\%n,$$

which implies

$$\left(f_{\binom{r}{\leq i-1}+1}\right)\%n, \ \ldots, \ \left(f_{\binom{r}{\leq i-1}+\binom{r}{i}}\right)\%n$$

must be different. □

4.4 Choosing the Affine Layer $B(x)$ for the Exponential Growth

Impact of Theorem 3 on the choice of $B(x)$. Let $\mathcal{B}_{r,w}$ be the set of tuples $(b_1, \ldots, b_w)$ defined as:

$$\mathcal{B}_{r,w} = \left\{ (b_1, \ldots, b_w) \mid \sum_{i=1}^{w} b_i = r, b_i \geq 0 \right\}.$$

Note that by definition, we have

$$\begin{aligned} |\mathcal{B}_{r,w}| &= |\mathcal{B}_{r,w-1}| + |\mathcal{B}_{r-1,w-1}| + |\mathcal{B}_{r-2,w-1}| + \ldots + |\mathcal{B}_{0,w-1}| \\ &= |\mathcal{B}_{r,w-1}| + |\mathcal{B}_{r-1,w}|. \end{aligned}$$

Given such a set, we can deduce the following result.

Corollary 2 *A necessary condition for the exponential growth in the first r rounds is that*

$$|\mathcal{B}_{r-1,w}| \geq \binom{r}{\lceil \frac{r}{2} \rceil}.$$

Proof. For any given $(h_1, h_2, \ldots, h_w) \in \mathbb{N}^w$, by definition, we have

$$|\mathcal{V}^R_{r,w}| \leq |\mathcal{V}_{r,w}| \leq |\mathcal{B}_{r-1,w}|. \tag{7}$$

The necessary condition $|\mathcal{B}_{r-1,w}| \geq \binom{r}{\lceil \frac{r}{2} \rceil}$ directly follows Corollary 1 by involving the condition on $|\mathcal{B}_{r-1,w}| \geq |\mathcal{V}_{r,w}| \geq |\mathcal{V}^R_{r,w}|$ specified in Equation 7. □

By simple computation:

$$|\mathcal{B}_{2,2}| = 3 \geq \binom{3}{2} = 3, \qquad |\mathcal{B}_{3,2}| = 4 < \binom{4}{2} = 6.$$

According to Corollary 2, the above inequality implies that it is impossible to achieve the sharp exponential growth of the algebraic degree at the 4th round when $w = 2$. In a similar way,

$$\begin{aligned} |\mathcal{B}_{5,3}| = 21 \geq \binom{6}{3} = 20, \qquad & |\mathcal{B}_{6,3}| = 28 < \binom{7}{4} = 35, \\ |\mathcal{B}_{8,4}| = 165 \geq \binom{9}{5} = 126, \qquad & |\mathcal{B}_{9,4}| = 220 < \binom{10}{5} = 252, \end{aligned}$$

(and so on). As before, these imply that it is impossible to achieve the sharp exponential growth of the algebraic degree at the 7th and 10th round when $w = 3$ and $w = 4$, respectively (and so on). In other words, the sharp exponential growth can be achieved for at most the first 3, 6 and 9 rounds when $w = 2, 3, 4$, respectively.

In summary, even without considering the concrete affine layers, i.e., the concrete values of $(h_1, \ldots, h_w)$, the above results have revealed the influence of the density (w) of the affine layers on the growth of the algebraic degree from a pure theoretical perspective.

Equivalent Affine Layers. Corollary 1 and Corollary 2 can work as efficient filters to decide whether a specific $B(x)$ can achieve the exponential growth of the algebraic degree. To further reduce the number of candidates, it will be helpful if we can identify equivalent affine layers with the same influence on the growth of the algebraic degree with our techniques.

The following results hold:

Corollary 3 *Given an integer $\varepsilon \geq 0$, the following two affine layers have the same effect on the growth of the algebraic degree:*

$$B(x) = c_0 + \sum_{i=1}^{w} c_i x^{2^{h_i}}, \quad B'(x) = c'_0 + \sum_{i=1}^{w} c'_i x^{2^{(h_i+\varepsilon)\%n}}.$$

Proof. First, let us introduce the set ${\mathcal{V}'}_{r,w}$ for a given integer $\varepsilon \geq 0$. Given $\mathcal{V}_{r,w}$ defined as in Equation 3, let ${\mathcal{V}'}_{r,w}$ and ${\mathcal{W}'}_r^S$ be defined as:

$${\mathcal{V}'}_{r,w} = \left\{ e \mid e = \sum_{i=1}^{w} b_i(h_i + \varepsilon), \sum_{i=1}^{w} b_i = r-1, b_i \geq 0 \right\} = \{(r-1)\varepsilon + e \mid e \in \mathcal{V}_{r,w}\},$$

$${\mathcal{W}'}_r^S = \left\{ \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{r,v} 2^{(r-i)d+f_v}, v = j + \binom{r}{\leq i-1}, 0 \leq a_{r,v} \leq 1, f_v \in {\mathcal{V}'}_{r,w} \right\},$$

where $(f_1, \ldots, f_{2^r})$ also satisfy Equation 4. Then, we have $\{e \mid e \in {\mathcal{W}'}_r^S\} = \{2^{(r-1)\varepsilon} \cdot e \mid e \in \mathcal{W}_r^S\}$. Since $\mathrm{Hw}(M_n(e)) = \mathrm{Hw}(M_n(2^j \cdot e))$ for $j \in \mathbb{N}$, we have

$$\max\left\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^S\right\} = \max\left\{\mathrm{Hw}(M_n(e)) \mid e \in {\mathcal{W}'}_r^S\right\}.$$

□

Corollary 4 *The following two affine layers have the same effect on the growth of the algebraic degree:*

$$B(x) = c_0 + \sum_{i=1}^{w} c_i x^{2^{h_i}}, \quad B'(x) = c'_0 + \sum_{i=1}^{w} c'_i x^{2^{h'_i}}$$

where $(h_{i+1} - h_i)\%n = (h'_{i+1} - h'_i)\%n$ for $i \in [1, w-1]$

Proof. First, according to Corollary 3, $B(x)$ is equivalent to $B_1(x) = c_{0,1} + \sum_{i=1}^{w} c_{i,1} x^{2^{(h_i - h_1)\%n}}$. Similarly, $B'(x)$ is equivalent to $B_2(x) = c_{0,2} + \sum_{i=1}^{w} c_{i,2} x^{2^{(h'_i - h'_1)\%n}}$. Moreover, we also have

$$\begin{aligned}(h'_i - h'_1)\%n &= ((h'_i - h'_{i-1}) + (h'_{i-1} - h'_{i-2}) + \ldots + (h'_2 - h'_1))\%n \\ &= ((h_i - h_{i-1}) + (h_{i-1} - h_{i-2}) + \ldots + (h_2 - h_1))\%n = (h_i - h_1)\%n.\end{aligned}$$

This implies that $B_1(x)$ and $B_2(x)$ are equivalent and hence $B(x)$ and $B'(x)$ are equivalent. □

As a concrete consequence, the two previous Corollaries imply that about $\frac{1}{w} \cdot \binom{n}{w-1}$ candidates out of $\binom{n}{w}$ have a different impact on the growth of the algebraic degree. Indeed, Corollary 3 indicates that we only need to consider such $(h_1, \ldots, h_w)$ that $0 \in \{h_1, \ldots, h_w\}$. For simplicity, let $0 = h_1 < \ldots < h_w$. Corollary 4 further suggests that even if we only need to consider such $(h_1, \ldots, h_w)$, there are still w equivalent cases, i.e. $(0, h_2, \ldots, h_w)$ is equivalent to $((0 - h_i)\%n, (h_2 - h_i)\%n, \ldots, (h_w - h_i)\%n)$ for $\forall i \in [1, w]$.

Consistency with previous works [7, 14, 23]. Let $h' = \max\{h_1, \ldots, h_w\}$. With the techniques proposed in these papers, an upper bound of δ_r is given by

$$\delta_r \leq \lfloor \log_2((2^d + 1)^r \cdot (2^{h'})^{r-1} + 1) \rfloor.$$

Here, we point out that this result can also be derived via the set $\mathcal{W}_r^S$. Specifically, since we always have $2^{(r-i)d+f_v} \leq 2^{(r-i)d} \cdot 2^{(r-1)h'}$, it follows that

$$\max\{e \mid e \in \mathcal{W}_r^S\} = 2^{(r-1)h'} \cdot \left(\sum_{i=0}^{r} \binom{r}{i} \cdot (2^d)^{r-i} \right) = 2^{(r-1)h'} \cdot (1 + 2^d)^r.$$

Hence, we have $\delta_r = \max\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^S\} \leq \lfloor \log_2((2^d+1)^r \cdot (2^{h'})^{r-1}+1) \rfloor$, which is exactly the same bound given before. While $\lfloor \log_2((2^d+1)^r \cdot (2^{h'})^{r-1}+1) \rfloor$ is an absolute upper bound, it may be loose because it corresponds to $\max\{\mathrm{Hw}(e) \mid e \in \mathcal{W}_r^S\}$ rather than $\max\{\mathrm{Hw}(M_n(e)) \mid e \in \mathcal{W}_r^S\}$, i.e. the modular addition cannot be handled. In our technique, we will deal with the modular operation.

5 Finding Affine Layers for the Exponential Increase

In the previous sections, we presented necessary conditions for the exponential growth of the algebraic degree. As next step, we analyze how to efficiently find $(h_1, \ldots, h_w)$ that satisfies such a requirement. We point out that this problem is nontrivial, since – as stated in Theorem 3 – we have to guarantee that there exists a valid assignment to $(f_1, \ldots, f_{2^r})$ for a given $(h_1, \ldots, h_w)$ such that the following 2^r numbers are all different, where we define $r + 1$ classes for these 2^r numbers and there are $\binom{r}{i-1}$ numbers in Class i:

$$\begin{aligned}
&\text{Class } 1: \ (rd + f_1)\%n, \\
&\text{Class } 2: \ ((r-1)d + f_{1+1})\%n, \ldots, \left((r-1)d + f_{1+\binom{r}{1}}\right)\%n, \\
&\ldots, \\
&\text{Class } i+1: \ \left((r-i)d + f_{\binom{r}{\leq i-1}+1}\right)\%n, \ldots, \left((r-i)d + f_{\binom{r}{\leq i-1}+\binom{r}{i}}\right)\%n, \\
&\ldots, \\
&\text{Class } r+1: \ f_{2^r}\%n.
\end{aligned}$$

Remark 4. We point out that giving a sufficient condition is a much more difficult task. Indeed, even giving the capability to write down the accurate polynomial

representation after r rounds, many more factors play a role. E.g., the coefficients in the polynomial representation depend on the concrete value of the secret key, hence, the maximum algebraic degree could be different for different keys (remember that it is defined by the monomials with nonzero coefficients).

Let us recall that there are some special inner conditions on $(f_1, \ldots, f_{2^r})$:

$$f_{\binom{r}{\leq i}+\ell} = f_{\binom{r}{\leq i}-\binom{r-1}{i}+\ell} \text{ for } 0 \leq i \leq r-1, 1 \leq \ell \leq \binom{r-1}{i}. \tag{8}$$

Specifically, the inner conditions on $(f_1, \ldots, f_{2^r})$ state that for two consecutive classes, the last $\binom{r-1}{i} = \binom{r}{i} - \binom{r-1}{i-1}$ different f_v in Class $i+1$ have to be identical to the first $\binom{r-1}{i}$ different f_v in Class $i+2$, respectively. This can be observed from Fig. 2 by replacing q with r.

5.1 The Pre-processing Phase

To efficiently check the conditions in Theorem 3, we first precompute $r+1$ arrays of size n, which are denoted by $A_1, \ldots, A_{r+1}$ where $A_i = (A_i[n-1], \ldots, A_i[0]) \in \mathbb{F}_2^n$. Specifically, A_{i+1} corresponds to the Class $i+1$, and it is computed via the following rule:

$$\begin{aligned}
&\text{Set } A_{i+1} \text{ as all zero} \\
&\text{for all } u \in \mathcal{V}_{r,w}^R: \\
&\quad j = ((r-i) \times d + u) \% n \\
&\quad A_{i+1}[j] = 1
\end{aligned}$$

In other words, $A_{i+1}[j] = 1$ means that the value j can appear in the Class $i+1$, while $A_{i+1}[j] = 0$ means that the value j does not appear in the Class $i+1$. By simple computation, we also have that

$$A_{i+1}[j] = A_i[(j+d)\%n], \ \forall j \in [0, n-1].$$

Given these $r+1$ precomputed arrays, we can build a model to decide whether there exists an assignment to $(f_1, \ldots, f_{2^r})$ such that all the 2^r elements in the $r+1$ classes are different. This corresponds to finding $\binom{r}{i}$ nonzero positions from A_{i+1} for each $i \in [0, r]$ – by taking the inner conditions in Equation 8 into account – such that all these 2^r positions are different.

Before going on, we introduce an important set $\mathcal{Z}$. With these $r+1$ arrays $(A_1, \ldots, A_{r+1})$, we can construct a set $\mathcal{Z}$ defined as:

$$\mathcal{Z} := \{j \mid \exists i \in [1, r+1], \ A_i[j] \neq 0\}. \tag{9}$$

This set $\mathcal{Z}$ stores all possible numbers that can be chosen for the 2^r numbers in the $r+1$ classes. The following necessary condition trivially follows:

Corollary 5 *A necessary condition for the exponential growth in the first r rounds is $|\mathcal{Z}| \geq 2^r$.*

Different from Corollary 1 and Corollary 2, the influence of d on the growth of the algebraic degree can be reflected by Corollary 5.

5.2 Building the Model

Given the arrays A_i defined as before, we can now construct our model. First, we introduce two binary arrays for each A_i denoted by A_i^{up} and A_i^{dw}, respectively. A_i^{up} records the choices for the first $\binom{r-1}{i-2}$ elements in the Class i, while A_i^{dw} records the choices for the last $\binom{r-1}{i-1} = \binom{r}{i-1} - \binom{r-1}{i-2}$ elements in the Class i. Note that:

- if $A_i^{up}[j] = 1$ or $A_i^{dw}[j] = 1$, then the position j will be chosen from A_i as a number in the first $\binom{r-1}{i-2}$ (or the last $\binom{r-1}{i-1}$) elements in Class i;
- the inner conditions in Equation 8 imply that

$$A_{i+1}^{up}[j] = A_i^{dw}[(j+d)\%n], \ \forall i \in [1,r], \forall j \in [0,n-1]; \tag{10}$$

- the arrays A_1^{up} and A_{r+1}^{dw} should always be set to 0 because A_1 and A_{r+1} represent the first and last class, respectively. This implies that the following conditions are necessary:

$$A_1^{up}[j] = 0, \quad A_{r+1}^{dw}[j] = 0, \forall j \in [0,n-1]; \tag{11}$$

- the arrays A_i^{up} and A_i^{dw} have to satisfy

$$A_i^{up}[j] = A_i^{dw}[j] = 0 \ \text{ if } \ A_i[j] = 0, \ \forall i \in [1,r+1], \forall j \in [0,n-1], \tag{12}$$

  because we can only choose positions from nonzero positions in A_i.
- the arrays A_i^{dw} have to satisfy

$$\sum_{j=0}^{n-1} A_i^{dw}[j] \leq \binom{r-1}{i-1}, \ \forall i \in [1,r], \tag{13}$$

  because we can choose at most $\binom{r-1}{i-1}$ positions from A_i for the last $\binom{r-1}{i-1}$ elements in Class i, i.e. it is possible that some elements in Class i will be the same. Note that combined with Equation 10, there is always:

$$\sum_{j=0}^{n-1} A_i^{up}[j] = \sum_{j=0}^{n-1} A_{i-1}^{dw}[j] \leq \binom{r-1}{i-2}, \ \forall i \in [2,r+1], \tag{14}$$

  i.e. we can choose at most $\binom{r-1}{i-2}$ positions from A_i for the first $\binom{r-1}{i-2}$ elements in Class i. In other words, the constraint in Equation 14 can be omitted.

Finally, we introduce another binary array $X = (X[n-1], \ldots, X[0]) \in \mathbb{F}_2^n$, where $X[j] = 1$ (resp., $X[j] = 0$) means j has (resp., not) been chosen for the 2^r numbers. Hence, $X[j] = 1$ holds if and only if there exists (i,j) such that $A_i^{up}[j] + A_i^{dw}[j] \geq 1$. As a result, the following constraints can be derived:

$$X[j] \geq A_i^{up}[j], \ \forall i \in [1,r+1], \forall j \in [0,n-1],$$
$$X[j] \geq A_i^{dw}[j], \ \forall i \in [1,r+1], \forall j \in [0,n-1],$$

$$X[j] \leq \sum_{i=1}^{r+1}(A_i^{up}[j] + A_i^{dw}[j]),\ \forall j \in [0, n-1].$$

The objective function can be set to

$$\text{maximize } \sum_{i=0}^{n-1} X[i].$$

Interpretation of the objective function. The objective function indeed corresponds to the maximal number of different elements in the $r+1$ classes under all possible valid assignments to $(f_1, \ldots, f_{2^r})$. Let us denote the solution of the objective function by π. Taking Algorithm 2 into account, π indeed means that under all possible assignments to $(f_1, \ldots, f_{2^r})$, the corresponding vectors $(\nu_{r,n-1}, \ldots, \nu_{r,0})$ must satisfy

$$|\{i \mid \nu_{r,i} \neq 0, 0 \leq i \leq n-1\}| \leq \pi. \tag{15}$$

According to Theorem 3, $\pi = 2^r$ has to hold to ensure the sharp exponential growth of the algebraic degree, in which case we can find a valid assignment to $(f_1, \ldots, f_{2^r})$ such that all the 2^r elements in the $r+1$ classes are different.

Remark 5. To ensure the maximal degree n can be reached at round r, we can turn to testing whether $\pi = n$ holds. Of course, this is not a necessary condition but it greatly simplifies the problem and helps designers efficiently pick proper $(h_1, \ldots, h_w)$ for the fastest growth of the algebraic degree.

5.3 Practical Tests

For the efficiency of ciphers, w should be kept as small as possible while it can still ensure the exponential/fastest growth of the algebraic degree. In our following experiments, we find that the model is very light and can be efficiently solved. Our aim is to find proper $(h_1, \ldots, h_w)$ for any specified (n, d) with small w. All the experiments in this paper are performed on a PC by using 4 threads. The processor is 11th Gen Intel(R) Core(TM) i7-1195G7 and the RAM is 32GB.

The case $(n, d) = (63, 32)$. For example, for $(n, d) = (63, 32)$ which is the parameter used in Chaghri, according to Corollary 2, $w \geq 3$ has to hold to ensure the sharp exponential growth. With our new technique, we found in total 80 desired solutions of (h_1, h_2, h_3) without equivalent relations that can make $\delta_5 = 32$ and $\delta_6 = 63$, as shown in Table 1. For the running time, we observed that for each possible candidate of (h_1, h_2, h_3), it takes about $0.01 \sim 0.07$ second to output the solution π. This is definitely much better than the previous coefficient grouping technique by the plain enumeration [30].

It is found that the recommended parameter $(h_1, h_2, h_3) = (0, 2, 8)$ to patch Chaghri in [30] is not included, which implies that $(h_1, h_2, h_3) = (0, 2, 8)$ is probably not a good choice.

Table 1: List of choices of (h_1, h_2, h_3) for $(n, d) = (63, 32)$

Choices of (h_1, h_2, h_3) for $(n, d) = (63, 32)$
(0,2,9),(0,2,14),(0,2,20),(0,2,22),(0,2,24),(0,2,25),(0,2,26),(0,2,27),(0,2,38), (0,2,39),(0,2,40),(0,2,41),(0,2,43),(0,2,45),(0,2,51),(0,2,56),(0,3,27),(0,3,39), (0,4,10),(0,4,17),(0,4,26),(0,4,29),(0,4,38),(0,4,41),(0,4,50),(0,4,57),(0,5,19), (0,5,24),(0,5,28),(0,5,40),(0,5,44),(0,5,49),(0,6,14),(0,6,15),(0,6,54),(0,6,55), (0,7,22),(0,7,27),(0,7,34),(0,7,36),(0,7,43),(0,7,48),(0,8,18),(0,8,26),(0,8,45), (0,8,53),(0,9,26),(0,9,28),(0,9,34),(0,9,35),(0,9,37),(0,9,38),(0,9,44),(0,9,46), (0,10,23),(0,10,25),(0,10,27),(0,10,28),(0,10,29),(0,10,44),(0,10,45),(0,10,46),(0,10,48), (0,10,50),(0,11,29),(0,11,34),(0,11,36),(0,11,38),(0,11,40),(0,11,45),(0,12,26),(0,12,30)
running time: 0.01 second $\sim$ 0.04 second per (h_1, h_2, h_3)
List of some choices of (h_1, h_2, h_3, h_4) for $(n, d) = (129, 1)$
(0,1,6,54),(0,1,6,55),(0,1,6,56),(0,1,6,79),(0,1,6,80),(0,1,6,81),(0,1,7,27),(0,1,7,28), (0,1,7,29),(0,1,7,34),(0,1,7,35),(0,1,7,36),(0,1,7,40),(0,1,7,41),(0,1,7,54),(0,1,7,55), (0,1,7,56),(0,1,7,57),(0,1,7,79),(0,1,7,80),(0,1,7,81),(0,1,7,82),(0,1,7,95),(0,1,7,100), (0,1,7,101),(0,1,7,102),(0,1,7,103),(0,1,7,107),(0,1,7,108),(0,1,7,109),(0,1,8,28),(0,1,8,29)
running time: 0.01 second $\sim$ 0.04 second per (h_1, h_2, h_3, h_4)

The case $(n, d) = (129, 1)$. According to Corollary 2, for $(n, d) = (129, 1)$ which is the parameter used in MiMC, to ensure the sharp exponential growth, $w \geq 4$ has to hold. There are so many possible such (h_1, h_2, h_3, h_4) that can reach this goal, i.e. $\delta_7 = 128$ and $\delta_8 = 129$ can be achieved with these $(h_1, \ldots, h_4)$. For the running time, we observed that for each possible candidate of (h_1, h_2, h_3, h_4), it takes about $0.02 \sim 0.15$ second to output the solution π, as shown in Table 1.

To summarize, we presented a light strategy for efficiently determining whether a complex affine layer can be used to achieve the goal of the exponential/fastest growth of the algebraic degree. It is found that this problem can be converted into checking whether it is possible to select $\min\{2^r, n\}$ numbers from $r + 1$ sets under several constraints. As a result, we no more need to enumerate the set $\mathcal{P}_r$ as in [30], and test whether the best case occurs.

6 Degree Evaluation for Arbitrary Affine Layers

Until now, we mainly consider the problem from the designers' perspective, i.e. how to efficiently find proper $B(x)$ to achieve the fastest growth of the algebraic degree. If we consider the attackers' perspective, we then need to give an upper bound for δ_r for any $B(x)$.

The main obstacle here is how to avoid the enumeration of all possible $(f_1, \ldots, f_{2^r})$ and the corresponding vectors ν_r constructed with Algorithm 2. Once we can overcome it, we will be closer to efficiently finding a meaningful upper bound for δ_r for any $B(x)$.

6.1 Three Common Features

We avoid the enumeration of all possible $(\nu_{r,n-1}, \ldots, \nu_{r,0})$ by capturing the common features in them.

Feature 1. The first feature has been given by Theorem 2, i.e., each vector $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0})$ satisfies

$$\sum_{i=0}^{n-1} \nu_{r,i} = 2^r.$$

Feature 2. The second feature is given by Equation 15, i.e. each such vector $\nu_r = (\nu_{r,n-1}, \ldots, \nu_{r,0})$ satisfies

$$|\{i \mid \nu_{r,i} \neq 0,\ 0 \leq i \leq n-1\}| \leq \pi.$$

Feature 3. By the definition of $\mathcal{Z}$ as in Equation 9, we also know that the positions i for which $\nu_{r,i}$ can be different from zero must belong to $\mathcal{Z}$, that is,

$$\{i \mid \nu_{r,i} \neq 0,\ 0 \leq i \leq n-1\} \subseteq \mathcal{Z}.$$

6.2 The Problem Reduction

To give an upper bound for δ_r for any $B(x)$, we use the problem reduction, i.e. finding an equivalent problem to upper bound δ_r. With the above common features in all possible $(v_{r,n-1}, \ldots, \nu_{r,0})$, the problem reduces to

$$\begin{aligned}
&\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=0}^{n-1} 2^i a_i\right)\right) \\
&\text{subject to :} \\
&\forall i \in [0, n-1] : a_i \leq \nu_{r,i};\ \sum_{i=0}^{n-1} \nu_{r,i} = 2^r;\ |\{i \mid \nu_{r,i} \neq 0\}| \leq \pi;\ \{i \mid \nu_{r,i} \neq 0\} \subseteq \mathcal{Z}.
\end{aligned}$$

For convenience, let $\mathcal{Z} = \{p_1, \ldots, p_{|\mathcal{Z}|}\}$. Then, we can obtain an equivalent description of the above optimization problem by omitting the variables $(\nu_{r,n-1}, \ldots, \nu_{r,0})$, as shown below:

$$\begin{aligned}
&\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{|\mathcal{Z}|} 2^{p_i} a_{p_i}\right)\right), \\
&\text{subject to } \forall i \in [1, |\mathcal{Z}|] : a_{p_i} \geq 0;\ \sum_{i=1}^{|\mathcal{Z}|} a_{p_i} \leq 2^r;\ |\{p_i \mid a_{p_i} \neq 0\}| \leq \pi.
\end{aligned}$$

Hence, to upper bound δ_r for any given $B(x)$, the first step is to compute the set $\mathcal{Z}$ and the solution of the objective function π as shown in the above section. The next step is to solve the above optimization problem. It can be observed that this optimization problem is very similar to **OP-1**. The differences are (i) there is only a trivial upper bound for each a_{p_i}, i.e. $a_{p_i} \leq 2^r$; (ii) there is a concrete upper bound for the sum $\sum_{i=1}^{|\mathcal{Z}|} a_{p_i}$; and (iii) there is an upper bound for the number of nonzero a_{p_i}. In this sense, this optimization problem is more general and should be more difficult to solve. However, by using off-the-shelf solvers like Gurobi[9], we can still efficiently solve it. Modelling this optimization problem is almost the same as modelling **OP-1**, which has been fully described in [30]. For completeness, the pseudo-code for this general optimization problem can be referred to Algorithm 3 in Appendix A.

Remark 6. It can be observed that we consider all possible vectors ν_r satisfying the 3 common features. It is possible that some of them will not appear in the actual set of all possible ν_r obtained by the naive enumeration. Hence, the solution of the optimization problem is a possibly loose upper bound of δ_r.

6.3 Practical Tests

One interesting experiment is to see how the upper bound of the algebraic degree increases in the univariate setting when $w = 2$. We tried some (n, d, h_1, h_2) and the results are shown in Table 2. A graphic illustration is given in Fig. 3. Note that with the method [30] to enumerate $\mathcal{P}_r$ to find the upper bound, it can work for at most the first 6 rounds since the time complexity is $w^{2^{r-1}} = 2^{2^{r-1}}$ when $w = 2$. However, in the experiments with our technique, testing this even for 20 rounds takes less than 1 minute.

In particular, it is found that upper bound of the algebraic degree increases linearly for $(n, d, h_1, h_2) = (129, 1, 0, 63)$ and it increases by 7 each round after the 7th round. This is difficult to explain since $\log_2((2^1 + 1) \cdot 2^{63}) \approx 64.6$ is much larger than 7. We thus believe that more theoretic interpretations should be developed to understand this, which is left as an interesting problem.

As for the multivariate case, we should emphasize that once the maximal degree is reached for the univariate case, we will reach the maximal degree for the multivariate degree very fast, which can be seen from the attacks on Chaghri [30] and can also be seen from the relations between **OP-1** and **OP-t** which share the same vector ν_r. Details can be referred to Appendix B.

7 Future works

In this paper, we have developed a new technique to evaluate the growth of the algebraic degree for a special SPN cipher over $\mathbb{F}_{2^n}$. With this technique, we are able to answer some important related questions. Still, there are still some problems worth further investigating.

[9] `www.gurobi.com`

Table 2: Some experimental results for upper bounding the algebraic degree

(n, d, h_1, h_2)	r	≤ 3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
(63,32,0,3)	δ_r	2^r	15	27	42	52	62	63	–	–	–	–	–	–	–	–	–	–	–
time (seconds)		≤ 0.1	0.1	0.1	0.4	0.2	0.3	5.2	–	–	–	–	–	–	–	–	–	–	–
(63,32,0,6)	δ_r	2^r	15	27	47	63	–	–	–	–	–	–	–	–	–	–	–	–	–
time (seconds)		≤ 0.1	0.11	0.1	0.5	1.6	–	–	–	–	–	–	–	–	–	–	–	–	–
(63,32,0,36)	δ_r	2^r	15	27	48	63	–	–	–	–	–	–	–	–	–	–	–	–	–
time (seconds)		≤ 0.1	0.1	0.1	1.0	4.3	–	–	–	–	–	–	–	–	–	–	–	–	–
(129,1,0,6)	δ_r	2^r	15	27	40	49	$49 + 8 \times (r-7)$									129	–	–	–
time (seconds)		< 0.1	0.1	0.1	0.3	0.1	0.5	0.7	0.3	0.3	1.7	1.9	1.5	1.9	2.3	5.8	–	–	–
(129,1,0,9)	δ_r	2^r	15	27	48	66	78	89	100	111	122	129	–	–	–	–	–	–	–
time (seconds)		< 0.1	0.1	0.1	0.2	1.5	1.4	2.4	1.3	1.7	1.9	5	–	–	–	–	–	–	–
(129,1,0,36)	δ_r	2^r	15	27	48	72	94	116	127	129	–	–	–	–	–	–	–	–	–
time (seconds)		< 0.1	0.1	0.1	0.3	0.6	5.1	18.3	23.3	13.9	–	–	–	–	–	–	–	–	–
(129,1,0,42)	δ_r	2^r	15	26	37	47	57	66	75	84	93	102	111	120	129	–	–	–	–
time (seconds)		< 0.1	0.1	0.1	0.2	0.3	0.4	0.7	0.5	8.9	1.5	4.2	8.3	23.6	11.1	–	–	–	–
(129,1,0,63)	δ_r	2^r	15	24	32	40	$40 + 7 \times (r-7)$												129
time (seconds)		< 0.1	0.1	0.1	0.1	0.3	0.2	0.5	0.9	0.9	1.0	1.1	29.2	4.8	8.6	32.9	70.7	23.2	48.9

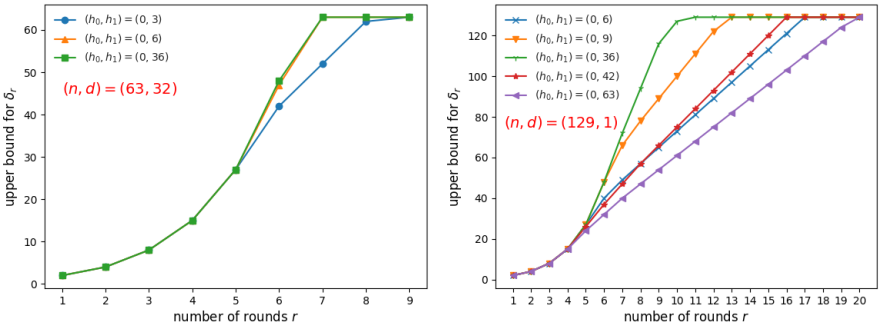

Fig. 3: Graphic illustration of the growth of the algebraic degree

Table 3: Comparison between the algebraic degree (obtained experimentally via "AZP" or "OZP") with the upper bound obtained with the strategy proposed in this paper ("our bound") for case1/case2/case3.

(n, d, h_1, h_2)	methods	case	# rounds				
			1	2	3	4	5
(129,1,0,6)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	23
		case2	2	4	8	13	23
		case3	2	4	8	14	23
(129,1,0,9)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	23
		case2	2	4	8	13	23
		case3	2	4	8	14	23
(129,1,0,36)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	23
		case2	2	4	8	13	23
		case3	2	4	8	14	23
(129,1,0,42)	our bound		2	4	8	15	26
	AZP/OZP	case1	2	4	8	14	22
		case2	2	4	8	13	21
		case3	2	4	8	14	22
(129,1,0,63)	our bound		2	4	8	15	24
	AZP/OZP	case1	2	4	8	14	20
		case2	2	4	8	13	20
		case3	2	4	8	14	20
(63,32,0,3)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	25
		case2	2	4	8	14	24
		case3	2	4	8	14	25
(63,32,0,6)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	25
		case2	2	4	8	14	24
		case3	2	4	8	14	25
(63,32,0,36)	our bound		2	4	8	15	27
	AZP/OZP	case1	2	4	8	14	25
		case2	2	4	8	14	24
		case3	2	4	8	14	25

First of all, as mentioned several times, the reduced well-structured optimization problems are solved via blackbox solvers. It is meaningful to develop efficient algorithms for them to further understand the efficiency of the new technique.

Secondly, we have mainly discussed how to efficiently find $B(x)$ such that the fastest growth of the algebraic degree can be reached and how to upper bound the algebraic degree for arbitrary $B(x)$. It is natural to ask how tight the upper bounds are.

In order to answer this question, we performed some practical experiments[10] on the studied SPN cipher parameterized with $m = 3$, where the coefficients $(c_1, \ldots, c_w)$ in $B(x)$ and the round constants/keys will vary in each experiment. For the matrix M, we considered three cases:

case1: M is a simple non-MDS matrix fixed as $[[0, 1, \beta], [\beta, 0, 1], [1, \beta, 0]]$ for the efficiency of the experiments, where β here denotes the root of the irreducible polynomial over $\mathbb{F}_{2^n}$.
case2: M is a simple MDS matrix fixed as $[[\beta, 1, 1 + \beta], [1, 1, 1], [1 + \beta, \beta, 1]]$ for the efficiency of the experiments.
case3: M is a randomly generated invertible matrix.

For finding the actual growth of the algebraic degree, we simply choose subspaces of dimension dim, and test whether the corresponding sums of the outputs are zero. In this way, we can deduce that the algebraic degree is dim -1 for the smallest dim that can make the sums of the outputs zero. For completeness, we test two different zero-sum properties:

- all-zero property (AZP): all the sums of the output words are zero;
- one-zero property (OZP): one of the sums of the output words is zero.

First, we test the parameters displayed in Table 1 to achieve the exponential growth of the algebraic degree as discussed in Sect. 5. In case1/case2/case3, we obtain that the algebraic degree at round i is exactly 2^i for $1 \leq i \leq 4$, no matter if it is detected by AZP or OZP. Due to the limitation of our computing resources[11], we could only verify this for the first 4 rounds.

Second, to bound the algebraic degree for arbitrary $B(x)$, we test the parameters considered in Sect. 6 for up to 5 rounds. It is found that the upper bounds are tight for the first 3 rounds and become loose in the 4th and 5th rounds, though the gap is not so large, as shown in Table 3. This result is expected, since the model used to evaluate the upper bound of the algebraic degree for arbitrary $B(x)$ is not perfect due to the usage of relaxed constraints. We leave the problem to improve it without affecting its efficiency too much open for future work.

Moreover, it can also be observed from Table 3 that the same algebraic degree is obtained by detecting AZP or OZP. We also observe that the algebraic degree obtained in the 3 cases is only slightly different, which somehow demonstrates

[10] The source code is available at `https://github.com/LFKOKAMI/CG_test`.

[11] It turns out that case2 consumes much more time than case1/case2. Encrypting 2^{23} inputs takes about 1 hour in case1/case2, while encrypting 2^{21} inputs takes about 1 hour in case3.

the influence of M on the growth of the algebraic degree, though we could not capture it with our technique. This is also an interesting problem for further study.

Acknowledgments. We also thank the reviewers of CRYPTO 2023 for providing many insightful comments. Fukang Liu is supported by Grant-in-Aid for Research Activity Start-up (Grant No. 22K21282). Lorenzo Grassi is supported by the German Research Foundation (DFG) within the framework of the Excellence Strategy of the Federal Government and the States – EXC 2092 CaSa – 39078197. Takanori Isobe is supported by JST, PRESTO Grant Number JPMJPR2031. These research results were also obtained from the commissioned research (No.05801) by National Institute of Information and Communications Technology (NICT), Japan.

References

1. M. R. Albrecht, C. Cid, L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, and M. Schofnegger. Algebraic Cryptanalysis of STARK-Friendly Designs: Application to MARVELlous and MiMC. In *ASIACRYPT (3)*, volume 11923 of *LNCS*, pages 371–397. Springer, 2019.
2. M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In *ASIACRYPT (1)*, volume 10031 of *LNCS*, pages 191–219, 2016.
3. M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. *IACR Cryptol. ePrint Arch.*, page 687, 2016.
4. T. Ashur, M. Mahzoun, and D. Toprakhisar. Chaghri - A FHE-friendly Block Cipher. In *CCS*, pages 139–150. ACM, 2022.
5. T. Beyne, A. Canteaut, I. Dinur, M. Eichlseder, G. Leander, G. Leurent, M. Naya-Plasencia, L. Perrin, Y. Sasaki, Y. Todo, and F. Wiemer. Out of Oddity - New Cryptanalytic Techniques Against Symmetric Primitives Optimized for Integrity Proof Systems. In *CRYPTO (3)*, volume 12172 of *LNCS*, pages 299–328. Springer, 2020.
6. C. Boura, A. Canteaut, and C. D. Cannière. Higher-Order Differential Properties of Keccak and *Luffa*. In *FSE*, volume 6733 of *LNCS*, pages 252–269. Springer, 2011.
7. C. Bouvier, A. Canteaut, and L. Perrin. On the algebraic degree of iterated power functions. *Des. Codes Cryptogr.*, 91(3):997–1033, 2023.
8. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Trans. Comput. Theory*, 6(3):13:1–13:36, 2014.
9. A. Canteaut, S. Carpov, C. Fontaine, T. Lepoint, M. Naya-Plasencia, P. Paillier, and R. Sirdey. Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. *J. Cryptol.*, 31(3):885–916, 2018.
10. J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *ASIACRYPT (1)*, volume 10624 of *LNCS*, pages 409–437. Springer, 2017.
11. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 Seconds. In *ASIACRYPT (1)*, volume 10031 of *LNCS*, pages 3–33, 2016.

12. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast Fully Homomorphic Encryption Over the Torus. *J. Cryptol.*, 33(1):34–91, 2020.
13. J. Cho, J. Ha, S. Kim, B. Lee, J. Lee, J. Lee, D. Moon, and H. Yoon. Transciphering Framework for Approximate Homomorphic Encryption. In *ASIACRYPT (3)*, volume 13092 of *LNCS*, pages 640–669. Springer, 2021.
14. C. Cid, L. Grassi, A. Gunsing, R. Lüftenegger, C. Rechberger, and M. Schofnegger. Influence of the Linear Layer on the Algebraic Degree in SP-Networks. *IACR Trans. Symmetric Cryptol.*, 2022(1):110–137, 2022.
15. O. Cosseron, C. Hoffmann, P. Méaux, and F.-X. Standaert. Towards Globally Optimized Hybrid Homomorphic Encryption - Featuring the Elisabeth Stream Cipher. Cryptology ePrint Archive, Paper 2022/180, 2022. `https://eprint.iacr.org/2022/180`.
16. J. Cui, K. Hu, M. Wang, and P. Wei. On the Field-Based Division Property: Applications to MiMC, Feistel MiMC and GMiMC. In *ASIACRYPT (3)*, volume 13793 of *Lecture Notes in Computer Science*, pages 241–270. Springer, 2022.
17. I. Dinur. Cryptanalytic Applications of the Polynomial Method for Solving Multivariate Equation Systems over GF(2). In *EUROCRYPT (1)*, volume 12696 of *LNCS*, pages 374–403. Springer, 2021.
18. I. Dinur, Y. Liu, W. Meier, and Q. Wang. Optimized Interpolation Attacks on LowMC. In *ASIACRYPT (2)*, volume 9453 of *LNCS*, pages 535–560. Springer, 2015.
19. C. Dobraunig, M. Eichlseder, L. Grassi, V. Lallemand, G. Leander, E. List, F. Mendel, and C. Rechberger. Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In *CRYPTO (1)*, volume 10991 of *LNCS*, pages 662–692. Springer, 2018.
20. C. Dobraunig, L. Grassi, L. Helminger, C. Rechberger, M. Schofnegger, and R. Walch. Pasta: A Case for Hybrid Homomorphic Encryption. Cryptology ePrint Archive, Paper 2021/731, 2021. `https://eprint.iacr.org/2021/731`.
21. C. Dobraunig, D. Kales, C. Rechberger, M. Schofnegger, and G. Zaverucha. Shorter Signatures Based on Tailor-Made Minimalist Symmetric-Key Crypto. In *CCS*, pages 843–857. ACM, 2022.
22. S. Duval, V. Lallemand, and Y. Rotella. Cryptanalysis of the FLIP Family of Stream Ciphers. In *CRYPTO (1)*, volume 9814 of *LNCS*, pages 457–475. Springer, 2016.
23. M. Eichlseder, L. Grassi, R. Lüftenegger, M. Øygarden, C. Rechberger, M. Schofnegger, and Q. Wang. An Algebraic Attack on Ciphers with Low-Degree Round Functions: Application to Full MiMC. In *ASIACRYPT (1)*, volume 12491 of *LNCS*, pages 477–506. Springer, 2020.
24. C. Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In *STOC*, pages 169–178. ACM, 2009.
25. C. Gentry, S. Halevi, and N. P. Smart. Homomorphic Evaluation of the AES Circuit. In *CRYPTO*, volume 7417 of *LNCS*, pages 850–867. Springer, 2012.
26. C. Gentry, A. Sahai, and B. Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *CRYPTO (1)*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.
27. J. Ha, S. Kim, W. Choi, J. Lee, D. Moon, H. Yoon, and J. Cho. Masta: An HE-Friendly Cipher Using Modular Arithmetic. *IEEE Access*, 8:194741–194751, 2020.
28. P. Hebborn and G. Leander. Dasta - Alternative Linear Layer for Rasta. *IACR Trans. Symmetric Cryptol.*, 2020(3):46–86, 2020.

29. X. Lai. *Higher Order Derivatives and Differential Cryptanalysis*, pages 227–233. Springer US, Boston, MA, 1994.
30. F. Liu, R. Anand, L. Wang, W. Meier, and T. Isobe. Coefficient Grouping: Breaking Chaghri and More. In *EUROCRYPT (4)*, volume 14007 of *Lecture Notes in Computer Science*, pages 287–317. Springer, 2023.
31. F. Liu, T. Isobe, and W. Meier. Cryptanalysis of Full LowMC and LowMC-M with Algebraic Techniques. In *CRYPTO (3)*, volume 12827 of *LNCS*, pages 368–401. Springer, 2021.
32. F. Liu, W. Meier, S. Sarkar, and T. Isobe. New Low-Memory Algebraic Attacks on LowMC in the Picnic Setting. *IACR Trans. Symmetric Cryptol.*, 2022(3):102–122, 2022.
33. F. Liu, S. Sarkar, W. Meier, and T. Isobe. Algebraic Attacks on Rasta and Dasta Using Low-Degree Equations. In *ASIACRYPT (1)*, volume 13090 of *LNCS*, pages 214–240. Springer, 2021.
34. F. Liu, S. Sarkar, W. Meier, and T. Isobe. The Inverse of χ and Its Applications to Rasta-Like Ciphers. *J. Cryptol.*, 35(4):28, 2022.
35. F. Liu, S. Sarkar, G. Wang, W. Meier, and T. Isobe. Algebraic Meet-in-the-Middle Attack on LowMC. In *ASIACRYPT (1)*, volume 13791 of *Lecture Notes in Computer Science*, pages 225–255. Springer, 2022.
36. F. Liu and L. Wang. An $\mathcal{O}(n)$ Algorithm for Coefficient Grouping. Cryptology ePrint Archive, Paper 2022/992, 2022. `https://eprint.iacr.org/2022/992`.
37. P. Méaux, A. Journault, F. Standaert, and C. Carlet. Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. In *EUROCRYPT (1)*, volume 9665 of *LNCS*, pages 311–343. Springer, 2016.
38. M. Naehrig, K. E. Lauter, and V. Vaikuntanathan. Can Homomorphic Encryption be Practical? In *CCSW*, pages 113–124. ACM, 2011.
39. K. Nyberg. Differentially Uniform Mappings for Cryptography. In *EUROCRYPT*, volume 765 of *LNCS*, pages 55–64. Springer, 1993.
40. F. Rodríguez-Henríquez, A. D. Pérez, N. A. Saqib, and Çetin Kaya Koç. *Cryptographic Algorithms on Reconfigurable Hardware*. Springer US, Boston, MA, 2007.
41. Y. Todo. Structural Evaluation by Generalized Integral Property. In *EUROCRYPT (1)*, volume 9056 of *LNCS*, pages 287–314. Springer, 2015.

A The Pseudo-code for the General Optimization Problem

In this section, we present a high-level description of the MILP model to solve the general optimization problem, as shown in Algorithm 3.

We give some explanations for the code:

- The function `MODADD` is used to model the modular addition operation

$$M_n(2^{s_a} \cdot a + 2^{s_b} \cdot b),$$

  where a and b are both n bits.
- The function `ADD` is used to model the common addition operation

$$a + b,$$

  where a and $a + b$ are at most l_a and len bits, respectively.

Algorithm 3 The MILP model for the general optimization problem where $\mathcal{Z} = (p[0], \ldots, p[|\mathcal{Z}| - 1])$.

procedure ModAdd(model, $a, s_a, b, s_b, c_1, c_2, d_1, d_2, n$)//$d_2 = M_n(2^{s_a}a + 2^{s_b}b)$
    model.addConstr($c_1[0] = 0$)
    for all $i \in [0, n-1]$ **do**
        model.addConstr($a[(i - s_a)\%n] + b[(i - s_b)\%n] + c_1[i] = d_1[i] + 2 \times c_1[i+1]$)
    model.addConstr($c_2[0] = c_1[n]$)
    for all $i \in [0, n-1]$ **do**
        model.addConstr($d_1[i] + c_2[i] = d_2[i] + 2 \times c_2[i+1]$)

procedure Add(model, a, b, c, d, l_a, len)//$d = a + b$
    model.addConstr($c[0] = 0$)
    for all $i \in [0, l_a - 1]$ **do**
        model.addConstr($a[i] + b[i] + c[i] = d[i] + 2 \times c[i+1]$)
    for all $i \in [l_a, len - 1]$ **do**
        model.addConstr($b[i] + c[i] = d[i] + 2 \times c[i+1]$)

procedure Model($\pi, \mathcal{Z}, n, r$)
    $len = \lceil \log_2 n \rceil + r + 1$
    $k = |\mathcal{Z}|$
    Claim 2-dimensional binary vectors a, su_m, su_c, ca_m, ca_c
    //$su_m[2 \times i + 2] = M_n(su_m[2 \times i] + 2^{p[i]}a_i)$
    //$su_m[2 \times k] = M_n(\sum_{i=0}^{k-1} 2^{p[i]}a_i)$
    //$su_c[i+1] = su_c[i] + a_i$
    //$su_c[k] = \sum_{i=0}^{k-1} a_i$
    Claim a binary vector u of size k
    for all $i \in [r+1, n-1], j \in [0, k-1]$ **do**
        model.addConstr($a[j][i] = 0$)//$a[j] \leq 2^r$
    for all $i \in [0, n-1]$ **do**
        model.addConstr($su_m[0][i] = 0$)
    for all $i \in [0, len - 1]$ **do**
        model.addConstr($su_c[0][i] = 0$)
    for all $i \in [0, k-1]$ **do**
        $j = 2 \times i$
        MODADD(model, $su_m[j], 0, a[i], p[i], ca_m[j], ca_m[j+1], su_m[j+1], su_m[j+2], n$)
        ADD(model, $a[i], su_c[i], ca_c[i], su_c[i+1], r+1, len$)
    for all $i \in [r+1, len - 1]$ **do**
        model.addConstr($su_c[k][i] = 0$)//$su_c[k] \leq 2^r$
    for all $i \in [0, r-1]$ **do**
        model.addConstr($su_c[k][i] + su_c[k][r] \leq 1$)//$su_c[k] \leq 2^r$
    for all $i \in [0, k-1]$ **do**
        for all $j \in [0, n-1]$ **do**
            model.addConstr($u[i] \geq a[i][j]$)
        model.addConstr($u[i] \leq \sum_{j=0}^{n-1} a[i][j]$)
    model.addConstr($\sum_{i=0}^{k-1} u[i] \leq \pi$)//$|\{i \mid a[i] \neq 0\}| \leq \pi$
    model.setObjective($\sum_{i=0}^{n-1} su_m[2 \times k][i]$, MAXIMIZE)

– The codes in Line 31 – Line 34 correspond to modelling

$$M_n \left(\sum_{i=0}^{|\mathcal{Z}|-1} 2^{p[i]} \cdot a_i \right)$$

and

$$\sum_{i=0}^{|\mathcal{Z}|-1} a_i.$$

– The codes in Line 35 – Line 38 correspond to the constraint

$$\sum_{i=0}^{|\mathcal{Z}|-1} a_i \leq 2^r.$$

– The codes in Line 39 – Line 43 correspond to the constraint

$$|\{i \mid a[i] \neq 0\}| \leq \pi.$$

B The Multivariate Case

As final step, we re-consider the proposed results for the multivariate case.

B.1 From the Univariate to the Multivariate Setting

We point out that if the maximum degree is reached in the univariate case, then it is very likely that it is reached in the multivariate case as well.

Maximum degree in the univariate case: impact on the multivariate case. As already mentioned, once the maximal degree is reached for the univariate case, we will reach the maximal degree for the multivariate degree very fast. Specifically, once the maximal degree is reached at round r, then there exists a vector ν_r that – after the reduction algorithm [36] – satisfies

$$\nu_{r,i} \geq 1, \forall i \in [0, n-1].$$

Then, at round $r + r'$, there must exist a vector that – after the reduction algorithm – satisfies

$$\nu_{r,i} \geq 2^{r'}, \forall i \in [0, n-1].$$

This is because, according to the findings in [30], all the vectors in $\mathcal{P}_{r+1}$ are computed by summing up all possible combinations of two vectors in $\mathcal{P}_r$, which can also be observed from the proof of Theorem 1, especially from the construction of the set $\mathcal{W}_q^S$ from $\mathcal{W}_q$. Then, further according to the reduction algorithm [36], it implies that maximal degree tn must be reached if $t \leq 2^{r'}$ at most at round $r + r'$. (Note that the maximal degree tn may be reached before round $r + r'$.)

Multivariate setting. Here, we develop techniques for the multivariate case, which is going to be a simple extension of the techniques for the univariate case.

In the multivariate setting, we consider the input $(x_1, \ldots, x_l)$ such that each x_i is linear in t variables $(y_1, \ldots, y_t)$. Similarly, we use a set $\mathcal{W}_r^S$ to describe the tuples of exponents $(e_1, \ldots, e_t)$ of all possible monomials (where each monomial is of the form $\prod_{i=1}^{t} y_i^{e_i}$) in the multivariate polynomial representation after the S-box operation in the r-th round. In this way, we have

$$\mathcal{W}_r^S = \left\{ (e_1, \ldots, e_t) \mid e_u = \sum_{i=0}^{r} \sum_{j=1}^{\binom{r}{i}} a_{u,r,v} 2^{(r-i)d+f_v}, \right.$$
$$\left. v = j + \binom{r}{\leq i-1}, 0 \leq \sum_{u=1}^{t} a_{u,r,v} \leq 1, f_v \in \mathcal{V}_{r,w} \right\},$$

where $(f_1, \ldots, f_{2^r})$ still satisfy Equation 8. This can be proved in the same way as that for the univariate case. Indeed, we only need to add a part to trace the condition $0 \leq \sum_{u=1}^{t} a_{u,r,v} \leq 1$ in the proof. Some intuitive explanations for the first 2 round are given below.

Let $\mathcal{W}_r$ denote the set of all possible tuples of exponents after r rounds. In this way, we have

$$\mathcal{W}_0 = \{(e_1, \ldots, e_t) \mid 0 \leq e_1 + \ldots + e_t \leq 1\},$$
$$\mathcal{W}_1^S = \left\{ (e_1, \ldots, e_t) \mid e_u = a_{u,1,1} 2^d + a_{u,1,2}, 0 \leq \sum_{u=1}^{t} a_{u,1,1} \leq 1, 0 \leq \sum_{u=1}^{t} a_{u,1,2} \leq 1 \right\},$$
$$\mathcal{W}_1 = \left\{ (e_1, \ldots, e_t) \mid e_u = a_{u,1,1} 2^{d+h_i} + a_{u,1,2} 2^{h_i}, \right.$$
$$\left. 0 \leq \sum_{u=1}^{t} a_{u,1,1} \leq 1, 0 \leq \sum_{u=1}^{t} a_{u,1,2} \leq 1, 1 \leq i \leq w \right\},$$
$$\mathcal{W}_2^S = \left\{ (e_1, \ldots, e_t) \mid e_u = \sum_{i=0}^{2} \sum_{j=1}^{\binom{1}{i}} a_{u,2,v} 2^{(2-i)d+f_v}, \right.$$
$$\left. v = j + \binom{r}{\leq i-1}, 0 \leq \sum_{u=1}^{t} a_{u,2,v} \leq 1, f_v \in \mathcal{V}_{2,w} \right\},$$

Based on this representation, we can also construct a vector ν_r for each assignment to $(f_1, \ldots, f_{2^r})$ via Algorithm 2. For each obtained vector ν_r, bounding the algebraic degree can be reduced to solving **OP-t** parametrized by this ν_r.

B.2 Achieving the Exponential Increase for the Multivariate Case

In order to achieve exponential growth of the algebraic degree in the case of t variables $(y_1, y_2, \ldots, y_t)$, at each round r, two conditions must be satisfied:

1. the maximal degree $(t-1)n$ for variables $(y_1, \ldots, y_{t-1})$ must be reached;
2. the maximal degree $\min\{2^r, tn\}$ is reached for variables $(y_1, \ldots, y_t)$.

If the first condition cannot hold, we can always mount a higher-order differential attack by considering $t-1$ variables. In such a case, the time complexity will not exceed $2^{(t-1)n}$, in which case there is no need to consider attacks with t variables. If the second condition cannot hold, the algebraic degree either does not increase exponentially or we can attack r rounds with time complexity 2^{tn}.

Let us define $(r+1)t$ rows $\mathrm{Row}-(1,1), \ldots, \mathrm{Row}-(t,1), \ldots, \mathrm{Row}-(t,r+1)$ as below:

$$
\begin{aligned}
&\mathrm{Row}-(u,1):\ a_{u,r,1}(rd+f_1)\%n,\\
&\ldots,\\
&\mathrm{Row}-(u,i+1):\ a_{u,r,\binom{r}{\le i-1}+1}\left((r-i)d+f_{\binom{r}{\le i-1}+1}\right)\%n,\ldots,\\
&\qquad\qquad a_{u,r,\binom{r}{\le i-1}+\binom{r}{i}}\left((r-i)d+f_{\binom{r}{\le i-1}+\binom{r}{i}}\right)\%n,\\
&\ldots,\\
&\mathrm{Row}-(u,r+1):\ a_{u,r,2^r}f_{2^r}\%n.
\end{aligned}
$$

Moreover, we define the $r+1$ rows $(\mathrm{Row}-(i,1), \ldots, \mathrm{Row}-(i,r+1))$ as Slice$-i$ and define the t rows $(\mathrm{Row}-(1,i), \ldots, \mathrm{Row}-(t,i))$ as Lane$-i$. In this way, there are in total t slices and 2^r lanes. We suggest to have a 3-dimensional structure in mind when studying the multivariate case.

Achieving the above goal for the growth of the algebraic degree is equivalent to check whether there exists a valid assignment to $(f_1, \ldots, f_{2^r})$ and $(a_{1,r,1}, \ldots, a_{t,r,2^r})$ such that

Condition 1: In each Lane$-i$, there is $\sum_{u=1}^{t} a_{u,r,i} \le 1$. In this way, the condition $\sum_{u=1}^{t} a_{u,r,i} \le 1$ for $\forall i \in [1, 2^r]$ is satisfied. Note that there are at most 2^r nonzero coefficients among all the $t2^r$ coefficients $(a_{1,r,1}, \ldots, a_{t,r,2^r})$ due to this constraint.

Condition 2: For each $i \in [1, t-1]$, in Slice$-i$, there should exist n nonzero coefficients among all the 2^r coefficients $(a_{i,r,1}, \ldots, a_{i,r,2^r})$, i.e. $\sum_{j=1}^{2^r} a_{i,r,j} = n$. Moreover, for the 2^r numbers in Slice$-i$, those n numbers with nonzero $a_{i,r,j}$ should be different from each other. In this way, the first condition is satisfied, i.e. the maximal degree $(t-1)n$ is reached for variables $(y_1, \ldots, y_{t-1})$.

Condition 3: For Slice$-t$, there are $2^r - (t-1)n$ nonzero coefficients among $(a_{t,r,1}, \ldots, a_{t,r,2^r})$ and those $2^r - (t-1)n$ numbers in Slice$-t$ with nonzero $a_{t,r,j}$ should include $\min\{2^r-(t-1)n, n\}$ different elements. In this way, the second condition will be satisfied that the maximal degree $\min\{2^r, tn\}$ is reached for variables $(y_1, \ldots, y_t)$.

Building the model. Similar to the univariate case, we precompute $r+1$ arrays $A_1, \ldots, A_{r+1}$. For each vector A_i, we introduce t additional binary arrays

$A^{up}_{1,i}, \ldots, A^{up}_{t,i}$ and another t additional arrays $A^{dw}_{1,i}, \ldots, A^{dw}_{t,i}$, all of which are of length n. Moreover, we also introduce t binary arrays $X_1, X_2, \ldots, X_t$ of length n. Specifically, $A^{up}_{u,i}/A^{dw}_{u,i}$ is used to represent the choice for the first/last $\binom{r-1}{i-2}/\binom{r-1}{i-1}$ elements in Row$-(u,i)$, while X_i is used to record the choice for Slice$-i$.

The relations inside these binary arrays are specified as:

$$A^{up}_{u,1}[j] = A^{dw}_{u,r+1}[j] = 0, \ \forall j \in [0, n-1], \forall u \in [1, t],$$
$$X_u[j] \geq A^{up}_{u,i}[j] + A^{dw}_{u,i}[j], \ \forall i \in [1, r+1], \forall j \in [0, n-1], \forall u \in [1, t],$$
$$X_u[j] \leq \sum_{i=0}^{r} (A^{up}_{u,i}[j] + A^{dw}_{u,i}[j]), \ \forall i \in [1, r+1], \forall j \in [0, n-1], \forall u \in [1, t]$$

The inner conditions in Equation 8 are described by the following constraints:

$$\sum_{u=1}^{t} A^{up}_{u,i+1}[j] = \sum_{u=1}^{t} A^{dw}_{u,i}[(j+d)\%n], \ \forall i \in [1, r], \forall j \in [0, n-1]. \tag{16}$$

Moreover, since we can choose at most $\binom{r}{i-1}$ numbers from A_i, it follows that

$$\sum_{j=0}^{n-1} (A^{up}_{u,i}[j] + A^{dw}_{u,i}[j]) \leq \binom{r}{i-1}, \ \forall i \in [1, r+1], \forall u \in [1, t].$$

Due to Equation 16, this constraint is equivalent to

$$\sum_{j=0}^{n-1} A^{dw}_{u,i}[j] \leq \binom{r-1}{i-1}, \ \forall i \in [1, r+1], \forall u \in [1, t]. \tag{17}$$

Similar to the univariate case:

- we also have

  $\forall i \in [1, r+1], \forall j \in [0, n-1], \forall u \in [1, t] : A_{u,i}[j]^{up} = A_{u,i}[j]^{dw} = 0$ if $A_i[j] = 0$;
- Condition 1 is equivalent to

  $$\sum_{u=1}^{t} A^{up}_{u,i}[j] \leq 1, \ \sum_{u=1}^{t} A^{dw}_{u,i}[j] \leq 1, \ \forall i \in [1, r+1], \forall j \in [0, n-1];$$
- Condition 2 is equivalent to

  $$\sum_{j=0}^{n-1} X_u[j] = n, \ \forall u \in [1, t-1].$$
- to check Condition 3, we need to check if $\min\{2^r - (t-1)n, n\}$ is the solution of the objective function:

  $$\text{maximize} \sum_{j=0}^{n-1} X_t[j].$$

If this is the case, the used affine layer parameterized by $(h_1, \ldots, h_w)$ can help achieve the sharp exponential growth. Otherwise, we should not use such an affine layer.

B.3 Degree Evaluation for Arbitrary Affine Layers for t Variables

It has been stated above that for each assignment to $(f_1, \ldots, f_{2^r})$, bounding the algebraic degree in the case of t variables is equivalent to solving **OP-t** parametrized by the vector ν_r derived from this assignment.

We observe that the three common features for all possible vectors ν_r in the univariate case still hold in the multivariate case because they are only related to the construction of the set $\mathcal{V}_{r,w}$. Hence, similarly, we can consider a general optimization problem that can cover all possible such vectors ν_r, as stated below:

$$\begin{aligned}
&\text{maximize } \sum_{u=1}^{t} \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{|\mathcal{Z}|} 2^{p_i} a_{u,p_i}\right)\right),\\
&\text{subject to}\\
&a_{u,p_i} \geq 0 \ \forall i \in [1, |\mathcal{Z}|], \forall u \in [1,t], \sum_{u=1}^{t}\sum_{i=1}^{|\mathcal{Z}|} a_{u,p_i} \leq 2^r,\ |\{p_i \mid a_{u,p_i} \neq 0\}| \leq \pi.
\end{aligned}$$

Moreover, when considering attacks using t variables, we should only care about the case when the maximal degree $(t-1)n$ has been reached for the first $t-1$ variables. Hence, we indeed only need to consider the following general optimization problem called **OP-M**:

$$\begin{aligned}
&\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{|\mathcal{Z}|} 2^{p_i} a_{t,p_i}\right)\right),\\
&\text{subject to } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{|\mathcal{Z}|} 2^{p_i} a_{j,p_i}\right)\right) = n \ \forall j \in [1, t-1],\\
&\qquad a_{u,p_i} \geq 0 \ \forall i \in [1, |\mathcal{Z}|], \forall u \in [1,t], \sum_{u=1}^{t}\sum_{i=1}^{|\mathcal{Z}|} a_{u,p_i} \leq 2^r,\ |\{p_i \mid a_{u,p_i} \neq 0\}| \leq \pi.
\end{aligned}$$

This optimization problem is too heavy. Indeed, we can solve it step by step. First, solve the following optimization problem called **OP-S**:

$$\begin{aligned}
&\text{minimize } \sum_{i=1}^{|\mathcal{Z}|} a_{p_i},\\
&\text{subject to } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{\kappa} 2^{p_i} a_{p_i}\right)\right) = n, a_{p_i} \geq 0 \ \forall i \in [1, |\mathcal{Z}|], |\{p_i \mid a_{p_i} \neq 0\}| \leq \pi.
\end{aligned}$$

It is difficult to model the objective function of **OP-S**. Hence, we can use a binary-search-like method to traverse a threshold $\mathcal{T}$ and for each $\mathcal{T}$, we solve the following optimization problem called **OP-B**:

$$\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{\kappa} 2^{p_i} a_{p_i}\right)\right),$$

$$\text{subject to } \sum_{i=1}^{|\mathcal{Z}|} a_{p_i} \leq \mathcal{T}, a_{p_i} \geq 0 \ \forall i \in [1, |\mathcal{Z}|], |\{p_i \mid a_{p_i} \neq 0\}| \leq \pi.$$

If the solution of **OP-B** is larger than n, we decrease $\mathcal{T}$. If the solution is smaller than n, we increase $\mathcal{T}$. Finally, we will reach the minimal $\mathcal{T} = \mathcal{T}_1$ that can make the solution of **OP-B** be n. Thus, we need to solve **OP-B** for $\log_2(2^r) = r$ times. Note that by sacrificing the accuracy for the efficiency, we can also only consider $\mathcal{T}$ of the form 2^i where $i \in [0, r]$ and aim to find the maximal $\mathcal{T} = \mathcal{T}_2$ such that the solution of **OP-B** is smaller than n.

After this step, we can solve the following optimization problem called **OP-F**:

$$\begin{aligned}
&\text{maximize } \mathrm{Hw}\left(M_n\left(\sum_{i=1}^{\kappa} 2^{p_i} a_{p_i}\right)\right), \\
&\text{subject to } \sum_{i=1}^{|\mathcal{Z}|} a_{p_i} \leq 2^r - (t-1)\mathcal{T}, a_{p_i} \geq 0 \ \forall i \in [1, |\mathcal{Z}|], |\{p_i \mid a_{p_i} \neq 0\}| \leq \pi.
\end{aligned}$$

Especially, if $\mathcal{T} = \mathcal{T}_1$, the solution of **OP-F** is identical to the solution of **OP-M**. If $\mathcal{T} = \mathcal{T}_2$, the solution of **OP-F** is an upper bound for the solution of **OP-M**.

It can be observed that **OP-F** and **OP-B** are almost the same, i.e. they only vary in the constraint on the sum $\sum_{i=1}^{|\mathcal{Z}|} a_{p_i}$. In a word, by reduction, upper bounding the algebraic degree in the multivariate case is again reduced to solving a well-structured optimization problem of the form as **OP-F** and **OP-B**. It is thus interesting to investigate whether there are efficient ad-hoc algorithms for this optimization problem.