Gröbner Basis Cryptanalysis of Ciminion and Hydra

Matthias Johann Steiner 🆔

Alpen-Adria-Universität Klagenfurt, Klagenfurt am Wörthersee, Austria
mattsteiner@edu.aau.at

Abstract. Ciminion and Hydra are two recently introduced symmetric key Pseudo-Random Functions for Multi-Party Computation applications. For efficiency, both primitives utilize quadratic permutations at round level. Therefore, polynomial system solving-based attacks pose a serious threat to these primitives. For Ciminion, we construct a quadratic degree reverse lexicographic (DRL) Gröbner basis for the iterated polynomial model via linear transformations. With the Gröbner basis we can simplify cryptanalysis, as we no longer need to impose genericity assumptions to derive complexity estimates. For Hydra, with the help of a computer algebra program like SageMath we construct a DRL Gröbner basis for the iterated model via linear transformations and a linear change of coordinates. In the Hydra proposal it was claimed that $r_{\mathcal{H}} = 31$ rounds are sufficient to provide 128 bits of security against Gröbner basis attacks for an ideal adversary with $\omega = 2$. However, via our Hydra Gröbner basis standard term order conversion to a lexicographic (LEX) Gröbner basis requires just 126 bits with $\omega = 2$. Moreover, using a dedicated polynomial system solving technique up to $r_{\mathcal{H}} = 33$ rounds can be attacked below 128 bits for an ideal adversary.

Keywords: Gröbner basis · Ciminion · Hydra

1 Introduction

With secure *Multi-Party Computation* (MPC) several parties can jointly compute a function without revealing their private inputs. In practice, data often has to be securely transferred from and to third parties before it can be used in MPC protocols. Also, one may need to store intermediate results securely in a database. For efficiency, it is recommended to perform encryption via an MPC-friendly *Pseudo-Random Function* (PRF) [GRR+16] which utilizes a secret-shared symmetric key. Compared to classical bit-based PRFs their MPC counterparts have two novel design criteria:

- They must be native over large finite fields $\mathbb{F}_q$, typically $\log_2(q) \geq 64$ and q is prime.

- They must admit a low number of multiplications for evaluation.

Two such recently introduced MPC-friendly PRFs are Ciminion [DGGK21] and Hydra [GØSW23]. These constructions are based on the iteration of degree 2 Feistel and Lai–Massey permutations. Due to the low degree at round level, both designs consider Gröbner basis attacks as one of the most threatening attack vectors.

In this paper we perform dedicated Gröbner basis cryptanalysis for both designs. Let us quickly sketch the main steps of a Gröbner basis attack on a symmetric design:

(1) Model the function with a system of polynomials. For Ciminion and Hydra we analyze an *iterated* polynomial model $\mathcal{R}(\mathbf{x}_{i-1}, \mathbf{y}) - \mathbf{x}_i = \mathbf{0}$, where $\mathcal{R} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is

the round function, the $\mathbf{x}_i$'s are intermediate state variables with $\mathbf{x}_0$ the input state and $\mathbf{x}_r$ the output state of the function, and $\mathbf{y}$ the cryptographic secret variables.

(2) Compute a Gröbner basis with respect to an efficient term order, e.g. the *degree reverse lexicographic* (DRL) term order.

(3) Construct a univariate polynomial, e.g. via term order conversion to the *lexicographic* (LEX) term order or via the Eigenvalue Method.

(4) Factor the univariate polynomial.

For `Ciminion` and `Hydra` the designers conjectured that Step (2) dominates the complexity of a Gröbner basis attack. In addition, to estimate the complexity of this step it was assumed that the polynomial systems are *regular* or *semi-regular* [BFS04]. We on the other hand will construct DRL Gröbner bases for both designs via linear transformations and possibly a linear change of variables. Thus, we can directly proceed with Step (3). Typically, this step is also computationally cheaper than finding the DRL Gröbner basis for generic polynomial systems, which is the main reason why we yield better complexity estimations than the designers.

1.1 Our Results & Relationship to Other Works

The research goals of this work can be summarized in three bullet points:

- Increase the understanding of polynomial models for MPC-friendly designs.

- Provide clean cryptanalysis which avoids strong genericity assumptions.

- Improve the complexity estimations of `Ciminion` and `Hydra` from a designer's point of view.

Ciminion. The PRF `Ciminion` is an iterated Feistel design based on the Toffoli gate [Tof80]. For the iterated `Ciminion` polynomial model we construct a DRL Gröbner basis via linear transformations, see Theorem 3.1. Hence, we trivialize the hardness assumption of Step (2).

It is worthwhile mentioning that in a recent work Bariant [Bar23] constructed the univariate LEX polynomial of `Ciminion` via iterated univariate polynomial multiplications. The complexities of constructing and factoring the LEX polynomial are linear-logarithmic in its degree. On the other hand, constructing the univariate LEX polynomial via Gröbner basis methods scales at least quadratic-logarithmic in its degree. Therefore, to the best of our knowledge Bariant's attack currently constitutes the most competitive attack on certain `Ciminion` parameters. We also review this attack in more detail in Section 3.2.

There are two simple countermeasures against Bariant's attack: increasing the number of rounds or additional key additions. For the latter, we also discuss in Section 3.2 how key additions can be performed while maintaining the `Ciminion` DRL Gröbner basis from Theorem 3.1. Additionally, in Section 3.3.1 we propose `Ciminion2`, a slightly modified variant which incorporates key additions that are compatible with Theorem 3.1.

In Section 3.4 we discuss Gröbner basis cryptanalysis of `Ciminion` and `Ciminion2` respectively. Due to our DRL Gröbner basis we yield a clean analysis as well as state-of-the-art complexity estimations for `Ciminion`. In particular, we improve upon the designer's original analysis. In Table 1 we present the minimal `Ciminion` round numbers that achieve 128 bits of security for an ideal adversary[1] against the designer's estimation

---

[1]In our setting an ideal adversary has access to a hypothetical matrix multiplication algorithm which achieves the linear algebra constant $\omega = 2$.

("Fully Substituted Model"), Bariant's attack and our dedicated Eigenvalue Method for `Ciminion` and `Ciminion2`.

Table 1: Minimal number of rounds required for `Ciminion` and `Ciminion2` to achieve at least 128 bits of security against various Gröbner basis attacks. All computations are performed over the prime $q = 2^{127} + 45$ with $\omega = 2$.

	Complexity (bits)		
$r_C + r_E$	Bariant's Attack [Bar23]	Eigenvalue Method	Fully Substituted Model [DGGK21, §4.4]
33	46.67	63.09	130
66	81.22	129.09	262
112	128.47	221.09	446

Hydra.   The heads of the `Hydra` can be considered as iterated Lai–Massey ciphers, where the key as well as the input are hidden from the adversary and only the output is released. In a recent work Steiner [Ste24] introduced the notion of polynomial systems in *generic coordinates* [CG21] to cryptographic polynomial models. In essence, if a polynomial system is in generic coordinates, then we have a proven complexity for the computation of the DRL Gröbner basis [CG21]. In [Ste24, §6.2] the iterated polynomial models of generalized Feistel ciphers were analyzed for being in generic coordinates. For many instances of Feistel ciphers generic coordinates can be verified via the rank of an associated linear system. In Theorem 4.1 we will construct such a linear system for the iterated `Hydra` model. Hence, with the aid of a computer algebra program like `SageMath` [Sag24] we yield a computer-aided proof for the complexity of DRL Gröbner basis computations. Additionally, if the linear system for generic coordinates verification has full rank, then we can perform a linear change of coordinates to transform the `Hydra` polynomial model into a DRL Gröbner basis, see Section 4.2.

Via the change of coordinates we produce a quadratic DRL Gröbner basis in $2 \cdot r_{\mathcal{H}} - 2$ polynomials and variables together with four additional quadratic polynomials. This leaves us with two possible routes to recover the key:

(A) We can recompute the DRL Gröbner basis for $2 \cdot r_{\mathcal{H}} + 2$ quadratic polynomials.

(B) We ignore the additional equations and just use the DRL Gröbner basis to construct a univariate polynomial.

The first approach was also considered by the `Hydra` designers [GØSW23, §7.4], although without the linear transformation to the DRL Gröbner basis. Under the semi-regularity assumption [BFS04] the designers derived the *degree of regularity*, which determines the complexity of the computation. In principle, we can apply the same strategy, but with one important benefit: After the linear transformation we only have to consider *square-free* monomials in a DRL Gröbner basis computation. This effectively saves both time and space. In analogy to Gröbner basis computations over $\mathbb{F}_2$ we call this the *Boolean Semi-Regular* Gröbner basis attack.

Alternatively, we can simply restrict to the DRL Gröbner basis and proceed with term order conversion via the probabilistic FGLM algorithm [FGHR14]. It turns out that this approach invalidates the designers' claim that $r_{\mathcal{H}} = 31$ rounds cannot be attacked by an ideal adversary below 128 bits. In addition, via a dedicated Eigenvalue Method an ideal adversary can attack up to $r_{\mathcal{H}} = 33$ rounds.

In Section 4.4 we discuss the complexity estimations for `Hydra` in more detail, a summary of the considered attacks is given in Table 2. Let $r_{\mathcal{H}}^*$ be the least round number which achieves 128 bits of security against an ideal adversary. The `Hydra` designers require for the

head round number that $r_{\mathcal{H}} = \lceil 1.25 \cdot \max\{24, 2 + r_{\mathcal{H}}^*\} \rceil$ [GØSW23, §5.5]. In the original analysis the designers found that $r_{\mathcal{H}}^* = 29$ and $r_{\mathcal{H}} = 39$, however our analysis yields $r_{\mathcal{H}}^* = 34$ which implies the increase $r_{\mathcal{H}} = 45$ to maintain the originally intended security margin.

Table 2: Complexities of Gröbner basis attacks on Hydra which beat the designers' analysis, and security margin reduction for full rounds. All computations are performed over the prime $q = 2^{127} + 45$ with $\omega = 2$.

$r_{\mathcal{H}}$	Complexity (Bits)			
	Term Order Conversion	Eigenvalue Method	Boolean Semi-Regular Estimate	Semi-Regular Estimate [GØSW23]
29	117.81	111.09	123.29	130.80
31	125.91	119.09	135.91	141.77
33	134.00	127.09	145.75	152.75
39	158.25	151.09	172.66	182.22

Implementation. We implemented the Ciminion and Hydra polynomial systems and their DRL Gröbner bases in the computer algebra systems SageMath [Sag24] and OSCAR [OSC24].[2] In particular, for Hydra we have implemented the linear system for the generic coordinates verification as well as the linear change of coordinates to produce a quadratic DRL Gröbner basis. We stress that our SageMath implementations are able to handle practical parameters like 128 bit prime numbers and full rounds Hydra with $r_{\mathcal{H}} = 39$.

1.2 Organization of the Paper

In Section 2 we introduce the technical requirements of this paper. Ciminion and Hydra are formally introduced in Sections 2.1 and 2.2 respectively. In Section 2.3 we quickly review Gröbner basis algorithms and the notion of generic coordinates. As preparation for Hydra cryptanalysis, in Section 2.4 we consider polynomial systems $\mathcal{F} \subset K[x_1, \ldots, x_n]$ which contain a subset $\mathcal{F}_{\text{Bool}} = \{g_1, \ldots, g_n\} \subset \mathcal{F}$ with $\text{LM}_{DRL}(g_i) = x_i^2$ for all $1 \leq i \leq n$. For such polynomial systems we can reduce Gaussian elimination on the Macaulay matrix $M_{\leq d}$ to a submatrix $M_{\leq d}^{\text{Bool}}$ which only represents the square-free monomials. The complexity of square-free Gaussian elimination is derived in Theorem 2.10, and the one of the auxiliary division by remainder step is given in Corollary 2.12. Finally, in Section 2.5 we develop a dedicated strategy, called the *Eigenvalue Method*, to find a $\mathbb{F}_q$-valued key guess via the quadratic DRL Gröbner bases for Ciminion and Hydra.

In Section 3 we analyze the iterated Ciminion polynomial system. In particular, in Theorem 3.1 we construct a Ciminion DRL Gröbner basis via linear transformations. Bariant's attack and compatibility of countermeasures with the DRL Gröbner basis are discussed in Section 3.2. Finally, in Section 3.4 we discuss Gröbner basis cryptanalysis of Ciminion.

In Section 4 we analyze the iterated polynomial system for the Hydra heads. A linear system to verify generic coordinates for Hydra is developed in Theorem 4.1. In case the linear system has full rank we discuss in Section 4.2 that a linear change of coordinates yields a quadratic DRL Gröbner basis together with four additional quadratic polynomials. Note that we do not perform this change of coordinates by hand, instead we outsource it to our SageMath or OSCAR implementations. In Section 4.4 we finish this paper with Gröbner basis cryptanalysis for Hydra.

---

[2] https://github.com/sca-research/Groebner-Basis-Cryptanalysis-of-Ciminion-and-Hydra.git

2   Preliminaries

We denote fields with K and their algebraic closure by $\overline{K}$. The finite field with q elements is denoted as $\mathbb{F}_q$. If the field and the number of variables are clear from context, then we abbreviate the polynomial ring as $P = K[x_1, \ldots, x_n]$. Matrices $\mathbf{M} \in K^{m \times n}$ are denoted with upper bold letters and vectors $\mathbf{v} \in K^n$ by lower bold letters. Matrix-vector and matrix-matrix products are written as $\mathbf{M}\mathbf{v}$.

Let $f \in K[x_1, \ldots, x_n]$ be a polynomial, and let x_0 be an additional variable, the homogenization of f with respect to x_0 is defined as

$$f^{\mathrm{hom}}(x_0, \ldots, x_n) = x_0^{\deg(f)} \cdot f\left(\frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0}\right) \in K[x_0, \ldots, x_n]. \tag{1}$$

Analogously, we denote the homogenization of ideals $I^{\mathrm{hom}} = \left\{ f^{\mathrm{hom}} \mid f \in I \right\}$ and of finite systems of polynomials $\mathcal{F}^{\mathrm{hom}} = \left\{ f_1^{\mathrm{hom}}, \ldots, f_m^{\mathrm{hom}} \right\}$.

Let $f \in K[x_1, \ldots, x_n]$ be a polynomial, we can always decompose it as sum $f = f_d + \ldots + f_0$, where f_i is homogeneous of degree i. We call f_d the homogeneous highest degree component of f and denote it as f^{top}. Analogously, we denote $\mathcal{F}^{\mathrm{top}} = \left\{ f_1^{\mathrm{top}}, \ldots, f_m^{\mathrm{top}} \right\}$. Note that we have $f^{\mathrm{top}} = f^{\mathrm{hom}} \mod (x_0)$.

Let $I \subset R$ be an ideal in a commutative ring, the radical of I is defined as $\sqrt{I} = \{ f \in R \mid \exists n \in \mathbb{Z}_{\geq 1} \colon f^n \in I \}$. It is well-known that the radical is also an ideal.

2.1   Ciminion

Let $\mathbb{F}_q$ be a finite field, the Ciminion [DGGK21] PRF over $\mathbb{F}_q$ is most easily described via an illustration, see Figure 1. A nonce $\aleph \in \mathbb{F}_q$ and the first key pair (K_1, K_2) are fed into the permutation $p_C : \mathbb{F}_q^3 \to \mathbb{F}_q^3$. Then the outputs are fed into the permutation $p_E : \mathbb{F}_q^3 \to \mathbb{F}_q^3$. The first two outputs of p_E can then be added to messages $P_1, P_2 \in \mathbb{F}_q$ to encrypt them, and the third output remains private. If one wishes to encrypt an additional message pair $P_3, P_4 \in \mathbb{F}_q$ one feeds the outputs of p_C after a key addition with (K_3, K_4) into the rolling function $rol : \mathbb{F}_q^3 \to \mathbb{F}_q^3$ and then again into p_E.

The permutations p_C and p_E are based on an iterated Feistel network

$$\mathcal{R}^{(i)} : \mathbb{F}_q^3 \to \mathbb{F}_q^3,$$
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 1 \\ 1 & c_4^{(i)} & c_4^{(i)} \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z + x \cdot y \end{pmatrix} + \begin{pmatrix} c_1^{(i)} \\ c_2^{(i)} \\ c_3^{(i)} \end{pmatrix}, \tag{2}$$

where $c_1^{(i)}, c_2^{(i)}, c_3^{(i)} \in \mathbb{F}_q$ and $c_4^{(i)} \in \mathbb{F}_q \setminus \{0, 1\}$. Note that the Feistel network $(x, y, z)^{\mathsf{T}} \mapsto (x, y, z + x \cdot y)^{\mathsf{T}}$ is also known as Toffoli gate which plays a special role in reversible computing [Tof80]. Finally, the rolling function is defined as

$$rol : \mathbb{F}_q^3 \to \mathbb{F}_q^3, \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} z + x \cdot y \\ x \\ y \end{pmatrix}. \tag{3}$$
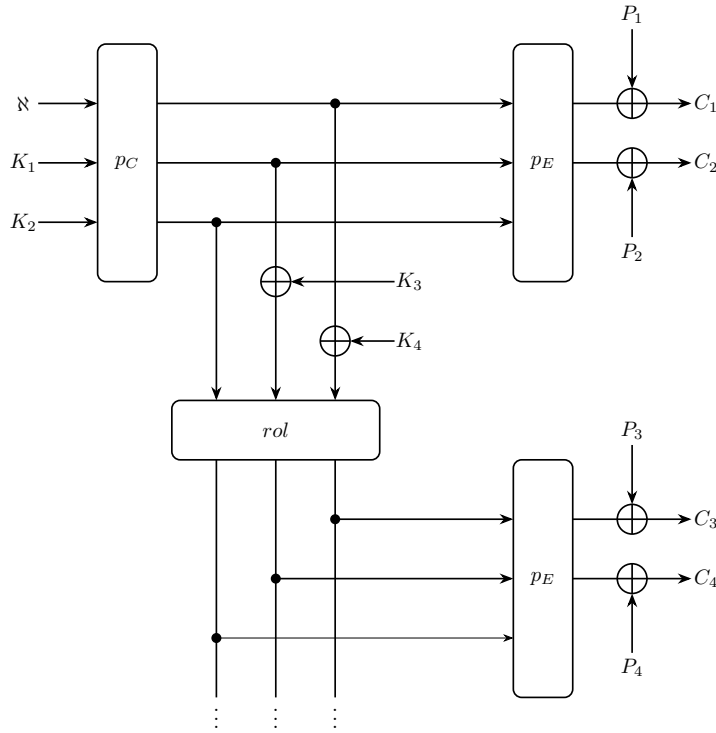
Figure 1: Encryption with `Ciminion`.

Formulas to compute the round numbers for `Ciminion` are given in [DGGK21, Table 1]. E.g., for $s = 128$ and data limit $2^{\frac{s}{2}}$ we have the round numbers $r_C = 90$ and $r_E = 14$.

It is now straight-forward to set up the iterated polynomial system for the first key pair (K_1, K_2) given the first plain/ciphertext pair. For ease of notation we abbreviate $r = r_C + r_E$.

Definition 2.1 (Iterated polynomial system for `Ciminion`)**.** *Let $\mathbb{F}_q$ be a finite field, and let $\left(\aleph, (p_1, p_2), (c_1, c_2)\right) \in \mathbb{F}_q \times \mathbb{F}_q^2 \times \mathbb{F}_q^2$ be a nonce/plain/ciphertext sample given by a `Ciminion` encryption function. We call the polynomial system $\mathcal{F}_{Ciminion} = \left\{\mathbf{f}^{(i)}\right\}_{1 \le i \le r} \subset$ $\mathbb{F}_q\left[y_1, y_2, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(r-1)}, x\right]$ the iterated `Ciminion` polynomial system*

$$\mathbf{f}^{(i)} = \begin{cases} \mathcal{R}^{(1)}\left(\aleph, y_1, y_2\right) - \mathbf{x}^{(1)}, & i = 1, \\ \mathcal{R}^{(i)}\left(\mathbf{x}^{(i-1)}\right) - \mathbf{x}^{(i)}, & 2 \le i \le r - 1, \\ \mathcal{R}^{(r)}\left(\mathbf{x}^{(r-1)}\right) - \begin{pmatrix} c_1 - p_1 \\ c_2 - p_2 \\ x \end{pmatrix}, & i = r. \end{cases}$$

2.2 Hydra

Let $\mathbb{F}_q$ be a finite field, the Hydra [GØSW23] PRF over $\mathbb{F}_q$ is also easily described via an illustration, see Figure 2. A nonce $\aleph \in \mathbb{F}_q$ and an initial value $\mathrm{IV} \in \mathbb{F}_q^3$ are added to a key $\mathrm{K} \in \mathbb{F}_q^4$, and then fed into the keyed *body* function $\mathcal{B} : \mathbb{F}_q^4 \times \mathbb{F}_q^4 \to \mathbb{F}_q^8$ which produces the state $\mathbf{y}||\mathbf{z} = (\mathbf{y}, \mathbf{z})^\mathsf{T} \in \mathbb{F}_q^4 \times \mathbb{F}_q^4$. Then $\mathbf{y}||\mathbf{z}$ is fed into the first keyed *head* function $\mathcal{H}_{\mathrm{K}} : \mathbb{F}_q^8 \times \mathbb{F}_q^4 \to \mathbb{F}_q^8$. After adding $\mathbf{y}||\mathbf{z}$ to the output of the head the result is released as the first Hydra sample. Say one has already requested $(i - 1)$ Hydra samples, for an additional sample $\mathbf{y}||\mathbf{z}$ is fed into the ith rolling function $\mathcal{R}_i : \mathbb{F}_q^8 \to \mathbb{F}_q^8$, and then again

into the head $\mathcal{H}_K$. After adding the output of $\mathcal{R}_i$ to the output of $\mathcal{H}_K$ the result is released as i^{th} Hydra sample.
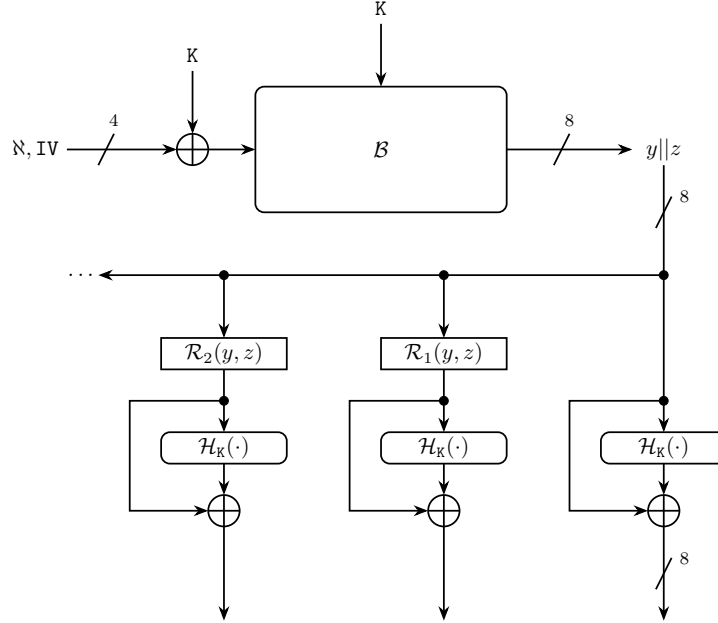

Figure 2: The Hydra PRF with hidden body.

For our polynomial model we are only going to consider the heads of the Hydra. Thus, we do not describe the details of the body function $\mathcal{B}$, interested readers can find the description in [GØSW23, §5.2]. Let $\mathbf{M}_\mathcal{E} \in \mathrm{GL}_4(\mathbb{F}_q)$ be an MDS matrix, e.g. the Hydra designers recommend $\mathrm{circ}(2, 3, 1, 1)$ or $\mathrm{circ}(3, 2, 1, 1)$.[3] For the keyed permutation $\mathcal{H}_K$, let

$$K' = \begin{pmatrix} K \\ \mathbf{M}_\mathcal{E}K \end{pmatrix} \in \mathbb{F}_q^8, \tag{4}$$

$$f(x_1, \ldots, x_n) = \left(\sum_{i=1}^{8} (-1)^{\lfloor \frac{i-1}{4} \rfloor} \cdot x_i \right)^2. \tag{5}$$

Moreover, let $\mathbf{M}_\mathcal{J} \in \mathrm{GL}_8(\mathbb{F}_q)$ be an invertible matrix. The heads round function is defined as

$$\begin{aligned} \mathcal{J}_i : \mathbb{F}_q^8 \times \mathbb{F}_q^4 &\to \mathbb{F}_q^8, \\ (\mathbf{x}, K) &\mapsto \mathbf{M}_\mathcal{J} \left(x_i + f(\mathbf{x}) \right)_{1 \le i \le 8} + K' + \mathbf{c}^{(i)}, \end{aligned} \tag{6}$$

where $\mathbf{c}^{(i)} \in \mathbb{F}_q^8$ is a constant. The heads function is now defined as the following $r_\mathcal{H}$-fold composition

$$\begin{aligned} \mathcal{H}_K : \mathbb{F}_q^4 \times \mathbb{F}_q^4 \times \mathbb{F}_q^4 &\to \mathbb{F}_q^8, \\ (\mathbf{y}, \mathbf{z}, K) &\mapsto \mathcal{J}_{r_\mathcal{H}} \circ \cdots \circ \mathcal{J}_1 \left(\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}, K \right). \end{aligned} \tag{7}$$

---

[3]We consider circulant matrices with respect to a right shift, i.e.

$$\mathrm{circ}(a_1, \ldots, a_n) = \begin{pmatrix} a_1 & a_2 & \ldots & a_{n-1} & a_n \\ a_n & a_1 & \ldots & a_{n-2} & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_2 & a_3 & \ldots & a_n & a_1 \end{pmatrix}.$$

Let $\mathbf{M}_{\mathcal{I}} \in \mathrm{GL}_4\left(\mathbb{F}_q\right)$ be an invertible matrix of the form

$$\mathbf{M}_{\mathcal{I}} = \begin{pmatrix} \mu_{1,1} & 1 & 1 & 1 \\ \mu_{2,1} & \mu_{2,2} & 1 & 1 \\ \mu_{3,1} & 1 & \mu_{3,3} & 1 \\ \mu_{4,1} & 1 & 1 & \mu_{4,4} \end{pmatrix}. \tag{8}$$

Note that the $\mu_{i,j} \in \mathbb{F}_q$ have to be chosen under additional constraints to prevent invariant subspaces [GØSW23, §5.2]. Finally, for the rolling function let

$$g(\mathbf{y}, \mathbf{z}) = \left(\sum_{i=1}^{4} (-1)^{i-1} \cdot y_i \right) \cdot \left(\sum_{i=1}^{4} (-1)^{\lfloor \frac{i-1}{2} \rfloor} \cdot z_i \right), \tag{9}$$

$$\mathbf{M}_{\mathcal{R}} = \mathrm{diag}\left(\mathbf{M}_{\mathcal{I}}, \mathbf{M}_{\mathcal{I}}\right). \tag{10}$$

Then, the rolling function is defined as

$$\mathcal{R} : \mathbb{F}_q^4 \times \mathbb{F}_q^4 \to \mathbb{F}_q^8,$$
$$\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \mapsto \mathbf{M}_{\mathcal{R}} \begin{pmatrix} \left(y_i + g(\mathbf{y}, \mathbf{z})\right)_{1 \le i \le 4} \\ \left(z_i + g(\mathbf{z}, \mathbf{y})\right)_{1 \le i \le 4} \end{pmatrix}, \tag{11}$$

and the ith rolling function is simply the composition $\mathcal{R}_i = \mathcal{R} \circ \mathcal{R}_{i-1} + \mathbf{c}_{\mathcal{R}}^{(i)}$, where $\mathbf{c}_{\mathcal{R}}^{(i)} \in \mathbb{F}_q$ is a random constant. Note that the degree increasing parts in $\mathcal{J}_i$ and $\mathcal{R}$ are generalized Lai–Massey permutations [GOPS22, RS24].

For κ bits of security, the number of rounds for the Hydra heads with data limit $2^{\frac{\kappa}{2}}$ are computed via, see [GØSW23, §5.5],

$$r_{\mathcal{H}} = \lceil 1.25 \cdot \max\left\{24, 2 + r_{\mathcal{H}}^*\right\} \rceil, \tag{12}$$

where $r_{\mathcal{H}}^*$ is the smallest integer that satisfies [GØSW23, Equation 12] and [GØSW22, Equation 15]. Note that [GØSW23, Equation 12] is derived from the Gröbner basis cryptanalysis of Hydra. E.g., for $q = 2^{127} + 45$ and $\kappa = 128$ the Hydra designers propose $r_{\mathcal{H}} = 39$.

Given two Hydra outputs it is straight-forward to set up an overdetermined polynomial system for Hydra. For ease of writing we denote the ith intermediate state variables for the jth output as $\mathbf{x}_j^{(i)} = \left(x_{j,1}^{(i)}, \ldots, x_{j,8}^{(i)}\right)^{\intercal}$ and analogously for $\mathbf{y} = (y_1, \ldots, y_4)^{\intercal}$, $\mathbf{z} = (z_1, \ldots, z_4)^{\intercal}$ and $\mathbf{k} = (k_1, \ldots, k_4)^{\intercal}$.

Definition 2.2 (Iterated polynomial system for Hydra). *Let $\mathbb{F}_q$ be a finite field, and let $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{F}_q^8$ be the first two outputs of a Hydra function. We call the following polynomial system $\mathcal{F}_{Hydra} = \left\{ \mathbf{f}_1^{(i)}, \mathbf{f}_{\mathcal{R}}, \mathbf{f}_2^{(i)} \right\}_{1 \le i \le r_{\mathcal{H}}} \subset \mathbb{F}_q\left[\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_1^{(r_{\mathcal{H}}-1)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_2^{(r_{\mathcal{H}}-1)}, \mathbf{k} \right]$ the iterated Hydra polynomial system*

$$\mathbf{f}_1^{(i)} = \begin{cases} \mathcal{J}_1\left(\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}, \mathbf{k}' \right) - \mathbf{x}_1^{(1)}, & i = 1, \\ \mathcal{J}_i\left(\mathbf{x}_1^{(i-1)}, \mathbf{k}' \right) - \mathbf{x}_1^{(i)}, & 2 \le i \le r_{\mathcal{H}} - 1, \\ \mathcal{J}_{r_{\mathcal{H}}}\left(\mathbf{x}_1^{(r_{\mathcal{H}}-1)}, \mathbf{k}' \right) + \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} - \mathbf{c}_1, & i = r_{\mathcal{H}}, \end{cases}$$

$$\mathbf{f}_{\mathcal{R}} = \mathcal{R}\left(\begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \right) + \mathbf{c}_{\mathcal{R}}^{(1)} - \mathbf{x}_2^{(0)},$$

$$\mathbf{f}_2^{(i)} = \begin{cases} \mathcal{J}_1\left(\mathbf{x}_2^{(0)}, \mathbf{k}'\right) - \mathbf{x}_2^{(1)}, & i = 1, \\ \mathcal{J}_i\left(\mathbf{x}_2^{(i-1)}, \mathbf{k}'\right) - \mathbf{x}_2^{(i)}, & 2 \leq i \leq r_{\mathcal{H}} - 1, \\ \mathcal{J}_{r_{\mathcal{H}}}\left(\mathbf{x}_2^{(r_{\mathcal{H}}-1)}, \mathbf{k}'\right) + \mathbf{x}_2^{(0)} - \mathbf{c}_2, & i = r_{\mathcal{H}}, \end{cases}$$

where $\mathbf{k}' = \begin{pmatrix} \mathbf{k} \\ \mathbf{M}_{\mathcal{E}}\mathbf{k} \end{pmatrix}.$

This polynomial system consists of $16 \cdot r_{\mathcal{H}} + 8$ many polynomials in $16 \cdot r_{\mathcal{H}} + 4$ many variables. It can also easily be extended to an arbitrary number of output samples.

2.3 Gröbner Bases

Gröbner bases are a fundamental tool in computational algebra. They were first introduced by Bruno Buchberger in his PhD thesis [Buc65]. For their definition one needs to induce an order on the monomials of a polynomial ring $P = K[x_1, \ldots, x_n]$. First notice that a monomial $m = \prod_{i=1}^{n} x_i^{a_i} \in P$ can be identified with its exponent vector $\mathbf{a} = (a_1, \ldots, a_n)^{\intercal} \in \mathbb{Z}_{\geq 0}^n$. Therefore, a term order $>$ on P is a binary relation on $\mathbb{Z}_{\geq 0}^n$ such that, see [CLO15, Chapter 2 §2 Definition 1]:

(i) $>$ is a total ordering in $\mathbb{Z}_{\geq 0}^n$.

(ii) If $\mathbf{a} > \mathbf{b}$ and $\mathbf{c} \in \mathbb{Z}_{\geq 0}^n$, then $\mathbf{a} + \mathbf{c} > \mathbf{b} + \mathbf{c}$.

(iii) Every non-empty subset of $\mathbb{Z}_{\geq 0}^n$ has a smallest element under $>$, i.e. $>$ is a well-ordering on $\mathbb{Z}_{\geq 0}^n$.

Now let $I \subset P = K[x_1, \ldots, x_n]$ be an ideal, and let $>$ be a term order on P. For $f \in P$ we denote with $\text{LM}_>(f)$ the leading monomial of f, i.e. the largest monomial with non-zero coefficient in f under $>$. A finite ideal basis $\mathcal{G} = \{g_1, \ldots, g_m\} \subset I$ such that

$$\left(\text{LM}_>(g_1), \ldots, \text{LM}_>(g_m)\right) = \left(\text{LM}_>(f) \mid f \in I\right) \tag{13}$$

is called a $>$-*Gröbner basis* of I. I.e., the leading monomials of $\mathcal{G}$ are minimal under $>$.

We abbreviate the ideal of leading terms of I as $\text{LM}_>(I) = \left(\text{LM}_>(f) \mid f \in I\right)$. Also, for ease of writing we often denote the variables of P in vector form $\mathbf{x} = (x_1, \ldots, x_n)^{\intercal}$. If not specified otherwise, then we always assume that the variables within a variable vector $\mathbf{x}$ are naturally ordered with respect to $>$, i.e. $x_1 > \ldots > x_n$.

With Gröbner bases one can solve many important computational problems like the membership problem or the computation of the set of zeros of an ideal. For a general introduction to Gröbner bases we refer to [KR00, KR05, CLO15].

In this paper we work with the lexicographic (LEX) and the degree reverse lexicographic (DRL) term order. For $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_{\geq 0}^n$, we have that $\mathbf{a} >_{LEX} \mathbf{b}$ if the first non-zero entry of $\mathbf{a} - \mathbf{b}$ is positive. We have that $\mathbf{a} >_{DRL} \mathbf{b}$ if either $\sum_{i=1}^{n} a_i > \sum_{i=1}^{n} b_i$ or $\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i$ and the last non-zero entry of $\mathbf{a} - \mathbf{b}$ is negative.

In more generality, a term order $>$ on P is called *degree compatible* if for any $f, g \in P$ the degree inequality $\deg(f) > \deg(g)$ implies that $f > g$. Obviously, the DRL term order is degree compatible.

2.3.1 Linear Algebra-Based Gröbner Basis Algorithms

Gröbner bases can be constructed via Gaussian elimination on the so-called *Macaulay matrices*. Let $\mathcal{F} \subset P = K[x_1, \ldots, x_n]$ be homogeneous, and let $>$ be a term order on P. The *homogeneous Macaulay matrix* M_d for $\mathcal{F}$ has columns indexed by the monomials of

degree d in P sorted via $>$, and its rows are indexed by $s \cdot f$, where $f \in \mathcal{F}$ and $s \in P$ is a monomial such that $\deg(s \cdot f) = d$. The entry of row $s \cdot f$ at column t is the coefficient of t in the polynomial $s \cdot f$. If $\mathcal{F}$ is inhomogeneous, then one replaces the degree equalities by inequalities to obtain the *inhomogeneous Macaulay matrix* $M_{\leq d}$. Obviously, if we perform Gaussian elimination on $M_0, \ldots, M_d$ (respectively $M_{\leq d}$) for d large enough, then we construct a $>$-Gröbner basis for $\mathcal{F}$. This idea can be traced back to Lazard [Laz83]. The least d which yields a Gröbner basis is of special interest.

Definition 2.3 (Solving degree, [CG21, Definition 6]). *Let $\mathcal{F} = \{f_1, \ldots, f_m\} \subset K[x_1, \ldots, x_n]$ and let $>$ be a term order. The solving degree of $\mathcal{F}$ is the least degree d such that Gaussian elimination on the Macaulay matrix $M_{\leq d}$ produces a Gröbner basis of $\mathcal{F}$ with respect to $>$. We denote it by $\mathrm{sd}_>(\mathcal{F})$.*

If $\mathcal{F}$ is homogeneous, then we consider the homogeneous Macaulay matrix M_d and let the solving degree of $\mathcal{F}$ be the least degree d such that Gaussian elimination on $M_0, \ldots, M_d$ produces a Gröbner basis of $\mathcal{F}$ with respect to $>$.

Algorithms which perform Gaussian elimination on the Macaulay matrices are generally referred to as *linear algebra-based Gröbner basis algorithms*. Modern examples of such algorithms are Faugère's F4 [Fau99] and Matrix-F5 [Fau02] algorithms. Since the solving degree is a priori unknown for most polynomial systems, F4/5 construct the Macaulay matrices for increasing values of d. Therefore, they require a stopping criterion to decide whether a Gröbner basis has already been found. On the other hand, we can always stop F4/5 after the final Gaussian elimination in the solving degree, since the row space basis must already contain a Gröbner basis. For this reason, we also consider the solving degree as complexity measure for advanced linear algebra-based algorithms.

2.3.2 Polynomial Systems in Generic Coordinates

Polynomial systems in generic coordinates have a proven DRL solving degree upper bound. For their definition we recall the *degree of regularity* of a polynomial system.

Definition 2.4 (Degree of regularity, [BFS04, Definition 4]). *Let K be a field, and let $\mathcal{F} \subset K[x_1, \ldots, x_n]$. If $\dim(\mathcal{F}^{\mathrm{top}}) = 0$, then the degree of regularity is defined as*

$$d_{\mathrm{reg}}(\mathcal{F}) = \min\left\{ d \geq 0 \; \middle| \; \dim_{\mathbb{F}_q}\left(\left\{f \in (\mathcal{F}^{\mathrm{top}}) \mid \deg(f) = d\right\}\right) = \binom{n+d-1}{d} \right\}.$$

If $\dim(\mathcal{F}^{\mathrm{top}}) > 0$, then $d_{\mathrm{reg}}(\mathcal{F}) = \infty$.

We present an equivalent definition for generic coordinates which requires slightly less knowledge about commutative algebra and projective geometry than the original definition from [CG21, CG22, Definition 5].

Definition 2.5 (cf. [CG21, CG22, Definition 5], [Ste24, Theorem 3.2]). *Let K be an infinite field, and let $\mathcal{F} \subset P = K[x_1, \ldots, x_n]$. We say that $(\mathcal{F}^{\mathrm{hom}}) \subset P[x_0]$ is in generic coordinates if either $d_{\mathrm{reg}}(\mathcal{F}) < \infty$ or $\sqrt{\mathcal{F}^{\mathrm{hom}}} = (x_0, \ldots, x_n)$.*

Let K be any field, and let $K \subset L$ be an infinite field extension. I is in generic coordinates over K if $I \otimes_K L[x_0, \ldots, x_n] \subset L[x_0, \ldots, x_n]$ is in generic coordinates.

Let $\mathcal{F} \subset K[x_1, \ldots, x_n]$ be an inhomogeneous polynomial system, to prove that $(\mathcal{F}^{\mathrm{hom}})$ is in generic coordinates one can use the following procedure based on [Ste24, Theorem 3.2]:

(1) Compute $(\mathcal{F}^{\mathrm{top}}) = (\mathcal{F}^{\mathrm{hom}}) \mod (x_0)$.

(2) Construct a monomial $x_i^d \in (\mathcal{F}^{\mathrm{top}})$ for some $1 \leq i \leq n$ and $d \geq 1$.

(3) Replace $(\mathcal{F}^{\mathrm{top}}) = (\mathcal{F}^{\mathrm{top}}) \mod (x_i)$ and return to Step (2).

If Step (2) is successful for all $1 \leq i \leq n$, then the polynomial system $\mathcal{F}^{\text{hom}}$ is in generic coordinates.

As already mentioned, polynomial systems in generic coordinates have a proven DRL solving degree upper bound.

Theorem 2.6 ([CG21, Theorem 9, 10, Corollary 2]). *Let K be an algebraically closed field, and let $\mathcal{F} = \{f_1, \ldots, f_m\} \subset K[x_1, \ldots, x_n]$ be an inhomogeneous polynomial system such that $(\mathcal{F}^{\text{hom}})$ is in generic coordinates. Let $d_i = \deg(f_i)$ and assume that $d_1 \geq \ldots \geq d_m$, then for $l \in \min\{n + 1, m\}$*

$$\text{sd}_{DRL}(\mathcal{F}) \leq d_1 + \ldots + d_l - l + 1.$$

In the Gröbner basis and algebraic cryptanalysis literature, the latter inequality is often referred to as the *Macaulay bound.*

2.3.3 Complexity Estimates via the Solving Degree

It is well-known that the complexity of Gaussian elimination of a matrix $\mathbf{A} \in K^{n \times m}$ with $r = \text{rank}(\mathbf{A})$ is given by $\mathcal{O}(n \cdot m \cdot r^{\omega - 2})$ field operations [Sto00, §2.2], where $2 \leq \omega < 2.371552$ is a linear algebra constant [WXXZ24].

Given an inhomogeneous polynomial system $\mathcal{F} \subset P = K[x_1, \ldots, x_n]$, it is well-known that the number of monomials in P of degree d is $\binom{n+d-1}{d}$. Thus, the number of columns of $M_{\leq d}$ is bounded by $\sum_{i=0}^{d} \binom{n+i-1}{i} = \binom{n+d}{d}$, and the number of rows by $|\mathcal{F}| \cdot \binom{n+d}{d}$. So, the complexity of Gaussian elimination on the inhomogeneous Macaulay matrices $M_{\leq d}$ is bounded by

$$\mathcal{O}\left(|\mathcal{F}| \cdot \binom{n+d}{d}^{\omega}\right). \tag{14}$$

2.4 Boolean Macaulay Matrices

Let $\mathcal{F} \subset \mathbb{F}_2[x_1, \ldots, x_n]$ be a polynomial system. If we are only interested in $\mathbb{F}_2$-valued solutions, then it is sufficient to compute a Gröbner basis of the ideal $(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$. On the other hand, via the field equations $F = (x_1^2 - x_1, \ldots, x_n^2 - x_n)$ we can reduce any monomial in the Macaulay matrix to a square-free monomial. Effectively, this reduces the full Macaulay matrix to a square-free submatrix, the so-called *Boolean Macaulay matrix.* Gaussian elimination for Boolean Macaulay matrices with respect to DRL was analyzed in [BFSS13] by Bardet et al.

Interestingly, for `Ciminion` and `Hydra` we will construct zero-dimensional DRL Gröbner bases in Sections 3.1 and 4.2 with leading monomials x_i^2. Hence, for Gröbner basis computations we can use these bases to reduce the Macaulay matrix to a "Boolean" one over an arbitrary finite field $\mathbb{F}_q$. This motivates the following definitions.

Definition 2.7. *Let K be a field, let $\mathcal{F} \subset P = K[x_1, \ldots, x_n]$, and let $>$ be a degree compatible term order on P. $\mathcal{F}$ is said to be a K-Boolean polynomial system if*

(i) for every $1 \leq i \leq n$ there exists $f \in \mathcal{F}$ such that $\text{LM}_>(f) = x_i^2$, and

(ii) $\mathcal{F}$ is a $>$-Gröbner basis.

Note that Condition (ii) is introduced so that the remainder of multivariate polynomial division does not depend on the ordering of the elements of $\mathcal{F}$, see [CLO15, Chapter 2 §6 Proposition 1].

Definition 2.8 (cf. [BFSS13, Definition 1]). *Let K be a field, let $\mathcal{F}_{Bool}, \mathcal{F} \subset P = K[x_1, \ldots, x_n]$ be polynomial systems, and let $>$ be a degree compatible term order on P. Assume that $\mathcal{F}_{Bool}$ is K-Boolean with respect to $>$. For the (inhomogeneous) K-Boolean Macaulay matrix $M_{\leq d}^{Bool}$ of $\mathcal{F} \cup \mathcal{F}_{Bool}$:*

- *Columns are indexed by square-free monomials $s \in P \setminus \mathrm{LM}_> (\mathcal{F}_{Bool})$ with $\deg(m) \leq d$ sorted via $>$.*

- *Rows are indexed by polynomials $t \cdot f$ where $t \in P \setminus \mathrm{LM}_> (\mathcal{F}_{Bool})$ is a square-free monomial and $f \in \mathcal{F}$ such that $\deg(t \cdot f) \leq d$.*

- *In the row $t \cdot f$ one writes the coefficients of the remainder $r_{t \cdot f} = t \cdot f \mod (\mathcal{F}_{Bool})$ with respect to $>$.*

Note that any monomial $t \in P$ can be reduced to a square-free polynomial $\hat{t} \equiv t \mod (\mathcal{F}_{\mathrm{Bool}})$ with $\deg(\hat{t}) \leq \deg(t)$ due to degree compatibility of $>$. Since remainders with respect to a Gröbner basis are unique, see [CLO15, Chapter 2 §6 Proposition 1], for the remainder of $t \cdot f$ it does not matter whether we reduce t first and then $\hat{t} \cdot f$ or $t \cdot f$ at once. So it is indeed sufficient to consider only products with square-free monomials.

Obviously, the remainder $r_{t \cdot f} = t \cdot f \mod (\mathcal{F}_{\mathrm{Bool}})$ is a polynomial consisting of only square-free terms. Moreover, by elementary properties of multivariate polynomial division with respect to a degree compatible term order, see [CLO15, Chapter 2 §3 Theorem 3], we have that

$$t \cdot f = \sum_{i=1}^{m} g_i \cdot f_i^{\mathrm{Bool}} + r_{t \cdot f}, \tag{15}$$

where $\deg\left(g_i \cdot f_i^{\mathrm{Bool}}\right), \deg(r_{t \cdot f}) \leq \deg(t \cdot f)$. I.e., the K-Boolean Macaulay matrix can be constructed via row operations from the Macaulay matrix $M_{\leq d}$ of $\mathcal{F} \cup \mathcal{F}_{\mathrm{Bool}}$. Alternatively, we can obtain the K-Boolean Macaulay matrix by pivoting non-square-free monomials in $M_{\leq d}$ via rows coming from $\mathcal{F}_{\mathrm{Bool}}$, and then performing an auxiliary Gaussian elimination step that stops after eliminating all non-square-free monomials in the rows coming from $\mathcal{F}$.

Let $M_{\leq d}$ be a Macaulay matrix, we denote with $\mathrm{rowsp}(M_{\leq d})$ the row space basis, i.e. the polynomial system obtained via Gaussian elimination on $M_{\leq d}$. Analogously, we use this notation for Boolean Macaulay matrices.

Definition 2.9. *Let K be a field, let $\mathcal{F}_{Bool}, \mathcal{F} \subset P = K[x_1, \ldots, x_n]$ be polynomial systems, and let $>$ be a degree compatible term order on P. Assume that $\mathcal{F}_{Bool}$ is K-Boolean with respect to $>$. The K-Boolean solving degree of $\mathcal{F} \cup \mathcal{F}_{Bool}$ is the least degree d such that $\mathrm{rowsp}\left(M_{\leq d}^{Bool}\right) \cup \mathcal{F}_{Bool}$ contains a Gröbner basis with respect to $>$. We denote it by $\mathrm{sd}_>^{Bool}(\mathcal{F} \cup \mathcal{F}_{Bool})$.*

Note that $\mathrm{rowsp}\left(M_{\leq d}^{\mathrm{Bool}}\right)$ might not contain a zero-dimensional Gröbner basis of the full system. A trivial example is $\mathcal{F} = \left\{x_1^2, x_1 \cdot x_2, x_2^2\right\}$. Therefore, to extract the Gröbner basis from $\mathrm{rowsp}\left(M_{\leq d}^{\mathrm{Bool}}\right) \cup \mathcal{F}_{\mathrm{Bool}}$ we set up the ideal of $>$-leading terms and then compute its minimal generating set. With the minimal generating set we can then filter the $>$-Gröbner basis.[4]

The next theorem follows straight-forward from $M_{\leq d}^{\mathrm{Bool}} \subset M_{\leq d}$ (as K-vector spaces).

Theorem 2.10. *Let K be a field, let $\mathcal{F}_{Bool}, \mathcal{F} \subset P = K[x_1, \ldots, x_n]$ be polynomial systems, and let $>$ be a degree compatible term order on P. Let $D = \min_{f \in \mathcal{F}} \deg(f)$, and let $d = \mathrm{sd}_>^{Bool}(\mathcal{F} \cup \mathcal{F}_{Bool})$. Assume that $\mathcal{F}_{Bool}$ is K-Boolean with respect to $>$. Then*

(1) $\mathrm{sd}_>^{Bool}(\mathcal{F} \cup \mathcal{F}_{Bool}) \leq \mathrm{sd}_>(\mathcal{F} \cup \mathcal{F}_{Bool})$.

(2) If $d - D < n$, then Gaussian elimination on $M_{\leq d}^{Bool}$ requires

$$\mathcal{O}\left(|\mathcal{F}| \cdot \left(\sum_{i=0}^{d-D} \binom{n}{i}\right) \cdot \left(\sum_{i=0}^{d} \binom{n}{i}\right)^{\omega-1}\right)$$

---

[4]It is obvious that all leading terms of $\mathrm{rowsp}\left(M_{\leq d}\right)$ not coming from $\mathrm{rowsp}\left(M_{\leq d}^{\mathrm{Bool}}\right)$ are already contained in $\mathrm{LM}_> (\mathcal{F}_{\mathrm{Bool}})$. Since $\mathcal{F}_{\mathrm{Bool}}$ is a $>$-Gröbner basis it is indeed sufficient to just consider $\mathrm{rowsp}\left(M_{\leq d}^{\mathrm{Bool}}\right) \cup \mathcal{F}_{\mathrm{Bool}}$.

field operations.

(3) If $d - D \geq n$, then Gaussian elimination on $M_{\leq d}^{Bool}$ requires at most

$$\mathcal{O}\left(|\mathcal{F}| \cdot 2^{\omega \cdot n}\right)$$

field operations.

Proof. For (1), in $M_{\leq d}$ we pivot non-square-free columns via rows coming from $\mathcal{F}_{\text{Bool}}$, then we perform the auxiliary Gaussian elimination step that produces $M_{\leq d}^{\text{Bool}}$ as submatrix, and finally we perform Gaussian elimination on $M_{\leq d}^{\text{Bool}}$. Since Gaussian elimination does not depend on the order of reductions $\text{rowsp}\left(M_{\leq d}^{\text{Bool}}\right) \cup \mathcal{F}_{\text{Bool}}$ has to contain the $>$-Gröbner basis.

For (2), it is well-known that in degree i there exist $\binom{n}{i}$ square-free monomials in P. The polynomial $f \in \mathcal{F}$ contributes $\sum_{i=0}^{d-\deg(f)} \binom{n}{i}$ many rows to the K-Boolean Macaulay matrix. Then, the total number of rows in $M_{\leq d}^{\text{Bool}}$ is bounded by

$$\sum_{f \in \mathcal{F}} \sum_{i=0}^{d-\deg(f)} \binom{n}{i} \leq \sum_{f \in \mathcal{F}} \sum_{i=0}^{d-D} \binom{n}{i} = |F| \cdot \sum_{i=0}^{d-D} \binom{n}{i}.$$

The number of columns is bounded by $\sum_{i=0}^{d} \binom{n}{i}$, so we also have that $\text{rank}\left(M_{\leq d}^{\text{Bool}}\right) \leq \sum_{i=0}^{d} \binom{n}{i}$, and the claim follows from the complexity of Gaussian elimination [Sto00, §2.2].

For (3), if $d - D \geq n$, then there exists at least one polynomial $f \in \mathcal{F}$ which has to be multiplied with any square-free monomial $s \in P$. So, f contributes $\sum_{i=0}^{n} \binom{n}{i} = 2^n$ many rows. Obviously, any other $g \in \mathcal{F}$ contributes at most $\leq 2^n$ many rows. Hence, the total number of rows is bounded by $|\mathcal{F}| \cdot 2^n$, and the number of columns as well as the rank are bounded by 2^n. Again, the claim follows by [Sto00, §2.2]. □

Remark 2.11. This theorem can be refined by replacing the binomial coefficients $\binom{n}{i}$ with the number of monomials in $P \setminus \text{LM}_>\left(\mathcal{F}_{\text{Bool}}\right)$ of degree i.

So far our analysis does not differ from the one over $\mathbb{F}_2$, however the binary field has a special feature. Division with respect to the field equations $\left(x_1^2 - x_1, \ldots, x_n^2 - x_n\right)$ is "for free", i.e. its complexity is negligible compared to Gaussian elimination. For general K-Boolean polynomial systems this is not necessarily the case. The complexity of general polynomial division with respect to quadratic K-Boolean polynomial systems follows as corollary to [Ten19, Proposition 3.10].

Corollary 2.12. *Let K be a field, let $\mathcal{F}_{Bool} \subset P = K[x_1, \ldots, x_n]$ be a quadratic polynomial system, and let $>$ be a degree compatible term order on P. Assume that $\mathcal{F}_{Bool}$ is K-Boolean with respect to $>$.*

(1) Let $f \in P$, then computation of $r_f = f \mod (\mathcal{F}_{Bool})$ with respect to $>$ requires

$$\mathcal{O}\left(\frac{n^2 + 3 \cdot n + 2}{2} \cdot \binom{n + \deg(f)}{\deg(f)}\right)$$

field operations.

(2) Let $\mathcal{F} \subset P$, $D_{\min} = \min_{f \in \mathcal{F}} \deg(f)$ and $D_{\max} = \max_{f \in \mathcal{F}} \deg(f)$, then construction of $M_{\leq d}^{Bool}$ for $\mathcal{F} \cup \mathcal{F}_{Bool}$ requires

$$\mathcal{O}\left(|\mathcal{F}| \cdot \frac{n^2 + 3 \cdot n + 2}{2} \cdot \sum_{i=0}^{d-D_{\min}} \binom{n}{i} \cdot \binom{n + D_{\max} + i}{D_{\max} + i}\right)$$

field operations.

Proof. For (1), in P there exist $\sum_{i=0}^{\deg(f)} \binom{n+i-1}{i} = \binom{n+\deg(f)}{\deg(f)}$ monomials of degree $\leq \deg(f)$ and $\binom{n+1}{2} + \binom{n}{1} + 1$ monomials of degree ≤ 2. Now the claim follows from [Ten19, Proposition 3.10].

For (2), with (1) we can estimate the complexity of the construction as

$$
\sum_{f \in \mathcal{F}} \sum_{i=0}^{d-\deg(f)} \binom{n}{i} \cdot \frac{n^2 + 3 \cdot n + 2}{2} \cdot \binom{n + \deg(f) + i}{\deg(f) + i}
$$

$$
\overset{(*)}{\leq} \frac{n^2 + 3 \cdot n + 2}{2} \cdot \sum_{f \in \mathcal{F}} \sum_{i=0}^{d-D_{\min}} \binom{n}{i} \cdot \binom{n + D_{\max} + i}{D_{\max} + i}
$$

$$
= |\mathcal{F}| \cdot \frac{n^2 + 3 \cdot n + 2}{2} \cdot \sum_{i=0}^{d-D_{\min}} \binom{n}{i} \cdot \binom{n + D_{\max} + i}{D_{\max} + i},
$$

where $(*)$ follows from $d - \deg(f) \leq d - D_{\min}$ and the fact that $\binom{n+d}{d}$ is strictly increasing in d, hence $\binom{n+\deg(f)+i}{\deg(f)+i} \leq \binom{n+D_{\max}+i}{D_{\max}+i}$. $\qquad\square$

Of course, the K-Boolean Macaulay matrix generalizes to more advanced linear algebra-based Gröbner basis algorithms like F4/5 [Fau99, Fau02].

2.5   The Eigenvalue Method

Now we discuss how the zeros of a polynomial system can be found via linear algebra once a zero-dimensional DRL Gröbner basis of the input system is known [KR16, Chapter 6]. Let $I \subset P = K[x_1, \dots, x_n]$ be a zero-dimensional ideal, i.e. P/I is a finite dimensional K-vector space [KR00, Proposition 3.7.1], and let $d = \dim_K (P/I)$. If $>$ is a term order on P and $\mathcal{G} \subset I$ is $>$-Gröbner basis, then a K-vector space basis $\mathcal{B}$ of P/I is given by the monomials not contained in $\mathrm{LM}_> (\mathcal{G}) = \mathrm{LM}_> (I)$. Now let $f \in P$, then the multiplication map

$$
\theta_f : P/I \to P/I, \qquad x \mapsto x \cdot f, \tag{16}
$$

is K-linear, so it can be represented by a matrix. This matrix is called the multiplication matrix $\mathbf{M}_f$ for f. Given a $>$-Gröbner basis its computation is straight-forward: Index the columns of $\mathbf{M}_f$ by the K-vector space basis $\mathcal{B}$, and the rows by $b \cdot f$, where $b \in \mathcal{B}$. The row $b \cdot f$ is then simply the coefficient vector of $b \cdot f$ in P/I, which can be computed via $b \cdot f \mod (\mathcal{G})$ with respect to $>$.

Now assume in addition that K is algebraically closed, then for every point in the variety $\mathbf{x} \in \mathcal{V}(I) = \{\mathbf{x} \in K^n \mid \forall g \in I : g(\mathbf{x}) = 0\}$ the point $f(\mathbf{x})$ is an eigenvalue of $\mathbf{M}_f$, see [KR16, Proposition 6.2.1]. Therefore, we can compute the variety $\mathcal{V}(I)$ as follows: Compute the eigenvalues of $\mathbf{M}_{x_1}, \dots, \mathbf{M}_{x_n}$, iterate over all possible combinations, and verify whether a combination is indeed a solution to I. This approach is known as *Eigenvalue Method* [KR16, §6.2.A].

It is well-known that the eigenvalues of a matrix $\mathbf{M} \in K^{N \times N}$ are given by the roots of its characteristic polynomial

$$
\chi_{\mathbf{M}}(x) = \det (\mathbf{I}_{N \times N} \cdot x - \mathbf{M}) \in K[x]. \tag{17}
$$

So the complexity of the Eigenvalue Method boils down to n determinant computations with polynomial entries.

2.5.1 Solving Structured Polynomial Systems Over Finite Fields

Let $\mathcal{F} = \{f_1, \ldots, f_n\} \subset P = K[x_1, \ldots, x_n]$ be such that $\mathrm{LM}_{DRL}(f_i) = x_i^{d_i}$ for all $1 \leq i \leq n$. Then, trivially $\mathcal{F}$ is a DRL Gröbner basis, see e.g. [CLO15, Chapter 2 §9 Theorem 3, Proposition 4] with $\dim_K(\mathcal{F}) = \prod_{i=1}^{n} d_i$. Let $D = \prod_{i=1}^{n-1} d_i$, and let $\mathcal{B}' \subset K[x_1, \ldots, x_{n-1}]/\left(x_1^{d_1}, \ldots, x_{n-1}^{d_{n-1}}\right)$ be some K-vector space basis, then $\mathcal{B} = \bigcup_{i=0}^{d_n-1} x_n^i \cdot \mathcal{B}'$ is a K-vector space basis for $P/(\mathcal{F})$. In particular, the multiplication matrix $\mathbf{M}_{x_n}$ is then of the form

$$
\mathbf{M}_{x_n} = \begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{I} \\ \mathbf{A}_0 & \mathbf{A}_1 & \ldots & \ldots & \mathbf{A}_{d_n-1} \end{pmatrix} \begin{matrix} x_n \cdot \mathcal{B}' \\ x_n^2 \cdot \mathcal{B}' \\ \vdots \\ x_n^{d_n-1} \cdot \mathcal{B}' \\ x_n^{d_n} \cdot \mathcal{B}' \end{matrix} , \tag{18}
$$

where $\mathbf{0} \in K^{D \times D}$ denotes the zero matrix, $\mathbf{I} \in K^{D \times D}$ the identity matrix, and $\mathbf{A}_0, \ldots, \mathbf{A}_{d_n-1} \in K^{D \times D}$ some matrices. By [BBL$^+$24, Lemma 2] we then have that

$$
\chi_{\mathbf{M}_{x_n}}(x) = \det\left(x_n^{d_n} \cdot \mathbf{I} - \sum_{i=0}^{d_n-1} x_n^i \cdot \mathbf{A}_i \right), \tag{19}
$$

and computation of this determinant requires [BBL$^+$24, §3.2]

$$
\mathcal{O}\left(d_n \cdot \log_2(d_n)^2 \cdot \left(1 + \log_2\left(\log_2(d_n)\right)\right) \cdot \left(\frac{D}{d_n}\right)^{\omega} \right) \tag{20}
$$

field operations, where $2 \leq \omega < 2.371552$ is again the linear algebra constant [WXXZ24].

In this paper we solely work over finite fields $\mathbb{F}_q$, and we are only interested in the $\mathbb{F}_q$-valued roots of $\mathcal{F}$. Given a univariate polynomial $f \in \mathbb{F}_q$ with $\deg(f) = d$ we can always achieve this restriction by passing to the GCD with the field equation $g = \gcd(f, x^q - x)$. By [BBLP22, §3.1] computation of this GCD requires

$$
\mathcal{O}\left(\begin{cases} d \cdot \log_2(d) \cdot \log\left(\log(d)\right) \cdot \left(\log(d) + \log_2(q)\right), & d < q, \\ d \cdot \log(d)^2 \cdot \log\left(\log(d)\right), & d \geq q \end{cases} \right) \tag{21}
$$

field operations. Provided that only a few roots of f come from $\mathbb{F}_q$ the final factoring step can be neglected.

In this paper we will encounter quadratic DRL Gröbner bases $\mathcal{F} = \{f_1, \ldots, f_n\} \subset P = \mathbb{F}_q[x_1, \ldots, x_n]$ of the form

$$
f_i = \mathcal{Q}(x_i, \ldots, x_n) + \mathcal{A}_i(x_1, \ldots, x_n), \tag{22}
$$

where $\mathcal{Q}_i \in P$ is a quadratic polynomial which has the term x_i^2 and $\mathcal{A}_i \in P$ is an affine polynomial which has the term x_{n-i+1} for $1 \leq i \leq n$. It is easy to see that $\mathcal{F}$ is a DRL Gröbner basis for $x_1 > \ldots > x_n$. For such systems we can give an optimized Eigenvalue Method to find the $\mathbb{F}_q$-valued solutions. We use a $\mathbb{F}_q$-vector space basis such that the multiplication matrix $\mathbf{M}_{x_n}$ has the shape from Equation (18). Then, the characteristic polynomial is constructed in $\mathcal{O}\left(2^{\omega \cdot (n-1)+1}\right)$ field operations, and its $\mathbb{F}_q$-valued roots are extracted via the GCD with the field equation. Let $\alpha \in \mathbb{F}_q$ be such a solution for x_n, then we can substitute $x_n = \alpha$ back into $\mathcal{F}$. In particular, from f_n we obtain the affine equation

$$
\alpha^2 + \tilde{\mathcal{A}}_n = 0. \tag{23}
$$

Now we rearrange this equation to have x_1 on the right-hand side. Then, we can additionally eliminate the variable x_1 in the equations $f_1 = \ldots = f_{n-1} = 0$. Since x_1 is not present in the quadratic terms of $f_2, \ldots, f_{n-1} \in \mathbb{F}_q[x_2, \ldots, x_{n-1}]$, they are again a zero-dimensional DRL Gröbner basis. The polynomial f_1 is an additional overdetermined polynomial which we ignore. Now we iterate this procedure to solve for $x_{n-1}, x_{n-2}, \ldots, x_{\lfloor \frac{n}{2} \rfloor}$, then we lift to a global solution guess for $x_1, \ldots, x_n$, and finally verify it on $\mathcal{F}$. Let $\mathcal{D}(d, q)$ denote the complexity of the GCD between a degree d polynomial and the field equation of $\mathbb{F}_q$, moreover let N be a bound on the number of $\mathbb{F}_q$-valued solutions at each iteration in our solving algorithm. Then, this dedicated Eigenvalue Method requires

$$\mathcal{O}\left(\left\{ \begin{array}{ll} \displaystyle\sum_{i=1}^{\frac{n}{2}} N^{\frac{n}{2}-i} \cdot \left(2^{\omega \cdot (2 \cdot i - 1) + 1} + \mathcal{D}\left(2^{2 \cdot i}, q\right) \right), & 2 \mid n, \\[4mm] \displaystyle\sum_{i=1}^{\frac{n-1}{2}} N^{\frac{n-1}{2}-i} \cdot \left(2^{\omega \cdot 2 \cdot i + 1} + \mathcal{D}\left(2^{2 \cdot i + 1}, q\right) \right), & 2 \nmid n \end{array} \right\} \right) \tag{24}$$

$$= \mathcal{O}\left(\left\{ \begin{array}{ll} 2^{\omega + 1} \cdot \dfrac{2^{\omega \cdot n} - N^{\frac{n}{2}}}{2^{2 \cdot \omega} - N} + \displaystyle\sum_{i=1}^{\frac{n}{2}} N^{\frac{n}{2}-i} \cdot \mathcal{D}\left(2^{2 \cdot i}, q\right), & 2 \mid n, \\[4mm] 2^{2 \cdot \omega + 1} \cdot \dfrac{2^{\omega \cdot (n-1)} - N^{\frac{n-1}{2}}}{2^{2 \cdot \omega} - N} + \displaystyle\sum_{i=1}^{\frac{n-1}{2}} N^{\frac{n-1}{2}-i} \cdot \mathcal{D}\left(2^{2 \cdot i + 1}, q\right), & 2 \nmid n \end{array} \right\} \right) \tag{25}$$

field operations.

3   Analysis of `Ciminion`

At round level `Ciminion` is a three branch Feistel network. In [Ste24, §4.1] a DRL Gröbner basis for Feistel-$2n/n$ polynomial systems has been computed by substituting the linear branches into the non-linear ones. In principle, we are going to replicate this strategy to construct a DRL Gröbner basis for `Ciminion`.

3.1   A DRL Gröbner Basis for `Ciminion`

For simplicity let us denote the matrix in the ith `Ciminion` round by $\mathbf{A}_i \in \mathbb{F}_q^{3 \times 3}$, see Equation (2), then

$$\mathbf{A}_i = \begin{pmatrix} 0 & 0 & 1 \\ 1 & c_4^{(i)} & c_4^{(i)} \\ 0 & 1 & 1 \end{pmatrix}, \qquad \mathbf{A}_i^{-1} = \begin{pmatrix} 0 & 1 & -c_4^{(i)} \\ -1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \tag{26}$$

where $c_4^{(i)} \in \mathbb{F}_q \setminus \{0, 1\}$.

Theorem 3.1. *Let $\mathbb{F}_q$ be a finite field, and let $\mathcal{F}_{\text{Ciminion}} = \left\{ \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(r)} \right\} \subset \mathbb{F}_q\left[y_1, y_2, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r-1)}, x\right]$ be a `Ciminion` polynomial system. For the DRL term order $y_1 > y_2 > \mathbf{x}^{(1)} > \ldots > \mathbf{x}^{(r-1)} > x$, the following procedure computes a DRL Gröbner basis for `Ciminion`:*

(1) For every round $1 \leq i \leq r$ compute $\mathbf{g}^{(i)} = \mathbf{A}_i^{-1} \mathbf{f}_i$.

(2) Collect the linear polynomials from the $\mathbf{g}^{(i)}$'s in $\mathcal{L}$, and perform Gaussian elimination on $\mathcal{L}$.

(3) Reduce the non-linear polynomials in the $\mathbf{g}^{(i)}$'s modulo $\mathcal{L}$.

Proof. By inspection of the round function (Equation (2)) we have that

$$\mathbf{g}^{(1)} = \begin{pmatrix} \aleph \\ y_1 \\ \aleph \cdot y_1 + y_2 \end{pmatrix} + \mathbf{A}_1^{-1}\left(\mathbf{c}^{(1)} - \mathbf{x}^{(1)}\right), \tag{27}$$

$$\mathbf{g}^{(i)} = \begin{pmatrix} x_1^{(i-1)} \\ x_2^{(i-1)} \\ x_3^{(i-1)} + x_1^{(i-1)} \cdot x_2^{(i-1)} \end{pmatrix} + \mathbf{A}_i^{-1}\left(\mathbf{c}^{(i)} - \mathbf{x}^{(i)}\right), \tag{28}$$

$$\mathbf{g}^{(r)} = \begin{pmatrix} x_1^{(r-1)} \\ x_2^{(r-1)} \\ x_3^{(r-1)} + x_1^{(r-1)} \cdot x_2^{(r-1)} \end{pmatrix} + \mathbf{A}_r^{-1}\left(\mathbf{c}^{(r)} - \begin{pmatrix} c_1 - p_1 \\ c_2 - p_2 \\ x \end{pmatrix}\right). \tag{29}$$

Hence, $\mathcal{L}$ contains $\mathbf{g}^{(1)}$ and the first two components of all remaining $\mathbf{g}^{(i)}$'s.

For Gaussian elimination on $\mathcal{L}$ let us start in the last round $i = r$. By Equations (26) and (29) the linear polynomials with leading monomials $x_1^{(r-1)}$, $x_2^{(r-1)}$ contain a linear term in x and a constant term. Moving to $i = r - 1$ next the linear polynomials from $\mathbf{g}^{(r-1)}$ depend on $\mathbf{x}^{(r-2)}$ and $\mathbf{x}^{(r-1)}$, and via Gaussian elimination the variables $x_1^{(r-1)}$ and $x_2^{(r-1)}$ get eliminated. Thus, the linear polynomials with leading monomials in $x_1^{(r-2)}$ and $x_2^{(r-2)}$ can only depend on $x_3^{(r-1)}$ and x. Moreover, by Equation (26) the coefficients for $x_3^{(r-1)}$ are non-zero. Via a downwards induction we can then conclude for $2 \leq i \leq r - 1$ that after Gaussian elimination

$$\text{LM}_{DRL}\left(g_1^{(i)}\right) = x_1^{(i-1)},$$
$$\text{LM}_{DRL}\left(g_2^{(i)}\right) = x_2^{(i-1)},$$
$$g_1^{(i)}, g_2^{(i)} \in \mathbb{F}_q\left[x_3^{(i)}, \ldots, x_3^{(r-1)}, x\right],$$

and that the coefficients of $x_3^{(i)}$ in $g_1^{(i)}$ and $g_2^{(i)}$ are non-zero.

The first round needs special care since the first row of $\mathbf{A}_1^{-1}$ is non-zero on the variable $x_2^{(1)}$. (If it were zero on $x_1^{(1)}$ and $x_2^{(1)}$, the first polynomial in $\mathbf{g}^{(1)}$ would have leading monomial $x_3^{(1)}$, and we could proceed.) Expansion of Equation (27) yields

$$\mathbf{g}^{(1)} = \begin{pmatrix} \aleph \\ y_1 \\ \aleph \cdot y_1 + y_2 \end{pmatrix} + \mathbf{A}_1^{-1}\mathbf{c}^{(1)} - \begin{pmatrix} x_2^{(1)} - c_4^{(1)} \cdot x_3^{(1)} \\ -x_1^{(1)} + x_3^{(1)} \\ x_1^{(1)} \end{pmatrix}.$$

Producing a polynomial with leading monomial y_2 is straight-forward, we simply have to take $g_3^{(1)} - \aleph \cdot g_2^{(1)}$. On the other hand, there is second polynomial with leading monomial $x_2^{(1)}$ after inversion of the matrices

$$g_2^{(2)} = x_2^{(1)} + \left(\mathbf{A}_2^{-1}\mathbf{c}^{(2)}\right)_2 - \left(-x_1^{(2)} + x_3^{(2)}\right).$$

Then

$$g_1^{(1)} + g_2^{(2)} = \aleph + \left(\mathbf{A}_1^{-1}\mathbf{c}^{(1)}\right)_1 + c_4^{(1)} \cdot x_3^{(1)} + \left(\mathbf{A}_2^{-1}\mathbf{c}^{(2)}\right)_2 + x_1^{(2)} - x_3^{(2)},$$

and by the choice of our DRL term order and $c_4^{(1)} \neq 0$ this polynomial has leading monomial $x_3^{(1)}$. Obviously, after Gaussian elimination of $\mathcal{L}$ the terms of $g_1^{(1)} + g_2^{(2)}, g_2^{(1)}, g_3^{(1)} - \aleph \cdot g_2^{(1)}$ after the leading monomials will also only depend on the variables $x_3^{(2)}, \ldots, x_3^{(r-1)}, x$.

Lastly, let us consider the reduction modulo $\mathcal{L}$. By our previous observations all linear equations are of the form

$$y_1 + \mathcal{A}_1 \left(x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right) = 0,$$
$$y_2 + \mathcal{A}_2 \left(x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right) = 0,$$
$$x_3^{(1)} + \mathcal{A}_3 \left(x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right) = 0,$$
$$x_1^{(i-1)} + c_4^{(i)} \cdot x_3^{(i)} + \ldots = 0,$$
$$x_2^{(i-1)} - x_3^{(i)} + \ldots = 0,$$
$$x_1^{(r-1)} + c_4^{(r)} \cdot x + \ldots = 0,$$
$$x_2^{(r-1)} - x + \ldots = 0,$$

where $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ are affine polynomials and $2 \leq i \leq r - 1$. So, for $2 \leq i \leq r$ we can directly conclude from Equations (28) and (29) that

$$\hat{g}_3^{(i)} = \left(g_3^{(i)} \mod (\mathcal{L}) \right) \in \mathbb{F}_q \left[x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right],$$

has leading monomial $x_3^{(i)^2}$ and respectively x^2 for the last round.

Thus, we have constructed a polynomial system with pairwise coprime leading monomials, so by [CLO15, Chapter 2 §9 Theorem 3, Proposition 4] we have constructed a Gröbner basis. □

Note that if we only keep the $r - 1$ quadratic polynomials of $\mathcal{G}$, then we still have a fully determined polynomial system in $r - 1$ variables. Therefore, we call the quadratic polynomials of $\mathcal{G}$ the downsized `Ciminion` polynomial system.

For the next corollary, recall that a LEX Gröbner basis $\mathcal{F} = \{ f_1, \ldots, f_n \} \subset K[x_1, \ldots, x_n]$ is said to be in x_n-*shape position* if it is of the form

$$f_i = x_i - \tilde{f}_n(x_n), \qquad 1 \leq i \leq n - 1, \tag{30}$$
$$f_n = \tilde{f}_n(x_n) \tag{31}$$

It is well-known that the LEX Gröbner basis of a radical ideal can be brought into shape position for some ordering of the variables, see [KR00, Theore 3.7.25].

Corollary 3.2. *In the situation of Theorem 3.1, let* $\mathcal{G}_{Ciminion}$ *denote the* `Ciminion` *DRL Gröbner basis.*

(1) $\dim_{\mathbb{F}_q} \left(\mathcal{F}_{Ciminion} \right) = 2^{r-1} = 2^{r_C + r_E - 1}$.

(2) *Let* $I \subset \overline{\mathbb{F}}_q \left[y_1, y_2, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r-1)}, x \right] [x_0]$ *be a homogeneous ideal such that* $\mathcal{Z}_+(I) \neq \emptyset$ *and* $\mathcal{G}_{Ciminion}^{hom} \subset I$. *Then* I *is in generic coordinates.*

(3) $\mathrm{sd}_{DRL} \left(\mathcal{G}_{Ciminion} \right) \leq r = r_C + r_E$.

(4) *The LEX Gröbner basis for* `Ciminion` *under* $y_1 > y_2 > \mathbf{x}^{(1)} > \ldots > \mathbf{x}^{(r-1)} > x$ *is in* x-*shape position.*

Proof. (1), (2) and (3) are consequences of the DRL Gröbner basis from Theorem 3.1 and application of [Ste24, Theorem 3.2] and Theorem 2.6. For the last claim we take a look at the shape of the quadratic polynomials in the DRL Gröbner basis

$$g_3^{(2)} = \mathcal{Q}_2 \left(x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right) + \mathcal{A}_2 \left(x_3^{(2)}, \ldots, x_3^{(r-1)}, x \right),$$

$$g_3^{(i)} = \mathcal{Q}_i\left(x_3^{(i)}, \ldots, x_3^{(r-1)}, x\right) + \mathcal{A}_i\left(x_3^{(i-1)}, \ldots, x_3^{(r-1)}, x\right), \qquad 3 \le i \le r$$

$$g_3^{(r)} = \mathcal{Q}(x) + \mathcal{A}_r\left(x_3^{(r-1)}, x\right),$$

where the $\mathcal{Q}_i$'s are quadratic polynomials which contain the monomial $x_3^{(i)2}$ and respectively x^2 for the last round, and the $\mathcal{A}_i$'s are affine polynomials that contain the monomial $x_3^{(i-1)}$ for all $i \ge 3$ and without any restrictions on $\mathcal{A}_2$. Then, by iterative substitution of $g_3^{(r)}$ into $g_3^{(r-1)}$ to eliminate $x_3^{(r-1)}$, $g_3^{(r-1)}$ into $g_3^{(r-2)}$ to eliminate $x_3^{(r-2)}$, etc. we produce a LEX Gröbner basis in x-shape position. □

3.2　Bariant's Attack

Recently Bariant introduced an attack on `Ciminion` via univariate greatest common divisor computations [Bar23]. As in our polynomial model, he introduced an auxiliary variable X for the truncated component. If one inverts $p_C \circ p_E$ using $(c_1 - p_1, c_2 - p_2, X)^\intercal$ as output state, then one obtains three polynomials, see [Bar23, Fig. 4],

$$\aleph = f_1(X), \qquad K_1 = f_2(X), \qquad K_2 = f_3(X). \tag{32}$$

Since the first polynomial is univariate in X, one can compute its greatest common divisor with $X^q - X$ to solve for X and henceforth also solve for K_1 and K_2.

　　Observe that Bariant's univariate polynomial is exactly the univariate polynomial of the LEX Gröbner basis.

Lemma 3.3. *Let $\mathbb{F}_q$ be a finite field, let $I_{\mathtt{Ciminion}} \subset \mathbb{F}_q\left[y_1, y_2, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(r-1)}, x\right]$ be a `Ciminion` ideal, let $\tilde{f} \in \mathbb{F}_q[x]$ be the univariate LEX polynomial from Corollary 3.2, and let $f \in \mathbb{F}_q[x]$ be Bariant's univariate polynomial. Then $f = \alpha \cdot \tilde{f}$ for some $\alpha \in \mathbb{F}_q^\times$.*

Proof. `Ciminion` is an iterated Feistel permutation therefore we have that

$$\mathcal{R}^{(i)}\left(\mathbf{x}^{(i-1)}\right) \equiv \mathbf{x}^{(i)} \mod I_{\mathtt{Ciminion}} \qquad \Longleftrightarrow \qquad \mathbf{x}^{(i-1)} \equiv \mathcal{R}^{(i)^{-1}}\left(\mathbf{x}^{(i)}\right) \mod I_{\mathtt{Ciminion}},$$

where $2 \le i \le r - 1$, and analogously for $i = 1, r$. Since the inverse round functions are in $I_{\mathtt{Ciminion}}$, we can iteratively substitute them into each other. This yields exactly the polynomials from Equation (32), so $f \in I_{\mathtt{Ciminion}}$. By Corollary 3.2 we can construct a `Ciminion` LEX Gröbner basis in x-shape position, then for any $g \in \mathbb{F}_q[x]$ we have $g \in I_{\mathtt{Ciminion}} \Leftrightarrow \tilde{f} \mid g$ [CLO15, Chapter 4 §5 Exercise 13]. Since $\deg\left(\tilde{f}\right) = \deg\left(f\right)$ they can only differ up to a multiplicative constant. □

　　Moreover, the construction of the univariate LEX polynomial can be effectively expressed in the cost of univariate polynomial multiplication over finite fields [CK91], which is quasi-linear in the degree.

　　As a countermeasure, Bariant proposes to add a linear combination of K_1 and K_2 to each component after p_C [Bar23, §4]. We note that in general this modification breaks our Gröbner basis from Theorem 3.1. As a settlement, we propose to add a linear combination of K_1 and K_2 only to the non-linear parts of the round functions, i.e.

$$\mathcal{R}^{(i)} : \mathbb{F}_q^3 \times \mathbb{F}_q^2 \to \mathbb{F}_q^3,$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}, \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 1 \\ 1 & c_4^{(i)} & c_4^{(i)} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z + x \cdot y + \alpha \cdot K_1 + \beta \cdot K_2 \end{pmatrix} + \mathbf{c}^{(i)}, \tag{33}$$

where $\alpha, \beta \in \mathbb{F}_q^\times$. This modification does not affect the proof of Theorem 3.1, hence we maintain the DRL Gröbner basis for the modified `Ciminion` polynomial system. However,

this modification breaks the construction of the univariate LEX polynomial via univariate polynomial multiplication, because the quadratic DRL Gröbner basis polynomial for the $r_C{}^{\text{th}}$ round will now also contain linear terms for the variables $x_3^{(2)}, \dots, x_3^{(r-1)}, x$ instead of only $x_3^{(r_C-1)}, \dots, x_3^{(r-1)}, x$. While the LEX Gröbner basis of the modified Ciminion might still be in x-shape position, we can now construct it only via an FGLM algorithm.

Note if the modification of Equation (33) is performed in all rounds of p_E, then the *Middle State-Output Relation* polynomial system of Bariant et al. [BBLP22, §5] becomes underdetermined, so this attack also becomes more difficult.

3.3 Implementation

We implemented Ciminion in the computer algebra systems SageMath [Sag24] and OSCAR [OSC24].[5] The SageMath implementation can handle 128 bit prime numbers, in particular the Gröbner basis from Theorem 3.1 can be computed for primes of that size. For $p = 2^{127} + 45$, $r_C = 90$ and $r_E = 14$ it is computed in negligible time.

3.3.1 Ciminion2

We also implemented a modified version of Ciminion to protect against Bariant's attack, called Ciminion2, which adds $K_1 + K_2$ in every non-linear round of p_E and p_C, i.e. only in the first round of p_E the key addition is omitted. The round function for the first sample then modifies to

$$
\mathcal{R}^{(i)} : \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} \mapsto \begin{pmatrix} 0 & 0 & 1 \\ 1 & c_4^{(i)} & c_4^{(i)} \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z + x \cdot y + K_1 + K_2 \end{pmatrix} + \mathbf{c}^{(i)}, \qquad (34)
$$

where $2 \leq i \leq r_C + r_E$. For higher samples one then adds $K_{2 \cdot j-1} + K_{2 \cdot j}$ in p_E. For Ciminion2 a DRL Gröbner basis for the first sample is also computed in negligible time in SageMath for practical parameters. Due to the additional key addition the quadratic polynomials in the DRL Gröbner basis modify to

$$
g_3^{(i)} = \mathcal{Q}_i \left(x_3^{(i)}, \dots, x_3^{(r-1)}, x \right) + \mathcal{A}_i \left(x_3^{(2)}, \dots, x_3^{(r-1)}, x \right), \qquad 2 \leq i \leq r-1, \qquad (35)
$$

$$
g_3^{(r)} = \mathcal{Q}_r(x) + \mathcal{A}_i \left(x_3^{(2)}, \dots, x_3^{(r-1)}, x \right), \qquad (36)
$$

where $\mathcal{Q}_i$ is quadratic and has the term $x_3^{(i)^2}$ and respectively x^2 for the last round, and $\mathcal{A}_i$ is affine and has non-zero coefficients for $x_3^{(2)}, \dots, x_3^{(r-1)}, x$.

3.4 Cryptanalysis

We now analyze the complexity of a Gröbner basis attack on Ciminion and Ciminion2 via our Gröbner basis, and compare it to the designer's estimation and Bariant's attack.

3.4.1 Estimation of the Designers

For Gröbner basis cryptanalysis of Ciminion [DGGK21, §4.4] the designers considered a weaker scheme where the keys are added after the permutations p_E instead of before the rolling functions, see [DGGK21, Fig. 7]. The designers claimed that the weaker scheme is simpler to analyze than Ciminion. We reject this claim, after all we have computed a DRL Gröbner basis for Ciminion with rather simple linear variable eliminations, see Theorem 3.1.

---

[5] https://github.com/sca-research/Groebner-Basis-Cryptanalysis-of-Ciminion-and-Hydra.git

The designers assumed that weakened `Ciminion` polynomial systems are regular, moreover for regular systems it is well-known that the degree of regularity d_{reg} is equal to the Macaulay bound (Theorem 2.6). Then, the DRL Gröbner basis complexity is bounded by [BFS04, Theorem 7]

$$\mathcal{O}\left(\binom{n + d_{\mathrm{reg}}}{d_{\mathrm{reg}}}^{\omega}\right) \tag{37}$$

field operations. For `Ciminion` cryptanalysis, it is then conjectured that the DRL Gröbner basis computation is the most costly part in an attack.

The designers did not analyze the iterated model, they only studied a model of two equations in two key variables, we call this model the *fully substituted model*. By construction the polynomials in the fully substituted model have degree $2^{r_C + r_E - 1}$. Under the regularity assumption we then have that $d_{\mathrm{reg}} = 2 \cdot \left(2^{r_C + r_E - 1} - 1\right) + 1 = 2^{r_C + r_E} - 1$, so Equation (37) evaluates to

$$\mathcal{O}\left(\binom{2^{r_C + r_E} + 1}{2^{r_C + r_E} - 1}^{\omega}\right) = \mathcal{O}\left(\left(\frac{2^{r_C + r_E} \cdot (2^{r_C + r_E} + 1)}{2}\right)^{\omega}\right). \tag{38}$$

In Table 3 this estimate is evaluated for sample we round numbers.

3.4.2 Complexity of Bariant's Attack

Bariant's attack is split into two parts: construction of the univariate polynomial via an iterated sequence of squarings and the GCD with the field equation. Multiplication of two univariate polynomials $f, g \in \mathbb{F}_q[x]$ with $\deg(f), \deg(g) \le d$ can be performed in $\mathcal{O}\left(d \cdot \log_2(d) \cdot \log_2\left(\log_2(d)\right)\right)$ field operations [CK91], and the complexity for the GCD was given in Equation (21). The construction step then requires [Bar23, §3.2]

$$\mathcal{O}\left(2^{r_C + r_E} \cdot (r_C + r_E) \cdot \log_2(r_C + r_E)\right) \tag{39}$$

field operations, and the GCD requires

$$\mathcal{O}\left(2^{r_C + r_E - 1} \cdot (r_C + r_E - 1) \cdot \left(r_C + r_E - 1 + \log_2(q)\right) \cdot \log_2(r_C + r_E)\right). \tag{40}$$

Therefore, in general the GCD step dominates the complexity. In Table 3 Bariant's attack is evaluated for sample we round numbers.

Note that the GCD step implicitly assumes that $2^{r_C + r_E - 1} < q$, though if the degree of the `Ciminion` function exceeds q, then an exhaustive search over the truncated output always performs better than the GCD computation.

3.4.3 Eigenvalue Method

For standard `Ciminion` it is sufficient to find a $\mathbb{F}_q$-valued root for x, because we can then invert $p_C \circ p_E$ to obtain a key guess. Obviously, our `Ciminion` DRL Gröbner basis (Theorem 3.1) satisfies the structural assumption from Section 2.5.1, so the complexity of constructing a univariate polynomial in x is, see Equation (20),

$$\mathcal{O}\left(2^{\omega \cdot (r_C + r_E - 2) + 1}\right). \tag{41}$$

However, for standard `Ciminion` this complexity is always dominated by Bariant's attack, see Section 3.4.2.

For `Ciminion2` on the other hand solving for x is not sufficient, because we cannot invert $p_C \circ p_E$ anymore due to additional key additions. Therefore, for `Ciminion2` we have to find a $\mathbb{F}_q$-valued solution for all quadratic variables to derive a key guess. Since the

`Ciminion2` DRL Gröbner basis satisfies the structure of Equation (22), the complexity of the Eigenvalue Method is given by Equation (25)

$$
\mathcal{O}\left(
\left\{
\begin{array}{ll}
2^{\omega+1} \cdot \dfrac{2^{\omega \cdot (r_C + r_E - 1)} - N^{\frac{r_C + r_E - 1}{2}}}{2^{2 \cdot \omega} - N}, & 2 \mid (r_C + r_E - 1), \\[2ex]
2^{2 \cdot \omega + 1} \cdot \dfrac{2^{\omega \cdot (r_C + r_E - 2)} - N^{\frac{r_C + r_E - 2}{2}}}{2^{2 \cdot \omega} - N}, & 2 \nmid (r_C + r_E - 1)
\end{array}
\right\}
\right),
\tag{42}
$$

where N is a bound on the number of $\mathbb{F}_q$-valued solutions in every iteration. In addition, we absorbed the terms for the GCDs with the field equations into the implied constant, because these steps are always dominated by the construction of the respective characteristic polynomial. Since a bound on N is in general not known we consider a worst-case scenario approach. We assume that the first $\mathbb{F}_q$-valued solution that an adversary recovers in every iteration leads to the true key, i.e. we assume that $N = 1$ in Equation (42).

In Table 3 we evaluated the discussed `Ciminion` complexity estimations. All our estimations are from a designer's point of view, which assume an ideal adversary who has access to a hypothetical matrix multiplication algorithm which achieves $\omega = 2$. As already mentioned, for standard `Ciminion` Bariant's attack is the most performative attack, and at least $r_C + r_E \geq 112$ is required to achieve 128 bits of security. For `Ciminion2` on the other hand this attack is not feasible, so we have to fall back to the Eigenvalue Method. Then, we require at least $r_C + r_E \geq 66$ to achieve 128 bits of security. Finally, the estimation of the `Ciminion` designers grossly underestimates the capabilities of Gröbner basis attacks. After all, we can construct an iterated `Ciminion` DRL Gröbner basis in negligible time, whereas the designers estimated this complexity to be in $\mathcal{O}\left(2^{2 \cdot \omega \cdot (r_C + r_E)}\right)$.

Table 3: Complexity estimations for various Gröbner basis attacks on `Ciminion` and `Ciminion2` with $q = 2^{127} + 45$ and $\omega = 2$.

	Complexity (bits)		
$r_C + r_E$	Bariant's Attack [Bar23]	Eigenvalue Method	Fully Substituted Model [DGGK21, §4.4]
32	45.60	61.09	126
33	46.67	63.09	130
65	80.18	127.09	258
66	81.22	129.09	262
111	127.45	219.09	442
112	128.47	221.09	446

Recall from [DGGK21, Table 1] that $r_C \in \left\{ s + 6, \left\lceil \frac{2 \cdot (s+6)}{3} \right\rceil \right\}$, where s is the required security level. For `Ciminion2` this already implies that $\dim_{\mathbb{F}_q}\left(\mathcal{F}_{\texttt{Ciminion2}}\right) \geq 2^{\frac{s}{2}}$, and the complexity of the Eigenvalue Method will scale approximately with $\gtrsim 2^{\omega \cdot \frac{s}{2}} \geq 2^s$. So, for now we conclude that `Ciminion2` resists Gröbner basis attacks within the security level, even under ideal assumptions on the adversary.

4   Analysis of Hydra

The `Hydra` heads are based on the Lai–Massey permutation. For the Gröbner basis analysis we recall that the Lai–Massey permutation

$$
\mathcal{L} : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + F(x - y) \\ y + F(x - y) \end{pmatrix}
\tag{43}
$$

is linearly equivalent to the Feistel permutation

$$\mathcal{F} : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y + F(x) \end{pmatrix} \tag{44}$$

via the matrices $\mathbf{A}_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$ and $\mathbf{A}_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, i.e. $\mathcal{L} = \mathbf{A}_2 \circ \mathcal{F} \circ \mathbf{A}_1$. Inspired by this equivalence we can reduce the Hydra polynomial system to only one non-linear equation per round, i.e. the Hydra polynomial system transforms to a Feistel-like polynomial system.

In [Ste24, §6.2] the iterated polynomial systems representing generalized Feistel ciphers were analyzed. For these polynomial systems linear systems were derived, and in case the systems are of full rank the iterated polynomial systems are in generic coordinates, see [Ste24, Theorem 6.5]. For a system in generic coordinates we then have a proven upper bound on the solving degree, see Theorem 2.6.

Based on the aforementioned linear equivalence, we are going to replicate this strategy to construct a linear system for Hydra. Further, if the Hydra linear system has full rank, then we can transform the system into a DRL Gröbner basis via a linear change of coordinates.

4.1 Generic Coordinates for Hydra

Recall that the Hydra iterated polynomial system (Definition 2.2) consists of $16 \cdot r_{\mathcal{H}} + 8$ polynomials in $16 \cdot r_{\mathcal{H}} + 4$ variables. Let

$$\mathbf{A}_n = \begin{pmatrix} 1 & 0 & \dots & 0 & -1 \\ 0 & 1 & \dots & 0 & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix} \in \mathrm{GL}_n\left(\mathbb{F}_q\right), \tag{45}$$

and let $\mathbf{B} = \mathrm{diag}\left(\mathbf{A}_4, \mathbf{A}_4\right) \in \mathrm{GL}_8\left(\mathbb{F}_q\right)$. Then the first seven components of $\mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_j^{(i)}$, where $1 \leq i \leq r_{\mathcal{H}}$ and $j \in \{1, 2\}$, are affine and the eighth component is quadratic. Moreover, the fourth and the eighth components of $\mathbf{B} \mathbf{M}_{\mathcal{R}}^{-1} \mathbf{f}_{\mathcal{R}}$ are quadratic while the other ones are affine. In particular, by extracting the $2 \cdot 7 \cdot r_{\mathcal{H}}$ affine components of the $\mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_j^{(i)}$'s we can eliminate the variables

$$y_1, \dots, y_4, z_1, \dots, z_3, x_{j,1}^{(i)}, \dots, x_{j,7}^{(i)}, \tag{46}$$

where $0 \leq i \leq r_{\mathcal{H}} - 1$ and $j \in \{1, 2\}$, in the quadratic equations. (This can be explicitly seen in the proof of Theorem 4.1.) Thus, after the elimination we reduce to $2 \cdot r_{\mathcal{H}} + 8$ polynomials, of which 6 are also affine, in $2 \cdot r_{\mathcal{H}} + 4$ variables.

Next we derive a linear system to verify being in generic coordinates for Hydra.

Theorem 4.1. *Let $\mathbb{F}_q$ be a finite field, let $\overline{\mathbb{F}_q}$ be its algebraic closure, let*

$$\mathcal{F}_{Hydra} = \left\{ \mathbf{f}_1^{(i)}, \mathbf{f}_{\mathcal{R}}, \mathbf{f}_2^{(i)} \right\}_{1 \leq i \leq r_{\mathcal{H}}} \subset P = \overline{\mathbb{F}_q}\left[\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(1)}, \dots, \mathbf{x}_1^{(r_{\mathcal{H}}-1)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_2^{(r_{\mathcal{H}}-1)}, \mathbf{k} \right]$$

be a Hydra polynomial system, and let

$$\mathcal{G}_{Hydra} = \left\{ \mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_1^{(i)}, \mathbf{B} \mathbf{M}_{\mathcal{R}}^{-1} \mathbf{f}_{\mathcal{R}}, \mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_2^{(i)} \right\}_{1 \leq i \leq r_{\mathcal{H}}}.$$

Then every homogeneous ideal $I \subset P[x_0]$ such that $x_0 \notin \sqrt{I}$ and $\mathcal{G}_{Hydra}^{\mathrm{hom}} \subset I$ is in generic coordinates if the following linear system has rank $16 \cdot r_{\mathcal{H}} + 4$

$$y_1 - z_4 + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' - \mathbf{x}_1^{(1)} \right) \right)_1 + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{x}_1^{(1)} - \mathbf{k}' \right) \right)_8 = 0,$$

$$\vdots$$

$$z_3 - z_4 + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' - \mathbf{x}_1^{(1)} \right) \right)_7 + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{x}_1^{(1)} - \mathbf{k}' \right) \right)_8 = 0,$$

$$\sum_{l=1}^{4} y_l - z_l = 0,$$

$$x_{2,k}^{(0)} - x_{2,8}^{(0)} + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' - \mathbf{x}_2^{(1)} \right) \right)_k + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{x}_2^{(1)} - \mathbf{k}' \right) \right)_8 = 0,$$

$$\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(0)} = 0,$$

$$x_{j,k}^{(i-1)} - x_{j,8}^{(i-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' - \mathbf{x}_j^{(i)} \right) \right)_k + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{x}_j^{(i)} - \mathbf{k}' \right) \right)_8 = 0,$$

$$\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{j,l}^{(i-1)} = 0,$$

$$x_{1,k}^{(r_{\mathcal{H}}-1)} - x_{1,8}^{(r_{\mathcal{H}}-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' + (\mathbf{y},\mathbf{z})^{\mathsf{T}} \right) \right)_k - \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' + (\mathbf{y},\mathbf{z})^{\mathsf{T}} \right) \right)_8 = 0,$$

$$\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{1,l}^{(r_{\mathcal{H}}-1)} = 0,$$

$$x_{2,k}^{(r_{\mathcal{H}}-1)} - x_{2,8}^{(r_{\mathcal{H}}-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' + \mathbf{x}_2^{(0)} \right) \right)_k - \left(\mathbf{M}_{\mathcal{J}}^{-1} \left(\mathbf{k}' + \mathbf{x}_2^{(0)} \right) \right)_8 = 0,$$

$$\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(r_{\mathcal{H}}-1)} = 0,$$

$$y_1 - y_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_1 + \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_4 = 0,$$

$$\vdots$$

$$y_3 - y_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_3 + \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_4 = 0,$$

$$z_1 - z_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_5 + \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_8 = 0,$$

$$\vdots$$

$$z_3 - z_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_7 + \left(\mathbf{M}_{\mathcal{R}}^{-1} \mathbf{x}_2^{(0)} \right)_8 = 0,$$

where $2 \leq i \leq r_{\mathcal{H}} - 1$, $j \in \{1, 2\}$ and $1 \leq k \leq 7$.

Proof. By [Ste24, Theorem 3.2] $\left(\mathcal{G}_{\mathsf{Hydra}}^{\mathrm{hom}} \right)$ is in generic coordinates if

$$\sqrt{\mathcal{G}_{\mathsf{Hydra}}^{\mathrm{top}}} = \left(\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_1^{(r_{\mathcal{H}}-1)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_2^{(r_{\mathcal{H}}-1)}, \mathbf{k} \right). \tag{47}$$

To shorten some writing let

$$\mathbf{g}_j^{(i)} = \left(\mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_j^{(i)} \right)^{\mathrm{hom}} \mod (x_0),$$

$$\mathbf{g}_{\mathcal{R}} = \left(\mathbf{B} \mathbf{M}_{\mathcal{R}}^{-1} \mathbf{f}_{\mathcal{R}} \right)^{\mathrm{hom}} \mod (x_0).$$

Then for $i = 1$ we have that

$$
\mathbf{g}_1^{(1)} = \left(\begin{array}{c} \left\{ y_k - z_4 + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' - \mathbf{x}_1^{(1)} \right) \right)_k + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{x}_1^{(1)} - \mathbf{k}' \right) \right)_8 \right\}_{1 \le k \le 7} \\ \left(\sum_{l=1}^{4} y_l - z_l \right)^2 \end{array} \right),
$$

$$
\mathbf{g}_2^{(1)} = \left(\begin{array}{c} \left\{ x_{2,k}^{(0)} - x_{2,8}^{(0)} + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' - \mathbf{x}_2^{(1)} \right) \right)_k + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{x}_2^{(1)} - \mathbf{k}' \right) \right)_8 \right\}_{1 \le k \le 7} \\ \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(0)} \right)^2 \end{array} \right).
$$

For $2 \le i \le r_{\mathcal{H}} - 1$ and $j \in \{1, 2\}$ we have that

$$
\mathbf{g}_j^{(i)} = \left(\begin{array}{c} \left\{ x_{j,k}^{(i-1)} - x_{j,8}^{(i-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' - \mathbf{x}_j^{(i)} \right) \right)_k + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{x}_j^{(i)} - \mathbf{k}' \right) \right)_8 \right\}_{1 \le k \le 7} \\ \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{j,l}^{(i-1)} \right)^2 \end{array} \right).
$$

For $i = r_{\mathcal{H}}$ we have that

$$
\mathbf{g}_1^{(r_{\mathcal{H}})} =
$$
$$
\left(\begin{array}{c} \left\{ x_{1,k}^{(r_{\mathcal{H}}-1)} - x_{1,8}^{(r_{\mathcal{H}}-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' + (\mathbf{y}, \mathbf{z})^{\mathsf{T}} \right) \right)_k - \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' + (\mathbf{y}, \mathbf{z})^{\mathsf{T}} \right) \right)_8 \right\}_{1 \le k \le 7} \\ \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{1,l}^{(r_{\mathcal{H}}-1)} \right)^2 \end{array} \right),
$$
$$
\mathbf{g}_2^{(r_{\mathcal{H}})} =
$$
$$
\left(\begin{array}{c} \left\{ x_{2,k}^{(r_{\mathcal{H}}-1)} - x_{2,8}^{(r_{\mathcal{H}}-1)} + \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' + \mathbf{x}_2^{(0)} \right) \right)_k - \left(\mathbf{M}_{\mathcal{J}}^{-1}\left(\mathbf{k}' + \mathbf{x}_2^{(0)} \right) \right)_8 \right\}_{1 \le k \le 7} \\ \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(r_{\mathcal{H}}-1)} \right)^2 \end{array} \right).
$$

Finally, for the rolling function we have that

$$
\mathbf{g}_{\mathcal{R}} = \left(\begin{array}{c} \left\{ y_k - y_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1}\mathbf{x}_2^{(0)} \right)_k + \left(\mathbf{M}_{\mathcal{R}}^{-1}\mathbf{x}_2^{(0)} \right)_4 \right\}_{1 \le k \le 3} \\ g(\mathbf{y}, \mathbf{z}) \\ \left\{ z_k - z_4 - \left(\mathbf{M}_{\mathcal{R}}^{-1}\mathbf{x}_2^{(0)} \right)_{4+k} + \left(\mathbf{M}_{\mathcal{R}}^{-1}\mathbf{x}_2^{(0)} \right)_8 \right\}_{1 \le k \le 3} \\ g(\mathbf{z}, \mathbf{y}) \end{array} \right).
$$

Except for the rolling function, all quadratic polynomials are squares of linear polynomials. Collecting the linear polynomials from $\mathcal{G}_{\mathsf{Hydra}}^{\mathrm{top}}$ together with the linear polynomials from the squares, we yield the linear system from the assertion. It is clear that all these linear polynomials are elements of $\sqrt{\mathcal{G}_{\mathsf{Hydra}}^{\mathrm{top}}}$. Since the highest degree components are homogeneous, we always have that $\sqrt{\mathcal{G}_{\mathsf{Hydra}}^{\mathrm{top}}} \subset \left(\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(i)}, \dots, \mathbf{x}_1^{(r_{\mathcal{H}}-1)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_2^{(r_{\mathcal{H}}-1)}, \mathbf{k} \right)$. Thus, if the linear system from the assertion has full rank, then we have established the equality from Equation (47). $\qquad\square$

Naturally, this theorem can be extended to an arbitrary number of Hydra samples. Now it is straight-forward to compute the Macaulay bound (Theorem 2.6) for Hydra.

Corollary 4.2. *Let $\mathbb{F}_q$ be a finite field, let $\overline{\mathbb{F}_q}$ be its algebraic closure, let*

$$\mathcal{F}_{Hydra} = \left\{ \mathbf{f}_1^{(i)}, \mathbf{f}_\mathcal{R}, \mathbf{f}_2^{(i)} \right\}_{1 \le i \le r_\mathcal{H}} \subset P = \overline{\mathbb{F}_q}\left[\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(1)}, \dots, \mathbf{x}_1^{(r_\mathcal{H}-1)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_2^{(r_\mathcal{H}-1)}, \mathbf{k} \right]$$

be a Hydra polynomial system. Let

$$\mathcal{G}_{Hydra} = \left\{ \mathbf{A}_8 \mathbf{M}_\mathcal{J}^{-1} \mathbf{f}_1^{(i)}, \mathbf{B}\mathbf{M}_\mathcal{R}^{-1} \mathbf{f}_\mathcal{R}, \mathbf{A}_8 \mathbf{M}_\mathcal{J}^{-1} \mathbf{f}_2^{(i)} \right\}_{1 \le i \le r_\mathcal{H}},$$

and assume that $\left(\mathcal{G}_{Hydra}^{hom} \right)$ is in generic coordinates. If the affine polynomials in $\mathcal{G}_{Hydra}$ have rank $2 \cdot 7 \cdot r_\mathcal{H} + 6$, then

$$\mathrm{sd}_{DRL}\left(\mathcal{G}_{Hydra} \right) \le 2 \cdot r_\mathcal{H}.$$

Proof. If the affine polynomials have rank $2 \cdot 7 \cdot r_\mathcal{H} + 6$, then we can reduce to $2 \cdot r_\mathcal{H} + 2$ quadratic polynomials in $2 \cdot r_\mathcal{H} - 2$ variables. This variable elimination does not affect being in generic coordinates, so the Macaulay bound (Theorem 2.6) evaluates to

$$\mathrm{sd}_{DRL}\left(\mathcal{G}_{Hydra} \right) \le 2 \cdot (2 \cdot r_\mathcal{H} - 1) - (2 \cdot r_\mathcal{H} - 1) + 1 = 2 \cdot r_\mathcal{H}. \qquad \square$$

4.2 Extracting a Zero-Dimensional DRL Gröbner Basis for Hydra

Due to the structure of the Hydra heads we can actually produce a DRL Gröbner basis via a simple linear change of coordinates. Again, let $\mathcal{G}_{Hydra}$ be as in Theorem 4.1, we assume that

(i) the linear system from Theorem 4.1 has full rank, and

(ii) the linear polynomials in $\mathcal{G}_{Hydra}^{top}$ have rank $2 \cdot 7 \cdot r_\mathcal{H} + 6$.

I.e., $\left(\mathcal{G}_{Hydra}^{hom} \right)$ is in generic coordinates, and we can use the linear polynomials in $\mathcal{G}_{Hydra}$ to reduce the system to $2 \cdot r_\mathcal{H} + 2$ quadratic polynomials in $2 \cdot r_\mathcal{H} - 2$ variables.

By construction $2 \cdot r_\mathcal{H}$ of these quadratic polynomials come from the Hydra heads and their quadratic terms are squares of linear equations. In particular, before the variable elimination the quadratic terms are, see Definition 2.2 and the proof of Theorem 4.1,

$$g_{1,8}^{(1)} = \left(\sum_{l=1}^{4} y_l - z_l \right)^2 \qquad\qquad = \mathcal{L}_{1,1}^2, \tag{48}$$

$$g_{2,8}^{(1)} = \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(0)} \right)^2 \quad = \mathcal{L}_{2,1}^2, \tag{49}$$

$$g_{j,8}^{(i)} = \left(\sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{j,l}^{(i-1)} \right)^2 = \mathcal{L}_{j,i}^2, \qquad 2 \le i \le r_\mathcal{H}, \; j \in \{1,2\}, \tag{50}$$

Recall that these linear equations $\mathcal{L} = \{ \mathcal{L}_{1,i}, \mathcal{L}_{2,i} \}_{1 \le i \le 2 \cdot r_\mathcal{H}}$ also appeared in the linear system from Theorem 4.1. By Assumption (i), $\mathcal{L}$ together with the linear equations from $\mathcal{G}_{Hydra}^{top}$ has rank $16 \cdot r_\mathcal{H} + 4$. After eliminating $2 \cdot 7 \cdot r_\mathcal{H} + 6$ variables with the affine equations from $\mathcal{G}_{Hydra}$, the polynomials from Equations (48) to (50) modify to

$$g_{j,8}^{(i)} = \left(\tilde{\mathcal{L}}_{j,i} + a_{j,i} \right)^2, \qquad 1 \le i \le r_\mathcal{H}, \; j \in \{1,2\}, \tag{51}$$

where $\tilde{\mathcal{L}}_{j,i}$ is a linear polynomial in the remaining $2 \cdot r_\mathcal{H} - 2$ variables and $a_{j,i} \in \mathbb{F}_q$. Moreover, the rank of a matrix does not depend on the order of Gaussian elimination, so

by Assumption (i) the new linear polynomials $\tilde{\mathcal{L}} = \left\{ \tilde{\mathcal{L}}_{1,i}, \tilde{\mathcal{L}}_{2,i} \right\}_{1 \leq i \leq 2 \cdot r_{\mathcal{H}}}$ still must contain a maximal linearly independent subset of rank $2 \cdot r_{\mathcal{H}} - 2$.

Let $\mathbf{x}_{\mathrm{nl}}$ denote the remaining $2 \cdot r_{\mathcal{H}} - 2$ non-linear variables. We can now collect this maximal linearly independent subset of $\tilde{\mathcal{L}}$ in a matrix $\mathbf{M} \in \mathbb{F}_q^{(2 \cdot r_{\mathcal{H}} - 2) \times (2 \cdot r_{\mathcal{H}} - 2)}$. Columns of $\mathbf{M}$ are indexed by the elements of $\mathbf{x}_{\mathrm{nl}}$, and in the rows of $\mathbf{M}$ we write the coefficients of the maximal linearly independent equations. Next we pick new variables $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_{2 \cdot r_{\mathcal{H}} - 2})^\mathsf{T}$, and finally we perform a linear change of variables $\hat{\mathbf{x}} = \mathbf{M} \mathbf{x}_{\mathrm{nl}}$. By construction $2 \cdot r_{\mathcal{H}}$ polynomials in $\mathcal{G}_{\mathsf{Hydra}}$ are of the form

$$\tilde{\mathcal{L}}_{j,i}(\mathbf{x}_{\mathrm{nl}})^2 + \mathcal{A}_{j,i}(\mathbf{x}_{\mathrm{nl}}) = 0, \tag{52}$$

where $\mathcal{A}_{j,i} \in \mathbb{F}_q[\mathbf{x}_{\mathrm{nl}}]$ is affine, $1 \leq i \leq 2 \cdot r_{\mathcal{H}}$ and $j \in \{1, 2\}$. Therefore, after the change of coordinates we produce $2 \cdot r_{\mathcal{H}} - 2$ polynomials of the form

$$g_i = \hat{x}_i^2 + \sum_{j=1}^{2 \cdot r_{\mathcal{H}} - 2} \alpha_{i,j} \cdot \hat{x}_j + \alpha_{i,0}, \tag{53}$$

where $\alpha_{i,j}, \alpha_{i,0} \in \mathbb{F}_q$, together with four additional quadratic polynomials. In particular, $\hat{\mathcal{G}} = \{g_i\}_{1 \leq i \leq 2 \cdot r_{\mathcal{H}} - 2} \subset \mathbb{F}_q[\hat{\mathbf{x}}]$ is a DRL Gröbner basis with $\dim_{\mathbb{F}_q}\left(\hat{\mathcal{G}}\right) = 2^{2 \cdot r_{\mathcal{H}} - 2}$.

We summarize this construction in a more formal manner in the next corollary.

Corollary 4.3. *Let $\mathbb{F}_q$ be a finite field, let*

$$\mathcal{F}_{\mathsf{Hydra}} = \left\{ \mathbf{f}_1^{(i)}, \mathbf{f}_{\mathcal{R}}, \mathbf{f}_2^{(i)} \right\}_{1 \leq i \leq r_{\mathcal{H}}} \subset P = \mathbb{F}_q\left[\mathbf{y}, \mathbf{z}, \mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_1^{(r_{\mathcal{H}} - 1)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_2^{(r_{\mathcal{H}} - 1)}, \mathbf{k}\right]$$

be a Hydra polynomial system, and let

$$\mathcal{G}_{\mathsf{Hydra}} = \left\{ \mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_1^{(i)}, \mathbf{B} \mathbf{M}_{\mathcal{R}}^{-1} \mathbf{f}_{\mathcal{R}}, \mathbf{A}_8 \mathbf{M}_{\mathcal{J}}^{-1} \mathbf{f}_2^{(i)} \right\}_{1 \leq i \leq r_{\mathcal{H}}},$$

$$\mathcal{G}_{lin} = \left\{ g \in \mathcal{G}_{\mathsf{Hydra}} \mid \deg(g) = 1 \right\},$$

$$\mathcal{G}_{squ} = \left\{ g \in \mathcal{G}_{\mathsf{Hydra}} \mid \deg(g) = 2 \right\}.$$

Assume that

(i) *the linear system from Theorem 4.1 has full rank, and*

(ii) $\mathrm{rank}\left(\mathcal{G}_{lin}^{\mathrm{top}}\right) = 2 \cdot 7 \cdot r_{\mathcal{H}} + 6.$

Then there exists a linear change of coordinates $\mathcal{G}_{sub} \subset P\left[\hat{x}_i \mid 1 \leq i \leq 2 \cdot r_{\mathcal{H}} - 2\right]$ such that:

(1) $\hat{\mathcal{G}}_{squ} = \mathcal{G}_{squ} \mod (\mathcal{G}_{lin}, \mathcal{G}_{sub}) \subset \mathbb{F}_q\left[\hat{x}_i \mid 1 \leq i \leq 2 \cdot r_{\mathcal{H}} - 2\right]$, *where division is performed with respect to the DRL term order* $\mathbf{y} > \mathbf{z} > \mathbf{x}_1^{(1)} > \ldots > \mathbf{x}_1^{(r_{\mathcal{H}} - 1)} > \mathbf{x}_2^{(0)} > \ldots > \mathbf{x}_2^{(r_{\mathcal{H}} - 1)} > \mathbf{k} > \hat{\mathbf{x}}.$

(2) *For every $1 \leq i \leq 2 \cdot r_{\mathcal{H}} - 2$ there exists $g_i \in \hat{\mathcal{G}}_{squ}$ such that*

$$g_i = \hat{x}_i^2 + \mathcal{A}_i,$$

where $\mathcal{A}_i \in \mathbb{F}_q\left[\hat{x}_i \mid 1 \leq i \leq 2 \cdot r_{\mathcal{H}} - 2\right]$ is affine.

(3) $\dim_{\mathbb{F}_q}\left(\mathcal{G}_{lin} \cup \mathcal{G}_{sub} \cup \hat{\mathcal{G}}_{squ}\right) \leq 2^{2 \cdot r_{\mathcal{H}} - 2}.$

Besides being tedious to do by hand, success of the change of coordinates depends on the matrices used in the Hydra heads. Therefore, we did not attempt a full proof by hand, instead for specific instances we outsourced it to a computer algebra program where the construction can be performed in a matter of seconds.

4.3 Implementation

We have implemented Hydra and its iterated polynomial system in the computer algebra systems SageMath [Sag24] and OSCAR [OSC24].[6] In both systems we implemented the linear system from Theorem 4.1 and the change of coordinates from Section 4.2. With the SageMath implementation we can compute the DRL Gröbner basis for full rounds $r_{\mathcal{H}} = 39$ over 128 bit prime fields in negligible time. The OSCAR computer algebra system offers an efficient F4 implementation that can handle primes of size $q < 2^{31}$, therefore we recommend to use OSCAR for experimental computations over small prime fields.

For completeness, we mention that we implemented the rolling function without a constant addition, i.e. we set $\mathbf{c}_{\mathcal{R}}^{(i)} = \mathbf{0}$ for all $i \geq 1$. However, we emphasize that the constant addition does not affect either the linear system from Theorem 4.1 or the change of coordinates from Section 4.2.

Example 4.4 (A Concrete Instance). In [GØSW22, Appendix C] the Hydra designers propose concrete matrices over the prime field $q = 2^{127} + 45$

$$\mathbf{M}_{\mathcal{E}} = \mathrm{circ}\,(3, 2, 1, 1)\,, \qquad \mathbf{M}_{\mathcal{I}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 3 & 1 & 3 & 1 \\ 4 & 1 & 1 & 2 \end{pmatrix}, \tag{54}$$

$$\mathbf{M}_{\mathcal{J}} = \begin{pmatrix} 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 3 & 1 & 1 & 1 & 1 & 1 & 1 \\ 4 & 1 & 4 & 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 8 & 1 & 1 & 1 & 1 \\ 7 & 1 & 1 & 1 & 7 & 1 & 1 & 1 \\ 8 & 1 & 1 & 1 & 1 & 5 & 1 & 1 \\ 5 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\ 4 & 1 & 1 & 1 & 1 & 1 & 1 & 6 \end{pmatrix}. \tag{55}$$

For security level $\kappa = 128$ bits the designers suggest to use $r_{\mathcal{H}} = 39$, see [GØSW23, §5.5].

For this instance using our SageMath implementation we verified that the Hydra polynomial system is in generic coordinates for $m \in \{2, 3, 4, 5, 10\}$ many samples. For $m = 2$ we can compute a Hydra DRL Gröbner basis with the change of coordinates from Section 4.2.

Recall the quadratic terms from the Hydra heads, see Equations (48) to (50). For $r_{\mathcal{H}} \geq 2$ and after the elimination of $2 \cdot 7 \cdot r_{\mathcal{H}} + 6$ variables, see Equation (51), we observed that

$$\mathrm{rank}\left(\tilde{\mathcal{L}}_{1,8}^{(3)}, \ldots, \tilde{\mathcal{L}}_{1,8}^{(r_{\mathcal{H}})}, \tilde{\mathcal{L}}_{2,8}^{(1)}, \ldots, \tilde{\mathcal{L}}_{2,8}^{(r_{\mathcal{H}})}\right) = 2 \cdot r_{\mathcal{H}} - 2. \tag{56}$$

So, before the elimination of $2 \cdot 7 \cdot r_{\mathcal{H}} + 6$ variables our change of coordinates corresponds to, see Equations (48) to (50),

$$\hat{x}_i = \sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{1,l}^{(i-1)}, \qquad 3 \leq i \leq r_{\mathcal{H}}, \tag{57}$$

$$\hat{x}_{i+r_{\mathcal{H}}-2} = \sum_{l=1}^{8} (-1)^{\left\lfloor \frac{l-1}{4} \right\rfloor} \cdot x_{2,l}^{(j-1)}, \qquad 1 \leq i \leq r_{\mathcal{H}}. \tag{58}$$

We also note that we did not find a round number $r_{\mathcal{H}} \geq 2$ for which the Gröbner basis extraction fails.

4.4 Cryptanalysis

Next we discuss the cryptanalytic impact of our theoretical analysis. Throughout, this section we will always assume that we can perform the linear change of coordinates from Section 4.2. In particular, we stress that for the Hydra instance from Example 4.4 we are always able to construct a DRL Gröbner basis. Therefore, we have a quadratic Hydra DRL Gröbner basis in $2 \cdot r_{\mathcal{H}} - 2$ equations and variables together with four additional quadratic polynomials at hand. This opens the door for two different attack strategies:

(A) Recomputation of the DRL Gröbner basis for the overdetermined system of $2 \cdot r_{\mathcal{H}} + 2$ quadratic polynomials.

(B) Polynomial system solving techniques for the DRL Gröbner basis of $2 \cdot r_{\mathcal{H}} - 2$ quadratic polynomials.

For comparison, we also review the complexity estimation of the designers.

4.4.1 Estimation of the Designers

The Hydra designers considered the iterated polynomial model, see [GØSW23, §7.4] and [GØSW22, Appendix I.2], and reduced it to $2 \cdot r_{\mathcal{H}} + 2$ quadratic polynomials in $2 \cdot r_{\mathcal{H}} - 2$ variables. The designers assumed that this downsized system is semi-regular [BFS04]. Then, the degree of regularity d_{reg} is given by the index of the first negative coefficient in the power series

$$H(t) = \frac{(1 - t^2)^{2 \cdot r_{\mathcal{H}} + 2}}{(1 - t)^{2 \cdot r_{\mathcal{H}} - 2}}, \tag{59}$$

and the complexity of computing the DRL Gröbner basis is then bounded by [BFS04, Theorem 7]

$$\mathcal{O}\left(\binom{2 \cdot r_{\mathcal{H}} - 2 + d_{\mathrm{reg}}}{d_{\mathrm{reg}}}^{\omega}\right). \tag{60}$$

For $r_{\mathcal{H}}^* = 29$ the designers estimate $d_{\mathrm{reg}} = 23$, with $\omega = 2$ this yields a complexity estimate of $\approx 2^{130.8}$ bits. Moreover, it is claimed that $r_{\mathcal{H}} = r_{\mathcal{H}}^* + 2 = 31$ rounds are sufficient to provide 128 bits of security against Gröbner basis attacks.

4.4.2 Recomputing the DRL Gröbner Basis

After constructing the Hydra DRL Gröbner basis we have one big advantage compared to the designer estimation: We yield a $\mathbb{F}_q$-Boolean polynomial system, see Section 2.4. Thus, in the Macaulay matrix we only have to consider square-free monomials. Effectively, this saves time and space.

Proven Solving Degree Bound. Recall from Corollary 4.2 that $\mathrm{sd}_{DRL}\left(\mathcal{G}_{\mathsf{Hydra}}\right) \leq 2 \cdot r_{\mathcal{H}}$. So, in the worst case we have to construct the full $\mathbb{F}_q$-Boolean Macaulay matrix, see Theorem 2.10. The complexity for constructing the $\mathbb{F}_q$-Boolean Macaulay matrix was given in Corollary 2.12, for the Hydra parameters $|F| = 4$, $n, d = 2 \cdot r_{\mathcal{H}} - 2$ and $D = 2$ we yield the complexity

$$\mathcal{O}\left(4 \cdot r_{\mathcal{H}} \cdot (2 \cdot r_{\mathcal{H}} - 1) \cdot \sum_{i=0}^{2 \cdot r_{\mathcal{H}} - 2} \binom{2 \cdot r_{\mathcal{H}} + i}{i + 2} \cdot \binom{2 \cdot r_{\mathcal{H}} - 2}{i}\right). \tag{61}$$

Further, by Theorem 2.10 Gaussian elimination on the $\mathbb{F}_q$-Boolean Macaulay matrix can be performed in $\mathcal{O}\left(4 \cdot 2^{\omega \cdot (2 \cdot r_{\mathcal{H}} - 2)}\right)$ field operations.

In Table 4 we evaluated these complexities for sample round numbers.

Semi-Regularity Assumption. If we assume that the Hydra polynomial system is semi-regular after the change of variables (Section 4.2), then Equation (61) improves to

$$\mathcal{O}\left(4 \cdot r_{\mathcal{H}} \cdot (2 \cdot r_{\mathcal{H}} - 1) \cdot \sum_{i=0}^{d_{\text{reg}}-2} \binom{2 \cdot r_{\mathcal{H}} - 2}{i} \cdot \binom{2 \cdot r_{\mathcal{H}} + i}{i + 2}\right), \tag{62}$$

and the complexity for Gaussian elimination improves to

$$\mathcal{O}\left(4 \cdot \left(\sum_{i=0}^{d_{\text{reg}}-2} \binom{2 \cdot r_{\mathcal{H}} - 2}{i}\right) \cdot \left(\sum_{i=0}^{d_{\text{reg}}} \binom{2 \cdot r_{\mathcal{H}} - 2}{i}\right)^{\omega - 1}\right). \tag{63}$$

In Table 4 we evaluated the complexities together with the one from the designers (Section 4.4.1). Construction of the $\mathbb{F}_q$-Boolean Macaulay matrix dominates this approach. As expected, under the semi-regularity assumption the Boolean approach is slightly more performative than the designers' estimation. In particular, an ideal adversary with $\omega = 2$ could break up to $r_{\mathcal{H}} = 29$ rounds below 128 bits. Moreover, if the $\mathbb{F}_q$-Boolean Macaulay matrix can be constructed with negligible cost, then an ideal adversary could break up to $r_{\mathcal{H}} = 34$ rounds.

Table 4: Hydra complexity estimations for recomputing the DRL Gröbner basis. The column d_{reg} shows the degree of regularity under the semi-regularity assumption. All estimations use $\omega = 2$.

		Complexity (bits)				
		Proven Solving Degree		Semi-Regular		
$r_{\mathcal{H}}$	d_{reg}	Boolean Matrix Construction	Boolean Gaussian Elimination	Boolean Matrix Construction	Boolean Gaussian Elimination	Semi-Regular Estimate [GØSW23]
28	22	148.72	110	118.28	102.10	125.31
29	23	153.89	114	123.29	106.25	130.80
30	24	159.05	118	128.27	110.40	136.29
31	25	164.21	122	133.25	114.54	141.77
32	26	169.37	126	138.21	118.66	147.26
33	27	174.52	130	143.16	122.79	152.75
34	28	179.67	134	148.10	126.90	158.23
35	28	184.82	138	150.37	129.66	160.24
39	32	205.41	154	170.12	146.16	182.22
45	37	236.24	178	197.03	169.55	211.72

Empirical F4 Working Degree. The Hydra designers performed small scale Gröbner basis experiments for Hydra with the F4 [Fau99] algorithm, see [GØSW22, Table 4]. With our OSCAR implementation we replicated these experiments to verify that F4's highest working degree matches the semi-regular degree of regularity before and after the change of coordinates. Our results are recorded in Table 5. Our experiments have been performed on an AMD EPYC-Rome (48) CPU with 94 GB RAM.

In all our experiments F4's highest working degree indeed matches the semi-regular degree of regularity. As expected, after the change of coordinates the computation is faster, because F4 will implicitly eliminate all monomials which are divisible by a square.

Table 5: Hydra highest working degree and total running time of F4 over the prime field $\mathbb{F}_{7741}$. The column d_{reg} shows the degree of regularity under the semi-regularity assumption.

$r_{\mathcal{H}}$	d_{reg}	Before Change of Coordinates		After Change of Coordinates		Grassi et al.'s data [GØSW22, Table 4]	
		Highest F4 working degree	Total time (s)	Highest F4 working degree	Total time (s)	Highest F4 working degree	Total time (s)
3	3	3	0.01	3	0.01	-	-
4	3	3	0.01	3	0.01	-	-
5	4	4	0.02	4	0.01	-	-
6	5	5	0.16	5	0.03	5	0.1
7	5	5	1.35	5	0.25	5	0.9
8	6	6	14.88	6	2.68	6	13.4
9	7	7	384.13	7	67.19	7	293.8
10	8	8	8676.30	8	3194.95	8	8799.5
11	8	-	-	-	-	8	229606.8

4.4.3 Term Order Conversion & Eigenvalue Method

With a Hydra DRL Gröbner basis at hand, see Section 4.2, we can proceed to direct system solving techniques. For example, we can use a generic algorithm like the state-of-the-art probabilistic FGLM algorithm [FGHR14] to convert to a LEX Gröbner basis, and then factor the univariate polynomial. Assuming that the Hydra DRL Gröbner basis is in shape position the complexity of term order conversion is bounded by [FGHR14, Proposition 3]

$$\mathcal{O}\Big((2 \cdot r_{\mathcal{H}} - 2) \cdot 2^{2 \cdot r_{\mathcal{H}} - 2} \cdot \big(2^{(\omega-1) \cdot (2 \cdot r_{\mathcal{H}} - 2)} + (2 \cdot r_{\mathcal{H}} - 2)^2 \cdot \log_2(2 \cdot r_{\mathcal{H}} - 2) \big) \Big) \qquad (64)$$

field operations. Next we have to factor the univariate polynomial of degree $2^{2 \cdot r_{\mathcal{H}} - 2}$ via the GCD with the field equation. However, even for full rounds $r_{\mathcal{H}} = 39$ this degree is well below 2^{128}. Therefore, the GCD complexity from Equation (21) will not exceed 128 bits, and henceforth can be ignored.

On the other hand, the Hydra DRL Gröbner basis satisfies the structure assumption from Equation (22), so we can pass to the dedicated Eigenvalue Method to find a $\mathbb{F}_q$-valued solution. For Hydra, the complexity from Equation (25) then evaluates to

$$\mathcal{O}\left(2^{\omega+1} \cdot \frac{2^{\omega \cdot (2 \cdot r_{\mathcal{H}} - 2)} - N^{r_{\mathcal{H}} - 1}}{2^{2 \cdot \omega} - N} \right), \qquad (65)$$

where N is a bound on the number of $\mathbb{F}_q$-valued solutions in every iteration. Again, the GCD complexity can be considered negligible. Since a bound on N is in general not available, we consider a worst-case scenario approach. We assume that the first root which an adversary recovers in every iteration leads to the true Hydra key, i.e. we use $N = 1$ in Equation (65).

In Table 6 we evaluated the complexities for direct system solving techniques and compared them to the other strategies. Our estimations are again from a designer's point of view which assumes an ideal adversary who can achieve the linear algebra constant $\omega = 2$. In this scenario, term order conversion breaks up to $r_{\mathcal{H}} = 31$ rounds below 128 bits. This invalidates the claim of the Hydra designers [GØSW23, §7.4] that $r_{\mathcal{H}} = 31$ rounds are sufficient to provide 128 bits of security against Gröbner basis attacks for an ideal adversary. Moreover, via the dedicated Eigenvalue Method up to $r_{\mathcal{H}} = 33$ do not achieve 128 bits of security. For full rounds $r_{\mathcal{H}} = 39$ the security margin is reduced by ≈ 30 bits, however the security claim is not affected.

Let $r_{\mathcal{H}}^*$ be the minimum round number which achieves 128 bits of security in a Gröbner basis attack, i.e. according to our analysis we have $r_{\mathcal{H}}^* = 34$. Recall from Equation (12) that the head round number is set as $r_{\mathcal{H}} = \lceil 1.25 \cdot \max\{24, 2 + r_{\mathcal{H}}^*\} \rceil$. Thus, our analysis suggests that this round number is increased to $r_{\mathcal{H}} = 45$ to recover the security margin originally intended by the designers.

In case one wants to be extra conservative, one can consider the construction of the $\mathbb{F}_q$-Boolean Macaulay matrix to be for free. Then, we have $r_{\mathcal{H}}^* = 35$, see Table 4, which implies $r_{\mathcal{H}} = 47$.

Table 6: Hydra complexity estimations for term order conversion and the Eigenvalue Method. The column d_{reg} shows the degree of regularity under the semi-regularity assumption. All estimations use $\omega = 2$.

		Complexity (Bits)			
$r_{\mathcal{H}}$	d_{reg}	Term Order Conversion	Eigenvalue Method	Boolean Semi-Regular Estimate	Semi-Regular Estimate [GØSW23]
28	22	113.75	107.09	118.28	125.31
29	23	117.81	111.09	123.29	130.80
30	24	121.86	115.09	128.27	136.29
31	25	125.91	119.09	133.25	141.77
32	26	129.95	123.09	138.21	147.26
33	27	134.00	127.09	143.16	152.75
34	28	138.04	131.09	148.10	158.23
35	28	142.09	135.09	150.37	160.24
39	32	158.25	151.09	170.12	182.22
45	37	182.46	175.09	197.03	211.72

Small Scale Experiments. With our `OSCAR` implementation we computed the characteristic polynomial of the Hydra multiplication matrix $\mathbf{M}_{\hat{x}_{2 \cdot r_{\mathcal{H}}-2}}$, see Equation (19), for small $r_{\mathcal{H}}$ and measured the time of the overall computation. In addition, we computed the polynomial matrix determinant from Equation (19) for uniformly random matrices $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{F}_q^{(2 \cdot r_{\mathcal{H}}-3) \times (2 \cdot r_{\mathcal{H}}-3)}$. For construction and determinant of the matrix we used `OSCAR`'s functions for multivariate polynomial division and determinant respectively. Our results are recorded in Table 7. In addition, we performed term order conversion to LEX for our Hydra DRL Gröbner basis with `OSCAR`'s `fglm` function. For comparison, we also generated a random multivariate quadratic polynomial system in the same number of variables as Hydra, computed the DRL Gröbner basis, and finally converted to LEX via FGLM. Our FGLM results are recorded in Table 8. All experiments in this section have been performed on an AMD EPYC-Rome (48) CPU with 94 GB RAM.

For the characteristic polynomial, we observed that the construction time for the multiplication matrix is negligible compared to the one for the characteristic polynomial. Moreover, Hydra characteristic polynomials are computed faster than for random matrices $\mathbf{A}_0$ and $\mathbf{A}_1$.

For term order conversion, we observed that FGLM on Hydra systems requires less time than for random multivariate quadratic systems. We also note that in all our experiments the Hydra LEX Gröbner basis was in $\hat{x}_{2 \cdot r_{\mathcal{H}}-2}$-shape position.

Table 7: Hydra running times for construction of the multiplication matrix and its characteristic polynomial over the prime field $\mathbb{F}_{7741}$. The column "Determinant Random Matrix" shows the running time of the determinant from Equation (19) with uniformly random $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{F}_{7741}^{(2 \cdot r_{\mathcal{H}} - 3) \times (2 \cdot r_{\mathcal{H}} - 3)}$.

	Time (s)		
$r_{\mathcal{H}}$	Construction Hydra Matrix	Characteristic Polynomial Hydra Matrix	Determinant Random Matrix
5	< 0.1	0.3	0.4
6	0.7	73.9	82.8
7	11.1	8336.7	11722.9

Table 8: Hydra FGLM running times over the prime field $\mathbb{F}_{7741}$. The column "Random Quadratic F4" shows the running time for computing a DRL Gröbner basis with F4 for a random multivariate quadratic polynomial system in the same number of variables as the Hydra system. The column "Random Quadratic FGLM" shows the running time of FGLM for the random DRL Gröbner basis.

	Time (s)		
$r_{\mathcal{H}}$	Hydra FGLM	Random Quadratic F4	Random Quadratic FGLM
5	0.3	0.1	0.4
6	20.0	0.8	26.1
7	1265.7	15.6	1598.5
8	78345.1	1123.9	99492.9

Acknowledgments

References

[Bar23]   Augustin Bariant. Algebraic cryptanalysis of full Ciminion. Cryptology ePrint Archive, Paper 2023/1283, 2023. Accepted into Selected Areas in Cryptography 2024. URL: https://eprint.iacr.org/2023/1283.

[BBL+24]   Augustin Bariant, Aurélien Boeuf, Axel Lemoine, Irati Manterola Ayala, Morten Øygarden, Léo Perrin, and Håvard Raddum. The algebraic FreeLunch: Efficient Gröbner basis attacks against arithmetization-oriented primitives. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IV*, volume 14923 of *LNCS*, pages 139–173. Springer, Cham, August 2024. doi:10.1007/978-3-031-68385-5_5.

[BBLP22]   Augustin Bariant, Clémence Bouvier, Gaëtan Leurent, and Léo Perrin. Algebraic attacks against some arithmetization-oriented primitives. *IACR Trans. Symm. Cryptol.*, 2022(3):73–101, 2022. doi:10.46586/tosc.v2022.i3.73-101.

[BFS04] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proceedings of the International Conference on Polynomial System Solving*, pages 71–74, 2004.

[BFSS13] Magali Bardet, Jean-Charles Faugére, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic Boolean systems. *J. Complex.*, 29(1):53–75, 2013. `doi:10.1016/j.jco.2012.07.001`.

[Buc65] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Universität Innsbruck, 1965.

[CG21] Alessio Caminata and Elisa Gorla. Solving multivariate polynomial systems and an invariant from commutative algebra. In Jean Claude Bajard and Alev Topuzoğlu, editors, *Arithmetic of Finite Fields*, pages 3–36, Cham, 2021. Springer International Publishing. `doi:10.1007/978-3-030-68869-1_1`.

[CG22] Alessio Caminata and Elisa Gorla. Solving multivariate polynomial systems and an invariant from commutative algebra, 2022. Version: 7. `arXiv:1706.06319`.

[CK91] David G. Cantor and Erich Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28(7):693–701, 07 1991. `doi:10.1007/BF01178683`.

[CLO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer International Publishing, 4 edition, 2015. `doi:10.1007/978-3-319-16721-3`.

[DGGK21] Christoph Dobraunig, Lorenzo Grassi, Anna Guinet, and Daniël Kuijsters. Ciminion: Symmetric encryption based on Toffoli-gates over large finite fields. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 3–34. Springer, Cham, October 2021. `doi:10.1007/978-3-030-77886-6_1`.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra*, 139(1):61–88, 1999. `doi:10.1016/S0022-4049(99)00005-5`.

[Fau02] 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*, ISSAC '02, page 75–83. Association for Computing Machinery, 2002. `doi:10.1145/780506.780516`.

[FGHR14] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Sub-cubic change of ordering for Gröbner basis: A probabilistic approach. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, page 170–177, New York, NY, USA, 2014. Association for Computing Machinery. `doi:10.1145/2608628.2608669`.

[GOPS22] Lorenzo Grassi, Silvia Onofri, Marco Pedicini, and Luca Sozzi. Invertible quadratic non-linear layers for MPC-/FHE-/ZK-friendly schemes over $\mathbb{F}_p^n$: Application to Poseidon. *IACR Trans. Symm. Cryptol.*, 2022(3):20–72, 2022. `doi:10.46586/tosc.v2022.i3.20-72`.

[GØSW22] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From Farfalle to Megafono via Ciminion: The PRF Hydra for MPC applications. Cryptology ePrint Archive, Report 2022/342, 2022. URL: https://eprint.iacr.org/2022/342.

[GØSW23] Lorenzo Grassi, Morten Øygarden, Markus Schofnegger, and Roman Walch. From Farfalle to Megafono via Ciminion: The PRF Hydra for MPC applications. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 255–286. Springer, Cham, April 2023. doi:10.1007/978-3-031-30634-1_9.

[GRR+16] Lorenzo Grassi, Christian Rechberger, Dragos Rotaru, Peter Scholl, and Nigel P. Smart. MPC-friendly symmetric key primitives. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 430–443. ACM Press, October 2016. doi:10.1145/2976749.2978332.

[KR00] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1 edition, 2000. doi:10.1007/978-3-540-70628-1.

[KR05] Martin Kreuzer and Lorenzo Robbiano. *Computational Commutative Algebra 2*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1 edition, 2005. doi:10.1007/3-540-28296-3.

[KR16] Martin Kreuzer and Lorenzo Robbiano. *Computational Linear and Commutative Algebra*. Springer International Publishing, Cham, 1 edition, 2016. doi:10.1007/978-3-319-43601-2.

[Laz83] Daniel Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In J. A. van Hulzen, editor, *Computer Algebra, EUROCAL '83, European Computer Algebra Conference, London, England, March 28-30, 1983, Proceedings*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin Heidelberg, 1983. doi:10.1007/3-540-12868-9_99.

[OSC24] OSCAR – Open Source Computer Algebra Research system, Version 1.2.2, 2024. URL: https://www.oscar-system.org.

[RS24] Arnab Roy and Matthias Johann Steiner. Generalized triangular dynamical system: An algebraic system for constructing cryptographic permutations over finite fields. Cryptology ePrint Archive, Paper 2024/1316, 2024. Accepted into Selected Areas in Cryptography 2024. URL: https://eprint.iacr.org/2024/1316.

[Sag24] SageMath, the Sage Mathematics Software System (Version 10.5), 2024. URL: https://www.sagemath.org.

[Ste24] Matthias Johann Steiner. Solving degree bounds for iterated polynomial systems. *IACR Trans. Symm. Cryptol.*, 2024(1):357–411, 2024. doi:10.46586/tosc.v2024.i1.357-411.

[Sto00] Arne Storjohann. *Algorithms for matrix canonical forms*. Doctoral thesis, ETH Zurich, Zürich, 2000. Diss., Technische Wissenschaften ETH Zürich, Nr. 13922, 2001. doi:10.3929/ethz-a-004141007.

[Ten19] Andrea Tenti. *Sufficiently overdetermined random polynomial systems behave like semiregular ones*. PhD thesis, University of Bergen, 2019. URL: https://hdl.handle.net/1956/21158.

[Tof80]   Tommaso Toffoli. Reversible computing. In Jaco de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg. `doi:10.1007/3-540-10003-2_104`.

[WXXZ24]   Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In David P. Woodruff, editor, *35th SODA*, pages 3792–3835. ACM-SIAM, January 2024. `doi:10.1137/1.9781611977912.134`.