

# Gröbner Basis Attack on STARK-Friendly Symmetric-Key Primitives: JARVIS, MiMC and GMiMC<sub>erf</sub>

Gizem Kara, Oğuz Yayla  
Institute of Applied Mathematics,  
Middle East Technical University,  
06800, Çankaya, Ankara Turkey  
{kara.gizem,oguz}@metu.edu.tr

## Abstract

A number of arithmetization-oriented ciphers emerge for use in advanced cryptographic protocols such as secure multi-party computation (MPC), fully homomorphic encryption (FHE) and zero-knowledge proofs (ZK) in recent years. The standard block ciphers like AES and the hash functions SHA2/SHA3 are proved to be efficient in software and hardware but not optimal to use in this field, for this reason, new kind of cryptographic primitives were proposed recently. However, unlike traditional ones, there is no standard approach to design and analyze such block ciphers and the hash functions, therefore their security analysis needs to be done carefully. In 2018, StarkWare launched a public STARK-Friendly Hash (SFH) Challenge to select an efficient and secure hash function to be used within ZK-STARKs, transparent and post-quantum secure proof systems. The block cipher JARVIS is one of the first ciphers designed for STARK applications but, shortly after its publication, the cipher has been shown vulnerable to Gröbner basis attack. This paper aims to describe a Gröbner basis attack on new block ciphers, MiMC, GMiMC<sub>erf</sub> (SFH candidates) and the variants of JARVIS. We present the complexity of Gröbner basis attack on JARVIS-like ciphers. Then we give results from our experiments for the attack on reduced-round MiMC and a structure we found in the Gröbner basis attack for GMiMC<sub>erf</sub>.

## 1 Introduction

Block ciphers are the fundamental tools of modern cryptography. They are pseudo-random permutations operating on fixed-size blocks and used to secure different types of data. Their design and security considerations are well understood in the literature. However, the design of symmetric-key primitives for use in advanced cryptographic protocols such as secure multi-party computation (MPC), fully homomorphic encryption (FHE) or new proof systems like SNARKs, STARKs, Bulletproofs studied in the past few years because of the recent progress in practical applications of this field.

Secure multi-party computation (MPC) is a cryptographic protocol that enables the parties to evaluate output of a function securely without knowing anything about their private inputs. In MPC systems, the arithmetic operations on secret sharing values are often performed over a finite field with large prime characteristic  $\mathbb{F}_p$ . The problem of using traditional block ciphers like AES in MPC setting is the hardness of representing such block ciphers using arithmetic over finite fields. Their design strategy aims to provide mostly good performance in hardware or software implementations. Therefore, we have a new area of designing efficient symmetric primitives for use in MPC or ZK-proof systems. We refer reader to [6] that gives detailed information for the design of such primitives.

One of the design of pseudo-random functions (PRFs) for MPC applications is given in [5], where the designers propose a blockcipher LowMC with low multiplicative depth and low multiplicative complexity which operates over  $\text{GF}(2)$ . After that, several bit-oriented primitives have appeared like Kreyvium [14] or FLIP [26] considering the same design strategy as LowMC. Because most of the advanced cryptographic protocols support operations over large prime fields, MiMC family [2], including a block cipher and a cryptographic hash function, were presented by offering multiplications over large fields  $\text{GF}(2^n)$  and  $\text{GF}(p)$ . The block cipher MiMC was designed mainly for SNARK applications like Zerocash [28], but it is also competitive for use in STARKs and MPC applications. The

designers of MiMC improved cipher to Generalized MiMC (GMiMC) [4] in order to provide efficient performance also in the area of PQ-secure signature schemes, where MiMC was not so competitive in this area. MARVELlous family [7], the block cipher JARVIS and the hash function FRIDAY, are the first designs to propose efficiency in STARK applications, however it has been shown that these designs do not provide adequate security as claimed [3]. The paper [6] calls these new primitives as arithmetization-oriented algorithms.

The design strategies of standard block ciphers like AES (Advanced Encryption Standard) [27] or 3DES (Triple DES, Data Encryption Standard) [25] and the arithmetization-oriented ciphers are different. Therefore, their security analysis and the corresponding attack techniques are also different. Statistical attacks such as differential and linear cryptanalysis are widely used for the cryptanalysis of block ciphers. Algebraic attacks are a different type of cryptanalysis which aims to exploit algebraic structure of the cipher. This kind of attacks tries to represent the cipher as a system of polynomial equations and then solves to recover the key using a suitable method like SAT solvers, Gröbner basis methods, Mixed-Integer Linear Programming (MILP) Solvers or algebraic higher-order differentials. A common belief is that the statistical attacks are generally faster than the algebraic attacks because of the high complexity of algebraic attacks. Not a single proper block cipher has been broken using pure algebraic techniques faster than with other techniques [3]. Algebraic techniques were mostly considered against some public-key schemes and stream ciphers because they were proved to be successful against them. However, the target applications like MPC/FHE/ZK-STARKs are algebraic systems, and therefore algebraic attacks gain attention again from the cryptographers.

The design of arithmetization-oriented algorithms which are both efficient and secure still in progress. Two design strategies, MARVELlous [6] and HADES [22, 23], provide a generic way for the demand in design space relative to these target applications. After JARVIS was shown to be insecure againsts Gröbner basis attack, the designers of MARVELlous together with Ben-Sasson proposed a Marvellous family design strategy which includes two ciphers *Vision* for binary fields and *Rescue* for prime fields. These ciphers were candidates for STARK-Friendly Hash (SFH) Challenge [1]. The HADES design strategy proposed by Grassi et al. [23] and the HadesMiMC family of algorithms, the hash functions *Starkad* and *Poseidon* [22], were also candidate in SFH challenge. In this public competition, the security of four families of algorithms – MiMC, GMiMC, HadesMiMC and MARVELlous, was analyzed by the cryptanalysts. At the end of the selection process of STARK-Friendly hash function, the hash function *Rescue* is recommended by Ben-Sasson et al. [10].

## 1.1 Our motivation

The new arithmetic-oriented primitives designed for applications of advanced cryptographic protocols may be vulnerable to algebraic attacks, particularly Gröbner basis attacks. The security of these ciphers was examined against various algebraic attacks but not focus directly on Gröbner basis attacks. However, as said in [6], it is the common question for these new designs:

*“Consequently, the question of security against Gröbner basis attacks seems to be the crucial concern raised by arithmetization-oriented ciphers, and no such proposal is complete without explicitly addressing it”.*

The success of the attack strategy on JARVIS and FRIDAY motivated us to study Gröbner basis attack against variants of JARVIS and the other proposed ciphers, MiMC and GMiMC<sub>erf</sub>.

## 1.2 Structure of the paper

Sections 2 and 3 will present mathematical background for Gröbner bases and Gröbner basis attacks. In Section 4, we will briefly describe the block cipher JARVIS and in Section 4.2, we will mention successful Gröbner basis attack on JARVIS by Albrecht et al. [3], then we generalize the attack strategy on JARVIS-like ciphers. We will give a formula to estimate the complexity of the attack and using this formula we will show JARVIS with degree 8 polynomials is still vulnerable to Gröbner basis attack in Section 5. Furthermore, we will compare S-boxes of JARVIS and AES in Section 5.1 and estimate the complexity of the attack on JARVIS with AES S-box in Section 5.2. If we replace

the  $S$ -box of JARVIS with AES  $S$ -box, we see that the complexity of the attack with 8-bits input is around  $\approx 97$  bits for 10 rounds.

Section 6 will express our other target cipher MiMC and present results from our experiments for the Gröbner basis attack on reduced round MiMC. We see that MiMC with 82 rounds is resistant against Gröbner basis attack. Then, Section 7 will give a brief description of the block cipher GMiMC<sub>erf</sub> and describe our results for the Gröbner basis attack against the primitive. We will say that GMiMC<sub>erf</sub> is secure against Gröbner basis attack not because of the high complexity of basis computation but for a different reason. Section 8 will conclude our results in this paper and end up with discussion and future work section.

Note that all the experiments in this paper are performed in Sage 9.0. "Sage: Software for Algebra and Geometry Experimentation" is a free and open source computational algebra system [30]. The full source codes of the attacks given in this paper are provided in:

[https://github.com/gizemmkara/masters\\_thesis](https://github.com/gizemmkara/masters_thesis)

## 2 Mathematical background

In this section, we will give some main theorems and definitions to understand the concept of Gröbner basis and Gröbner basis attacks. For more detailed information, we refer to [16].

### 2.1 Monomial orders and monomial ideals

**Definition 2.1.** A *multivariate polynomial*  $f$  in  $k$  variables  $x_0, \dots, x_{k-1}$  with coefficients  $c_0, \dots, c_{k-1}$  over a field  $\mathbb{F}$  can be expressed as

$$f = \sum_{i \in \mathbb{Z}_{\geq 0}^k} c_i x^i$$

where  $x^i = x_0^{i_0} \cdot x_1^{i_1} \cdots x_{k-1}^{i_{k-1}}$  is a *monomial* with total degree  $i_0 + i_1 + \cdots + i_{k-1}$ . The degree of  $f$  is defined as the maximum value of the total degrees of the monomials.

For multivariate polynomials, the order of terms *monomial ordering* is not just important to write and read terms but also to decide the leading term of the polynomial and how to store and operate the polynomials in a computer since they affect the complexity. For example, while using division algorithm on **univariate polynomials**, a polynomial depends only one variable, over  $\mathbb{F}[x]$ , we write terms in decreasing order on degrees of the terms,  $\cdots > x^{t+2} > x^{t+1} > x^t > \cdots > x^2 > x^1 > x > 1$ . Also, in row-reduction algorithm for the matrices, we deal with the linear equations in  $k$  variables  $x_1, \dots, x_k$  in decreasing order, written as  $x_1 > \cdots > x_k$ . Now, we may define ordering in monomials.

**Definition 2.2 (Monomial ordering).** A *monomial ordering* on  $\mathbb{F}[x_1, \dots, x_k]$  is a relation  $>$  on  $\mathbb{Z}_{\geq 0}^k$  (i.e., exponents of monomials) or a relation on monomials  $x^a, a \in \mathbb{Z}_{\geq 0}^k$ , such that:

1. The relation  $>$  is a total ordering on  $\mathbb{Z}_{\geq 0}^k$ . That means for any pairs of  $x^a$  and  $x^b$  exactly one of the three statements,  $x^a > x^b, x^a = x^b, x^a < x^b$  should be satisfied.
2. If  $a > b$  and  $c \in \mathbb{Z}_{\geq 0}^k$ , then  $a + c > b + c$ .
3. The relation  $>$  has well-ordering which means every non-empty subset has a smallest element under  $>$ .

For example, the numerical order  $t + 1 > t > \cdots > 2 > 1 > 0$  on  $\mathbb{N}$ , satisfies the above conditions, hence the degree ordering on monomials over  $\mathbb{F}[x]$  is a monomial ordering. In computational algebra, the following three term orderings are mostly used

**Definition 2.3 (Lexicographic Order).** We say  $a >_{lex} b$  if the left most non-zero entry in  $a - b \in \mathbb{Z}^k$  is positive.

**Definition 2.4 (Graded Lexicographic Order).** We say  $a >_{grlex} b$  if the total degrees  $|a| > |b|$  or if  $|a| = |b|$  and  $a >_{lex} b$ .

**Definition 2.5 (Graded Reverse Lexicographic Order).** We say  $a >_{\text{grevlex}} b$  if the total degrees  $|a| > |b|$  or if  $|a| = |b|$  and the rightmost non-zero entry of vector difference  $a - b \in \mathbb{Z}^k$  is negative.

Before giving the definition of Gröbner basis, we first define the *monomial ideals*.

**Definition 2.6.** An ideal  $I \subseteq \mathbb{F}[x_1, \dots, x_k]$  is called a **monomial ideal** if it can be generated by monomials.

For example,  $I = \langle x^2y, xy^3 \rangle \subseteq \mathbb{F}[x, y]$  is a monomial ideal generated by the monomials  $x^2y$  and  $xy^3$ .

**Theorem 2.1 (Dickson's Lemma).** Every monomial ideal  $I \subseteq \mathbb{F}[x_1, \dots, x_k]$  is finitely generated, i.e.  $I$  has a finite basis.

*Proof.* See [16, Chapter 2, Section 4, Theorem 5]. □

**Definition 2.7.** Consider a non-zero ideal  $I \subseteq \mathbb{F}[x_1, \dots, x_k]$  and fix a monomial ordering. The set  $LT(I)$  is the set of leading terms of the polynomials in  $I$

$$LT(I) = \{LT(f) \mid f \in I\}.$$

The ideal generated by the elements of  $LT(I)$  is denoted by  $\langle LT(I) \rangle$ .

Note that for the ideal  $I$ , say  $I = \langle g_1, \dots, g_t \rangle$ , the ideals  $\langle LT(g_1), \dots, LT(g_t) \rangle$  and  $\langle LT(I) \rangle$  may be different.

## 3 Gröbner bases and Gröbner basis attack

### 3.1 Gröbner bases

The concept of Gröbner basis and the algorithm to construct it introduced by Buchberger [12] in 1965. Gröbner bases have many applications in computational algebra such as, ideal membership problem, ideal description problem and the problem of solving polynomial equations. We will mainly focus on the solving polynomial equations.

**Definition 3.1 (Polynomial Systems Solving (PoSSo) Problem).** Given a set of polynomial equations  $P = \{f_1, f_2, \dots, f_m\} \in \mathbb{F}[x_1, \dots, x_k]$ . Find -if any- common solutions of the polynomial system such that:

$$f_1(x_1, \dots, x_k) = f_2(x_1, \dots, x_k) = \dots = f_m(x_1, \dots, x_k) = 0.$$

When the number of variables is high, this problem is hard to solve.

**Definition 3.2 (Gröbner Basis).** Fix a monomial ordering on  $\mathbb{F}[x_1, \dots, x_k]$  and an ideal  $I$ . A finite subset  $G = \{g_1, \dots, g_t\}$  of an ideal  $I$  is a **Gröbner basis** of  $I$  if the ideal generated by the leading term of every element of  $I$  is generated by the leading terms of the  $g_i$ , i.e.

$$\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_t) \rangle$$

or informally, if any element of  $I$  is divisible by one of  $LT(g_i)$ .

**Theorem 3.1.** Every ideal  $I \subseteq \mathbb{F}[x_1, \dots, x_k]$  has a Gröbner basis  $G = \{g_1, \dots, g_t\}$  for a fixed monomial order. Furthermore, any Gröbner basis for the ideal  $I$  is a basis of  $I$ .

*Proof.* See [16, Chapter 2, Section 5, Corollary 6]. □

Buchberger formulated an algorithm, known as *Burchberger's algorithm*, for computing Gröbner basis. This algorithm comes from the idea behind Buchberger's criterion and used to determine if a given basis for an ideal is Gröbner or not.

**Definition 3.3** (*S-polynomial*). Let  $f, g \in \mathbb{F}[x_1, \dots, x_k]$  be two non zero polynomials. The *S-polynomial* of  $f$  and  $g$  is defined as the combination

$$S(f, g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g,$$

where  $x^\gamma$  is the least common multiple of the leading monomials of  $f$  and  $g$ , written as  $x^\gamma = \text{lcm}(LM(f), LM(g))$ .

**Theorem 3.2** (**Buchberger's Criterion**). Let  $I$  be an ideal. A basis  $G = \{g_1, \dots, g_t\}$  is a Gröbner basis of  $I$  if and only if for any pairs  $i \neq j$ , the remainder on the division of  $S(g_i, g_j)$  by  $G$  listed in some order is zero written as

$$\overline{S(g_i, g_j)}^G = 0.$$

*Proof.* See [16, Chapter 2, Section 7, Theorem 2]. □

This criterion leads the Buchberger's algorithm to construct a Gröbner basis for a given ideal, see Algorithm 1.

---

**Algorithm 1** Buchberger's Algorithm

---

**Input:**  $F = (f_1, \dots, f_t)$

$\triangleright F \subseteq \mathbb{F}[x_1, \dots, x_k]$

**Output:** A gröbner basis  $G = (g_1, \dots, g_s)$  for the ideal  $I = \langle F \rangle$

$G = F$

$G' = \text{set}()$

**while**  $G' \neq G$  **do**

$G' = G$

**for** each pair  $\{p, q\}, p \neq q$  in  $G'$  **do**

$r := \overline{S(p, q)}^{G'}$

**if**  $r \neq 0$  **then**

$G.\text{add}(r)$

**return**  $G$

---

This algorithm terminates as  $G' = G$  in finitely many steps due to the Ascending Chain Condition which stabilizes the ascending chain of ideals. The runtime of the algorithm is affected by the choice of monomial ordering, the order of which  $p, q$  are selected and the unnecessary reductions to 0.

One may view Buchberger's algorithm as a generalization of Euclidean algorithm for computing greatest common divisor of polynomials and Gaussian elimination to solve linear equations. There are other algorithms such as  $F_4$  and  $F_5$  to compute Gröbner basis effectively using some linear algebra techniques [18, 19].

### 3.2 Gröbner basis attack

Algebraic attack is a type of cryptographic attack that exploits the algebraic structure of the cipher to recover the secret by solving multivariate polynomial system of equations which consists of key, plaintext and ciphertext bits. Gröbner basis attack is an example for algebraic attacks. The first step of the attack is to represent the cipher as a system of polynomial equations. Then, the attacker computes the Gröbner basis for the ideal generated by corresponding equations and finally solve the system for unknown variables. The phases of Gröbner basis attack are detailed below.

1. Set up a multivariate polynomial system of equations that describes the cipher. Note that one can always find a polynomial representation of a function over a finite field, but the crucial point is to find the simplest description due to the complexity of algebraic attacks.
2. Compute a Gröbner basis for the polynomial system, which forms an ideal, in degree reverse lexicographic order (mostly preferred for performance reasons) using Gröbner basis algorithms such as Buchberger's,  $F_4$ ,  $F_5$  or Macaulay matrices. In general, this is the most expensive step.

3. Change the ordering in Gröbner basis from degrevlex order to the lex order via Gröbner basis conversion algorithms like FGLM [20], which works only for zero dimensional ideals, or Gröbner Walk algorithm [15]. Lex ordered coefficient matrix of Gröbner basis is in triangular shape and the last row gives the solution for univariate equation, that's why the lex order is used to eliminate the variables.
4. Factorize the last element in lexicographic Gröbner basis (lex ordered Gröbner basis guarantees there is at least one univariate polynomial) using polynomial factorization algorithms such as Berlekamp algorithm [21]. Finally, compute the full solution of the system by back substituting roots of the univariate polynomial.

A general algorithm for key recovery using Gröbner bases [13] is provided below:

---

**Algorithm 2** Gröbner basis attack [13]

---

1. Set up a polynomial system of equations  $P = \{p_i = 0\}$  for the cipher in question which consists of both cipher and key schedule equations.
2. Request a plaintext/ciphertext pair  $((P_0, \dots, P_{t-1}), (C_0, \dots, C_{t-1}))$ . This gives rise to the following additional system of linear equations  $G : \{g_i = 0\}$ :

$$\begin{array}{rcc}
 x_0^{(0)} + P_0 = 0 & \dots & x_0^{(r)} + C_0 = 0 \\
 x_1^{(0)} + P_1 = 0 & \dots & x_1^{(r)} + C_1 = 0 \\
 \vdots & & \vdots \\
 x_{t-1}^{(0)} + P_{t-1} = 0 & \dots & x_{t-1}^{(r)} + C_{t-1} = 0
 \end{array}$$

Let  $I$  be the ideal generated by the set of polynomials  $J = (\bigcup_i \{p_i\}) \cup (\bigcup_i \{g_i\})$ . We call this ideal as the key recovery ideal.

3. Compute a degree reverse lexicographic ordered Gröbner basis  $G'_{degrevlex}$  of  $I$ . For ciphers using a multiplicative inverse as  $S$ -box function, the system may be inconsistent, resulting in  $G'_{degrevlex} = 1$ .
  4. If  $G'_{degrevlex} = 1$  go to step 2, otherwise continue.
  5. Use a Gröbner basis order conversion algorithm to obtain a lexicographical Gröbner basis  $G_{lex}$  from  $G'_{degrevlex}$ . The variable ordering should be such that the key variables of the first round are the least elements.
  6. Compute the variety  $Z$  of  $I$  using the Gröbner basis  $G_{lex}$ .
  7. Request another plaintext/ciphertext pair  $(P, C)$ .
  8. Try all elements  $k \in Z$  as key candidates to encrypt  $P$ . If  $k$  does not encrypt  $P$  to  $C$ , remove  $k$  from  $Z$ , otherwise retain.
  9. If  $Z$  contains more than one element, go to step 7.
  10. Terminate
- 

Note that the above algorithm is very general, many changes are possible such as computing Gröbner basis with a different monomial ordering rather than *degrevlex* or *lex*. Variety of an ideal is the set of all common solutions of the elements in ideal. Observe that in Step 6, to compute variety  $Z$  of  $I$ , one needs to factor univariate polynomials and substitute the roots into other equations to check if that root is a solution for whole system. In the following sections, we will discuss the complexity of each step.

### 3.3 Complexity of Gröbner basis computation

For a generic system of  $m$  equations in  $k$  variables

$$f_1(x_1, \dots, x_k) = \dots = f_m(x_1, \dots, x_k) = 0$$

the complexity of computing Gröbner basis is

$$\mathcal{O}\left(\binom{k + d_{reg}}{d_{reg}}^\omega\right), \quad (1)$$

operations over the field  $\mathbb{F}$ , where  $2 \leq \omega < 3$  is the exponent for the complexity of matrix multiplication and  $d_{reg}$  is the degree of regularity [9, 11]. The degree of regularity is informally the highest degree reached during Gröbner basis computation and therefore is the key concept to analyze the complexity of polynomial solving algorithms. There is a common belief that this degree determines when the solving algorithm will terminate, that's why it is used to parametrize the complexity [29]. In general, computing degree of regularity for the overdetermined systems ( $m > k$ ) is a hard problem and still an active research area [3]. Notice that the complexity does not contain the number  $m$  of equations explicitly but, the degree of regularity depends on the number of equations.

For the *regular systems*, where the number of equations is equal to the number of variables,  $m = k$ , one can calculate this degree by using the formula:

$$d_{reg} = 1 + \sum_{i=1}^m (d_i - 1), \quad (2)$$

where  $d_i$  is the degree of  $f_i$ , see [8]. In general, for the semi-regular (random) systems with the number of equations greater than the number of variables, over-determined systems ( $m > k$ ), the degree of regularity can be computed using Hilbert series expansion of the ideal generated by the polynomials  $f_1, \dots, f_m$ . In this case,  $d_{reg}$  is defined [8] as the first non-positive coefficient in

$$H(t) = \frac{1}{(1-t)^k} \times \prod_{i=1}^m (1-t)^{d_i}.$$

### 3.4 Complexity of change of term ordering

The input of the FGLM algorithm is the Gröbner basis (degrevlex ordered in our case) of a zero-dimensional ideal  $I$ , having finitely many solutions, and it returns the Gröbner basis with respect to the lex order. The complexity of the FGLM algorithm [20] is

$$\mathcal{O}(k \cdot D^3),$$

where  $k$  is the number of variables and  $D$  is the degree of the ideal  $I$  which is the dimension of the vector space of the quotient ring  $\mathbb{F}[x_1, \dots, x_k]/I$ . In general, it is known that FGLM algorithm is faster than the Gröbner Walk algorithm [13].

### 3.5 Complexity of factorization

Finally, we need to factorize the last univariate polynomial and find its roots in lex ordered Gröbner basis we discovered. A polynomial of degree  $d$  over a finite field  $\mathbb{F}_{2^n}$  can be factorized using the improved version of Berlekamp algorithm [21]. The complexity of the algorithm is

$$\mathcal{O}(d^3 n^2 + dn^3).$$

In the following sections, we will describe three block ciphers, JARVIS, MiMC and GMiMC<sub>erf</sub>. We will present Gröbner basis attacks for each cipher, then we will analyze the complexity of the attack for variants of JARVIS. After that, we will show our experimental results for key recovery attack on MiMC and our attack strategy on GMiMC<sub>erf</sub>.

## 4 The block cipher JARVIS

Ashur and Dhooghe [7] proposed JARVIS as a STARK-friendly block cipher in 2018. Its design is inspired from the design of the AES in order to gain resistance against differential and linear cryptanalysis. They instantiate JARVIS to offer 128, 160, 192 and 256-bit security levels.

### 4.1 Description of JARVIS

JARVIS is a family of SPN block ciphers designed for STARK-applications. It uses *wide-trail strategy* similar to AES, which ensures the security againsts differential and linear cryptanalysis. JARVIS works on an entire  $n$ -bit state and an  $n$ -bit key over the finite field  $\mathbb{F}_{2^n}$ . The non-linear layer in JARVIS uses a single  $S$ -box over  $F_{2^n}$  and defined as a multiplicative inverse function

$$S : \mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n} \\ x \longrightarrow x^{2^n-2},$$

or in rational form

$$S(x) := \begin{cases} \frac{1}{x}, & x \neq 0 \\ 0, & x = 0. \end{cases}$$

The linear part in JARVIS is defined as the composition of two affine polynomials. These affine polynomials are created by adding a constant value to a linearized polynomial. Remember that an  $\mathbb{F}_2$  linearized permutation polynomial is defined as

$$L(x) = \sum_{i=0}^{n-1} l_i x^{2^i} \in \mathbb{F}_{2^n}[x].$$

The affine polynomial obtained from  $L(x)$  is

$$A(x) = l_{-1} + \sum_{i=0}^{n-1} l_i x^{2^i} \in \mathbb{F}_{2^n}[x].$$

In JARVIS, two monic affine polynomials  $B$  and  $C$  of degree 4 are chosen in the form

$$B(x) = x^4 + b_2 x^2 + b_1 x + b_0 \text{ and } C(x) = x^4 + c_2 x^2 + c_1 x + c_0,$$

so that the linear layer  $A(x)$  is splitted as

$$A(x) = C \circ B^{-1}(x), \tag{3}$$

where  $B^{-1}$  is the compositional inverse satisfying  $B^{-1}(B(x)) = x$ . Note that the compositional inverse of  $B$  is still an affine polynomial but it has much higher degree. The round function of JARVIS is depicted below in Figure 1.

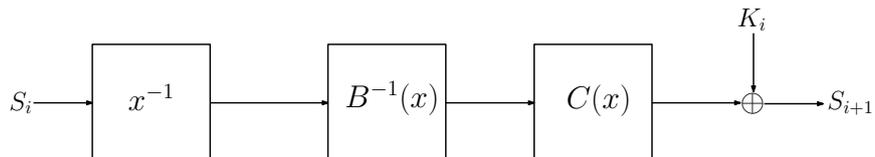


Figure 1: One round of the JARVIS block cipher

**Key Schedule:** The key schedule in JARVIS is similar to the round function. It uses the same  $S$ -box as in the round function whereas the affine part is omitted. The first key  $k_0$  is the master key and the round keys are generated by adding a round constant  $c_i$  to the output of the  $S$ -box in the key schedule. One round of the key schedule is shown in Figure 2.

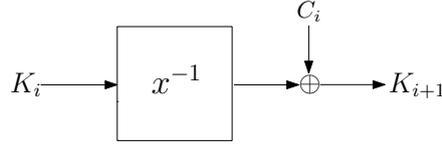


Figure 2: One round of the key schedule used in JARVIS block cipher

The designers of JARVIS propose the security levels for four different block sizes and different number of rounds  $r = 10, 11, 12, 14$  for the polynomials  $B$  and  $C$  with fixed round constants, see in Table 1.

Table 1: Instances of JARVIS [7]

Instances	$n$	number of rounds $r$
JARVIS-128	128	10
JARVIS-160	160	11
JARVIS-192	192	12
JARVIS-256	256	14

However, it has been shown that the specified number of rounds for JARVIS does not provide above security levels as claimed. In the following section, we will give successful Gröbner basis attack on JARVIS given in Albrecht et al. [3].

## 4.2 Gröbner basis attack on JARVIS

Albrecht et al. [3] showed that the JARVIS is not secure as claimed since the certain characteristics of JARVIS makes the cipher vulnerable to Gröbner basis attack. The one is that the  $S$ -box of JARVIS,  $S(x) = x^{2^n-2}$ , can be written as a degree-2 polynomial

$$S(x) = x^{-1} = y,$$

where  $x \cdot y = 1$  for any non zero element  $x \in \mathbf{F}_{2^n}$ . For a sufficiently large  $n$ , it is claimed that  $x$  is not equal to zero with a high probability. The other is that whereas the affine polynomial  $A$  has high degree, it is a decomposition of two low degree polynomials, see Equation (3), and setting up equations by avoiding the inverse computation of high degree  $B^{-1}$  makes the system vulnerable to the attack.

### 4.2.1 Gröbner basis attack on reduced round JARVIS

In the original proposal, the authors of [3] first present the Gröbner basis attack approach on reduced round JARVIS and then they improve the attack to apply the full round of JARVIS.

They describe the primitive by introducing an intermediate variable  $x_i$  for the  $i$ -th round where  $1 \leq i \leq r$ , see in Figure 3.

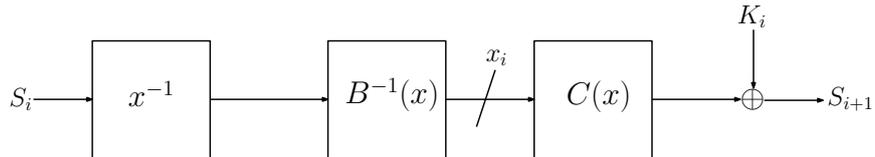


Figure 3: Introducing new intermediate variable  $x_i$  for the one round of the JARVIS block cipher

The two consecutive rounds of JARVIS is expressed by the equation

$$(C(x_i) + k_i) \cdot B(x_{i+1}) = 1 \quad (4)$$

for  $1 \leq i \leq r - 1$ , where  $r$  is the number of rounds and the equations for the plaintext  $p$  and the ciphertext  $c$  described as

$$(p + k_0) \cdot B(x_1) = 1, \quad (5)$$

$$C(x_r) + k_r = c. \quad (6)$$

The two consecutive round keys in JARVIS are defined by the equation

$$k_{i+1} = \frac{1}{k_i} + c_i$$

which can be written as

$$(k_{i+1} + c_i) \cdot k_i = 1, \quad 0 \leq i \leq r - 1. \quad (7)$$

Since  $B$  and  $C$  are both degree 4 polynomials, the equations in (4), (5), (6), (7) respectively result in:

- $(r - 1)$  equations of degree 8 with  $(2 \cdot r - 1)$  variables,  $x_1, \dots, x_r$  and  $k_1, \dots, k_{r-1}$ ,
- one equation of degree 5 in two variables  $k_0$  and  $x_1$ ,
- one degree-4 equation with two variables  $x_r$  and  $k_r$ ,
- $r$  equations having degree 2.

Overall, the above polynomial system of equations that describes the primitive has  $2r + 1$  equations in  $2r + 1$  variables  $x_1, \dots, x_r$  and  $k_0, \dots, k_r$ . Since the number of equations and the number of variables are equal and assuming system behaves like *regular sequences*, one may calculate the degree of regularity using (2) and estimate the complexity of computing Gröbner basis (1). Even for the number of rounds  $r = 6$ , this complexity is almost 120 bits and 85 bits for  $\omega = 2.8$  and  $\omega = 2$ , respectively. However, it is shown in [3] that these theoretical estimations are too pessimistic. In practice, the authors compute the Gröbner basis for the above polynomial system and apply the attack to full-round of JARVIS by improving the attack.

#### 4.2.2 Improved attack: A more efficient description of JARVIS

The authors of [3] improved the attack described in previous section by reducing the number of equations and the number of variables. In order to reduce the number of variables for round equations, they fix the intermediate variables  $x_i$  for the even number of rounds and express them using previous  $x_{i-1}$  and next following intermediate variables  $x_{i+1}$ . For each intermediate variable  $x_i$

$$B(x_i) = \frac{1}{C(x_{i-1}) + k_{i-1}} \text{ and } C(x_i) = \frac{1}{B(x_{i+1})} + k_i, \quad (8)$$

where  $2 \leq i \leq r - 1$ . In order to skip intermediate variables  $x_i$ , they define monic degree 4 affine polynomials  $D$  and  $E$  of the form

$$D(x) = x^4 + d_2x^2 + d_1x + d_0 \text{ and } E(x) = x^4 + e_2x^2 + e_1x + e_0$$

satisfying the equation

$$D(B) = E(C). \quad (9)$$

It has been shown that Equation (9) can be solved by equalizing the coefficients of polynomials, see [3]. After finding such suitable polynomials  $D$  and  $E$ , they apply these polynomials to  $B$  and  $C$  as given in (8) which yields the following polynomial system

$$D\left(\frac{1}{C(x_{i-1}) + k_{i-1}}\right) = E\left(\frac{1}{B(x_{i+1})} + k_i\right) \text{ for } 2 \leq i \leq r - 1, \quad (10)$$

$$D\left(\frac{1}{p + k_0}\right) = E\left(\frac{1}{B(x_2)} + k_1\right), \quad (11)$$

$$C(x_r) + k_r = c. \quad (12)$$

The degrees of each equations are as follows:

- For the intermediate round equations in (10), the left hand side is of degree 16, since  $D$  and  $C$  are degree 4 polynomials, and the right hand side is of degree 20, after equalizing denominators degree 36 polynomials obtained.
- The degree of Equation (11) is 24, degree 4 from left and degree 20 from the right hand side.
- Equation (12) is of degree 4.

Assuming the number of rounds  $r$  to be even, above polynomial system gives:

- $\frac{r}{2} - 1$  equations of degree 36
- one equation of degree 24
- one equation of degree 4

In total, above system expressed in  $\frac{r}{2} + 1$  equations with variables  $x_2, x_4, \dots, x_r$  and  $k_0, \dots, k_r$ . They also reduce the number of key variables by connecting each round key to the master key  $k_0$

$$k_{i+1} = \frac{\alpha_i \cdot k_0 + \beta_i}{\gamma_i \cdot k_0 + \delta_i}, \quad (13)$$

where  $\alpha_i, \beta_i, \gamma_i$  and  $\delta_i$  are constants and can be found explicitly by solving the recursive relation. This final improvement results in:

- $\frac{r}{2} - 1$  equations of degree 40
- one equation of degree 24
- one equation of degree 5

Overall, the improved attack strategy on JARVIS halves the number of equations and variables needed to describe the cipher. Hence, it yields a polynomial system with  $\frac{r}{2} + 1$  equations in  $\frac{r}{2} + 1$  variables  $x_2, \dots, x_r$  and  $k_0$ .

Table 2: Experimental results of the improved attack on JARVIS using Sage [3]

$r$	$k$	$d_{reg}$	$2 \log_2 \binom{k+d_{reg}}{d_{reg}}$	$d$	$2 \log_2 \binom{k+d}{d}$	$d_u$	Time
3	2	47	20	26	17	256	0.3s
4	3	67	31	40	27	1280	9.4s
5	3	86	34	40	27	6144	891.4s
6	4	106	45	41	34	28672	99989.0s

In Table 2,  $r$  denotes the number of rounds,  $k$  is the number of variables and  $d_{reg}$  is the degree of regularity calculated assuming the system behaves like regular (2). The estimated complexity in bits is  $2 \log_2 \binom{k+d_{reg}}{d_{reg}}$ , for  $\omega = 2$ ,  $d$  is the highest degree reached during the basis computation and the expected security based on the experiment in [3] is  $2 \log_2 \binom{k+d}{d}$ . The degree of the univariate polynomial obtained in the last step to solve the system is denoted by  $d_u$ .

## 5 Complexity estimates of Gröbner basis computation for the variants of JARVIS

The improved attack given in [3], as described in Section 4.2.2, motivated us to formulate the attack for the block ciphers having affine polynomial like JARVIS. Since the affine layer of JARVIS is the composition of two low degree polynomials  $B$  and  $C$ , we mentioned that one can find two low degree polynomials  $D$  and  $E$  which makes cipher vulnerable to Gröbner basis attack. The question is what if one replaces  $B$  and  $C$  with higher degree polynomials. In order to determine whether the higher degree polynomials choice makes the cipher resistant against Gröbner basis attacks or not, in this section, we try to generalize the complexity of the improved attack on JARVIS. We show that JARVIS with degree 8 affine polynomials is still vulnerable to Gröbner basis attack.

**Proposition 5.1.** *Let  $B$  and  $C$  be arbitrary affine polynomials used in JARVIS. Let  $D$  and  $E$  be the monic affine polynomials satisfying the equation  $D(B) = E(C)$ . Let  $d_b, d_c, d_d, d_e$  be the degrees of  $B, C, D, E$  respectively. Then the complexity of computing Gröbner basis with the improved attack on  $r$  rounds JARVIS in bits is*

$$\omega \log_2 \left( \begin{array}{c} ((\frac{r}{2} - 1)(d_d(d_c + 1) + d_e(d_b + 1) - 1) + (d_d + d_e(d_b + 1) + d_c)) + \frac{r}{2} + 1 \\ (\frac{r}{2} - 1)(d_d(d_c + 1) + d_e(d_b + 1) - 1) + d_d + d_e(d_b + 1) + d_c \end{array} \right) \quad (14)$$

where  $(\frac{r}{2} - 1)(d_d(d_c + 1) + d_e(d_b + 1) - 1) + (d_d + d_e(d_b + 1) + d_c)$  is the degree of regularity.

*Proof.* Assume that the degrees of the monic affine polynomials  $B, C, D, E$  be  $d_b, d_c, d_d$  and  $d_e$  respectively. The improved attack strategy yields the below equations

- $\frac{r}{2} - 1$  equations of degree  $d_d(d_c + 1) + d_e(d_b + 1)$  (from (10)),
- one equation of degree  $d_d + d_e(d_b + 1)$  (from (11)),
- one equation of degree  $d_c + 1$  (from (12)).

We know that the complexity of the Gröbner basis computation in bits is  $\omega \log_2 \binom{k+d_{reg}}{d_{reg}}$ , see (1). Since the number of equations and the number of variables are the same ( $m = k = \frac{r}{2} + 1$ ), assuming the system behaves like a regular system, we can estimate the degree of regularity using the closed formula (2). The result follows from putting the values we obtained from above system.  $\square$

We estimate the complexity of computing Gröbner basis for the affine polynomials  $B$  and  $C$  of degree 8 and corresponding polynomials  $D$  and  $E$  of degrees 2, 4 and 8 using (14), without regarding if there is a solution for the system ( $D(B) = E(C)$ ) or not. The results can be seen below in Table 3. In the table, the number of rounds, the number of variables, the expected security level and the degree of regularity are denoted by  $r, k, C_{GB}$  and  $d_{reg}$ , respectively. The security levels are estimated by setting  $\omega = 2.8$  and for  $\omega = 2$  in parenthesis.

Table 3: Complexity estimates for degree-8 polynomials  $B$  and  $C$

		$D, E$ are degree-2		$D, E$ are degree-4		$D, E$ are degree-8	
$r$	$k$	$d_{reg}$	$C_{GB}$	$d_{reg}$	$C_{GB}$	$d_{reg}$	$C_{GB}$
6	4	98	62 (44)	190	72 (52)	316	80 (57)
8	5	133	80 (57)	261	93 (67)	430	103 (74)
10	6	168	98 (70)	332	114 (82)	544	126 (90)
12	7	203	116 (83)	403	135 (97)	658	149(107)
14	8	238	135 (96)	474	157 (112)	772	172 (123)

**Remark 5.2.** *The complexity of the improved attack on JARVIS increases when the degrees of the polynomials increase. For example, when the number of rounds  $r = 6$  estimated complexity is  $\approx 45$  bits (see Table 2) for the polynomials  $B, C, D, E$  are all degree 4 (in original JARVIS), and complexity is  $\approx 57$  bits (see Table 3) for degree-8 polynomials.*

## 5.1 Comparison with the $S$ -box of the AES and decomposing AES $S$ -box

The non-linear part in JARVIS applies the same idea with the  $S$ -box of the AES,  $S\text{-box}_{AES}(z)$ . In this section, we try to decompose  $S\text{-box}_{AES}(z)$  for different degree affine polynomials. We provide some lemmas to decide appropriate degrees of the decomposition polynomials of AES  $S$ -box.

We know that AES  $S$ -box is the composition of an affine function  $A_{AES}(z)$  over  $\mathbb{F}_2$  and the multiplicative inverse of the input over  $\mathbb{F}_{2^8}$ . In particular

$$S\text{-box}_{AES}(z) = A_{AES}(z^{254}).$$

The multiplicative inverse is defined by the function  $F$  over  $\mathbb{F}_{2^8}$

$$\begin{aligned} F : \mathbb{F}_{2^8} &\longrightarrow \mathbb{F}_{2^8} \\ x &\longrightarrow x^{254}, \end{aligned}$$

where zero is mapped to zero. The affine function in AES can be expressed as a degree 128 polynomial over  $\mathbb{F}_{2^8}$ :

$$\begin{aligned} A_{AES}(z) = & 0x8F \cdot z^{128} + 0xB5 \cdot z^{64} + 0x01 \cdot z^{32} + 0xF4 \cdot z^{16} + 0x25 \cdot z^8 + \\ & 0xF9 \cdot z^4 + 0x09 \cdot z^2 + 0x05 \cdot z + 0x63. \end{aligned}$$

Then, the  $S$ -box of AES is represented as

$$\begin{aligned} S\text{-box}_{AES}(z) = & 0x05 \cdot z^{254} + 0x09 \cdot z^{253} + 0xF9 \cdot z^{251} + 0x25 \cdot z^{247} + 0xF4 \cdot z^{239} + \\ & 0x01 \cdot z^{223} + 0xB5 \cdot z^{191} + 0x8F \cdot z^{127} + 0x63. \end{aligned}$$

Since JARVIS is also composition of the inverse multiplication and the affine function,  $S$ -box  $S(z)$  of JARVIS can be written as

$$S(z) = A(z^{254}),$$

and the affine function  $A(z)$  is

$$A(z) = (C \circ B^{-1})(z),$$

where  $B$  and  $C$  are both monic permutation polynomials of degree 4. In the original paper [3], it is shown that the  $A_{AES}(z)$  can not be viewed as a decomposition of the polynomials such that

$$A_{AES}(z) = (\hat{C} \circ \hat{B}^{-1})(z),$$

both  $\hat{B}$  and  $\hat{C}$  have degree 4. The above equation implies

$$\begin{aligned} A_{AES}^{-1}(z) &= (\hat{B} \circ \hat{C}^{-1})(z), \\ A_{AES}^{-1}(\hat{C}(z)) &= \hat{B}(z), \end{aligned}$$

where

$$\begin{aligned} A_{AES}^{-1}(z) = & 0x6E \cdot z^{128} + 0xDB \cdot z^{64} + 0x59 \cdot z^{32} + 0x78 \cdot z^{16} + 0x5A \cdot z^8 + \\ & 0x7F \cdot z^4 + 0xFE \cdot z^2 + 0x5 \cdot z + 0x5 \end{aligned}$$

is the compositional inverse polynomial of  $A_{AES}$  which satisfies  $A_{AES}^{-1}(A_{AES}(z)) = z$  for every  $z \in \mathbb{F}_{2^8}$ .

**Lemma 5.1.** *There are no two affine polynomials  $\hat{B}$  and  $\hat{C}$  of degree-4*

$$\hat{B}(z) := \hat{b}_4 z^4 + \hat{b}_2 z^2 + \hat{b}_1 z + \hat{b}_0, \quad \hat{C}(z) := \hat{c}_4 z^4 + \hat{c}_2 z^2 + \hat{c}_1 z + \hat{c}_0. \quad (15)$$

such that  $A_{AES}^{-1}(\hat{C}(z))$  is equal to  $\hat{B}(z)$ .

*Proof.* Assume the equality holds for the polynomials of both degree 4, then we must have zero coefficients in resulting polynomial  $A_{AES}^{-1}(\hat{C}(z))$  for the degrees 8, 16, 32, 64, 128. That means, we need to solve the following multivariate polynomial system with 5 equations in 3 variables  $\hat{c}_4, \hat{c}_2, \hat{c}_1$ :

$$\begin{aligned} 0xFE \cdot \hat{c}_4^2 + 0x7F \cdot \hat{c}_2^4 + 0x5A \cdot \hat{c}_1^8 &= 0, \\ 0x7F \cdot \hat{c}_4^4 + 0x5A \cdot \hat{c}_2^8 + 0x78 \cdot \hat{c}_1^{16} &= 0, \\ 0x5A \cdot \hat{c}_4^8 + 0x78 \cdot \hat{c}_2^{16} + 0x59 \cdot \hat{c}_1^{32} &= 0, \\ 0x78 \cdot \hat{c}_4^{16} + 0x59 \cdot \hat{c}_2^{32} + 0xDB \cdot \hat{c}_1^{64} &= 0, \\ 0x59 \cdot \hat{c}_4^{32} + 0xDB \cdot \hat{c}_2^{64} + 0x6E \cdot \hat{c}_1^{128} &= 0. \end{aligned}$$

In practice, we have obtained that the only solution satisfies the above system is the trivial solution,  $\hat{c}_4 = \hat{c}_2 = \hat{c}_1 = 0$  as shown in [3]. Therefore, affine function of AES can not be decomposed by two degree 4 polynomials.  $\square$

In the following Lemma 5.2, we will show that  $A_{AES}(z)$  can be decomposed as  $A_{AES}(z) = (\hat{C} \circ \hat{B}^{-1})(z)$  if the degree of the product of the polynomials  $\hat{B}$  and  $\hat{C}$  is at least 128.

**Lemma 5.2.** *Let  $\hat{B}$  and  $\hat{C}$  be two affine polynomials of degree  $2^b$  and  $2^c$ , respectively such that*

$$\hat{B}(z) = \hat{b}_{2^b} z^{2^b} + \hat{b}_{2^{(b-1)}} z^{2^{(b-1)}} + \cdots + \hat{b}_2 z^2 + \hat{b}_1 z + \hat{b}_0$$

and

$$\hat{C}(z) = \hat{c}_{2^c} z^{2^c} + \hat{c}_{2^{(c-1)}} z^{2^{(c-1)}} + \cdots + \hat{c}_2 z^2 + \hat{c}_1 z + \hat{c}_0, \quad b, c \in \{0, \dots, 7\}.$$

Then,  $A_{AES}(z)$  results in  $\hat{C}(\hat{B}^{-1}(z))$  provided that  $6 < (b + c) \leq 14$ .

*Proof.* Assume that the degree of  $\hat{C}$  is  $2^c$  and the polynomial  $A_{AES}^{-1}(\hat{C}(z))$  is equal to  $\hat{B}$  having degree  $2^b$ , which implies we need to have zero coefficients for the degrees  $2^{(b+1)}, 2^{(b+2)}, \dots, 2^7$ . This results in a polynomial system of  $(7 - b)$  equations with  $(c + 1)$  variables  $\hat{c}_{2^c}, \dots, \hat{c}_4, \hat{c}_2, \hat{c}_1$ . In order to find a non-zero solution for this system, we need to have more unknowns than the equations. Therefore,  $b$  and  $c$  must satisfy,  $6 < (b + c) \leq 14$ .  $\square$

We have used the above lemma and decomposed the affine function of AES in practice for the following pairs of the degrees of  $\hat{B}$  and  $\hat{C}$ :

- degree of  $\hat{C} = 4, \hat{B} = 32$ ,
- degree of  $\hat{C} = 8, \hat{B} = 16, 32$ ,
- degree of  $\hat{C} = 16, \hat{B} = 8, 16, 32$ ,
- degree of  $\hat{C} = 32, \hat{B} = 4, 8, 16, 32$ .

For a more detailed explanation to see how we solve such a system for the above degrees, one can check [24].

## 5.2 Gröbner basis attack on JARVIS equipped with AES $S$ -box

In the previous section, we show that how the  $S$ -boxes of AES and JARVIS have similar decomposition. In this section, we will replace the non-linear operation in JARVIS with  $S\text{-box}_{AES}(z)$  and estimate the complexity of improved attack strategy given in [3]. Assume we have

$$S\text{-box}_{AES}(z) = (C \circ B^{-1})(z^{254}), z \in \mathbb{F}_{2^8} \quad (16)$$

for a known affine polynomials  $B$  and  $C$

$$\begin{aligned} B(z) &= b_{2^b} z^{2^b} + b_{2^{(b-1)}} z^{2^{(b-1)}} + \cdots + b_2 z^2 + b_1 z + b_0, \\ C(z) &= c_{2^c} z^{2^c} + c_{2^{(c-1)}} z^{2^{(c-1)}} + \cdots + c_2 z^2 + c_1 z + c_0, \quad b, c \in \{0, \dots, 7\}, \end{aligned}$$

where  $(b + c) > 6$  (see Lemma 5.2). The polynomial equations defining the JARVIS with AES  $S$ -box can be viewed as a system of equations such that the equality

$$D(B) = E(C) \quad (17)$$

is satisfied for the affine polynomials  $D$  and  $E$  of the form

$$\begin{aligned} D(z) &= d_{2^d} z^{2^d} + d_{2^{(d-1)}} z^{2^{(d-1)}} \cdots + d_2 z^2 + d_1 z + d_0 \text{ and} \\ E(z) &= e_{2^e} z^{2^e} + e_{2^{(e-1)}} z^{2^{(e-1)}} \cdots + e_2 z^2 + e_1 z + e_0, \quad d, e \in \{0, \dots, 7\}. \end{aligned}$$

We will consider two cases to estimate the complexity of the improved attack on JARVIS with  $S\text{-box}_{AES}(z)$ :

1. The key schedule is the same as in (2).
2. The key schedule in AES is used and all subkeys are captured by the attacker, but not the master key.

Before moving on we first need to find suitable  $D$  and  $E$  such that the system (17) has a solution. We give the following lemma to decide the degrees of the polynomials  $D$  and  $E$ .

**Lemma 5.3.** *Let  $B$  and  $C$  be given decomposition polynomials of the AES  $S$ -box as in (16) having degree  $d_b$  and  $d_c$  respectively where  $(b + c) > 6$  and  $d_e d_c \geq d_d d_b$ . Then, one can find two non-zero affine polynomials  $D$  and  $E$  of degrees  $d_d$  and  $d_e$  respectively satisfying the system (17) provided that  $d + 2 \geq c$ .*

*Proof.* Write the polynomial system for  $D(B) = E(C)$  by comparing the coefficients of  $D(B)$  and  $E(C)$  and assume that  $d_e d_c \geq d_d d_b$ . This system results in  $e + c + 2$  equations, since the number of equations determined by the highest degree, with  $d + e + 4$  variables  $d_{2^d}, d_{2^{d-1}}, \dots, d_2, d_1, d_0$  and  $e_{2^e}, e_{2^{e-1}}, \dots, e_2, e_1, e_0$ . In order to find non-zero solutions to recover the polynomials  $D$  and  $E$ , we must have at least as many variables as the number of equations, which implies  $d + e + 4 \geq e + c + 2$ .  $\square$

We apply suitable polynomials  $D$  and  $E$  which satisfy the above Lemma 5.3 and estimate the complexity of improved attack for both two cases, see in Tables 4 and 5, respectively.

Table 4: Complexity estimates of the improved attack on JARVIS with  $S\text{-box}_{AES}(z)$  and the same key schedule described as in (13).

$r$	$k$	$d_b$	$d_c$	$d_d$	$d_e$	$d_{reg}$	Complexity in bits
6	4	16	8	4	8	490	62
8	5	16	8	4	8	661	80
10	6	16	8	4	8	832	97
12	7	16	8	4	8	1003	115

In the table,  $r$  denotes the number of rounds and  $k$  is the number of variables. The degrees of the decomposition polynomials  $B$  and  $C$  of  $S\text{-box}_{AES}(z)$  and the degrees of the corresponding polynomials  $D$  and  $E$  are denoted by  $d_b, d_c, d_d, d_e$ , respectively. The expected degree of regularity  $d_{reg}$  and complexity estimation in bits are computed, assuming the system behaves like regular sequences, via the formula we give in Proposition 5.1 for  $\omega = 2$ .

Table 5: Complexity estimates of the improved attack on JARVIS with  $S\text{-box}_{AES}(z)$  and AES key schedule in the case of all subkeys are captured by the attacker, but not the master key.

$r$	$k$	$d_b$	$d_c$	$d_d$	$d_e$	$d_{reg}$	Complexity in bits $2 \log_2 \binom{k+d_{reg}}{d_{reg}}$
6	4	16	8	4	8	457	62
8	5	16	8	4	8	616	79
10	6	16	8	4	8	775	96
12	7	16	8	4	8	934	114

In Table 5, the attacker obtains all the key variables  $k_1, \dots, k_r$ , where  $r$  is the number of rounds. The improved attack for the polynomials  $B, C, D, E$  having degree 16, 8, 4, 8 denoted as  $d_b, d_c, d_d, d_e$  yields,  $\frac{r}{2} - 1$  equations of degrees 160 (from (10)), one equation having degree 132 (from (11)), one equation having degree 8 (from (12)). Since the number of equations is same as the number of variables we estimate  $d_{reg}$  using (2), and the expected the bit security computed for  $\omega = 2$ .

**Remark 5.3.** *We note that while the estimated complexity for JARVIS is  $\approx 45$  bits, for the number of rounds  $r = 6$ , this complexity becomes  $\approx 62$  bits, see Table 4, when JARVIS using the  $S$ -box of AES, with an input 8 bits. If we use AES key schedule and  $S$ -box of AES in JARVIS and assume the attacker captures all the subkeys, except the master key, the improved attack complexity is  $\approx 96$  bits for 10 rounds, see Table 5.*

## 6 The block cipher MiMC

The block cipher MiMC "Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity" [2], with its variants, published in 2016 and designed to provide high performance for the applications of secure multi-party computation (MPC), fully homomorphic encryption (FHE), zero knowledge proofs (ZK) and the other popular proof systems like SNARKs and STARKs. It minimizes multiplicative complexity to be efficient over larger fields. In this section, we will describe the block cipher MiMC- $n/n$  and present our experimental results from running the Gröbner basis attack on reduced rounds of MiMC. We will discuss why the cipher is secure against the attack.

### 6.1 MiMC- $n/n$

MiMC is an arithmetic-oriented block cipher works over a finite field  $\mathbb{F}_q$ , where  $q$  is either a prime number or a power of 2. We will mainly consider MiMC over  $\mathbb{F}_{2^n}$ , same description of the cipher is used for prime fields. The round function of MiMC- $n/n$  is described by a non-linear cubic function  $x \mapsto x^3$  where  $x \in \mathbb{F}_{2^n}$ . At each round, the same key  $k$  and the randomly chosen round constants  $c_i \in \mathbb{F}_{2^n}$  are added to the output of the function. The round function of MiMC can be found in Figure 4. Note

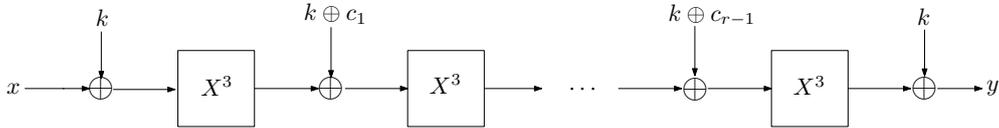


Figure 4:  $r$  rounds of the MiMC- $n/n$  block cipher

that the cube function is a permutation in  $\mathbb{F}_{2^n}$  only if  $n$  is an odd number or if  $\gcd(3, p-1) = 1$  when operate over prime field  $\mathbb{F}_p$ . The decryption in MiMC is done using the round constants in reverse order and inverting the non-linear function  $x^3$  ( $S^{-1}(x) := x^s$  where  $s = (2^{n+1} - 1)/3$  for odd  $n$  [2]). Because of the high degree of inverse cubing function, decryption part is more computationally expensive than the encryption; however, the target applications of MiMC, like cryptographic hash functions, not usually require to perform decryption. The designers give the security analysis for various algebraic attacks and the number of rounds  $r$  for MiMC- $n/n$  is decided by the interpolation attack as  $r = \lceil \frac{n}{\log_2 3} \rceil$ . It is claimed in [2] that 82 rounds is enough for MiMC-129/129 to be secure against GCD, interpolation and the other attacks.

### 6.2 Gröbner basis attack on MiMC

Since the MiMC- $n/n$  has a simple algebraic expression, several algebraic attacks performed in literature [17, 3]. The authors of [3] state that the equations describing MiMC are already form a Gröbner basis, therefore the first step of the attack (computing basis) is free but the recovered univariate polynomial has degree  $\approx 3^r$  for  $r$  rounds. Because of the cost of the factorization algorithm, they conclude that Gröbner basis attack has no threat on the security of MiMC.

The graphical representation of introducing new variables for MiMC- $n/n$  is given in Figure 5.

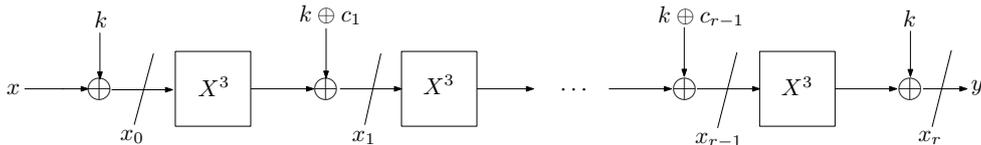


Figure 5: Introducing new intermediate variable  $x_i$  for  $r$  rounds of MiMC- $n/n$

As stated in [3], we express the intermediate rounds of MiMC as follows:

$$\begin{aligned} x_{i-1}^3 + x_i + c_i + k_0 &= 0, \\ x_{r-1}^3 + x_r + k_0 &= 0, \end{aligned}$$

for  $1 \leq i \leq r$ , where  $k_0$  is the key variable. In order to make the polynomial system dependent on plaintext  $p$  and ciphertext  $c$ , we write

$$\begin{aligned} p + k_0 + x_0 &= 0, \\ c + x_r &= 0. \end{aligned}$$

Since the above system already forms a Gröbner basis, we skip the first step of the attack and try to recover the key for the reduced rounds of MiMC-129/129 in practice, see Table 6.

Table 6: The number of rounds and the degree of the univariate equation after applying  $r$  rounds MiMC denoted as  $r$  and  $d_u$  respectively. FGLM and FACT times represents the time, in seconds, needed to compute FGLM and factorization algorithms for the corresponding number of rounds.

$r$	FGLM time	FACT time	$d_u$
3	0.4s	0.2s	27
4	8.8s	2.2s	81
5	266.0s	31.6s	243
6	11462.0s	248.0s	729

Although the equations for MiMC- $n/n$  form a Gröbner basis, time needed to run FGLM and factorization algorithms increases exponentially when the number of rounds increase. Therefore, we conclude that Gröbner basis attack has no threat on MiMC with 82 rounds.

## 7 The block cipher GMiMC

The block cipher GMiMC "*Generalized Feistel MiMC*", proposed in 2019, with its variants is the more efficient generalized version of MiMC and designed to benefit MPC, SNARK applications and PQ-secure signature schemes [4]. In this section, we will briefly describe GMiMC<sub>erf</sub>, a variant of GMiMC using expanding round function, and then give our Gröbner basis attack strategy. In the original proposal [4], the security analysis of the cipher against Gröbner basis attack is based on the difficulty of computing Gröbner basis. However, we discover a recursion in Gröbner basis of GMiMC<sub>erf</sub> with four branches for the univariate case and that enables us to skip the first step of the attack, see Section ?? to remember the steps of the attack. We will show that cipher secure against Gröbner basis attack not because of the complexity of computing Gröbner basis but for a different reason.

### 7.1 Description of GMiMC<sub>erf</sub>

GMiMC-with an expanding round function (erf) is an unbalanced Feistel cipher. One round of an unbalanced Feistel Network with an expanding round function can be written as:

$$(X_{t-1}, X_{t-2}, \dots, X_0) \leftarrow (X_{t-2} \oplus F(X_{t-1}), \dots, X_0 \oplus F(X_{t-1}), X_{t-1})$$

where  $X_j \in \mathbb{F}_{2^n}$  is an input to the  $j$ th branch,  $1 \leq j \leq t-1$ , of the Feistel network,  $F$  is the round function similar to MiMC defined as

$$F(x) := (x \oplus k_i \oplus C_i)^3,$$

$k_i$  is the round key and  $C_i$  is the randomly chosen and fixed round constant for the  $i$ -th round,  $1 \leq i \leq r$ . The graphical representation of the cipher can be found below in Figure 6.

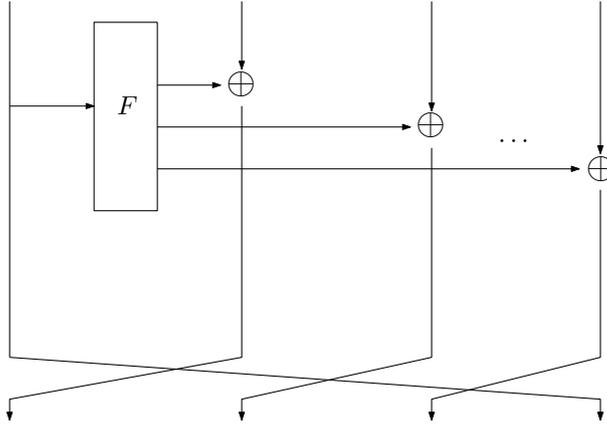


Figure 6: One round of an unbalanced Feistel Network GMiMC with an expanding round function

The description of the cipher over the prime finite field  $\mathbb{F}_p$  with order  $p$  is obtained by replacing XOR operation with the sum in modulo  $p$ . Throughout this paper, we consider the *univariate case*  $\kappa = n$  (or equivalently for the  $\mathbb{F}_p$  case,  $2^\kappa \simeq p$ ) where the key size, denoted by  $\kappa$ , is equal to the branch size  $n = \lceil \log_2 |\mathbb{F}| \rceil$  in bits. Key schedule for the univariate case in GMiMC<sub>erf</sub>, also for the other variants, is linear,  $k_i = k$  for any  $i$ .

## 7.2 Gröbner basis attack on GMiMC

The authors of [4] give a detailed security analysis of GMiMC over  $\mathbb{F}_p$  and discuss the minimum number of rounds that guarantees the security of the cipher for several attacks. They state that most of the attack techniques over  $\mathbb{F}_p$  can be performed similarly in  $\mathbb{F}_{2^n}$ . They claim that the minimum required number of rounds to be resistant against Gröbner basis attacks is given as

$$r = \lceil 0.631 \cdot \log_2(p) + 2\log_3(t) \rceil + 4t - 5.$$

They obtain this value by first observing the degrees of the polynomial equations describing cipher after  $r$  rounds, and then estimating the complexity of computing Gröbner basis for this degree. It is claimed that introducing new intermediate variables does not decrease the complexity of the attack since it causes to increase in number of variables.

### 7.2.1 Our attack strategy

In contrast to the block cipher MiMC- $n/n$ , the polynomial equations describing GMiMC<sub>erf</sub> do not form a Gröbner basis. Therefore, to perform the attack, one first needs to compute the Gröbner basis which is the most expensive step of the attack. Our idea was to find -if any- a recursion or a path in basis in order to skip basis computation. We discovered the recursion in degrevlex ordered Gröbner basis of GMiMC<sub>erf</sub> with four branches for the univariate case ( $2^\kappa \simeq p$ ) over arbitrary prime fields  $\mathbb{F}_p$  and so able to write a general recursive formula of the basis for  $r$  rounds of the cipher.

We describe the primitive by setting four intermediate variables  $x_{4(i-1)}$ ,  $x_{4(i-1)+1}$ ,  $x_{4(i-1)+2}$ ,  $x_{4(i-1)+3}$  for each round from leftmost to the rightmost branch where  $1 \leq i \leq r$  as depicted below in Figure 7.

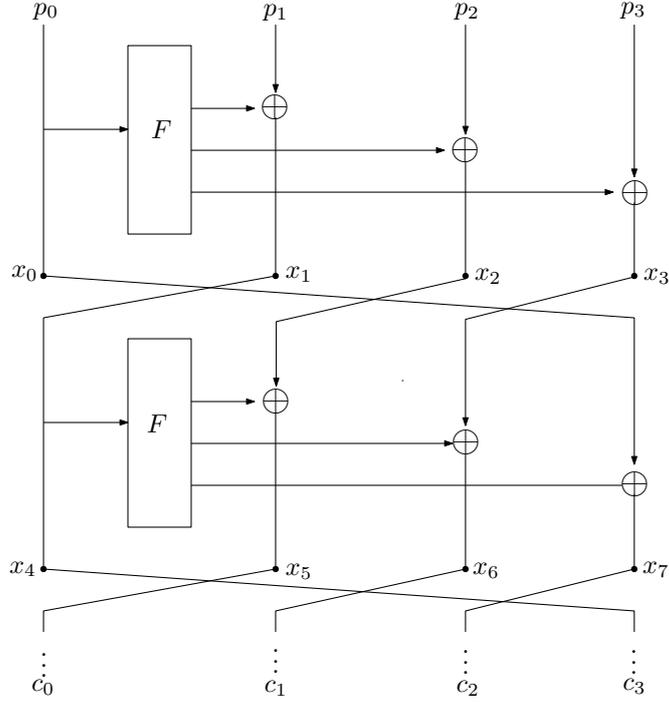


Figure 7: Introducing new intermediate variables  $x_{4(i-1)}$ ,  $x_{4(i-1)+1}$ ,  $x_{4(i-1)+2}$ ,  $x_{4(i-1)+3}$  for  $r$  rounds of  $\text{GMiMC}_{\text{era}}$  where  $1 \leq i \leq r$  with branch number  $t = 4$ .

Two consecutive rounds of  $\text{GMiMC}_{\text{era}}$  with 4 branches can be related as follows

$$\begin{aligned}
x_{4i} - x_{4(i-1)+1} &= 0, \\
x_{4i+1} - x_{4(i-1)+2} - F(x_{4(i-1)+1}, k_0, C_i) &= 0, \\
x_{4i+2} - x_{4(i-1)+3} - F(x_{4(i-1)+1}, k_0, C_i) &= 0, \\
x_{4i+3} - x_{4(i-1)} - F(x_{4(i-1)+1}, k_0, C_i) &= 0,
\end{aligned}$$

for  $1 \leq i \leq r - 1$ , where  $k_0$  and  $C_i$ 's are key and constant variables, respectively. To make the system dependent on the plaintext  $p$  and the ciphertext  $c \in (\mathbb{F}_p)^t$ , where  $p = (p_0, p_1, p_2, p_3)$  and  $c = (c_0, c_1, c_2, c_3)$ , we add 4 plaintext equations

$$\begin{aligned}
x_0 - p_0 &= 0, \\
x_1 - F(p_0, k_0, C_0) - p_1 &= 0, \\
x_2 - F(p_0, k_0, C_0) - p_2 &= 0, \\
x_3 - F(p_0, k_0, C_0) - p_3 &= 0,
\end{aligned}$$

and the 4 ciphertext equations

$$\begin{aligned}
x_{4(r-1)} - c_3 &= 0, \\
x_{4(r-1)+1} - c_0 &= 0, \\
x_{4(r-1)+2} - c_1 &= 0, \\
x_{4(r-1)+3} - c_2 &= 0.
\end{aligned}$$

Notice that the above system has six polynomial equations of degree 3 and two equations of degree 1 for the intermediate and the plaintext equations, and has four linear ciphertext equations. In practice, we observe that this system does not form a Gröbner basis for the primitive unlike to  $\text{MiMC-}n/n$ .

In order to discover a recursion or a structure in Gröbner basis for the above polynomial equations describing the cipher, we compute the basis for reduced rounds of  $\text{GMiMC}_{\text{era}}$  over prime fields  $\mathbb{F}_p$  having different prime orders  $p > 11$  and see that the basis is independent from the choice of order. We were able to compute the Gröbner basis in degrevlex order until 13 rounds using SageMath, which

was enough to discover the pattern in the basis. Gröbner basis  $G$  for  $\text{GMiMC}_{\text{erf}}$  with 4 branches over arbitrary prime field  $\mathbb{F}_p$  with respect to the degrevlex order has the following forms:

If  $r = 5 + 3k, k \in \mathbb{N}$ , then

$$\begin{aligned}
G = \{ & (x_{4(r-1)} + k_0 + C_0)^3 - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \\
& x_{4(r-8)} - x_{4(r-9)} + \cdots + 2x_4 + p_2 - p_1 - p_0, \\
& (x_{4(r-2)} + k_0 + C_0)^3 - x_{4(r-1)} + x_{4(r-2)} - x_{4(r-4)} + 2x_{4(r-5)} - x_{4(r-6)} - x_{4(r-7)} + \cdots + 2x_{12} - x_8 - x_4 - \\
& p_3 + p_1 + p_0, \\
& (x_{4(r-3)} + k_0 + C_0)^3 - x_{4(r-2)} + x_{4(r-3)} - x_{4(r-5)} + 2x_{4(r-6)} - x_{4(r-7)} - x_{4(r-8)} + \cdots + 2x_8 - x_4 + p_3 - p_2, \\
& (x_{4(r-4)} + k_0 + C_0)^3 - x_{4(r-3)} + x_{4(r-4)} - x_{4(r-6)} + 2x_{4(r-7)} - x_{4(r-8)} - x_{4(r-9)} + \cdots + 2x_4 + p_2 - p_1 - p_0, \\
& \vdots \\
& (x_{12} + k_0 + C_0)^3 - x_{16} + x_{12} - x_4 - p_3 + p_1 + p_0, \\
& (x_8 + k_0 + C_0)^3 - x_{12} + x_8 + p_3 - p_2, \\
& (x_4 + k_0 + C_0)^3 - x_8 + x_4 + p_2 - p_1, \\
& (p_0 + k_0 + C_0)^3 - x_4 + p_1, \\
& x_{4(r-1)+3} - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-2)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \cdots + 2x_4 + \\
& p_2 - p_1 - p_0, \\
& x_{4(r-1)+2} - x_{4(r-1)+1} + x_{4(r-2)} - 2x_{4(r-3)} + x_{4(r-4)} + x_{4(r-5)} - 2x_{4(r-6)} + \cdots + x_4 - p_3 + p_2 \},
\end{aligned}$$

If  $r = 6 + 3k, k \in \mathbb{N}$ , then

$$\begin{aligned}
G = \{ & (x_{4(r-1)} + k_0 + C_0)^3 - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \\
& x_{4(r-8)} - x_{4(r-9)} + \cdots - x_4 + p_3 - p_2, \\
& (x_{4(r-2)} + k_0 + C_0)^3 - x_{4(r-1)} + x_{4(r-2)} - x_{4(r-4)} + 2x_{4(r-5)} - x_{4(r-6)} - x_{4(r-7)} + \cdots + 2x_4 + p_2 - p_1 - p_0, \\
& (x_{4(r-3)} + k_0 + C_0)^3 - x_{4(r-2)} + x_{4(r-3)} - x_{4(r-5)} + 2x_{4(r-6)} - x_{4(r-7)} - x_{4(r-8)} + \cdots - x_4 - p_3 + p_1 + p_0, \\
& (x_{4(r-4)} + k_0 + C_0)^3 - x_{4(r-3)} + x_{4(r-4)} - x_{4(r-6)} + 2x_{4(r-7)} - x_{4(r-8)} - x_{4(r-9)} + \cdots - x_4 + p_3 - p_2, \\
& \vdots \\
& (x_{12} + k_0 + C_0)^3 - x_{16} + x_{12} - x_4 - p_3 + p_1 + p_0, \\
& (x_8 + k_0 + C_0)^3 - x_{12} + x_8 + p_3 - p_2, \\
& (x_4 + k_0 + C_0)^3 - x_8 + x_4 + p_2 - p_1, \\
& (p_0 + k_0 + C_0)^3 - x_4 + p_1, \\
& x_{4(r-1)+3} - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-2)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \cdots + 2x_8 - \\
& x_4 + p_3 - p_2, \\
& x_{4(r-1)+2} - x_{4(r-1)+1} + x_{4(r-2)} - 2x_{4(r-3)} + x_{4(r-4)} + x_{4(r-5)} - 2x_{4(r-6)} + \cdots + x_4 + p_3 - p_1 - p_0 \},
\end{aligned}$$

If  $r = 7 + 3k, k \in \mathbb{N}$ , then

$$\begin{aligned}
G = \{ & (x_{4(r-1)} + k_0 + C_0)^3 - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \\
& x_{4(r-8)} - x_{4(r-9)} + \cdots - x_8 - x_4 - p_3 + p_1 + p_0, \\
& (x_{4(r-2)} + k_0 + C_0)^3 - x_{4(r-1)} + x_{4(r-2)} - x_{4(r-4)} + 2x_{4(r-5)} - x_{4(r-6)} - x_{4(r-7)} + \cdots + 2x_8 - x_4 + p_3 - p_2, \\
& (x_{4(r-3)} + k_0 + C_0)^3 - x_{4(r-2)} + x_{4(r-3)} - x_{4(r-5)} + 2x_{4(r-6)} - x_{4(r-7)} - x_{4(r-8)} + \cdots + 2x_4 + p_2 - p_1 - p_0, \\
& (x_{4(r-4)} + k_0 + C_0)^3 - x_{4(r-3)} + x_{4(r-4)} - x_{4(r-6)} + 2x_{4(r-7)} - x_{4(r-8)} - x_{4(r-9)} + \cdots - x_4 - p_3 + p_1 + p_0, \\
& \vdots \\
& (x_{12} + k_0 + C_0)^3 - x_{16} + x_{12} - x_4 - p_3 + p_1 + p_0, \\
& (x_8 + k_0 + C_0)^3 - x_{12} + x_8 + p_3 - p_2, \\
& (x_4 + k_0 + C_0)^3 - x_8 + x_4 + p_2 - p_1, \\
& (p_0 + k_0 + C_0)^3 - x_4 + p_1, \\
& x_{4(r-1)+3} - x_{4(r-1)+1} + x_{4(r-1)} - x_{4(r-2)} - x_{4(r-3)} + 2x_{4(r-4)} - x_{4(r-5)} - x_{4(r-6)} + 2x_{4(r-7)} - \cdots - x_4 - \\
& p_3 + p_1 + p_0, \\
& x_{4(r-1)+2} - x_{4(r-1)+1} + x_{4(r-2)} - 2x_{4(r-3)} + x_{4(r-4)} + x_{4(r-5)} - 2x_{4(r-6)} + \cdots - 2x_4 - p_2 + p_1 + p_0 \},
\end{aligned}$$

where  $p = (p_0, p_1, p_2, p_3)$  is the plaintext and  $C_0$  is the round constant variable.

We note that we only consider the case where same key and round constant are used in each round and the number of branches is 4. We conclude that the Gröbner basis of  $\text{GMiMC}_{\text{erf}}$  has the above structure which makes the first step of the Gröbner basis attack for free. That means that one can always compute the Gröbner basis for any number of rounds and therefore there is no complexity

of computing Gröbner basis for  $\text{GMiMC}_{\text{eff}}$ . In order to recover the key, we used the above Gröbner basis elements as our polynomial equations with a single known p/c pair, and tried to change the term ordering in basis from *degrevlex* order to *lex* order via the FGLM algorithm (second step of the attack). The ideal generated by those equations is not zero-dimensional, so we used the Gröbner Walk algorithm and recovered the key until 13 rounds by solving the last univariate basis equation. However, the Gröbner Walk algorithm was slower than the FGLM algorithm and hence, our attack strategy did not speed up the Gröbner basis attack on  $\text{GMiMC}_{\text{eff}}$ . The natural question is to ask what happens if one makes the ideal zero-dimensional.

## 8 Conclusion

In this paper we focus on the Gröbner basis attack on three different symmetric-key primitives JARVIS-like ciphers, MiMC and  $\text{GMiMC}_{\text{eff}}$  which are designed to offer efficient solution for the applications of advanced cryptographic protocols.

We study the successful Gröbner basis attack against JARVIS by Albrecht et al. [3]. Then we extend this result for presenting our general formula to estimate the complexity of the attack on variants of JARVIS. We use this formula to analyze the security of JARVIS-like ciphers with higher degree polynomials. We choose the affine polynomials in JARVIS-128 as degree 8 polynomials rather than 4 and observe that although the expected bit security increases ( $\approx 90$  bits for 10 rounds), the cipher still does not provide the claimed security level in the original proposal [7]. Since the block cipher JARVIS is very similar to the AES  $S$ -box, we write the  $S$ -box of AES as a decomposition of two affine polynomials.

Next, we replace the JARVIS round function with the AES  $S$ -box which operates on inputs of 8-bits. We estimate the complexity of the improved attack in two cases, the first one is that we use the key schedule of JARVIS, and the other is that we regard the AES key schedule but the attacker obtained all subkeys. For both cases, we see that the improved attack complexity is around  $\approx 97$  bits when number of rounds is 10.

Also, we apply a Gröbner basis attack to MiMC block cipher. The first step of the attack is free because of the equations describing MiMC already forms a Gröbner basis as emphasized in [3]. We recovered the secret key until 6 rounds using SageMath. Our equations for MiMC result in a univariate polynomial of degree  $\approx 3^r$ . We conclude that our Gröbner basis attack strategy has no threat on MiMC due to the complexities of FGLM and factorization algorithms.

The polynomial equations we construct for the block cipher  $\text{GMiMC}_{\text{eff}}$  do not form a Gröbner basis unlike to MiMC. We consider the cipher with 4 branches and compute degrevlex ordered Gröbner basis until 13 rounds. However, the specified number of rounds for  $\text{GMiMC}$  is much higher. Therefore, our attack strategy is to make the first step of the attack free. In order to avoid basis computation, we try to find a structure in basis which leads us to obtain the Gröbner basis for any number of rounds. We find Gröbner bases in degrevlex. However, we couldn't change order of the terms to lexicographic order via FGLM algorithm since the dimension of the ideal was not zero. We use Gröbner Walk algorithm to recover univariate equation with single plaintext/ciphertext pair and solve for the key until 13 rounds but attack is still not applicable to the full round of  $\text{GMiMC}$  due to the performance reasons of Gröbner Walk algorithm.

### 8.1 Discussion and future work

The symmetric-key primitives become more algebraically simple to provide efficient solution in the applications of advanced cryptographic protocols in recent years. Security of these designs usually assured by the number of rounds to avoid corresponding algebraic attacks. Gröbner basis attack is one of those attacks which should be regarded especially due to the recent success of the attack on primitives JARVIS and FRIDAY. However, there is no generic systematic security argument for deciding resistance towards Gröbner basis attacks without experimentally running the flavor of the attack we give in Section 3. A systematic way to describe complexity of the attack should be a future work.

The paper [6] provides a novel framework to determine the security of the cipher against Gröbner basis attack. For most of the new designs, it becomes a standard that *the cipher's resistance against the Gröbner basis attack should be based on infeasible complexity of computing Gröbner basis in degrevlex order*. Note that the first step of the attack is free for the MiMC case. But, it turns out that the cipher is secure against the attack because of the complexity of order conversion and factorization algorithms.

We provide a complexity estimation for the JARVIS-like ciphers according to the attack strategy called *bridging equations over two rounds* in the original paper [3]. We apply this attack to AES *S*-box but that is not a comprehensive work due to the lack of time. Bridging more than two equations to reduce number of variables or generalization of the improved attack strategy on JARVIS and apply to AES or new arithmetization-oriented ciphers such as *GMiMC*, *Starkad/Poseidon* is a subject for the future work.

## References

- [1] Starkware industries: STARK-friendly hash challenge, <https://starkware.co/hash-challenge>, visited on September 15, 2021.
- [2] Martin Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 191–219. Springer, 2016.
- [3] Martin R Albrecht, Carlos Cid, Lorenzo Grassi, Dmitry Khovratovich, Reinhard Lüftenegger, Christian Rechberger, and Markus Schofnegger. Algebraic cryptanalysis of stark-friendly designs: application to MARVELlous and MiMC. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 371–397. Springer, 2019.
- [4] Martin R Albrecht, Lorenzo Grassi, Léo Perrin, Sebastian Ramacher, Christian Rechberger, Dragos Rotaru, Arnab Roy, and Markus Schofnegger. Feistel structures for MPC, and more. In *European Symposium on Research in Computer Security*, pages 151–171. Springer, 2019.
- [5] Martin R Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 430–454. Springer, 2015.
- [6] Abdelrahman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Transactions on Symmetric Cryptology*, pages 1–45, 2020.
- [7] Tomer Ashur and Siemen Dhooghe. MARVELlous: a STARK-friendly family of cryptographic primitives. *IACR Cryptol. ePrint Arch.*, 2018:1098, 2018.
- [8] M Bardet, JC Faugere, B Salvy, and BY Yang. Asymptotic behaviour of the index of regularity of quadratic semi-regular polynomial systems. In *8th International Symposium on Effective Methods in Algebraic Geometry*. Citeseer, 2005.
- [9] Magali Bardet, Jean-Charles Faugere, and Bruno Salvy. Complexity of Gröbner basis computation for semi-regular overdetermined sequences over  $\mathbb{F}_2$  with solutions in  $\mathbb{F}_2$ . 2003.
- [10] Eli Ben-Sasson, Lior Goldberg, and David Levit. Stark friendly hash – survey and recommendation. Cryptology ePrint Archive, Report 2020/948, 2020. <https://ia.cr/2020/948>.
- [11] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 67–74, 2012.

- [12] Bruno Buchberger. Bruno buchberger’s phd thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of symbolic computation*, 41(3-4):475–511, 2006.
- [13] Johannes Buchmann, Andrei Pyshkin, and Ralf-Philipp Weinmann. Block ciphers sensitive to Gröbner basis attacks. In *Cryptographers’ Track at the RSA Conference*, pages 313–331. Springer, 2006.
- [14] Anne Canteaut, Sergiu Carpov, Caroline Fontaine, Tancrede Lepoint, María Naya-Plasencia, Pascal Paillier, and Renaud Sirdey. Stream ciphers: A practical solution for efficient homomorphic-ciphertext compression. *Journal of Cryptology*, 31(3):885–916, 2018.
- [15] Stéphane Collart, Michael Kalkbrener, and Daniel Mall. Converting bases with the Gröbner walk. *Journal of Symbolic Computation*, 24(3-4):465–469, 1997.
- [16] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.
- [17] Maria Eichlseder, Lorenzo Grassi, Reinhard Lüftenegger, Morten Øy garden, Christian Rechberger, Markus Schofnegger, and Qingju Wang. An algebraic attack on ciphers with low-degree round functions: Application to full MiMC. *IACR Cryptol. ePrint Arch.*, 2020:182, 2020.
- [18] Jean-Charles Faugere. A new efficient algorithm for computing Gröbner bases (F4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.
- [19] Jean Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.
- [20] Jean-Charles Faugere, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.
- [21] Giulio Genovese. Improving the algorithms of Berlekamp and Niederreiter for factoring polynomials over finite fields. *Journal of Symbolic Computation*, 42(1-2):159–177, 2007.
- [22] Lorenzo Grassi, Daniel Kales, Dmitry Khovratovich, Arnab Roy, Christian Rechberger, and Markus Schofnegger. Starkad and poseidon: New hash functions for zero knowledge proof systems. 2019.
- [23] Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger. On a generalization of substitution-permutation networks: The HADES design strategy. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 674–704. Springer, 2020.
- [24] Gizem Kara. Gröbner basis attack on STARK-friendly symmetric-key primitives: JARVIS, MiMC and GMiMCerf. Master’s thesis, Middle East Technical University, 2021.
- [25] Phil Karn, Perry Metzger, and William Simpson. The ESP triple DES transform. *RFC1851*, 1995.
- [26] Pierrick Méaux, Anthony Journault, François-Xavier Standaert, and Claude Carlet. Towards stream ciphers for efficient FHE with low-noise ciphertexts. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 311–343. Springer, 2016.
- [27] Vincent Rijmen and Joan Daemen. Advanced encryption standard. *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pages 19–22, 2001.

- [28] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.
- [29] Igor Semaev and Andrea Tenti. Probabilistic analysis on Macaulay matrices over finite fields and complexity of constructing Gröbner bases. *IACR Cryptol. ePrint Arch.*, 2019:903, 2019.
- [30] William Stein and David Joyner. SAGE: System for algebra and geometry experimentation. *Acm Sigsam Bulletin*, 39(2):61–64, 2005.