Coefficient Grouping: Breaking Chaghri and More

Fukang Liu[1,2], Ravi Anand[2], Libo Wang[4,2], Willi Meier[5], Takanori Isobe[2,3]

[1] Tokyo Institute of Technology, Tokyo, Japan,
[2] University of Hyogo, Hyogo, Japan
[3] NICT, Tokyo, Japan
[4] Jinan University, Guangzhou, China
[5] FHNW, Windisch, Switzerland
liufukangs@gmail.com, ravianandsps@gmail.com, wanglibo12b@gmail.com,
willi.meier@fhnw.ch, takanori.isobe@ai.u-hyogo.ac.jp

Abstract. We propose an efficient technique called coefficient grouping to evaluate the algebraic degree of the FHE-friendly cipher Chaghri, which has been accepted for ACM CCS 2022. It is found that the algebraic degree increases linearly rather than exponentially. As a consequence, we can construct a 13-round distinguisher with time and data complexity of 2^{63} and mount a 13.5-round key-recovery attack. In particular, a higher-order differential attack on 8 rounds of Chaghri can be achieved with time and data complexity of 2^{38}. Hence, it indicates that the full 8 rounds are far from being secure. Furthermore, we also demonstrate the application of our coefficient grouping technique to the design of secure cryptographic components. As a result, a countermeasure is found for Chaghri and it has little overhead compared with the original design. Since more and more symmetric primitives defined over a large finite field are emerging, we believe our new technique can have more applications in the future research.

Keywords: Chaghri, degree evaluation, coefficient grouping, optimization problem, finite field

1 Introduction

In recent years, there is a new trend to design symmetric-key primitives for advanced protocols like secure multi-party computation (MPC), fully homomorphic encryption (FHE) and zero-knowledge proof systems (ZK) [2,3,4,5,10,12,15,16,17,20,21,23,29]. This is mainly motivated by the fact that traditional symmetric-key primitives like AES and SHA-2/SHA-3 are not efficient in these protocols. Therefore, when designing new symmetric-key primitives for them, designers need to be aware of the features of the target MPC/FHE/ZK schemes, e.g. which operations are cost-free and which are costly. For example, for many FHE schemes, a symmetric-key primitive with low multiplicative depth in decryption is desired.

It has been noticed by Canteaut et al. [10] that stream ciphers are a practical solution for efficient homomorphic ciphertext compression and many such stream ciphers have been proposed since then, like Kreyvrium [10], FLIP [29], Rasta [15], Dasta [23], Fasta [12], Masta [21] and Pasta [17]. Among them, Kreyvrium, FLIP, Rasta, Dasta, Fasta are designed over $\mathbb{F}_2$ while Masta and Pasta are designed over $\mathbb{F}_p$ where p is a large prime number. At ACM CCS 2022, an FHE-friendly block cipher called Chaghri [6] defined over $\mathbb{F}_{2^{63}}$ was proposed and it can outperform AES by about 65%.

Along with the new proposals, new cryptanalytic techniques have also been developed. There are some practical examples that several such primitives are broken with new cryptanalytic techniques. Specifically, the variant of MiMC designed over $\mathbb{F}_{2^n}$ is vulnerable against the higher-order differential attack [19]. Jarvis and Friday designed over a large finite field can be broken by Gröbner basis attacks [1]. The first version of FLIP can be practically broken by guess-and-determine attacks [18]. Some important parameters of LowMC and Agrasta are also shown to be insecure against algebraic attacks [14, 25, 26, 27, 30].

Due to the above design-and-break game, cryptographers have started to realize the importance to enrich the pool of cryptanalytic techniques for these new designs. Especially, as many such primitives are defined over a large finite field, it has become urgent to fill the shortcomings of the corresponding cryptanalytic techniques. At CRYPTO 2020, a major breakthrough was made where the higher-order differential attack was extended to finite fields of any characteristics [7]. At the same time, a more refined higher-order differential attack over $\mathbb{F}_{2^n}$ was discovered at ASIACRYPT 2020 [19]. These higher-order differential attacks rely on the degree evaluation. However, in both [7] and [19], the degree is computed in a rather straightforward way and they mainly exploit the low degree of the S-box, i.e. the S-box $x \mapsto x^3$. Although there are some follow-up works [9, 11], the corresponding general results still have some limitations and the degree evaluation still seems somewhat straightforward.

Some related works. Let us consider a MiMC-like construction defined over $\mathbb{F}_{2^n}$ with an S-box $x \mapsto x^d$ where $d = 2^j + 1$. Then, the general results in [9, 11] show that the algebraic degree after r rounds is upper bounded by $\lfloor r\log_2 d \rfloor - j + 1 \approx (r-1)j + 1$. This is obviously ineffective when j is large and n is small as $n < \lfloor r\log_2 d \rfloor - j + 1$ needs to hold to construct a meaningful higher-order differential distinguisher. However, as $(n, d) = (129, 3)$ is one parameter of MiMC, this is indeed quite effective and it implies that the algebraic degree increases linearly. Note that this was first observed in [19] and later generalized in [11].

In [11], the authors considered the SPN ciphers over $\mathbb{F}_{2^n}$. Although Chaghri is also based on the SPN structure, we emphasize that our method is still quite different from [11] and this will be very clear later. This is because we use a much more refined method to evaluate the algebraic degree for any such (n, d) while [9, 11] still rely on a very similar bound as in [19] which cannot be effective for large d and small n. Since in Chaghri $(n, d) = (63, 2^{32} + 1)$, we cannot obtain efficient attacks by simply using the bound given in [9, 11].

Another related work seems to be the bit-based division property [31], which is a powerful method for the degree evaluation. Recently, the field-based division property [13] has been proposed and used to analyze MiMC. Here, we emphasize that our method is in nature very different from the concept of division property, which should be clear after understanding our method. Moreover, we will give some discussions on the differences on page 13.

Our contributions. We mainly focus on the higher-order differential attacks on Chaghri. As mentioned above, due to the usage of $(n, d) = (63, 2^{32} + 1)$ in Chaghri, existing methods to bound the algebraic degree become ineffective and we almost cannot violate the designers' claim that the algebraic degree of Chaghri increases exponentially with them. Hence, new techniques are required to break Chaghri. The contributions of this paper are summarized below.

1. A novel and efficient technique called coefficient grouping is proposed for the degree evaluation of Chaghri. The efficiency comes from an efficient representation of the polynomial of any rounds of Chaghri in terms of the input. Specifically, this representation can be determined by a single vector of integers that can be computed in linear time. Furthermore, with this vector of integers, upper bounding the degree is reduced to some well-structured optimization problems that can be efficiently solved, e.g. some can be solved in linear time.
2. For SPN-based ciphers over $\mathbb{F}_{2^n}$, i.e. Chaghri, we demonstrate that it is necessary to first study the increase of the algebraic degree in the univariate case and then study it in the multivariate case. With this strategy and our method to evaluate the algebraic degree, we can break the full 8 rounds of Chaghri with a low data and time complexity of 2^{38}. Moreover, the attack can reach up to 13.5 rounds and this reveals that the original design of Chaghri is flawed. Our results are summarized in Table 1.
3. It is found that the vulnerability of Chaghri exists in the usage of a sparse affine transform (an $\mathbb{F}_2$-linearized affine polynomial), i.e. $B(x) = c_1 x^{2^3} + c_2$, where $c_1, c_2 \in \mathbb{F}_{2^{63}}$ are constants. This can be well explained by our coefficient grouping technique and further shows the advantage of our technique. Hence, we are motivated to design a slightly denser affine transform and further motivated to generalize our coefficient grouping technique to a more complex design. Based on it, we succeed in finding a new affine transform to achieve an almost exponential increase of the algebraic degree. The new affine transform is $B'(x) = c'_1 x + c'_2 x^{2^2} + c'_3 x^{2^8} + c'_4$. By replacing $B(x)$ with $B'(x)$, we can keep the number of rounds of Chaghri unchanged and this has little overhead compared with the original design[6].

Based on the above results, we believe our coefficient technique is useful for both cryptanalysis and design and worth further investigation. The source code of this work can be found in `https://github.com/LFKOKAMI/Chaghri-degEva.git`.

[6] The designers of Chaghri have revised their design based on our countermeasures.

Organization. In Sect. 2, we describe the used notations, the block cipher Chaghri and some basic knowledge related to this work. In Sect. 3, the coefficient grouping technique for Chaghri is described. Then, in Sect. 4, we give more details of our attacks on Chaghri in both the univariate and multivariate settings. In Sect. 5, the coefficient grouping technique is further generalized to a more complex design and we describe how to search for a secure affine transform with it. Finally, we conclude the paper in Sect. 6.

Table 1: Summary of our attacks on Chaghri

Attack Type	Rounds	Time	Data	Reference
Distinguisher	8 (full)	2^{38}	2^{38}	Sect. 4
	13	2^{63}	2^{63}	Sect. 4
	13.5	2^{123}	2^{123}	Sect. 4.3
Key recovery	13.5	$2^{96.6}$	2^{63}	Sect. 4.1

2 Preliminaries

2.1 Notation

The following notations will be used throughout this paper.

1. $|\mathcal{S}|$ denotes the size of the set $\mathcal{S}$.
2. $a\%b$ represents a mod b.
3. $a|b$ denotes that a divides b.
4. $[a, b]$ is a set of integers i satisfying $a \leq i \leq b$.
5. $H(a)$ is the hamming weight of a.
6. The function $\mathcal{M}_n(x)$ $(x \in \mathbb{N})$ is defined as follows:

$$\mathcal{M}_n(x) = \begin{cases} 2^n - 1 \text{ if } 2^n - 1|x, x \geq 2^n - 1, \\ x\%(2^n - 1) \text{ otherwise.} \end{cases}$$

By the definition of $\mathcal{M}_n(x)$, we have $\mathcal{M}_n(x_1 + x_2) = \mathcal{M}_n(\mathcal{M}_n(x_1) + \mathcal{M}_n(x_2))$, $\mathcal{M}_n(2^i) = 2^{i\%n}$ and $\mathcal{M}_n(2^i x) = \mathcal{M}_n(2^{i\%n}\mathcal{M}_n(x))$ for $i \geq 0$.

2.2 On the Finite Field $\mathbb{F}_{p^n}$

For a prime number p and a positive integer n, the finite field $\mathbb{F}_{p^n}$ can be represented as a set of numbers of size p^n. Let α be a primitive element of $\mathbb{F}_{p^n}$. Then each element x in the finite field $\mathbb{F}_{p^n}$ can be written as

$$x = \sum_{i=0}^{n-1} \beta_i \alpha^i,$$

where $\beta_i \in [0, p-1]$. Moreover, the set $\{1, \alpha, \ldots, \alpha^{n-1}\}$ is said to be a polynomial basis of $\mathbb{F}_{p^n}$.

For the element $x \in \mathbb{F}_{p^n}$, it is well-known that

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

Hence, for two monomials X^a and X^b in the polynomial ring $\mathbb{F}_{2^n}[X]$, there is $X^a \cdot X^b = X^{\mathcal{M}_n(a+b)}$, which is the main reason to define the function $\mathcal{M}_n(x)$.

Moreover, it is also well-known that

$$(x+y)^{p^i} = x^{p^i} + y^{p^i}$$

for $\forall x, y \in \mathbb{F}_{p^n}$ and $i \geq 0$.

The higher-order differential attack over $\mathbb{F}_{2^n}$**.** Throughout this paper, we mainly utilize the idea described in [19] to analyze Chaghri. Specifically, for a given function $\mathcal{F} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$, there always exists a vectorial Boolean function $\mathcal{G} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that

$$\begin{aligned} \sigma : \sum_{i=0}^{n-1} \beta_i \alpha^i &\mapsto (\beta_0, \beta_1, \ldots, \beta_{n-1}) \in \mathbb{F}_2^n, \\ \sigma(\mathcal{F}(x)) &= \mathcal{G}(\sigma(x)) \ \forall x \in \mathbb{F}_{2^n}, \end{aligned}$$

where $\{1, \alpha, \ldots, \alpha^{n-1}\}$ is a polynomial basis of $\mathbb{F}_{2^n}$.

Let $\deg(\mathcal{G})$ be the algebraic degree of $\mathcal{G}$. For the higher-order differential attack, given any affine vector subspace V of dimension $\deg(\mathcal{G}) + 1$ from $\mathbb{F}_2^n$, there is $\sum_{v \in V} \mathcal{G}(v) = 0$, which implies

$$\sum_{(\beta_0, \beta_1, \ldots, \beta_{n-1}) \in V} \mathcal{F}(\sum_{i=0}^{n-1} \beta_i \alpha^i) = 0.$$

It is well-known that $\deg(\mathcal{G})$ is related to the univariate representation of $\mathcal{F}$, as stated below:

Definition 1 (Univariate degree and algebraic degree). *Let $\mathcal{F}$ and $\mathcal{G}$ be as above. The univariate representation of $\mathcal{F}$ is*

$$\mathcal{F} = \sum_{i=0}^{2^n-1} u_i X^i,$$

where $u_i \in \mathbb{F}_{2^n}$ for $i \in [0, 2^n - 1]$. The univariate degree of $\mathcal{F}$ denoted by $D_{\mathcal{F}}^u$ is defined as:

$$D_{\mathcal{F}}^u = \max\{i : i \in [0, 2^n - 1], u_i \neq 0\}.$$

Then, $\deg(\mathcal{G})$ *can be computed as follows:*

$$\deg(\mathcal{G}) = \max\{H(i) : i \in [0, 2^n - 1], u_i \neq 0\}.$$

$\max\{H(i) : i \in [0, 2^n - 1], u_i \neq 0\}$ *is also called the algebraic degree of* $\mathcal{F}$ *denoted by* $D^a_{\mathcal{F}}$.

The multivariate case. The above higher-order differential attack can also be extended to the multivariate case. Specifically, let $\mathcal{F}(X_1, X_2, \ldots, X_t) : \mathbb{F}^t_{2^n} \to \mathbb{F}_{2^n}$ be a multivariate function in variables $(X_1, X_2, \ldots, X_t)$. Then, its multivariate representation is

$$\mathcal{F} = \sum_{i_1=0}^{2^n-1} \sum_{i_2=0}^{2^n-1} \cdots \sum_{i_t=0}^{2^n-1} u_{i_1,i_2,\ldots,i_t} X_1^{i_1} X_2^{i_2} \cdots X_t^{i_t}.$$

The algebraic degree is then defined as

$$D^a_{\mathcal{F}} = \max\{\sum_{j=1}^{t} H(i_j) : i_j \in [0, 2^n - 1], u_{i_1,i_2,\ldots,i_t} \neq 0\}.$$

Let

$$X_i = \sum_{j=0}^{n-1} \beta_{i,j}\alpha^j, \quad \beta^i = (\beta_{i,0}, \beta_{i,1}, \ldots, \beta_{i,n-1}) \in \mathbb{F}^n_2 \text{ where } i \in [1, t].$$

By choosing an affine subspace V of dimension $\dim(V) = D^a_{\mathcal{F}} + 1$ from $\mathbb{F}^{n\times t}_2$, there will be

$$\sum_{(\beta^1,\beta^2,\ldots,\beta^t)\in V} \mathcal{F}(X_1, \ldots, X_t) = 0,$$

which is trivial extension of the univariate case. Specifically, for any monomial $X_1^{\rho_1} X_2^{\rho_2} \cdots X_t^{\rho_t}$, there is $\sum_{i=1}^{t} H(\rho_i) \leq D^a_{\mathcal{F}}$ by definition. For any such affine subspace V, we can denote the corresponding affine subspace of β^i by V_i $(1 \leq i \leq t)$ and denote the dimension of V_i by $\dim(V_i)$. Then, there is $\sum_{i=1}^{t} \dim(V_i) = D^a_{\mathcal{F}} + 1 \geq 1 + \sum_{i=1}^{t} H(\rho_i)$. Therefore, there must exist an index i such that $\dim(V_i) \geq H(\rho_i) + 1$, which implies

$$\sum_{\beta^1\in V_1} \sum_{\beta^2\in V_2} \cdots \sum_{\beta^i\in V_i} \cdots \sum_{\beta^t\in V_t} X_1^{\rho_1} X_2^{\rho_2} \cdots X_i^{\rho_i} \cdots X_t^{\rho_t} = 0.$$

2.3 Description of Chaghri

The FHE-friendly block cipher Chaghri [6] is defined over a large finite field. There are in total 8 rounds and each round is composed of two steps. Denote the state of Chaghri by $a = (a_1, a_2, a_3) \in \mathbb{F}^3_{2^{63}}$. The round function $\mathtt{R}(a)$ of its decryption is described in Algorithm 1. Note that throughout this paper, we are considering the decryption of Chaghri because the designers choose the secure number of rounds for Chaghri by mainly analyzing the security of decryption and low multiplicative depth in decryption is desired in FHE schemes. For encryption, the algebraic degree of the S-box is very high and the affine layer is very dense. Hence, our attacks cannot be applied to the encryption of Chaghri.

Algorithm 1 The round function of Chaghri at the $(j+1)^{th}$ round where $0 \leq j \leq 7$

```
procedure R(a)
    $a_i = G(a_i)$ for $i \in \{1, 2, 3\}$
    $a_i = B(a_i)$ for $i \in \{1, 2, 3\}$
    $a = M \cdot (a_1, a_2, a_3)^T$
    $a_i = a_i + RK[2j+1]_i$ for $i \in \{1, 2, 3\}$
    $a_i = G(a_i)$ for $i \in \{1, 2, 3\}$
    $a_i = B(a_i)$ for $i \in \{1, 2, 3\}$
    $a = M \cdot (a_1, a_2, a_3)^T$
    $a_i = a_i + RK[2j+2]_i$ for $i \in \{1, 2, 3\}$
```

In Algorithm 1, the round key $RK[j] = (RK[j]_1, RK[j]_2, RK[j]_3) \in \mathbb{F}_{2^{63}}^3$ is generated from a master key $K = (K_1, K_2, K_3) \in \mathbb{F}_{2^{63}}^3$. The whitening key is $RK[0] = (RK[0]_1, RK[0]_2, RK[0]_3)$. We omit the key schedule function as it is not relevant to our attacks. In the following, we explain each component used in the round function, namely G, B and M.

The nonlinear function $G(x) : \mathbb{F}_{2^{63}} \to \mathbb{F}_{2^{63}}$***.*** $G(x)$ is defined as $G(x) = x^{2^{32}+1}$.

The affine transform $B(x) : \mathbb{F}_{2^{63}} \to \mathbb{F}_{2^{63}}$***.*** $B(x)$ is defined as $B(x) = c_1 x^{2^3} + c_2$ where $c_1, c_2 \in \mathbb{F}_{2^{63}}$ are constants.

The linear transform $M : \mathbb{F}_{2^{63}}^3 \to \mathbb{F}_{2^{63}}^3$***.*** M is a 3×3 MDS matrix. The designers do not specify a concrete choice for M and they claim any MDS matrix is suitable. We note here that our attacks apply to any choice of M.

Definition of one step. According to the round function described in Algorithm 1, the round function is $\mathtt{R}(a) = AK \circ M \circ B \circ G \circ AK \circ M \circ B \circ G(a)$. Similar to [6], one step of Chaghri is defined as $AK \circ M \circ B \circ G(a)$ and we call it the step function of Chaghri.

Notation for the internal state. We denote the internal state after i steps by $(z_{i,1}, z_{i,2}, z_{i,3})$. For example, the input state is $(z_{0,1}, z_{0,2}, z_{0,3})$, the internal state after 1 step is $(z_{1,1}, z_{1,2}, z_{1,3})$, and the internal state after 1 round is $(z_{2,1}, z_{2,2}, z_{2,3})$. In this paper, we consider R steps of Chaghri.

3 The Coefficient Grouping Technique

We give the intuitive explanation of our new technique with its application to Chaghri. For better understanding, we first only focus on its application to the univariate polynomial and then we discuss how it can be extended to the multivariate case.

Without loss of generality, we consider a general form of $G(x)$ and $B(x)$, as shown below:

$$G(x) = x^{2^{k_0}+2^{k_1}}, B(x) = c_1 x^{2^{k_2}} + c_2.$$

Moreover, we consider the finite field $\mathbb{F}_{2^n}$, i.e. the internal state $a = (a_1, a_2, a_3)$ of Chaghri satisfies $a_i \in \mathbb{F}_{2^n}$ for $i \in [1, 3]$. It should be emphasized that there are constraints on (k_0, k_1, n) to ensure that $G(x)$ is a permutation. Here we only care about its general form of algebraic degree 2. For Chaghri, $(k_0, k_1, k_2) = (32, 0, 3)$ and $n = 63$.

The main idea of our attacks. We consider an input state which can be represented as univariate polynomials in the variable $X \in \mathbb{F}_{2^n}$, as shown below:

$$z_{0,1} = A_{0,1}X + B_{0,1},\ z_{0,2} = A_{0,2}X + B_{0,2},\ z_{0,3} = A_{0,3}X + B_{0,3}, \tag{1}$$

where $A_{0,i}, B_{0,i} \in \mathbb{F}_{2^n}$ $(1 \leq i \leq 3)$ are randomly chosen constants. In this way, after an arbitrary number of steps, each state word can always be represented as a univariate polynomial in X. Our aim is to compute the upper bound $D_{r,i}$ for the algebraic degree of the univariate polynomial $P_{r,i}(X)$ where $z_{r,i} = P_{r,i}(X)$ $(1 \leq i \leq 3)$. We say the upper bound for the algebraic degree[7] of r-step Chaghri is D_r where $D_r = \max\{D_{r,1}, D_{r,2}, D_{r,3}\}$. Hence, if $D_r < n$, there exists a higher-order differential attack on r steps of Chaghri with time and data complexity 2^{D_r+1}.

Remark 1. In particular, this attack can be trivially extended for 1 more step by using 2^n data. Specifically, we can consider an input state of the following form:

$$z_{0,1} = X_1,\ z_{0,2} = A_2,\ z_{0,3} = A_3,$$

where $A_2, A_3 \in \mathbb{F}_{2^n}$ are randomly chosen constants and X is the variable. Then, by making $X = B \circ G(X_1 + RK[0]_1)$, the state $(z_{1,1}, z_{1,2}, z_{1,3})$ will be of the same form as in Equation 1. For such a state $(z_{1,1}, z_{1,2}, z_{1,3})$, after r more steps, the algebraic degree of the univariate polynomials in X is upper bounded by D_r. Since $D_r < n$ and X will traverse all the 2^n possible values when X_1 takes all the 2^n possible values, the higher-order differential attack indeed can reach $r + 1$ steps with time and data complexity of 2^n.

[7] From the perspective of attackers, D_r can be defined as $\min\{D_{r,1}, D_{r,2}, D_{r,3}\}$ to reduce the time complexity of the attacks. However, due to the strong diffusion of the MDS matrix, using $D_r = \max\{D_{r,1}, D_{r,2}, D_{r,3}\}$ is reasonable and can greatly simplify the attack. This can also be observed from our later analysis of the evolution of the polynomials through the step function of Chaghri, i.e. using $D_r = \max\{D_{r,1}, D_{r,2}, D_{r,3}\}$ is indeed tight according to the experiments.

3.1 Tracing the Form of the Univariate Polynomial

With the input form shown in Equation 1, the state words $(z_{r,1}, z_{r,2}, z_{r,3})$ can always be represented as univariate polynomials of the following form:

$$z_{r,1} = \sum_{i=1}^{|w_r|} A_{r,i} X^{w_{r,i}},\ z_{r,2} = \sum_{i=1}^{|w_r|} B_{r,i} X^{w_{r,i}}, z_{r,3} = \sum_{i=1}^{|w_r|} C_{r,i} X^{w_{r,i}}$$

where $A_{r,i}, B_{r,i}, C_{r,i} \in \mathbb{F}_{2^n}$ are key-dependent coefficients and we call the set

$$w_r = \{w_{r,1}, w_{r,2}, \dots, w_{r,|w_r|}\} \subseteq \mathbb{N}$$

the set of all possible exponents for the univariate polynomials after r steps. Note that $0 \in w_r$ since it represents the constant term. It should be mentioned that for $r = 0$, we have

$$w_0 = \{0, 1\}, \tag{2}$$

which corresponds to the input form specified in Equation 1.

According to the definition of the algebraic degree of a univariate polynomial, we have

$$D_r \leq \max\{H(w_{r,i}) : 1 \leq i \leq |w_r|\}. \tag{3}$$

Analyzing the evolution of the polynomial representations. We are interested in the univariate polynomials to represent $(z_{r+1,1}, z_{r+1,2}, z_{r+1,3})$, i.e. how the polynomials evolve through the step function of Chaghri. The detailed analysis is shown below.

$$\begin{aligned} G(z_{r,1}) &= (\sum_{i=1}^{|w_r|} A_{r,i} X^{w_{r,i}})^{2^{k_0}+2^{k_1}} \\ &= (\sum_{i=1}^{|w_r|} A_{r,i} X^{w_{r,i}})^{2^{k_0}} (\sum_{j=1}^{|w_r|} A_{r,j} X^{w_{r,j}})^{2^{k_1}} = \sum_{i=1}^{|w_r|} \sum_{j=1}^{|w_r|} A_{r,i,j} X^{\mathcal{M}_n(2^{k_0} w_{r,i} + 2^{k_1} w_{r,j})}, \end{aligned}$$

where $A_{r,i,j} \in \mathbb{F}_{2^n}$ are key-dependent coefficients.

$$\begin{aligned} B \circ G(z_{r,1}) &= c_1 \left(\sum_{i=1}^{|w_r|} \sum_{j=1}^{|w_r|} A_{r,i,j} X^{\mathcal{M}_n(2^{k_0} w_{r,i} + 2^{k_1} w_{r,j})} \right)^{2^{k_2}} + c_2 \\ &= \sum_{i=1}^{|w_r|} \sum_{j=1}^{|w_r|} A'_{r,i,j} X^{\mathcal{M}_n(2^{k_0+k_2} w_{r,i} + 2^{k_1+k_2} w_{r,j})}, \end{aligned}$$

where $A'_{r,i,j} \in \mathbb{F}_{2^{63}}$ are key-dependent coefficients. c_2 is removed due to $0 \in w_r$.

Similarly, it can be found that

$$B \circ G(z_{r,2}) = \sum_{i=1}^{|w_r|}\sum_{j=1}^{|w_r|} B'_{r,i,j} X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_1+k_2}w_{r,j})},$$
$$B \circ G(z_{r,3}) = \sum_{i=1}^{|w_r|}\sum_{j=1}^{|w_r|} C'_{r,i,j} X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_1+k_2}w_{r,j})},$$

where $B'_{r,i,j}, C'_{r,i,j} \in \mathbb{F}_{2^n}$ are key-dependent coefficients.

Therefore, we can obtain

$$z_{r+1,1} = \sum_{i=1}^{|w_r|}\sum_{j=1}^{|w_r|} A_{r+1,i,j} X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_1+k_2}w_{r,j})},$$
$$z_{r+1,2} = \sum_{i=1}^{|w_r|}\sum_{j=1}^{|w_r|} B_{r+1,i,j} X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_1+k_2}w_{r,j})},$$
$$z_{r+1,3} = \sum_{i=1}^{|w_r|}\sum_{j=1}^{|w_r|} C_{r+1,i,j} X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_1+k_2}w_{r,j})},$$

where $A_{r+1,i,j}, B_{r+1,i,j}, C_{r+1,i,j} \in \mathbb{F}_{2^{63}}$ are key-dependent coefficients.

Hence, we obtain a relation between the sets w_r and w_{r+1}, as shown below:

$$w_{r+1} = \{e | e = \mathcal{M}_n(2^{k_0+k_2}w_{r,i} + 2^{k_1+k_2}w_{r,j}), 1 \leq i, j \leq |w_r|\},$$

In this way, for each element $e \in w_{r+2}$, there must exist (i, j, s, t) where $1 \leq i, j, s, t \leq |w_r|$ such that

$$e = \mathcal{M}_n\left(2^{k_0+k_2}(2^{k_0+k_2}w_{r,i} + 2^{k_1+k_2}w_{r,j}) + 2^{k_1+k_2}(2^{k_0+k_2}w_{r,s} + 2^{k_1+k_2}w_{r,t})\right).$$

In other words,

$$\begin{aligned} w_{r+2} = \{e | e = \mathcal{M}_n(&2^{2k_0+2k_2}w_{r,i} + 2^{k_0+k_1+2k_2}(w_{r,j} + w_{r,s}) + 2^{2k_1+2k_2}w_{r,t}), \\ &1 \leq i, j, s, t \leq |w_r|\}. \end{aligned}$$

For the concrete parameters of Chaghri, we have

$$\begin{aligned} w_{r+1} &= \{e | e = \mathcal{M}_{63}(2^{35}w_{r,i} + 2^{3}w_{r,j}), 1 \leq i, j \leq |w_r|\}, \\ w_{r+2} &= \{e | e = \mathcal{M}_{63}\left(2^{7}w_{r,i} + 2^{38}(w_{r,j} + w_{r,s}) + 2^{6}w_{r,t}\right), 1 \leq i, j, s, t \leq |w_r|\}. \end{aligned}$$

Another representation of the set $w_{r+\ell}$. Based on the above discussions, it is now clear that there exists another general representation of the set $w_{r+\ell}$. Specifically, it must be of the following form:

$$\begin{aligned} w_{r+\ell} = \{e | e = \mathcal{M}_n(\sum_{i=1}^{N_{n-1}} 2^{n-1}w_{r,d_{i,n-1}} + \sum_{i=1}^{N_{n-2}} 2^{n-2}w_{r,d_{i,n-2}} + \ldots + \sum_{i=1}^{N_0} 2^{0}w_{r,d_{i,0}}), \\ \text{where } 1 \leq d_{i,j} \leq |w_r| \text{ for } 0 \leq j \leq n-1\} \end{aligned}$$

Proof. Proving this form is simple. Specifically, by induction, we only need to prove $w_{r+\ell+1}$ is also of this form when $w_{r+\ell}$ is as above. This is because w_0 is of this form, i.e. for $w_0 = \{w_{0,1}, w_{0,2}\} = \{0,1\} = \{e|e = 2^0 w_{0,i}, 1 \le i \le 2\}$, there is

$$N_0 = 1, N_i = 0 \ (1 \le i \le n-1). \tag{4}$$

Considering the relation between $w_{r+\ell}$ and $w_{r+\ell+1}$, we have

$$w_{r+\ell+1} = \{e|e = \mathcal{M}_n(2^{k_0+k_2} w_{r+\ell,i} + 2^{k_1+k_2} w_{r+\ell,j}), 1 \le i,j \le |w_{r+\ell}|\}.$$

Hence, we have

$$\begin{aligned} w_{r+\ell+1} = \{e|e = \mathcal{M}_n(&\sum_{i=1}^{N'_{n-1}} 2^{n-1} w_{r,d'_{i,n-1}} + \sum_{i=1}^{N'_{n-2}} 2^{n-2} w_{r,d'_{i,n-2}} + \ldots + \sum_{i=1}^{N'_0} 2^0 w_{r,d'_{i,0}}), \\ &\text{where } 1 \le d'_{i,j} \le |w_r| \text{ for } 0 \le j \le n-1\}, \end{aligned}$$

where

$$N'_i = N_{(i-k_1-k_2)\%n} + N_{(i-k_0-k_2)\%n} \text{ for } 0 \le i \le n-1. \tag{5}$$

□

In other words, each set w_r can be fully described with a vector of integers $(N^r_{n-1}, N^r_{n-2}, \ldots, N^r_0)$. For w_0, this vector is

$$N^0_0 = 1, N^0_i = 0 \ (1 \le i \le n-1).$$

Then, based on the recursive relation specified in Equation 5, i.e.

$$N^{r+1}_i = N^r_{(i-k_1-k_2)\%n} + N^r_{(i-k_0-k_2)\%n} \text{ for } 0 \le i \le n-1, r \ge 0, \tag{6}$$

for any w_r, the corresponding vector of integers $(N^r_{n-1}, N^r_{n-2}, \ldots, N^r_0)$ can be computed in linear time, i.e. with rn times of simple integer additions. Then, the set w_r can be described as follows:

$$\begin{aligned} w_r = \{e|e = \mathcal{M}_n(&\sum_{i=1}^{N^r_{n-1}} 2^{n-1} w_{0,d_{i,n-1}} + \sum_{i=1}^{N^r_{n-2}} 2^{n-2} w_{0,d_{i,n-2}} + \ldots + \sum_{i=1}^{N^r_0} 2^0 w_{0,d_{i,0}}), \\ &\text{where } 1 \le d_{i,j} \le |w_0| \text{ for } 0 \le j \le n-1\}. \end{aligned} \tag{7}$$

Application to the Chaghri parameters. For the concrete parameters of Chaghri, the corresponding $(N^1_{62}, N^1_{61}, \ldots, N^1_0)$ for w_1 is

$$N^1_3 = 1, N^1_{35} = 1, N^1_i = 0 \ (i \notin \{3, 35\}, 0 \le i \le 62).$$

While for w_2, the corresponding $(N^2_{62}, N^2_{61}, \ldots, N^2_0)$ is

$$N^2_6 = 1, N^2_7 = 1, N^2_{38} = 2, N^2_i = 0 \ (i \notin \{6, 7, 38\}, 0 \le i \le 62).$$

For any w_r, we can compute the corresponding $(N^r_{62}, N^r_{61}, \ldots, N^r_0)$ in linear time.

3.2 A Natural Optimization Problem

The last problem we need to deal with is how to compute D_r after giving the vector of integers $(N_{n-1}^r, N_{n-2}^r, \ldots, N_0^r)$. For our representation of w_r, it can be equivalently interpreted in the way that there are in total $N_{n-1}^r + N_{n-2}^r + \ldots + N_0^r$ possible variables that can independently take values from $w_0 = \{0, 1\}$. Hence, the problem to bound D_r becomes a natural optimization problem, as shown below:

$$\begin{aligned}\text{maximize } & H\Bigg(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i)\Bigg),\\ \text{subject to } & 0 \le \gamma_i \le N_i^r \text{ for } i \in [0, n-1].\end{aligned}$$

More specifically, due to $w_0 = \{0, 1\}$, w_r specified in Equation 7 is equivalent to

$$w_r = \{\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i) | 0 \le \gamma_i \le N_i^r \text{ for } i \in [0, n-1]\}.$$

After computing N_i^r for $i \in [0, n-1]$, which can be finished in linear time, this problem[8] can be easily encoded as an MILP problem. Specifically, for each integer $m \in [0, 2^n - 1]$, we can assign a bit vector $(m_{n-1}, m_{n-2}, \ldots, m_0)$ for m, i.e. $m = \sum_{i=0}^{n-1} 2^i m_i$. Then, $\mathcal{M}_n(2^j \cdot m)$ just makes m become

$$\Bigg(m_{(n-1-j)\%n}, m_{(n-2-j)\%n}, \ldots, m_{(0-j)\%n}\Bigg),$$

i.e. a change of the order of variables.

The addition is trivial. Specifically, for the addition $\mathcal{M}_n(x+y) = q$ where $x = (x_{n-1}, x_{n-2}, \ldots, x_0)$, $y = (y_{n-1}, y_{n-2}, \ldots, y_0)$ and $q = (q_{n-1}, q_{n-2}, \ldots, q_0)$, by introducing two $(n+1)$-bit vectors $g = (g_n, g_{n-1}, \ldots, g_0)$ and $g' = (g'_n, g'_{n-1}, \ldots, g'_0)$ as well as an n-bit vector $q' = (q'_{n-1}, q'_{n-2}, \ldots, q'_0)$ to represent the intermediate value, we have

$$\begin{cases} g_0 = 0,\ 2g_{i+1} + q'_i = x_i + y_i + g_i \text{ for } i \in [0, n-1], \\ \quad g'_0 = g_n,\ 2g'_{i+1} + q_i = q'_i + g'_i \text{ for } i \in [0, n-1]. \end{cases}$$

For the comparison $m \le b$ where $b = (b_{n-1}, b_{n-2}, \ldots, b_0) \in \mathbb{F}_2^n$ is a known integer, it can also be simply described with linear inequalities. Specifically, supposing $b_i = 1$ for any $i \in \{i_1, i_2, \ldots, i_{l-1}, i_l\}$ and $0 \le i_1 < i_2 < \ldots < i_l \le n-1$.

[8] Motivated by this work, an ad-hoc algorithm [28] has been developed to solve the above optimization problem in time $\mathcal{O}(n)$. However, in this following, there still remain some other optimization problems which cannot be handled by that $\mathcal{O}(n)$ algorithm [28]. Hence, we only consider the general-purpose solvers for the optimization problems in this paper.

Then $m \leq b$ can be described with the following $n - l$ linear (in)equalities:

$$\begin{cases} \begin{array}{r} m_j = 0 \text{ for } i_l < j \leq n-1, \\ (1 - m_{i_l}) - m_j \geq 0 \text{ for } i_{l-1} < j < i_l, \\ \sum\limits_{s=l-1}^{l} (1 - m_{i_s}) - m_j \geq 0 \text{ for } i_{l-2} < j < i_{l-1}, \\ \ldots \\ \sum\limits_{s=1}^{l} (1 - m_{i_s}) - m_j \geq 0 \text{ for } 0 \leq j < i_1. \end{array} \end{cases}$$

To maximize $H(m)$, we simply write

$$\text{maximize } m_0 + m_1 + \ldots + m_{n-1}.$$

In this way, a simple MILP model can be constructed and the solution of the model is exactly D_r according to Equation 3.

A useful theorem. In the following, we present a useful theorem for the case when the maximal degree is reached.

Theorem 1 *For a given vector of integers $(N_{n-1}, N_{n-2}, \ldots, N_0)$, if the solution to the following optimization problem called* ***Problem 1*** *is $h \times n$:*

$$\begin{aligned} \text{maximize } & \sum_{j=1}^{h} H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_{j,i})\Big), \\ \text{subject to } & \mathcal{C}_1(\gamma_{1,0}, \gamma_{1,1}, \ldots, \gamma_{h,n-1}, N_0, N_1, \ldots, N_{n-1}), \end{aligned}$$

the solution to the following optimization problem called ***Problem 2*** *must also be $h \times n$:*

$$\begin{aligned} \text{maximize } & \sum_{j=1}^{h} H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \nu_j \gamma_{j,i})\Big), \\ \text{subject to } & \mathcal{C}_1(\gamma_{1,0}, \gamma_{1,1}, \ldots, \gamma_{h,n-1}, N_0, N_1, \ldots, N_{n-1}), \end{aligned}$$

where $\nu_j \in \mathbb{N}^+$ for $j \in [1, h]$ and $\mathcal{C}_1(\gamma_{1,0}, \gamma_{1,1}, \ldots, \gamma_{h,n-1}, N_0, N_1, \ldots, N_{n-1})$ denotes the set of constraints.

Proof. Since the solution to **Problem 1** is $h \times n$, for each $j \in [1, h]$, there exists an assignment to $(\gamma_{j,n-1}, \gamma_{j,n-2}, \ldots, \gamma_{j,0})$ denoted by $(\gamma_{j,\hat{n}-1}, \gamma_{j,\hat{n}-2}, \ldots, \gamma_{\hat{j},0})$ such that

$$\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_{\hat{j},i}) = 2^n - 1.$$

Hence, for each $j \in [1, h]$, we have

$$\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \nu_j \hat{\gamma_{j,i}}) = \mathcal{M}_n\left(\nu_j \times \mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \hat{\gamma_{j,i}})\right) = 2^n - 1.$$

As the upper bound for the solution to **Problem 2** is $h \times n$ and we find an assignment to make its solution be $h \times n$, the solution to **Problem 2** is $h \times n$. □

Generalization to an arbitrary power function. In the above, we mainly analyze a power function $x \mapsto x^{2^{k_0}+2^{k_1}}$ with algebraic degree 2. It is easy to observe that a similar procedure can be applied to any power function $x \mapsto x^{\sum_{i=1}^{\rho} 2^{\phi_i}}$ over $\mathbb{F}_{2^n}$ with algebraic degree ρ. This is due to the following simple relation:

$$(\sum_{j=1}^{|w_r|} A_j X^{w_{r,j}})^{\sum_{i=1}^{\rho} 2^{\phi_i}} = \sum_{j_1=1}^{|w_r|} \sum_{j_2=1}^{|w_r|} \cdots \sum_{j_\rho=1}^{|w_r|} A_{j_1,j_2,\ldots,j_\rho} X^{2^{\phi_1} w_{r,j_1} + 2^{\phi_2} w_{r,j_2} + \cdots + 2^{\phi_\rho} w_{r,j_\rho}}.$$

By using the same $B(x) = c_1 x^{2^{k_2}} + c_2$, we still can simply use a vector of integers to represent the set of possible exponents. In addition, the recursive relation between the vectors $(N_{n-1}^{r+1}, N_{n-2}^{r+1}, \ldots, N_0^{r+1})$ and $(N_{n-1}^{r}, N_{n-2}^{r}, \ldots, N_0^{r})$ can be described as below:

$$N_j^{r+1} = \sum_{i=1}^{\rho} N_{(j-\phi_i-k_2)\%n}^{r} \text{ for } j \in [0, n-1],$$

which implies that these vectors can be computed in linear time. With these vectors, bounding the algebraic degree is then reduced to the same optimization problem. This obviously shows the effectiveness of our coefficient grouping technique.

Comparison with the literature. First, compared with the well-known degree evaluation technique developed for Keccak [8], our technique does not require to compute the upper bound of the degree at round i before computing the upper bound at round $i + 1$. This is mainly because we find an efficient representation of the polynomial of any rounds of Chaghri in terms of the input, which can be determined by a vector of integers and this vector can be computed in linear time. Second, different from the division property that has been well studied in recent years [13, 22, 24, 32], there is no need to use a heavy model to describe the monomial transitions through the round function for each round because they can be simply captured by an efficient recursive relation in our technique. At last, compared with more related works in [9, 11, 19], when the affine layer is simply $B(x) = c_1 x^{2^{k_2}} + c_2$ and $G(x)$ is any power map over $\mathbb{F}_{2^n}$, tighter bounds can be derived with our technique because our way to describe the polynomial representation after any rounds is more accurate, i.e. we can exclude many redundant monomials that will never appear with our technique while they are treated as possible to appear in these techniques [9, 11, 19].

4 Cryptanalysis of Full-round Chaghri

With the above model, the upper bounds for D_r are obtained in seconds, as listed in Table 2.

Table 2: The upper bounds for D_r

r	0	2	4	6	8	10	12	14	16	18	20	22	24	25	26
D_r	1	3	7	12	17	22	27	32	37	42	47	52	58	60	63

Consequently, we can mount a higher-order differential attack on full 8 rounds of Chaghri with data and time complexity of 2^{38}. It also suggests that there is a higher-order differential distinguisher for 12.5 rounds of Chaghri with time and data complexity of 2^{61}. Furthermore, according to Remark 1, we can extend one more step (0.5 round) and construct a higher-order differential distinguisher for 13 rounds of Chaghri with time and data complexity of 2^{63}.

4.1 The Key-recovery Attack on 13.5 Rounds of Chaghri

We have constructed a 13-round distinguisher with data and time complexity of 2^{63}. Then, we can append 0.5 round for the key recovery. To recover the round key $RK[27]$, an equivalent round key $RK[27]' = (RK[27]'_1, RK[27]'_2, RK[27]'_3)$ is considered, where

$$(RK[27]'_1, RK[27]'_2, RK[27]'_3)^T = M^{-1} \times (RK[27]_1, RK[27]_2, RK[27]_3)^T.$$

Since the operations B^{-1} and G^{-1} work on the internal state in a parallel way, the naive method is to independently guess $RK[27]'_i$ $(1 \leq i \leq 3)$ and compute the corresponding $z_{26,i}$ and check the sum of $z_{26,i}$. If the sum is zero, the guess is correct. Hence, the time complexity of this key-recovery attack is about $3 \times 2^{63} \times 2^{63} < 2^{128}$. Note that after recovering $RK[27]'$, we can compute $RK[27]$ and deduce the master key according to the key schedule function.

Indeed, the key-recovery attack can be more efficient by treating $B^{-1}(RK[27]'_i)$ $(1 \leq i \leq 3)$ as a variable Y_i. Note that $B(x)$ is an affine transform over $\mathbb{F}_{2^{63}}$ and hence $B^{-1}(x)$ is also an affine transform. Then, we can construct a univariate polynomial $P_i(Y_i)$ in terms of Y_i using the condition that the sum of $z_{26,i}$ is 0. The degree of P_i denoted by $\mathcal{D}$ is the degree of the inverse of G and we have $\mathcal{D} = 2^{32}-1$. Hence, we can estimate the time to construct $P_i(Y_i)$ as about $2^{H(\mathcal{D})} \times 2^{63} = 2^{32+63} = 2^{95}$ field operations. Then, similar to the idea in [19], recovering Y_i is reduced to finding the roots of the univariate polynomial P_i, the time complexity of which can be estimated as $O(\mathcal{D} \times \log(\mathcal{D}) \times \mathrm{loglog}(\mathcal{D}) \times \log(\mathcal{D}) \times \log(2^{63}\mathcal{D}))$ field operations. Hence, we estimate the time complexity to find the roots as 2^{51}. Hence, the time complexity and data complexity of our key-recovery attack on 13.5 rounds of Chaghri are $3 \times (2^{51} + 2^{95}) = 2^{96.6}$ and 2^{63}, respectively.

4.2 Further Refining the Upper Bounds

In this section, we show that before reaching the maximal algebraic degree n, it is possible to refine D_r with more careful analysis. Consider the input state of the following form

$$z_{0,1} = X_1,\ z_{0,2} = A_2,\ z_{0,3} = A_3, \tag{8}$$

where $A_2, A_3 \in \mathbb{F}_{2^n}$ are randomly chosen constants and X_1 is the variable. Let $X = X_1 + RK[0]_1$. In this way, for any number of steps, each state word of Chaghri can be represented as a univariate polynomial in X. For $(z_{1,1}, z_{1,2}, z_{1,3})$, we have

$$\begin{aligned} z_{1,1} &= A_{1,1}X^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + B_{1,1}, \\ z_{1,2} &= A_{1,2}X^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + B_{1,2}, \\ z_{1,3} &= A_{1,3}X^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + B_{1,3}, \end{aligned}$$

where $A_{1,i}, B_{1,i}$ $(i \in [1,3])$ are constants depending on the key.

Hence, for w_1, we have

$$w_1 = \{\mathcal{M}_n(2^{k_0+k_2} + 2^{k_1+k_2}), 0\}.$$

Then, we have

$$\begin{aligned} w_r = \{e|e = \mathcal{M}_n(&\sum_{i=1}^{N_{n-1}^r} 2^{n-1}w_{1,d_{i,n-1}} + \sum_{i=1}^{N_{n-2}^r} 2^{n-2}w_{1,d_{i,n-2}} + \ldots + \sum_{i=1}^{N_0^r} 2^0 w_{1,d_{i,0}}), \\ &\text{where } 1 \le d_{i,j} \le 2 \text{ for } 0 \le j \le n-1\}. \end{aligned}$$

By making $N_0^1 = 1$ and $N_i^1 = 0$ for $i \in [1, n-1]$, we can compute the corresponding $(N_{n-1}^r, N_{n-2}^r, \ldots, N_0^r)$ for $r \ge 1$ with the recursive relation specified in Equation 6. Computing D_r is then equivalent to the following optimization problem:

$$\begin{aligned} &\text{maximize } H\Bigg(\mathcal{M}_n\ \Big(\sum_{i=0}^{n-1} 2^i(2^{k_0+k_2}\gamma_i + 2^{k_1+k_2}\gamma_i)\ \Big)\Bigg), \\ &\text{subject to } 0 \le \gamma_i \le N_i^r \text{ for } i \in [0, n-1]. \end{aligned}$$

Table 3: The refined upper bounds for D_r in the univariate case

r	0	2	4	6	8	10	12	14	16	18	20	22	24	26	27
D_r	1	3	7	11	16	21	26	32	37	42	47	52	57	62	63

Refined or unrefined? This refined model is only slightly slower and all the results can still be obtained in seconds as well. The refined upper bounds are shown in Table 3. We have practically verified our attacks on Chaghri for up to 7 rounds. It is found that our refined bounds are correct and tight. It can be found that although the upper bound is slightly better for $r \leq 12$, the complexity to break 8 rounds of Chaghri remains the same. Moreover, the longest higher-order differential distinguisher still only covers 26 steps, which is indeed a direct result of Theorem 1, i.e. the constraints at step r in the unrefined model are the same with those at step $r+1$ in the refined model and we reach the maximal degree at step 26 in the unrefined model. Due to the high efficiency of the unrefined model, to detect how long a higher-order differential distinguisher can reach, we prefer the unrefined model.

4.3 On the Multivariate Case

After understanding our attack in the univariate case, it is natural to ask whether the distinguisher can be further extended for more steps with a larger set of inputs, e.g. a set of 2^{2n} different inputs. Specifically, with the following input form

$$z_{0,1} = X_1, Z_{0,2} = X_2, Z_{0,3} = A_3,$$

where $A_3 \in \mathbb{F}_{2^n}$ is a randomly chosen constant and X_1, X_2 are variables, whether the attack can be extended for more steps?

Let $X = B \circ G(X_1 + RK[0]_0)$ and $Y = B \circ G(X_2 + RK[0]_1)$. The state $(z_{1,1}, z_{1,2}, z_{1,3})$ can be represented as multivariate polynomials in (X, Y) as below:

$$z_{1,1} = A_{1,1}X + B_{1,1}Y + C_{1,1},\ z_{1,2} = A_{1,2}X + B_{1,2}Y + C_{1,2},$$
$$z_{1,3} = A_{1,3}X + B_{1,3}Y + C_{1,3}.$$

Note that in the following, we will not repeat emphasizing which are constants in the polynomial representation. Instead, we only say which are variables.

To construct the longest higher-order differential distinguisher with at most 2^{2n} data, it suffices to compute the maximal number of steps r where the maximal algebraic degree $2n$ is first reached for the following input state

$$z_{0,1} = A_{0,1}X + B_{0,1}Y + C_{0,1},\ z_{0,2} = A_{0,2}X + B_{0,2}Y + C_{0,2},$$
$$z_{0,3} = A_{0,3}X + B_{0,3}Y + C_{0,3}, \tag{9}$$

where X, Y are variables. As in the univariate case, 1 more step can always be appended before this distinguisher by using 2^{2n} data. This will result in an r-step distinguisher with data and time complexity of 2^{2n}.

For the input form in Equation 9, the general form of $(z_{r,1}, z_{r,2}, z_{r,3})$ can be written down, as shown below:

$$z_{r,1} = \sum_{i=1}^{|W_r|} A_{r,i}X^{w_{r,i}}Y^{u_{r,i}},\ z_{r,2} = \sum_{i=1}^{|W_r|} B_{r,i}X^{w_{r,i}}Y^{u_{r,i}}, z_{r,3} = \sum_{i=1}^{|W_r|} C_{r,i}X^{w_{r,i}}Y^{u_{r,i}},$$

where

$$W_r = \{(w_{r,1}, u_{r,1}), (w_{r,2}, u_{r,2}), \ldots, (w_{r,|W_r|}, u_{r,|W_r|})\}.$$

For W_0, we have

$$W_0 = \{(1,0), (0,1), (0,0)\},$$

which corresponds to the input state specified in Equation 9.

With similar analysis to trace the evolution of the polynomials through G and B, we have

$$\begin{aligned} W_{r+1} = \{&(e_0, e_1)| \\ &e_0 = \mathcal{M}_n(2^{k_0+k_2}w_{r,i} + 2^{k_1+k_2}w_{r,j}), e_1 = \mathcal{M}_n(2^{k_0+k_2}u_{r,i} + 2^{k_1+k_2}u_{r,j}), \\ &1 \leq i, j \leq |W_r|\}. \end{aligned}$$

Specifically, we have

$$\begin{aligned} B \circ G(z_{r,1}) &= c_1(\sum_{i=1}^{|W_r|} A_{r,i}X^{w_{r,i}}Y^{u_{r,i}})^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + c_2 \\ &= \sum_{i=1}^{|W_r|}\sum_{j=1}^{|W_r|} A_{r+1,i,j}X^{\mathcal{M}_n(2^{k_0+k_2}w_{r,i}+2^{k_0+k_2}w_{r,j})}Y^{\mathcal{M}_n(2^{k_0+k_2}u_{r,i}+2^{k_0+k_2}u_{r,j})}. \end{aligned}$$

With the similar deduction as in the univariate case, the set W_r can also be represented using a vector of integers $(N^r_{n-1}, N^r_{n-1}, \ldots, N^r_0)$, as shown below:

$$\begin{aligned} W_r = \{&(e_0, e_1)| \\ &e_0 = \mathcal{M}_n(\sum_{i=1}^{N^r_{n-1}} 2^{n-1}w_{0,d_{i,n-1}} + \sum_{i=1}^{N^r_{n-2}} 2^{n-2}w_{0,d_{i,n-2}} + \ldots + \sum_{i=1}^{N^r_0} 2^0 w_{0,d_{i,0}}, \\ &e_1 = \mathcal{M}_n(\sum_{i=1}^{N^r_{n-1}} 2^{n-1}u_{0,d_{i,n-1}} + \sum_{i=1}^{N^r_{n-2}} 2^{n-2}u_{0,d_{i,n-2}} + \ldots + \sum_{i=1}^{N^r_0} 2^0 u_{0,d_{i,0}}, \\ &\text{where } 1 \leq d_{i,j} \leq |W_0| = 3 \text{ for } 0 \leq j \leq n-1\}, \end{aligned}$$

where

$$\begin{cases} N^0_0 = 1, N^0_i = 0 \text{ for } i \in [1, n-1], \\ N^r_i = N^{r-1}_{(i-k_1-k_2)\%n} + N^{r-1}_{(i-k_0-k_2)\%n} \text{ for } 0 \leq i \leq n-1, \ r \geq 1. \end{cases}$$

Since

$$W_0 = \{(1,0), (0,1), (0,0)\},$$

i.e. $(w^0_i, u^0_i) \neq (1,1)$ for $i \in [1,3]$, computing the upper bound of the algebraic degree for the multivariate case is also a natural optimization problem[9], as shown

[9] Indeed, this problem can also be solved in $\mathcal{O}(n)$ time with the algorithm in [28].

below:

$$\begin{aligned}&\text{maximize } H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i)\Big) + H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\lambda_i)\Big),\\ &\text{subject to } 0 \le \gamma_i + \lambda_i \le N_i^r \text{ for } i \in [0, n-1].\end{aligned}$$

Why $0 \le \gamma_i + \lambda_i \le N_i^r$ should hold is due to $(w^0_{d_{i,j}}, u^0_{d_{i,j}}) \neq (1,1)$ for any index $d_{i,j} \in [1,3]$.

It is easy to observe that this model is almost the same as that for the univariate case. Applying it to the Chaghri parameters $(k_0, k_1, k_2, n) = (32, 0, 3, 63)$, we obtain the following upper bound for the algebraic degree D_r after r steps, as shown in Table 4. Note that we still use D_r to denote the upper bound for the algebraic degree for r-step Chaghri in the multivariate case. This indicates that the higher-order differential distinguisher can reach at most $26 + 1 = 27$ steps (i.e. 13.5 rounds) using 2^{126} data.

Table 4: The upper bounds for D_r in the multivariate case

r	0	2	4	6	8	10	12	14	16	18	20	22	24	26	27
D_r	1	4	10	20	30	40	50	60	70	80	90	100	111	121	126

The refined upper bounds. Similar to the refined upper bounds for the univariate case, we are interested whether the data complexity of the 13.5-round higher-order differential attack can be further optimized. Specifically, we re-evaluate the upper bound for the algebraic degree by considering the following input form:

$$z_{0,1} = X_1, z_{0,2} = X_2, z_{0,3} = A_3. \tag{10}$$

where only X_1, X_2 are variables. Moreover, we consider the case when X_1 traverses all the 2^n possible values because only in this case will we need to consider the multivariate polynomials to attack more steps. In this case, let $X = B \circ G(X_1 + RK[0]_1)$ and $Y = X_2 + RK[0]_2$. Hence, X will traverse all the 2^n possible values. In this way, we have

$$\begin{aligned}z_{1,1} &= A_{1,1}X + B_{1,1}Y^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + C_{1,1},\\ z_{1,2} &= A_{1,2}X + B_{1,2}Y^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + C_{1,2},\\ z_{1,3} &= A_{1,3}X + B_{1,3}Y^{\mathcal{M}_n(2^{k_0+k_2}+2^{k_1+k_2})} + C_{1,3},\end{aligned}$$

where only X, Y are variables. Hence, we have

$$W_1 = \{(0,0), (1,0), (0, \mathcal{M}_n(2^{k_0+k_2} + 2^{k_1+k_2}))\}.$$

Moreover, we have

$$\begin{aligned}W_r = \{(e_0, e_1)|\\ &e_0 = \mathcal{M}_n\big(\sum_{i=1}^{N_{n-1}^r} 2^{n-1} w_{1,d_{i,n-1}} + \sum_{i=1}^{N_{n-2}^r} 2^{n-2} w_{1,d_{i,n-2}} + \ldots + \sum_{i=1}^{N_0^r} 2^0 w_{1,d_{i,0}}\big),\\ &e_1 = \mathcal{M}_n\big(\sum_{i=1}^{N_{n-1}^r} 2^{n-1} u_{1,d_{i,n-1}} + \sum_{i=1}^{N_{n-2}^r} 2^{n-2} u_{1,d_{i,n-2}} + \ldots + \sum_{i=1}^{N_0^r} 2^0 u_{1,d_{i,0}}\big),\\ &\text{where } 1 \leq d_{i,j} \leq |W_1| = 3 \text{ for } 0 \leq j \leq n-1\},\end{aligned}$$

where

$$\begin{cases} N_0^1 = 1, N_i^1 = 0 \text{ for } i \in [1, n-1],\\ N_i^r = N_{(i-k_1-k_2)\%n}^{r-1} + N_{(i-k_0-k_2)\%n}^{r-1} \text{ for } 0 \leq i \leq n-1,\ r \geq 2.\end{cases}$$

In this way, computing D_r is equivalent to solving the following optimization problem:

$$\begin{aligned}&\text{maximize } H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)\Big) + H\Big(\mathcal{M}_n\ \big(\sum_{i=0}^{n-1} 2^i (2^{k_0+k_2}\lambda_i + 2^{k_1+k_2}\lambda_i)\ \big)\ \Big),\\ &\text{subject to } H\Big(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i \gamma_i)\Big) = n,\ 0 \leq \gamma_i + \lambda_i \leq N_i^r \text{ for } i \in [0, n-1].\end{aligned}$$

Note that γ_i represents that we assign nonzero values to γ_i variables $w_{1,d_{j,i}}$ and λ_i represents that we assign nonzero values to λ_i variables $u_{1,d_{j,i}}$. Since $(w_{1,d_{j,i}}, u_{1,d_{j,i}})$ cannot be assigned to nonzero values at the same time due to $W_1 = \{(0,0), (1,0), (0, \mathcal{M}_n(2^{k_0+k_2} + 2^{k_1+k_2}))\}$, we have the constraint $0 \leq \gamma_i + \lambda_i \leq N_i^r$. Moreover, since X will take all the 2^n possible values, we only are interested in the monomials of the form $X^{\rho_1}Y^{\rho_2}$ where $H(\rho_1) \geq n$, i.e. $H(\rho_1) = n$. This is because for the monomial $X^{\rho_1}Y^{\rho_2}$ where $H(\rho_1) < n$, when X takes all the 2^n possible values, the corresponding sum of $X^{\rho_1}Y^{\rho_2}$ is always 0. Hence, we add the constraint $H(\mathcal{M}_n(\sum_{i=0}^{n-1} 2^i\gamma_i)) = n$.

With this model for Chaghri, we obtain in seconds that

$$D_{27} = 122, \quad D_{28} = 126,$$

which are indeed consistent with Theorem 1, i.e. we cannot increase the length of the distinguisher with the refined model. However, $D_{27} = 122$ indicates that the data and time complexity of the 13.5-round distinguisher are both 2^{123}, which improves the results obtained from the unrefined model by a factor of 2^3.

5 Achieving an Almost Exponential Degree Increase

Based on our degree evaluation, it can be observed that the algebraic degree of Chaghri increases linearly in both the univariate case and multivariate case, which contradicts the designers' expectation that it increases exponentially. Therefore, it is natural to ask what countermeasures can be used to achieve an exponential increase of the algebraic degree. In this section, we focus on this problem.

For FHE-friendly ciphers, reducing the multiplicative depth is of great importance. Hence, we still keep the S-box of the form $G(x) = x^{2^{k_0}+2^{k_1}}$, which has algebraic degree 2. For the affine transform $B(x)$, as it is linear over $\mathbb{F}_{2^n}$ and it is almost cost-free for FHE protocols, we are interested whether choosing a different $B(x)$ can achieve an exponential increase of the algebraic degree.

In appendix A, a concrete example is given to explain the influence of the affine transform on the increase of the algebraic degree. In the following, we mainly deal with a general affine transform.

5.1 Searching for Secure Affine Transforms $B(x)$

We consider a general form of $B(x)$ where we omit the constant part, i.e. $B(x) = \sum_{i=1}^{|\mathcal{L}|} c'_i x^{2^{\varphi_i}}$, where $(c'_1, c'_2, \ldots, c'_{|\mathcal{L}|})$ are constants in $\mathbb{F}_{2^{63}}$ such that $B(x)$ is a permutation and $\mathcal{L} = \{\varphi_1, \varphi_2, \ldots, \varphi_{|\mathcal{L}|}\}$. For the S-box, we keep using $G(x) = x^{2^{32}+1}$.

To utilize our coefficient grouping technique for the above general $B(x)$, we need to adjust the general polynomial representation of $(z_{r,1}, z_{r,2}, z_{r,3})$. First, consider the univariate case and the form of $(z_{r,1}, z_{r,2}, z_{r,3})$ can be written as follows where only X is the variable:

$$\begin{aligned}
z_{r,1} &= \sum_{i=1}^{|E_{r,1}|} A_{1,i} X^{\omega_{r,1,i}} + \sum_{i=1}^{|E_{r,2}|} A_{2,i} X^{\omega_{r,2,i}} + \ldots \sum_{i=1}^{|E_{r,l_r}|} A_{3,i} X^{\omega_{r,l_r,i}},\\
z_{r,2} &= \sum_{i=1}^{|E_{r,1}|} B_{1,i} X^{\omega_{r,1,i}} + \sum_{i=1}^{|E_{r,2}|} B_{2,i} X^{\omega_{r,2,i}} + \ldots \sum_{i=1}^{|E_{r,l_r}|} B_{3,i} X^{\omega_{r,l_r,i}},\\
z_{r,3} &= \sum_{i=1}^{|E_{r,1}|} C_{1,i} X^{\omega_{r,1,i}} + \sum_{i=1}^{|E_{r,2}|} C_{2,i} X^{\omega_{r,2,i}} + \ldots \sum_{i=1}^{|E_{r,l_r}|} C_{3,i} X^{\omega_{r,l_r,i}},
\end{aligned}$$

where

$$E_{r,j} = \{\omega_{r,j,1}, \omega_{r,j,2}, \ldots, \omega_{r,j,|E_{r,j}|}\} \text{ for } 1 \le j \le l_r.$$

In this way, the set of all possible exponents for $(z_{r,1}, z_{r,2}, z_{r,3})$ denoted by E_r can be written as

$$E_r = \bigcup_{i=1}^{l_r} E_{r,i}.$$

For the initial input $(z_{0,1}, z_{0,2}, z_{0,3})$, we use the same form as specified in Equation 1. In this way, we have

$$E_0 = w_0 = \{0, 1\} = \{w_{0,1}, w_{0,2}\}.$$

Next, we study how the new general polynomial representation evolves through 1 step of Chaghri. First,

$$\begin{aligned}G(z_{r,i}) &= (\sum_{i=1}^{l_r}\sum_{j=1}^{|E_{r,i}|} A_{i,j}X^{\omega_{r,i,j}})^{2^{32}+1}\\ &= \sum_{i=1}^{l_r}\sum_{j=1}^{|E_{r,i}|}\sum_{s=1}^{l_r}\sum_{t=1}^{|E_{r,s}|} A_{i,j,s,t}X^{\mathcal{M}_{63}(2^{32}\omega_{r,i,j}+\omega_{r,s,t})},\\ B\circ G(z_{r,1}) &= \sum_{i=1}^{l_r}\sum_{j=1}^{|E_{r,i}|}\sum_{s=1}^{l_r}\sum_{t=1}^{|E_{r,s}|}\sum_{u=1}^{|\mathcal{L}|} A_{i,j,s,t,u}X^{\mathcal{M}_{63}(2^{32+\varphi_u}\omega_{r,i,j}+2^{\varphi_u}\omega_{r,s,t})}.\end{aligned}$$

Hence,

$$\begin{aligned}E_{r+1} = \{e|e = \mathcal{M}_{63}(2^{32+\varphi_u}\omega_{r,i,j} + 2^{\varphi_u}\omega_{r,s,t}),\\ 1\leq i,s\leq l_r,\ 1\leq j\leq |E_{r,i}|,\ 1\leq t\leq |E_{r,s}|, 1\leq u\leq |\mathcal{L}|\}.\end{aligned}$$

Based on the above recursive relation between E_r and E_{r+1}, with the coefficient grouping technique, E_r can be represented as follows:

$$\begin{aligned}E_r &= \bigcup_{j=1}^{l_r} E_{r,j},\\ E_{r,j} &= \{e|e = \mathcal{M}_{63}(\sum_{i=1}^{N_{62}^{r,j}} 2^{62}w_{0,d_{i,62}} + \sum_{i=1}^{N_{61}^{r,j}} 2^{61}w_{0,d_{i,61}} + \ldots + \sum_{i=1}^{N_{0}^{r,j}} 2^{0}w_{0,d_{i,0}}),\\ &\quad \text{where } 1\leq d_{i,i_0}\leq |w_0| \text{ for } 0\leq i_0\leq 62\}.\end{aligned}$$

Proof. For E_0, there are

$$\begin{aligned}E_0 &= E_{0,1} = w_0 = \{0,1\} = \{w_{0,1}, w_{0,2}\},\\ E_{0,1} &= \{e|e = \mathcal{M}_{63}(2^0 w_{0,i}),\ 1\leq i\leq |w_0|\}.\end{aligned}$$

Hence, it holds for $r = 0$. Supposing the above new representation of E_r holds, we now prove by induction that it also holds for E_{r+1}. In particular, a similar useful recursive relation can be derived.

Since

$$\begin{aligned}E_{r+1} = \{e|e = \mathcal{M}_{63}(2^{32+\varphi_u}\omega_{r,i,j} + 2^{\varphi_u}\omega_{r,s,t}),\\ 1\leq i,s\leq l_r,\ 1\leq j\leq |E_{r,i}|,\ 1\leq t\leq |E_{r,s}|, 1\leq u\leq |\mathcal{L}|\},\end{aligned}$$

we have

$$\begin{aligned}E_{r+1} &= \bigcup_{i=1}^{l_r}\bigcup_{s=1}^{l_r}\bigcup_{u=1}^{|\mathcal{L}|} E_{r+1,i,s,u},\\ E_{r+1,i,s,u} &= \{e|e = \mathcal{M}_{63}(2^{32+\varphi_u}\omega_{r,i,j} + 2^{\varphi_u}\omega_{r,s,t}),\ 1\leq j\leq |E_{r,i}|,\ 1\leq t\leq |E_{r,s}|\}.\end{aligned}$$

Since

$$E_{r,j} = \{e | e = \mathcal{M}_{63}(\sum_{i=1}^{N_{62}^{r,j}} 2^{62} w_{0,d_{i,62}} + \sum_{i=1}^{N_{61}^{r,j}} 2^{61} w_{0,d_{i,61}} + \ldots + \sum_{i=1}^{N_{0}^{r,j}} 2^{0} w_{0,d_{i,0}}),$$
$$\text{where } 1 \le d_{i,i_0} \le |w_0| \text{ for } 0 \le i_0 \le 62\},$$

we have

$$E_{r+1,i,s,u} = \{e | e = \mathcal{M}_{63}(\sum_{j=1}^{N_{62}^{r+1,i,s,u}} 2^{62} w_{0,d_{j,62}} + \ldots + \sum_{j=1}^{N_{0}^{r+1,i,s,u}} 2^{0} w_{0,d_{j,0}}),$$
$$\text{where } 1 \le d_{j,j_0} \le |w_0| \text{ for } 0 \le j_0 \le 62\},$$

where

$$N_t^{r+1,i,s,u} = N_{(t-32-\varphi_u)\%63}^{r,i} + N_{(t-\varphi_u)\%63}^{r,s} \text{ for } t \in [0, 62]. \tag{11}$$

□

With the above critical observation, we can always decompose E_r as a union of sets, each of which can be solely described with a vector of integers $(N_{62}, N_{61}, \ldots, N_0)$. Moreover, since $E_0 = w_0$, a single vector of integers $(N_{62}^{0,1}, N_{61}^{0,1}, \ldots, N_{61}^{0,1})$ is sufficient to describe E_0 where

$$N_0^{0,1} = 1, N_i^{0,1} = 0 \text{ for } i \in [1, 62].$$

Then, based on the recursive relation specified in Equation 11, for each E_r $(r \ge 1)$, we can compute the corresponding sets of vectors of integers to represent E_r. The algorithm is shown in Algorithm 2, where $\mathbf{N}^r$ and $\mathbf{N}^{r+1}$ are the sets of possible vectors of integers describing E_r and E_{r+1}, respectively.

In Algorithm 2, there is a function named `REDUCE`. This is used to remove the redundant vectors based on the fact that when there are two vectors $(N_{62}, N_{61}, \ldots, N_0)$ and $(N'_{62}, N'_{61}, \ldots, N'_0)$ such that $N_i \ge N'_i$ for each $i \in [0, 62]$, the set described with $(N'_{62}, N'_{61}, \ldots, N'_0)$ is just a subset of the set described with $(N_{62}, N_{61}, \ldots, N_0)$.

The main idea to search for a good affine transform. With Algorithm 2, it is now possible to describe how to search for a better affine transform. Specifically, for each E_r, there exist l_r vectors of integers $(N_{62}^{r,i}, N_{61}^{r,i}, \ldots, N_0^{r,i})$ to describe $E_{r,i}$ for $i \in [1, l_r]$. Moreover, if there exists a vector $(N_{62}^{r,i}, N_{61}^{r,i}, \ldots, N_0^{r,i})$ where there are $\mathbb{D}$ nonzero elements, it implies the upper bound for the algebraic degree after r steps is larger than $\mathbb{D}$. This is because it implies that there exists an element $e \in E_r$ such that $H(e) = \mathbb{D}$. Hence, to achieve an exponential increase for the first r $(1 \le r \le 5)$ steps, we need to ensure that there exists at least one vector $(N_{62}^{r,i}, N_{61}^{r,i}, \ldots, N_0^{r,i})$ where there are 2^r nonzero elements. For $r = 6$, we can slightly relax the constraint and expect that after 7 steps, the maximal degree 63 is reached, i.e. there exists a vector $(N_{62}^{7,i}, N_{61}^{7,i}, \ldots, N_0^{7,i})$ where all the

Algorithm 2 Enumerating vectors to represent E_{r+1}

```
procedure ENU($\mathbf{N}^r, \mathbf{N}^{r+1}, \mathcal{L}$)
    clear $\mathbf{N}^{r+1}$
    for $i$ in range ($\mathbf{N}^r$.size()) do
        $(N_{62}^0, N_{61}^0, \ldots, N_0^0) \leftarrow \mathbf{N}^r[i]$
        for $s$ in range ($\mathbf{N}^r$.size()) do
            $(N_{62}^1, N_{61}^1, \ldots, N_0^1) \leftarrow \mathbf{N}^r[s]$
            for $u \in [1, |\mathcal{L}|])$ do
                for $t \in [0, 62]$ do
                    $N_t = N^0_{(t-32-\varphi_u)\%63} + N^1_{(t-\varphi_u)\%63}$
                if REDUCE($N_{62}, N_{61}, \ldots, N_0, \mathbf{N}^{r+1}$)=1 then
                    add $(N_{62}, N_{61}, \ldots, N_0)$ to $\mathbf{N}^{r+1}$
procedure REDUCE($N_{62}, N_{61}, \ldots, N_0, \mathbf{N}$)
    for $i$ in range ($\mathbf{N}$.size()) do
        $(N'_{62}, N'_{61}, \ldots, N'_0) \leftarrow \mathbf{N}[i]$
        if $N_j \geq N'_j$ for all $j \in [0, 62]$ then
            $\mathbf{N}[i] = (N_{62}, N_{61}, \ldots, N_0)$
            return 0
        else if $N'_j \geq N_j$ for all $j \in [0, 62]$ then
            return 0
    return 1
```

elements are nonzero or there exists a vector $(N_{62}^{7,i}, N_{61}^{7,i}, \ldots, N_0^{7,i})$ such that the solution to the following optimization problem is 63:

$$\text{maximize } \; H\left(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \gamma_j)\right),$$

$$\text{subject to } \; 0 \leq \gamma_j \leq N_j^{7,i} \text{ for } j \in [0, 62].$$

Searching with heuristic strategies. For $r = 0$, there are

$$l_0 = 1, \; \mathbf{N}^0 = \{(0, 0, \ldots, 0, 1)\}.$$

Then, based on Algorithm 2, for any $r \geq 1$, we can always compute $\mathbf{N}^r$ for any given $\mathcal{L}$. However, the time complexity to compute $\mathbf{N}^r$ becomes exponential in r when $|\mathcal{L}| > 1$ due to the fast diffusion of the monomials. Even for small r, e.g. $r = 5$, if we aim to compute the full set of vectors, it cannot be finished in practical time. However, since we are only interested in vectors where there are a desired number of nonzero elements, we can use some heuristic strategies when computing $\mathbf{N}^r$.

Specifically, for the first r steps ($1 \leq r \leq 5$), we only add the vectors where there are 2^r nonzero elements to $\mathbf{N}^r$ when running Algorithm 2. The underlying reason is that to generate a monomial whose exponent is of hamming weight 2^r at step r, it is required to have two monomials (X^{e_0}, X^{e_1}) where $H(e_0) = H(e_1) = 2^{r-1}$ at step $r - 1$. When there exists an empty set $\mathbf{N}^r$ for

$1 \le r \le 5$, we abandon the current $\mathcal{L}$ and try another $\mathcal{L}$ since it implies we cannot reach the algebraic degree 2^r with the current $\mathcal{L}$. Based on this strategy, we find no candidates for $\mathcal{L}$ when $|\mathcal{L}| = 2$.

Hence, $|\mathcal{L}| = 3$ is taken into account. For $1 \le r \le 5$, we still use the above strategies. However, the size of $\mathbf{N}^r$ will increase exponentially. Hence, we further restrict that when the size of $\mathbf{N}^r$ is larger than 2^{13}, exit Algorithm 2 and compute $\mathbf{N}^{r+1}$. For $r = 6$, we only add the vectors where there are at least 55 nonzero elements to $\mathbf{N}^6$. For $r = 7$, when computing $\mathbf{N}^7$ with Algorithm 2, we test whether there is one $(N_{62}^{7,i}, N_{61}^{7,i}, \ldots, N_0^{7,i})$ which can lead to the maximal degree 63. If there is, exit and treat the current $\mathcal{L}$ as a good affine transform. It is found that $\mathcal{L} = \{0, 2, 8\}$ is such a candidate.

With $\mathcal{L} = \{0, 2, 8\}$, for the input of the form specified in Equation 1, the algebraic degree can reach 63 after 7 steps. Therefore, for the input of the form specified in Equation 8, the algebraic degree can reach 63 after 8 steps, which is a direct application of Theorem 1. In this way, an almost exponential increase of the algebraic degree is achieved in the univariate setting.

5.2 Evaluating the Algebraic Degree for the Multivariate Case

After obtaining a good affine transform $B(x)$ which can ensure an almost exponential increase of the algebraic degree in the univariate setting, we study how the algebraic degree increases in the multivariate setting. In general, after we reach the maximal algebraic degree in the univariate case, due to the strong diffusion of the MDS matrix and the affine transform, the maximal algebraic degree in the multivariate case can be reached in a few more steps. For Chaghri, we only care about the distinguisher with data complexity and time complexity below 2^{128} since Chaghri only provides 128-bit security. Hence, we only care about when the algebraic degree 128 is reached.

On two variables. We first consider the input of the form specified in Equation 9. Then, similar to the above analysis, the general polynomial representation of $(z_{r,1}, z_{r,2}, z_{r,3})$ can be written as follows:

$$z_{r,1} = \sum_{i=1}^{l_r} \sum_{j=1}^{|U_{r,i}|} A_{i,j} X^{\omega_{r,i,j}} Y^{\mu_{r,i,j}},\ z_{r,2} = \sum_{i=1}^{l_r} \sum_{j=1}^{|U_{r,i}|} B_{i,j} X^{\omega_{r,i,j}} Y^{\mu_{r,i,j}},$$

$$z_{r,3} = \sum_{i=1}^{l_r} \sum_{j=1}^{|U_{r,i}|} C_{i,j} X^{\omega_{r,i,j}} Y^{\mu_{r,i,j}},$$

where

$$U_{r,i} = \{(\omega_{r,i,1}, \mu_{r,i,1}), (\omega_{r,i,2}, \mu_{r,i,2}), \ldots, (\omega_{r,i,|U_{r,i}|}, \mu_{r,i,|U_{r,i}|})\} \text{ for } i \in [1, l_r]$$

and $U_r = \bigcup_{i=1}^{l_r} U_{r,i}$ is the set of all possible exponents for $(z_{r,1}, z_{r,2}, z_{r,3})$.

For the input form specified in Equation 9, we have

$$l_0 = 1,\ U_0 = U_{0,1} = W_0,$$

$$W_0 = \{(0,1),(1,0),(0,0)\} = \{(w_{0,1}, u_{0,1}), (w_{0,2}, u_{0,2}), (w_{0,3}, u_{0,3})\}.$$

Then, by tracing the evolution of the polynomials through 1 step of Chaghri, we can similarly derive

$$\begin{aligned} U_{r+1} = \{&(e_0, e_1)| \\ &e_0 = \mathcal{M}_{63}(2^{32+\varphi_u}\omega_{r,i,j} + 2^{\varphi_u}\omega_{r,s,t}),\ e_1 = \mathcal{M}_{63}(2^{32+\varphi_u}\mu_{r,i,j} + 2^{\varphi_u}\mu_{r,s,t}), \\ &1 \leq i, s \leq l_r,\ 1 \leq j \leq |U_{r,i}|,\ 1 \leq t \leq |U_{r,s}|, 1 \leq u \leq |\mathcal{L}|\}. \end{aligned}$$

With the coefficient grouping technique, similarly, $U_{r,j}$ $(1 \leq j \leq l_r)$ can be represented as

$$\begin{aligned} U_{r,j} = \{&(e_0, e_1)| \\ &e_0 = \mathcal{M}_{63}(\sum_{i=1}^{N_{62}^{r,j}} 2^{62} w_{0,d_{i,62}} + \sum_{i=1}^{N_{61}^{r,j}} 2^{61} w_{0,d_{i,61}} + \ldots + \sum_{i=1}^{N_{0}^{r,j}} 2^{0} w_{0,d_{i,0}}), \\ &e_1 = \mathcal{M}_{63}(\sum_{i=1}^{N_{62}^{r,j}} 2^{62} u_{0,d_{i,62}} + \sum_{i=1}^{N_{61}^{r,j}} 2^{61} u_{0,d_{i,61}} + \ldots + \sum_{i=1}^{N_{0}^{r,j}} 2^{0} u_{0,d_{i,0}}), \\ &1 \leq d_{i,i_0} \leq |W_0|, 0 \leq i_0 \leq 62\}, \end{aligned}$$

where $W_0 = \{(w_{0,1}, u_{0,1}), (w_{0,2}, u_{0,2}), (w_{0,3}, u_{0,3})\} = \{(0,1),(1,0),(0,0)\}$. Moreover, the recursive relation remains the same as in the univariate case, i.e. Equation 11. In other words, it is sufficient to describe U_r with a set of vectors of integers and we still denote the set by $\mathbf{N}^r$ to avoid the abuse of notation. Then,

$$\mathbf{N}^0 = \{(0,0,\ldots,0,1)\}$$

and Algorithm 2 can be directly used to compute $\mathbf{N}^r$ for $r \geq 1$.

Supposing there exists a vector $(N_{62}^{r,i}, N_{61}^{r,i}, \ldots, N_0^{r,i})$ in $\mathbf{N}^r$ such that the solution to the following optimization problem is 126, we reach the maximal degree for the input of the form in Equation 9 after r steps.

$$\begin{aligned} &\text{maximize}\ \ H\Big(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \gamma_j)\Big) + H\Big(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \lambda_j)\Big), \\ &\text{subject to}\ \ 0 \leq \gamma_j + \lambda_j \leq N_j^{r,i} \text{ for } j \in [0, 62]. \end{aligned}$$

Moreover, for the input of the form specified in Equation 10, the degree 126 can be reached after $r+1$ steps by applying Theorem 1.

For $\mathcal{L} = \{0, 2, 8\}$, the maximal degree 126 can be reached at $r = 9$ for the input specified in Equation 10. This implies that 9 steps are secure against the higher-order differential distinguishing attack with complexity below 2^{126}. Compared with the univariate case, only at most 1 more step can be reached. This is indeed as expected due to the strong diffusion effect of the affine transform and MDS matrix.

On three variables. Since the algebraic degree will reach 126 after 9 steps when there are 2 variables, we can argue that the algebraic degree will be much larger than 128 after 9 or 10 steps when considering 3 variables. For completeness, we also consider the case when there are 3 variables.

Consider the following input of the form:

$$\begin{aligned} z_{0,1} &= A_{0,1}X + B_{0,1}Y + C_{0,1}Z, \\ z_{0,2} &= A_{0,2}X + B_{0,2}Y + C_{0,2}Z, \\ z_{0,3} &= A_{0,3}X + B_{0,3}Y + C_{0,3}Z, \end{aligned}$$

where X, Y, Z are variables.

Then, we will have an initial set U_0 of all possible exponents where

$$U_0 = W_0 = \{(0,0,1), (0,1,0), (1,0,0), (0,0,0)\}.$$

To avoid the abuse of notation, we use the same notation as in the case for 2 variables. Then, it can be similarly derived that U_r can be fully described with a set of vectors of integers denoted by $\mathbf{N}^r$ where $\mathbf{N}^0 = \{(0,0,\ldots,0,1)\}$ and $\mathbf{N}^r$ ($r \geq 1$) can be computed with Algorithm 2. With $\mathbf{N}^r$ ($r \geq 1$), it is possible to give a lower bound for the algebraic degree after r steps for the above input polynomials in (X, Y, Z). Specifically, if there exists a vector $(N_{62}^{r,i}, N_{61}^{r,i}, \ldots, N_0^{r,i})$ in $\mathbf{N}^r$ such that the solution to the following optimization problem is $\texttt{SOL}$, the lower bound is $\texttt{SOL}$:

$$\begin{aligned} &\text{maximize } \; H\Big(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \gamma_j)\Big) + H\Big(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \lambda_j)\Big) + H\Big(\mathcal{M}_{63}(\sum_{j=0}^{62} 2^j \chi_j)\Big), \\ &\text{subject to } \; 0 \leq \gamma_j + \lambda_j + \chi_j \leq N_j^{r,i} \text{ for } j \in [0, 62]. \end{aligned}$$

As Chaghri only provides 128-bit security, we only need to ensure $\texttt{SOL} \geq 128$. It is found that $\texttt{SOL} \geq 187$ when $r = 8$ and $\texttt{SOL} = 189$ when $r = 9$, which imply that 9 steps are secure against our higher-order differential distinguishing attack.

5.3 New Parameters for Chaghri

According to [6], the total number of rounds T is chosen with the formula $T = 1.5 \times \max\{5, \eta\}$, where η is the maximal number of rounds that can be attacked with time complexity below 2^{128}. With $\mathcal{L} = \{0, 2, 8\}$, we have $\eta = 4$ and hence the total number of rounds T can be kept unchanged, i.e. $T = 8$. In the following, we give an optional assignment to (c_1', c_2', c_3', c_4') such that $B(x) = c_1'x + c_2'x^4 + c_3'x^{256} + c_4'$ is a permutation.

$$\begin{aligned} c_1' = \alpha^{60} &+ \alpha^{57} + \alpha^{54} + \alpha^{53} + \alpha^{49} + \alpha^{48} + \alpha^{45} + \alpha^{42} + \alpha^{40} + \alpha^{38} \\ &+ \alpha^{35} + \alpha^{33} + \alpha^{27} + \alpha^{26} + \alpha^{25} + \alpha^{24} + \alpha^{23} + \alpha^{21} + \alpha^{19} + \alpha^{18} \\ &+ \alpha^{16} + \alpha^{14} + \alpha^{13} + \alpha^{12} + \alpha^{9} + \alpha^{8} + \alpha^{6} + \alpha^{4}, \end{aligned}$$

$$\begin{aligned}
c_2' = {} & \alpha^{61} + \alpha^{60} + \alpha^{59} + \alpha^{58} + \alpha^{57} + \alpha^{55} + \alpha^{51} + \alpha^{50} + \alpha^{49} + \alpha^{48} \\
& + \alpha^{46} + \alpha^{45} + \alpha^{44} + \alpha^{41} + \alpha^{38} + \alpha^{35} + \alpha^{30} + \alpha^{29} + \alpha^{27} + \alpha^{26} \\
& + \alpha^{14} + \alpha^{12} + \alpha^{10} + \alpha^{9} + \alpha^{8} + \alpha^{7} + \alpha^{5} + \alpha, \\
c_3' = {} & \alpha^{62} + \alpha^{61} + \alpha^{57} + \alpha^{53} + \alpha^{52} + \alpha^{50} + \alpha^{48} + \alpha^{47} + \alpha^{46} + \alpha^{45} \\
& + \alpha^{44} + \alpha^{43} + \alpha^{38} + \alpha^{36} + \alpha^{35} + \alpha^{34} + \alpha^{32} + \alpha^{30} + \alpha^{27} + \alpha^{26} \\
& + \alpha^{24} + \alpha^{19} + \alpha^{18} + \alpha^{16} + \alpha^{14} + \alpha^{12} + \alpha^{11} + \alpha^{8} + \alpha^{7} + \alpha^{5} + \alpha^{2} + \alpha, \\
c_4' = {} & \alpha^{62} + \alpha^{55} + \alpha^{54} + \alpha^{52} + \alpha^{50} + \alpha^{49} + \alpha^{43} + \alpha^{40} + \alpha^{39} + \alpha^{38} + \alpha^{37} \\
& + \alpha^{36} + \alpha^{35} + \alpha^{34} + \alpha^{32} + \alpha^{31} + \alpha^{29} + \alpha^{26} + \alpha^{25} + \alpha^{24} + \alpha^{23} + \alpha^{22} \\
& + \alpha^{21} + \alpha^{18} + \alpha^{15} + \alpha^{12} + \alpha^{11} + \alpha^{10} + \alpha^{5} + \alpha^{2} + \alpha.
\end{aligned}$$

6 Conclusion

We perform an in-depth study on the increase of the algebraic degree of Chaghri by proposing a novel efficient technique called coefficient grouping. In its core, it is an efficient way to represent the polynomial of any rounds of Chaghri in terms of the input. Especially, such an efficient representation can be determined by a single vector of integers that can be computed in linear time. Benefiting from this representation, upper bounding the algebraic degree is reduced to some well-structured optimization problems that can be efficiently solved by either an $\mathcal{O}(n)$ algorithm [28] or the general-purpose solvers.

One important feature of this technique is to use efficient recursive relations to equivalently describe the heavy monomial transitions through the round function. However, in the well-known technique called division property, this has to be modelled round by round and the corresponding cost is high. Especially, with our technique applied to Chaghri, we can compute relatively tight upper bounds of the algebraic degree in linear time, which obviously distinguish it from other techniques.

With this technique, we can break the full 8 rounds of Chaghri with low complexity and can even break up to 13.5 rounds. This is in a way indicates that the lack of techniques to analyze such primitives defined over a large field is still a major issue. With the coefficient grouping technique, we further make a step towards this important question. Specifically, we not only attack a cipher with it, but also describe how to use it to search for secure cryptographic components.

Finally, we mention a few open questions:

- How to find a relatively sparse affine layer that can help achieve the exponential degree increase in a more efficient way?
- Is there a more refined (or even analytical) method to explain how much dense the affine layer should be to achieve an exponential degree increase?
- How to give tight upper bounds of the algebraic degree when the affine layer is dense?

Acknowledgement. We thank Mohammad Mahzoun and Tomer Ashur for carefully checking our results. We also thank the reviewers of EUROCRYPT

2023 for providing many insightful comments. Fukang Liu is supported by Grant-in-Aid for Research Activity Start-up (Grant No. 22K21282). 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. Libo Wang is supported by the National Natural Science Foundation of China under Grants 62102167, 62032025, and by the Guangdong Basic and Applied Basic Research Foundation under Grants 2022A1515010299, 2020A1515110364.

References

1. Albrecht, M.R., Cid, C., Grassi, L., Khovratovich, D., Lüftenegger, R., Rechberger, C., Schofnegger, M.: Algebraic Cryptanalysis of STARK-Friendly Designs: Application to MARVELlous and MiMC. In: ASIACRYPT (3). Lecture Notes in Computer Science, vol. 11923, pp. 371–397. Springer (2019). https://doi.org/10.1007/978-3-030-34618-8_13
2. Albrecht, M.R., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10031, pp. 191–219 (2016). https://doi.org/10.1007/978-3-662-53887-6_7
3. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 9056, pp. 430–454. Springer (2015). https://doi.org/10.1007/978-3-662-46800-5_17
4. Aly, A., Ashur, T., Ben-Sasson, E., Dhooghe, S., Szepieniec, A.: Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. IACR Trans. Symmetric Cryptol. **2020**(3), 1–45 (2020). https://doi.org/10.13154/tosc.v2020.i3.1-45
5. Ashur, T., Dhooghe, S.: MARVELlous: a STARK-Friendly Family of Cryptographic Primitives. IACR Cryptol. ePrint Arch. p. 1098 (2018), https://eprint.iacr.org/2018/1098
6. Ashur, T., Mahzoun, M., Toprakhisar, D.: Chaghri — an FHE-friendly Block Cipher. Cryptology ePrint Archive, Paper 2022/592 (2022), https://eprint.iacr.org/2022/592
7. Beyne, T., Canteaut, A., Dinur, I., Eichlseder, M., Leander, G., Leurent, G., Naya-Plasencia, M., Perrin, L., Sasaki, Y., Todo, Y., Wiemer, F.: Out of Oddity - New Cryptanalytic Techniques Against Symmetric Primitives Optimized for Integrity Proof Systems. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 12172, pp. 299–328. Springer (2020). https://doi.org/10.1007/978-3-030-56877-1_11
8. Boura, C., Canteaut, A., Cannière, C.D.: Higher-Order Differential Properties of Keccak and *Luffa*. In: FSE. Lecture Notes in Computer Science, vol. 6733, pp. 252–269. Springer (2011). https://doi.org/10.1007/978-3-642-21702-9_15
9. Bouvier, C., Canteaut, A., Perrin, L.: On the Algebraic Degree of Iterated Power Functions. IACR Cryptol. ePrint Arch. p. 366 (2022), https://eprint.iacr.org/2022/366
10. Canteaut, A., Carpov, S., Fontaine, C., Lepoint, T., Naya-Plasencia, M., Paillier, P., Sirdey, R.: Stream Ciphers: A Practical Solution for Efficient Homomorphic-Ciphertext Compression. In: FSE. Lecture Notes in Computer Science, vol. 9783, pp. 313–333. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_16

11. Cid, C., Grassi, L., Gunsing, A., Lüftenegger, R., Rechberger, C., Schofnegger, M.: Influence of the Linear Layer on the Algebraic Degree in SP-Networks. IACR Trans. Symmetric Cryptol. **2022**(1), 110–137 (2022). https://doi.org/10.46586/tosc.v2022.i1.110-137
12. Cid, C., Indrøy, J.P., Raddum, H.: FASTA - A Stream Cipher for Fast FHE Evaluation. In: CT-RSA. Lecture Notes in Computer Science, vol. 13161, pp. 451–483. Springer (2022). https://doi.org/10.1007/978-3-030-95312-6_19
13. Cui, J., Hu, K., Wang, M., Wei, P.: On the Field-Based Division Property: Applications to MiMC, Feistel MiMC and GMiMC. In: ASIACRYPT (3). Lecture Notes in Computer Science, vol. 13793, pp. 241–270. Springer (2022). https://doi.org/10.1007/978-3-031-22969-5_9
14. Dinur, I.: Cryptanalytic Applications of the Polynomial Method for Solving Multivariate Equation Systems over GF(2). In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 12696, pp. 374–403. Springer (2021). https://doi.org/10.1007/978-3-030-77870-5_14
15. Dobraunig, C., Eichlseder, M., Grassi, L., Lallemand, V., Leander, G., List, E., Mendel, F., Rechberger, C.: Rasta: A Cipher with Low ANDdepth and Few ANDs per Bit. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 10991, pp. 662–692. Springer (2018). https://doi.org/10.1007/978-3-319-96884-1_22
16. Dobraunig, C., Grassi, L., Guinet, A., Kuijsters, D.: Ciminion: Symmetric Encryption Based on Toffoli-Gates over Large Finite Fields. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 12697, pp. 3–34. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_1
17. Dobraunig, C., Grassi, L., Helminger, L., Rechberger, C., Schofnegger, M., Walch, R.: Pasta: A Case for Hybrid Homomorphic Encryption. IACR Cryptol. ePrint Arch. p. 731 (2021), https://eprint.iacr.org/2021/731
18. Duval, S., Lallemand, V., Rotella, Y.: Cryptanalysis of the FLIP Family of Stream Ciphers. In: CRYPTO (1). Lecture Notes in Computer Science, vol. 9814, pp. 457–475. Springer (2016). https://doi.org/10.1007/978-3-662-53018-4_17
19. Eichlseder, M., Grassi, L., Lüftenegger, R., Øygarden, M., Rechberger, C., Schofnegger, M., Wang, Q.: An Algebraic Attack on Ciphers with Low-Degree Round Functions: Application to Full MiMC. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 12491, pp. 477–506. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_16
20. Grassi, L., Lüftenegger, R., Rechberger, C., Rotaru, D., Schofnegger, M.: On a Generalization of Substitution-Permutation Networks: The HADES Design Strategy. In: EUROCRYPT (2). Lecture Notes in Computer Science, vol. 12106, pp. 674–704. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_23
21. Ha, J., Kim, S., Choi, W., Lee, J., Moon, D., Yoon, H., Cho, J.: Masta: An HE-Friendly Cipher Using Modular Arithmetic. IEEE Access **8**, 194741–194751 (2020). https://doi.org/10.1109/ACCESS.2020.3033564
22. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for Three-Subset Division Property Without Unknown Subset - Improved Cube Attacks Against Trivium and Grain-128AEAD. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 12105, pp. 466–495. Springer (2020). https://doi.org/10.1007/978-3-030-45721-1_17
23. Hebborn, P., Leander, G.: Dasta - Alternative Linear Layer for Rasta. IACR Trans. Symmetric Cryptol. **2020**(3), 46–86 (2020). https://doi.org/10.13154/tosc.v2020.i3.46-86

24. Hu, K., Sun, S., Wang, M., Wang, Q.: An Algebraic Formulation of the Division Property: Revisiting Degree Evaluations, Cube Attacks, and Key-Independent Sums. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 12491, pp. 446–476. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_15
25. Liu, F., Isobe, T., Meier, W.: Cryptanalysis of Full LowMC and LowMC-M with Algebraic Techniques. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 12827, pp. 368–401. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_13
26. Liu, F., Sarkar, S., Meier, W., Isobe, T.: Algebraic Attacks on Rasta and Dasta Using Low-Degree Equations. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 13090, pp. 214–240. Springer (2021). https://doi.org/10.1007/978-3-030-92062-3_8
27. Liu, F., Sarkar, S., Wang, G., Meier, W., Isobe, T.: Algebraic Meet-in-the-Middle Attack on LowMC. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 13791, pp. 225–255. Springer (2022). https://doi.org/10.1007/978-3-031-22963-3_8
28. Liu, F., Wang, L.: An $\mathcal{O}(n)$ Algorithm for Coefficient Grouping. Cryptology ePrint Archive, Paper 2022/992 (2022), https://eprint.iacr.org/2022/992
29. Méaux, P., Journault, A., Standaert, F., Carlet, C.: Towards Stream Ciphers for Efficient FHE with Low-Noise Ciphertexts. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 9665, pp. 311–343. Springer (2016). https://doi.org/10.1007/978-3-662-49890-3_13
30. Rechberger, C., Soleimany, H., Tiessen, T.: Cryptanalysis of Low-Data Instances of Full LowMCv2. IACR Trans. Symmetric Cryptol. **2018**(3), 163–181 (2018). https://doi.org/10.13154/tosc.v2018.i3.163-181
31. Todo, Y., Morii, M.: Bit-Based Division Property and Application to Simon Family. In: FSE. Lecture Notes in Computer Science, vol. 9783, pp. 357–377. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_18
32. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP Method to Searching Integral Distinguishers Based on Division Property for 6 Lightweight Block Ciphers. In: ASIACRYPT (1). Lecture Notes in Computer Science, vol. 10031, pp. 648–678 (2016). https://doi.org/10.1007/978-3-662-53887-6_24

A Influence of the Affine Transform

We use the same S-box $G(x) = x^{2^{32}+1}$ while we use a different affine transform $B(x) = c_1'x^8 + c_2'x + c_3'$, where $c_1', c_2', c_3' \in \mathbb{F}_{2^{63}}$ are some constants making $B(x)$ a permutation.

Consider an input state $(z_{0,1}, z_{0,2}, z_{0,3})$ of the following form:

$$z_{0,1} = A_{0,1}X + A_{0,2},\ z_{0,2} = B_{0,1}X + B_{0,2},\ z_{0,3} = C_{0,1}X + C_{0,2},$$

where X is the variable. Then, we have

$$\begin{aligned} G(z_{0,1}) &= A'_{0,1}X^{2^{32}+1} + A'_{0,2}, \\ B \circ G(z_{0,1}) &= c_1'(A'_{0,1}X^{2^{32}+1} + A'_{0,2})^8 + c_2'(A'_{0,1}X^{2^{32}+1} + A'_{0,2}) + c_3' \\ &= (A''_{0,1}X^{2^{35}+2^3} + A''_{0,2}) + (A'''_{0,3}X^{2^{32}+1} + A'''_{0,4}). \end{aligned}$$

Similar to the previous analysis, we have

$$\begin{aligned}
z_{1,1} &= A_{1,1}X^{2^{35}+2^3} + A_{1,2}X^{2^{32}+1} + A_{1,3},\\
z_{1,2} &= B_{1,1}X^{2^{35}+2^3} + B_{1,2}X^{2^{32}+1} + B_{1,3},\\
z_{1,3} &= C_{1,1}X^{2^{35}+2^3} + C_{1,2}X^{2^{32}+1} + C_{1,3}.
\end{aligned}$$

Then, we consider one more step, i.e. we consider $G(z_{1,1})$ and $B \circ G(z_{1,1})$. For $G(z_{1,1})$, we have

$$\begin{aligned}
G(z_{1,1}) &= (A_{1,1}X^{2^{35}+2^3} + A_{1,2}X^{2^{32}+1} + A_{1,3})^{2^{32}}(A_{1,1}X^{2^{35}+2^3} + A_{1,2}X^{2^{32}+1} + A_{1,3})\\
&= (A'_{1,1}X^{2^{35}+2^4} + A'_{1,2}X^{2^{32}+2} + A'_{1,3})(A_{1,1}X^{2^{35}+2^3} + A_{1,2}X^{2^{32}+1} + A_{1,3})\\
&= A''_{1,1}X^{2^{36}+2^3+2^4} + A''_{1,2}X^{2^{35}+2^4+2^{32}+1} + A''_{1,3}X^{2^{35}+2^4}\\
&\quad +A''_{1,4}X^{2^{32}+2+2^{35}+2^3} + A''_{1,5}X^{2^{33}+2+1} + A''_{1,6}X^{2^{32}+2}\\
&\quad +A''_{1,7}X^{2^{35}+2^3} + A''_{1,8}X^{2^{32}+1} + A''_{1,9}.
\end{aligned}$$

For $B \circ G(z_{1,1})$, we have

$$\begin{aligned}
B \circ G(z_{1,1}) = {} & A'''_{1,1}X^{2^{36}+2^3+2^4} + A'''_{1,2}X^{2^{35}+2^4+2^{32}+1} + A'''_{1,3}X^{2^{35}+2^4}\\
&+A'''_{1,4}X^{2^{32}+2+2^{35}+2^3} + A'''_{1,5}X^{2^{33}+2+1} + A'''_{1,6}X^{2^{32}+2}\\
&+A'''_{1,7}X^{2^{35}+2^3} + A'''_{1,8}X^{2^{32}+1} + A'''_{1,9}\\
&\underline{+A'''_{1,10}X^{2^{39}+2^6+2^7} + A'''_{1,11}X^{2^{38}+2^7+2^{35}+2^3} + A'''_{1,12}X^{2^{38}+2^7}}\\
&\underline{+A'''_{1,13}X^{2^{35}+2^4+2^{38}+2^6} + A'''_{1,14}X^{2^{36}+2^4+2^3} + A'''_{1,15}X^{2^{35}+2^4}}\\
&\underline{+A'''_{1,16}X^{2^{38}+2^6} + A'''_{1,17}X^{2^{35}+2^3}}
\end{aligned}$$

Therefore, for $(z_{2,1}, z_{2,2}, z_{2,3})$, we have

$$\begin{aligned}
z_{2,1} = {} & A_{2,1}X^{2^{36}+2^3+2^4} + A_{2,2}X^{2^{35}+2^4+2^{32}+1} + A_{2,3}X^{2^{35}+2^4}\\
&+A_{2,4}X^{2^{32}+2+2^{35}+2^3} + A_{2,5}X^{2^{33}+2+1} + A_{2,6}X^{2^{32}+2}\\
&+A_{2,7}X^{2^{35}+2^3} + A_{2,8}X^{2^{32}+1} + A_{2,9}\\
&\underline{+A_{2,10}X^{2^{39}+2^6+2^7} + A_{2,11}X^{2^{38}+2^7+2^{35}+2^3} + A_{2,12}X^{2^{38}+2^7}}\\
&\underline{+A_{2,13}X^{2^{35}+2^4+2^{38}+2^6} + A_{2,14}X^{2^{36}+2^4+2^3} + A_{2,15}X^{2^{35}+2^4}}\\
&\underline{+A_{2,16}X^{2^{38}+2^6} + A_{2,17}X^{2^{35}+2^3}},\\
z_{2,2} = {} & B_{2,1}X^{2^{36}+2^3+2^4} + B_{2,2}X^{2^{35}+2^4+2^{32}+1} + B_{2,3}X^{2^{35}+2^4}\\
&+B_{2,4}X^{2^{32}+2+2^{35}+2^3} + B_{2,5}X^{2^{33}+2+1} + B_{2,6}X^{2^{32}+2}\\
&+B_{2,7}X^{2^{35}+2^3} + B_{2,8}X^{2^{32}+1} + B_{2,9}\\
&\underline{+B_{2,10}X^{2^{39}+2^6+2^7} + B_{2,11}X^{2^{38}+2^7+2^{35}+2^3} + B_{2,12}X^{2^{38}+2^7}}
\end{aligned}$$

$$
\begin{aligned}
&\underline{+B_{2,13}X^{2^{35}+2^4+2^{38}+2^6} + B_{2,14}X^{2^{36}+2^4+2^3} + B_{2,15}X^{2^{35}+2^4}}\\
&\underline{+B_{2,16}X^{2^{38}+2^6} + B_{2,17}X^{2^{35}+2^3}},\\
z_{2,1} = {}&C_{2,1}X^{2^{36}+2^3+2^4} + C_{2,2}X^{2^{35}+2^4+2^{32}+1} + C_{2,3}X^{2^{35}+2^4}\\
&+C_{2,4}X^{2^{32}+2+2^{35}+2^3} + C_{2,5}X^{2^{33}+2+1} + C_{2,6}X^{2^{32}+2}\\
&+C_{2,7}X^{2^{35}+2^3} + C_{2,8}X^{2^{32}+1} + C_{2,9}\\
&\underline{+C_{2,10}X^{2^{39}+2^6+2^7} + C_{2,11}X^{2^{38}+2^7+2^{35}+2^3} + C_{2,12}X^{2^{38}+2^7}}\\
&\underline{+C_{2,13}X^{2^{35}+2^4+2^{38}+2^6} + C_{2,14}X^{2^{36}+2^4+2^3} + C_{2,15}X^{2^{35}+2^4}}\\
&\underline{+C_{2,16}X^{2^{38}+2^6} + C_{2,17}X^{2^{35}+2^3}}.
\end{aligned}
$$

Hence, for the new affine transform $B(x)$, after 2 steps, the algebraic degree becomes 4 in the univariate case. While for the original $B(x)$ in Chaghri, the algebraic degree is only 3 after 2 steps. This is mainly because the new affine transform can make more different non-zero monomials appear in its output. Then, due to the S-box operation, much more possible monomials will appear in its output and the probability that there exists a monomial whose exponent is of hamming weight 4 increases.