The Complexity of the Crossbred Algorithm

Damien Vidal, Sorina Ionica and Claire Delaplace

Laboratoire MIS, Université de Picardie Jules Verne, Amiens, France

Abstract. The Crossbred algorithm is currently the state-of-the-art method for solving overdetermined multivariate polynomial systems over $\mathbb{F}_2$. Since its publication in 2015, several record breaking implementations have been proposed and demonstrate the power of this hybrid approach. Despite these practical results, the complexity of this algorithm and the choice of optimal parameters for it are difficult open questions. In this paper, we prove a bivariate generating series for potentially admissible parameters of the Crossbred algorithm.

Keywords: Gröbner basis · polynomial system · MQ problem · exhaustive search · Crossbred

1 Introduction

Given a polynomial system of m polynomials and n variables over a finite field $\mathbb{F}_p$, solving the system is proven to be NP-complete [FY79]. However, not all polynomial systems are hard to solve since the behaviour of algorithms allowing to solve them depends on the relative values of m and n. If $m \geq n(n-1)/2$ or if $n \geq m(m+1)$, it is possible to solve a polynomial systems with these parameters in polynomial time [TW12]. Most often in cryptography, we are confronted to solving polynomial systems for which $m \geq n$. Commonly used methods to solve these systems are algorithms for computing Gröbner basis: Buchberger's algorithm [Buc65] or linear algebra-based algorithms (F_4 [Fau99], F_5 [Fau02], XL [CKPS00]).

In this paper, we focus on the *Multivariate Quadratic Problem* (*MQ*), which means that we consider polynomials of degree 2. In the case of small finite fields (for example $\mathbb{F}_2$, $\mathbb{F}_3$ or $\mathbb{F}_5$), exhaustive search becomes a viable way to solve a polynomial system (FES [BCC+10]). It is also used to assign certain variables before running the linear algebra-based algorithm for solving the system (FXL [CKPS00], BooleanSolve [BFSS13], Crossbred [JV17]). In particular, we are interested in the case where the polynomial system is defined over $\mathbb{F}_2$.

The complexity analysis of the Crossbred algorithm is not clear, but the authors of the algorithm claim it to be similar to that of FXL [CKPS00] or BooleanSolve [BFSS13], without giving a proof. However, Joux and Vitse's original implementation as well as more recent open-source implementations [NNY18, NNY17, BS23] were used to break records several times on overdetermined systems coming from the Fukuoka Type I MQ challenge [Yas15]. For a polynomial system $\mathcal{F}$, the running time of the algorithm heavily depends on three input parameters, D, d and k. In a precomputation step based on linear algebra on the degree D Macaulay matrix, the algorithm constructs r polynomials of total degree D and of degree d in the first k variables. After that, the last $n-k$ variables are assigned in $\mathcal{F}$ and the degree d Macaulay matrix is computed for the system obtained in this way. After specification, the new polynomials obtained in the precomputation step are also appended as rows in this matrix. If the resulting degree d system in the first k variables may be solved (for instance by linearization), then we are done.

E-mail: damien.vidal@u-picardie.fr (Damien Vidal), sorina.ionica@u-picardie.fr (Sorina Ionica), claire.delaplace@u-picardie.fr (Claire Delaplace)

A set of parameters D, d and k are called admissible if the degree d system obtained after specialization can be solved with echelonisation. Without a proper complexity analysis, it is far from obvious to determine admissible parameters for the algorithm, and even harder to determine optimal choices. To the best of our knowledge, all existing implementations over $\mathbb{F}_2$ have focused on $D \leq 5$ and $d = 1$, which means that the system obtained after specialization is linear. From a practical point of view, it is difficult to handle higher values of D and d, since as soon as $D \geq 4$ lots of linear dependencies start to appear in the Macaulay matrix and the matrices are large enough that it is no longer possible to construct them due to memory issues. This is a common problem when implementing Gröbner basis algorithms, most of the computation time is lost in useless operations. For example, in the F5 algorithm, Faugère [Fau02] and later Bardet in her PhD thesis [Bar04], proposed two criteria to remove all linear dependencies, for regular and semi-regular polynomial systems, respectively.

In this paper, we first restate these criteria for the special form of Macaulay matrices of degree D appearing in the Crossbred algorithm and show that we can remove linear dependencies after specialization. Secondly, we propose a simplified variant of the algorithm, called Block Crossbred , which may be seen as a homogeneous variant of the algorithm, before specialisation. We compute a bivariate generating series whose coefficients correspond to the number of newly generated polynomials in the precomputation step of the algorithm for input parameters D, d and k, under semi-regularity assumptions. From this analysis, we deduce the generating series for the Crossbred algorithm, again under a semi-regularity hypothesis. We conclude by showing sets of admissible parameters for Crossbred obtained by looking at the coefficients of the series. Finally, we implemented in Sage and ran the Crossbred algorithm on the smallest sets of these parameters, to confirm our theoretical findings.

Related work. Recently, there have been several attempts to study the complexity of Crossbred. first Duarte [Dua23] computes a bivariate generating series for the preprocessing step of the algorithm. The author introduces the notion of semi-regularity for homogenous systems of polynomials, but this notion is not used anywhere in his proof. On one hand, the polynomials appearing in the rows of the Macaulay matrices in Crossbred are affine, and on the other hand the proofs do not keep track of reductions to zero in the algorithm. Secondly, Nakamura [Nak24] claims a completely different series from the one in [Dua23, JV17]. Finally, the recent preprint [BCT+24] revisits the notion of semi-regularity and focuses on admissible parameters for Crossbred under semi-regularity assumptions, provided that the bivariate generating series conjectured in the literature is correct. We stress here that the correct assumption for Crossbred, which is a hybrid algorithm, is that of strong semi-regularity. Roughly speaking, this means that after assigning $n - k$ variables in the initial system, the derived system in k variables is semi-regular, for almost all assignments. Theorem 1 in [Nak24] and Theorem 2 in [BCT+24] came close to this idea, but the authors focus on a fixed assignment instead of looking at all possible assignments.

Our work is completely independent from that of Duarte, but we certainly do not claim originality for this approach. Most of the techniques used in this paper are standard in the literature (see [Fau02, Bar04]) and we adapted them to the case of Crossbred.

This paper is organised as follows. In Section 2 we introduce the notion of semi-regular sequences of polynomials and briefly survey linear algebra based algorithms and the Crossbred algorithm. From that, we state our criteria for reduction to zero and present the Block Crossbred algorithm in Section 3. In Section 4 we show that there are now reductions to zero in the Block Crossbred algorithm if the two criteria are used. Based on this result, we compute the generating series of our algorithm in Section 5. Finally, under semi-regularity assumptions, we deduce a proof for the bivariate generating series of the original Crossbred in Section 6. We apply our results to compute admissible parameters for Crossbred in Section 7.2.

Acknowledgment. The authors thank Charles Bouillaguet and Julia Sauvage for helpful discussion. This work was partially funded by the French Agence Nationale de la Recherche under the project Postcryptum (ANR20-ASTR-0011). The experiments were carried on the MatriCS platform of the Université de Picardie Jules Verne.

2 Background and Notation

In this section, we will introduce the notation and terminology used throughout the paper.

Notation We will use the polynomial ring $R = \mathbb{F}_p[x_1, \ldots, x_n]$, where $\mathbb{F}_p$ is any finite field. We choose an admissible monomial ordering on R. We write $x^b = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ with $b = (b_1, \ldots, b_n)$. Then $|b| = \sum_{i=1}^{n} b_i$ is the degree of the monomial (also called total degree) and is written $\deg(x^b)$. We denote by $\deg_k$ the degree over the first k variables (i.e. $\deg_k(x^b) = \sum_{i=1}^{k} b_i$). The leading term of a polynomial f with respect to the chosen order is denoted by $LT(f)$. The total degree and the degree over k of f are the total degree and the degree over the first k variables, respectively, of its leading term $LT(f)$ with respect to the chosen order.

We use the glex order with $x_1 \geq x_2 \ldots \geq x_n$ (i.e. $x^a >_{glex} x^b$ if $|a| > |b|$ or $|a| = |b|$ and the leftmost non-zero coefficient of $a - b$ is positive).

Macaulay matrices, initially introduced by Lazard [Laz], are at the heart of all linear algebra-based algorithms for computing Gröbner basis. The Macaulay matrix is defined as follows.

Definition 1. Fix an admissible monomial ordering on R. Given a homogeneous (affine) system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$ in R, we associate to it the Macaulay matrix of degree D (resp. $\leq D$), denoted by $Mac_{D,m}(\mathcal{F})$ (resp. $Mac_{\leq D,m}(\mathcal{F})$) and defined as follows: the columns of $Mac_{D,m}(\mathcal{F})$ (resp. $Mac_{\leq D,m}(\mathcal{F})$) are indexed by the monomials in $\mathbb{F}_p[x_1, \ldots, x_n]$ of degree D (resp. of degree $\leq D$), sorted in decreasing order from left to the right following the chosen order. Each row in this matrix is labeled by a tag $\langle u, f_i \rangle$, where u is a monomial in $\mathbb{F}_p[x_1, \ldots, x_n]$ and $f_i \in \mathcal{F}$ such that $deg(uf_i) = D$ (resp. $deg(uf_i) \leq D$), and contains the polynomial uf_i written as a vector of coefficients of monomials.

Example 2.1. Consider the polynomial system $\mathcal{F} = \{f_1, f_2\}$ with $f_1, f_2 \in \mathbb{F}_2[x_1, x_2, x_3]$ given by:

$$\begin{aligned} f_1 &= x_1x_3 + x_2x_3 + x_2 + 1, \\ f_2 &= x_1x_2 + x_1 + x_3 + 1. \end{aligned}$$

Since the goal is to compute roots of this polynomial system in $\mathbb{F}_2$, we add the polynomials $x_1^2 - x_1$, $x_2^2 - x_2$ and $x_3^2 - x_3$ to this system. This is equivalent to constructing the Macaulay matrix in $\mathbb{F}_2[x_1, x_2, x_3]/ < x_1^2 - x_1, x_2^2 - x_2, x_3^2 - x_3 >$. Then the corresponding Macaulay matrix of degree 3 is :

$$Mac_{\leq 3,2}(\mathcal{F}) = \begin{array}{c} \\ f_1 \\ f_2 \\ x_1f_1 \\ x_2f_1 \\ x_3f_1 \\ x_1f_2 \\ x_2f_2 \\ x_3f_2 \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1x_2x_3 & x_1x_2 & x_1x_3 & x_2x_3 & x_1 & x_2 & x_3 & 1 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

2.1 Linear algebra-based Gröbner basis algorithms and their complexity

Let I an ideal in R and let us fix a monomial order ordering on R. We denote by $LT(I)$ the set of all leading terms of non-zero polynomials in I. A Gröbner basis for I is a finite set of generators $G = \{g_1, \dots g_s\}$ such that the monomial ideal generated by elements of $LT(I)$ is given by:

$$\langle LT(I) \rangle = \{LT(g_1), \dots, LT(g_s)\}.$$

Gröbner basis algorithms based on linear algebra compute the row echelon form of the Macaulay matrix for a certain degree d of the system $\mathcal{F}$. The first nonzero element of each row corresponds to a leading monomial of an element of I, belonging to $LT(I)$. For large enough d, Dickson's lemma [DCO97, §2.4, Thm. 5] implies that the collection of those monomials up to degree d generates $LT(I)$ and thus the polynomials corresponding to those rows together form a Gröbner basis of I with respect to the chosen monomial ordering.

During the echelonization process, it may happen that a given row yields zero when reduced modulo the basis under construction. This is called *reduction to zero* in the literature. Ideally, one would like to avoid spending time on computations for rows whose result is zero. For this, several criteria have been proposed and allow to avoid the effective computation of useless reductions [Buc65, Fau02]. We briefly recall here the criteria used in the F5-like algorithms [Fau02, Bar04], which guarantee that there are no reductions to zero during the algorithm for semi-regular sequences of polynomials.

Let us first consider $\mathcal{F} = \{f_1, \dots, f_m\}$ a homogenous system of polynomials in $\mathbb{F}_p[x_1, \dots, x_n]$ with $\deg f_i = d_i$. The General Criterion [Fau02], used by the algorithm for polynomial systems defined over any field $\mathbb{F}_p$, states that a row in $Mac_{d,m}(\mathcal{F})$ labeled by (u, f_i) is a linear combination of previous rows if the monomial u is the leading term of a polynomial in $\langle f_1, \dots, f_{i-1} \rangle$. Therefore, the algorithm constructs the matrix $Mac_{d,i}(\mathcal{F})$ by adding to the matrix $Mac_{d,i-1}(\mathcal{F})$ all rows containing polynomials uf_i except for those where u is a leading term of a row in $\widetilde{Mac}_{d-d_i,i-1}(\mathcal{F})$, the row echelon form of $Mac_{d,i-1}(\mathcal{F})$.

Faugère shows that if the sequence of polynomials is regular, then the only reductions to zero during the execution of the algorithm for finite fields with $char(\mathbb{F}_p) > 2$ are those detected by the General Criterion. For a system of polynomials $\mathcal{F} = \{f_1, \dots, f_m\}$ in $\mathbb{F}_2[x_1, \dots, x_n]$, if the goal is to find solutions in $\mathbb{F}_2$, we may as well add to this system the equations $\{x_1^2 - x_1, \dots, x_n^2 - x_n\}$. Working with the system $\mathcal{F} \cup \{x_1^2 - x, \dots x_n^2 - x\}$ in $R = \mathbb{F}_2[x_1, \dots, x_n]$ is equivalent to working with the polynomial system $\mathcal{F}$ in $R_n = R/\langle x_1^2 - x, \dots x_n^2 - x \rangle$. Consequently, when running the $F5$ algorithm, we also need to remove all reductions to zero coming from the fact that $f^2 = f$, for any $f \in R_n$. The Frobenius Criterion [Fau02] states that a row of $Mac_{d,m}(\mathcal{F})$ labeled by (u, f_i) is a linear combination of previous rows if the monomial u is the leading term of a monomial in $\langle f_1, \dots, f_i \rangle$.

Bardet [Bar04] extends these two criteria for reductions to zero to sequences of polynomials where $m > n$ by introducing the notion of semi-regularity. We recall this notion here, but let us first introduce some more notation.

For $d \geq 0$, we denote by $\mathbb{F}_p[x_1, \ldots, x_n]_d$ the $\mathbb{F}_p$-vector space of homogeneous polynomials of degree d. Let I be an ideal of dimension 0 generated by the sequence $\mathcal{F} = \{f_1, \ldots f_m\}$ and denote by $I_d = \mathbb{F}_p[x_1, \ldots, x_n]_d \cap I$. Then there exists $D \geq 0$ such that

$$\dim_{\mathbb{F}_p}(I_d) = \dim_{\mathbb{F}_p}(\mathbb{F}_p[x_1, \ldots, x_n]_d),$$

for all $d \geq D$ [DCO97] and we define D_{reg} to be the smallest degree with this property.

Since the homogeneous part of highest degree of field equations $x_i^2 - x_i$ is x_i^2, we consider the ring $R_n^h = \mathbb{F}_2[x_1, \ldots, x_n]/\langle x_1^2, \ldots, x_n^2\rangle$. Any homogeneous polynomial of degree d in R_n^h verifies $f^2 = 0$. Following Bardet [Bar04], we directly state the definition of a semi-regular sequence of polynomials defined over $\mathbb{F}_2$.

Definition 2. A sequence of homogeneous polynomials $\{f_1, \ldots, f_m\}$ in R_n^h is called semi-regular over $\mathbb{F}_2$ if:

1. $\langle f_1, \ldots, f_m\rangle \neq R_n^h$,
2. For all $i \in \{1, \ldots, m\}$ if $g_i f_i = 0$ in $R_n^h/\langle f_1, \ldots, f_{i-1}\rangle$ and $\deg(g_i f_i) < D_{reg}$, then $g_i \in \langle f_1, \ldots f_{i-1}, f_i\rangle$.

Given a power series $S \in \mathbb{Z}[[X]]$, the notation $[S]$ denotes the series obtained by truncating S just before the index of its first non-positive coefficient. Bardet computes the Hilbert series of the ideal generated by a semi-regular sequence of homogeneous polynomials $\mathcal{F} = \{f_1, \ldots f_m\}$ of degrees $d_1, \ldots, d_m$ in R_n^h to be:

$$HF_{R/I} = \left[\frac{(1+X)^n}{\prod_{i=1}^m (1+X^{d_i})}\right]. \tag{1}$$

The degree of regularity of the system is given by the index of the first non-positive coefficient of the series in Equation (1).

As shown by Bardet [Bar04], if the sequence is semi-regular and the two criteria are used for recursively constructing the Macaulay matrices, then there are no reductions to zero in the matricial version of the $F5$ algorithm (called Matrix F5 in [Bar04]), until the degree $d = D_{reg} - 1$ is reached. When degree D_{reg} is reached, the algorithm ouputs a Gröbner basis with respect to the chosen monomial order. Consequently, the complexity the Gröbner basis computation using the Matrix F5 algorithm is:

$$\mathcal{O}\left(\binom{n + D_{reg} - 1}{n}^{\omega}\right),$$

where ω is a linear algebra constant.

If the system is affine, it suffices to examine its homogeneous part of highest degree to ensure that there are no degree falls during the execution of the algorithm. Following again [Bar04], we give the following definition.

Definition 3. Let $\{f_1, \ldots, f_m\}$ be an affine sequence of polynomials and denote by f_i^{top} the homogeneous part of highest degree of f_i, $1 \leq i \leq m$. Then $\{f_1, \ldots f_m\}$ is called semi-regular if the sequence $\{f_1^{top}, \ldots, f_m^{top}\}$ is semi-regular.

In practice, if the system $\mathcal{F}$ is affine, Gröbner basis algorithms will perform Gaussian elimination on Macaulay matrices $Mac_{\leq d,m}(\mathcal{F})$, with $d \geq 0$. Then we call *witness degree* the smallest degree for which linear algebra will produce a Gröbner basis. To estimate this index, we look at the affine Hilbert series of the ideal I. To define this series, we consider:

$$I_{\leq d} = I \cap R_{\leq d},$$

where $R_{\leq d} = \oplus_{0 \leq d' \leq d} R_{d'}$. Then the affine Hilbert series is defined as follows:

$$HF^a_{R/I}(X) = \sum_{d \geq 0} \dim(R_{\leq d}/I_{\leq d}) X^d. \tag{2}$$

Then the witness degree is the smallest integer $d > 0$ such that $R_{\leq d} = I_{\leq d}$.

The following result is folklore, but we state it here for completeness.

Lemma 1. *Let $\mathcal{F} = \{f_1, \ldots, f_m, x_1^2 - x, \ldots, x_n^2 - x_n\}$ be an affine semi-regular sequence defined over $\mathbb{F}_2$. Then its affine Hilbert series is given by the formula:*

$$\left[\frac{(1+X)^n}{(1-X)\prod_{i=1}^m (1+X^{d_i})} \right].$$

Proof. By [DCO97, Ch. 9, Th. 12 (ii)], $HF^a_{R/I}(X) = HF_{R^h/I^h}(X)$, where $R^h = \mathbb{F}_2[x_1, \ldots, x_n, h]$ and I^h is the homogenized ideal of I, i.e. the ideal generated by homogenizing the polynomials generating I with respect to a new variable h. By [BFSS13, Prop. 6], the Hilbert series of I^h is given by

$$HF_{R^h/I^h}(X) = \left[\frac{(1+X)^n}{(1-X)\prod_{i=1}^m (1+X^{d_i})} \right].$$

□

In other words, the witness degree of an affine semi-regular system of polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ is given by the first non-positive coefficient of the series whose truncation is given in Lemma 1.

Finally, to analyze the complexity of Crossbred, we will make the standard assumption that the input system $\mathcal{F}$ for the algorithm is strong semi-regular (see for instance [BFSS13]). Roughly speaking, this means that for almost all possible assignements $x_{k+1} = a_{k+1}, \ldots, x_n = a_n$, for some $k > 0$, the sequence

$$\{f_1(x_1, \ldots, x_k, a_{k+1}, \ldots, a_n), \ldots, f_m(x_1, \ldots, x_k, a_{k+1}, \ldots a_n)\}$$

is semi-regular. We slightly adapt here the definition in [BFSS13].

Definition 4. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a semi-regular sequence of polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ and let $0 \leq \gamma \leq 1$ such that $k = (1-\gamma)n$. We say that this sequence is γ-strong semi-regular if

$$\begin{aligned} \mathcal{S}(I) = \{ & (a_{k+1}, \ldots a_n) \in \mathbb{F}_2^{n-k} | \\ & \{f_1(x_1, \ldots, x_k, a_{k+1}, \ldots, a_n), \ldots, f_m(x_1, \ldots, x_k, a_{k+1}, \ldots a_n)\} \text{ is not semi-regular}\} \end{aligned}$$

has cardinality $O(2^{-\gamma n})$.

In Section 7.1 we show a series of experiments which support the claim that random systems are γ-strong semi-regular.

2.2 The Crossbred algorithm

In a nutshell, the Crossbred algorithm [JV17] for fixed input parameters D, d and k, works as follows:

1. Construct r new polynomials $p_1, \ldots, p_r$ of total degree D and of degree d over the first k variables. These polynomials are added to the original system.

2. Specify the last $n-k$ variables in the system obtained in this way.

3. Try to solve the system after specification. If no solution is found, we continue the exhaustive search and change the value of the last $n-k$ variables.

The precomputation step of Crossbred described at (1) performs linear algebra on certain submatrices of the Macaulay matrices introduced in Definition 1. We introduce these submatrices, as well as the submatrices we use in our simplified version of Crossbred in Section 3 in the following definition.

Definition 5. Given a homogeneous (resp. affine) system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$ in R, let $Mac^k_{D,d,m}(\mathcal{F})$ (resp. $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$) be the submatrix of the Macaulay matrix $Mac_{D,m}(\mathcal{F})$ (resp. $Mac_{\leq D,m}(\mathcal{F})$) whose rows correspond to products of the form uf_i, $1 \leq i \leq m$ with $\deg_k u = d-1$ (resp. $\deg_k u \geq d-1$). Let $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ (resp. $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$) be the submatrix of $Mac^k_{D,d,m}(\mathcal{F})$ (resp. $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$) whose columns correspond to monomials M with $\deg_k M = d+1$ or $\deg_k M = d-1$ (resp. $\deg_k M \geq d+1$).

Notation Given a polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$, we denote by f^* any polynomial in $\mathbb{F}_p[x_1, \ldots, x_k]$ obtained from f after specifying the variables $x_{k+1}, \ldots, x_n$. Similarly, given $\mathcal{F} = \{f_1, \ldots, f_m\}$, we denote by $\mathcal{F}^* = \{f_1^*, \ldots, f_m^*\}$.

The pseudocode of the Crossbread algorithm is given in Algorithm 1. Note that for step (3) there exist multiple ways to solve the resulting system. In this paper, we only consider the resolution via linearisation on the resulting Macaulay matrix of degree d of the specialized system. In other words, we think of each monomial in the system as an unknown, and try to solve the linear system obtained in this way.

```
Algorithm 1: The Crossbred algorithm
  Data: Polynomial system F of m equations of n variables, and D, d, k
  Result: A solution of the system (or nothing otherwise)
  Construct Mac^k_{≤D,≥d,m}(F) and M^k_{≤D,≥d,m}(F)
  Compute a basis (v_1, ..., v_r) of the left kernel of M^k_{≤D,≥d,m}(F)
  Construct polynomials p_1, ..., p_r corresponding to v_i · M^k_{≤D,≥d,m}(F)
  for i = (i_1, i_2, ..., i_{n-k}) ∈ F_2^{n-k} do
  |   Evaluate the last n − k variables in each f ∈ F at (i_1, i_2, ..., i_{n-k}) and
  |     compute F*
  |   Compute Mac_{d,m}(F*)
  |   Evaluate the polynomials p_i at (i_1, i_2, ..., i_{n-k}) and append them to the
  |     matrix Mac_{≤d,m}(F*)
  |   if the resulting system is consistent then
  |   |   return the solution
  |   end
  |   else
  |   |   continue
  |   end
  end
```

2.2.1 Another look at the preprocessing step

To better understand the preprocessing step of Algorithm 1, we look at it from a different angle. We construct the matrix $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ for the glex order. From left to right, we order the columns corresponding to monomials of degree D, $D-1$, …, $d+1$, d and $d-1$ over the first k variables.

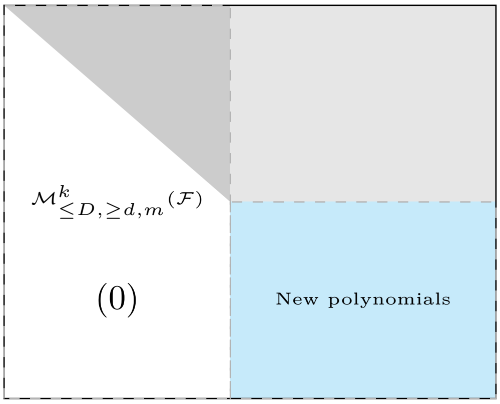

Figure 1: Partial Gaussian elimination of $Mac^k_{D,d,m}(\mathcal{F})$

Notation We denote by $M^k_{\leq D,\geq d}$ ($M^k_{D,d}$) the set of monomials m with $\deg(m) \leq D$ and $\deg_k(m) \geq d$ (with $\deg m = D$ and $\deg_k m = d$, respectively).

Instead of computing the kernel of $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$, an equivalent way to generate new polynomials in the preprocessing step of Algorithm 1 is to partially echelonize $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$, such that the submatrix given by its first $\#M^k_{\leq D,\geq d,m}$ rows is in row echelon form. New polynomials are obtained by taking the rows that have zero entries on the columns corresponding to monomials of degree $\geq d+1$ over the first k variables, as shown in Figure 1.

2.2.2 Limitations of the Crossbred algorithm

From a theoretical point of view, the complexity of the Crossbred algorithm is

$$\mathcal{O}\left(\binom{n+D-1}{D}^{\omega} + q^{n-k}\binom{k+d}{d}^{\omega}\right).$$

To the best of our knowledge, the question of determining a set of parameters (d, D) which are admissible, i.e. which ensure that the algorithm terminates, remains open. An even harder question is to determine an optimal choice of parameters d and D, which would minimize both its running and its memory cost.

In [JV17], the authors mostly experimented with $D = 3$ and expressly stated that when $D \geq 4$, linear dependencies appears between the rows of $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ Indeed, for larger degree D, we need to take in account trivial relations of the form $f_i f_j = f_j f_i$. As such, the Crossbred algorithm gets slower from this point as it does useless operations. A way to improve the algorithm would be to remove those dependencies before performing linear algebra operations. This problem is already addressed in [Fau02] where Faugère propose new criterion to remove those dependencies. More specifically, we consider the criterion as proposed in [Bar04] and adapt it to the Crossbred algorithm.

3 A simplified version of Crossbred

Our first goal is to remove linear dependencies of the form $f_i f_j = f_j f_i$ and $f^2 = f$ which appear while running the preprocessing step performed on the matrix $Mac^k_{\leq D,\geq d,k}(\mathcal{F})$ in the Crossbred algorithm. For that, let us we first consider a quadratic homogeneous system of polynomials $\mathcal{F} = \{f_1, \ldots, f_m\}$. We will adapt Algorithm 1 to generate recursively matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$ and $\mathcal{M}^k_{d_1,d_2,m}$, for all $d_1 \leq D$ and $d_2 \leq D$. During the process, we will apply two criteria to sieve out polynomials that give reductions to zero when performing linear algebra on this matrix.

Notation We denote by $\widetilde{Mac}^k_{D,d,m}(\mathcal{F})$ the row-reduced echelon form of the matrix $Mac^k_{D,d,m}(\mathcal{F})$.

The following proposition is an adaptation of Faugère's General Criterion [Fau02] to the matrices used in our algorithm and by extension, to those used in Crossbred.

Proposition 1. *(The General Criterion) Let $d \leq d_2 \leq d_1 \leq D$. For any row labeled by (t, f_j) in $\widetilde{Mac}^k_{d_1-2,d_2-2,m-1}(\mathcal{F})$ having the monomial t' with $\deg_k t' = d-1$ as a leading term, the row labeled by (t', f_m) is a linear combination of previous rows in $Mac_{d_1,m}(\mathcal{F})$ and the polynomial generated from this row in the preprocessing step of the Crossbread algorithm is a linear combination of rows of the matrix $Mac_{\leq d_2,m}(\mathcal{F}^*)$.*

Proof. Assume that $t' = LT(f)$, where $f = \sum_{j=1}^{m-1} g_j f_j$ with g_j polynomials with $\deg g_i f_i = d_1 - 2$ and $\deg_k g_i = d_2 - 3$. We may then write

$$\begin{aligned} t' f_m &= f f_m - (f - LT(f)) f_m \\ &= \sum_{j=1}^{m-1} (g_j f_m) f_j - (f - LT(f)) f_m. \end{aligned}$$

Hence we have:

$$t' f_m = \sum_{j=1}^{m-1} \left(\sum_i u_{i,j} + v_{i,j} + w_{i,j} \right) f_j - \left(\sum_i u_{i,m} + v_{i,m} + w_{i,m} \right) f_m, \tag{3}$$

with $deg_k(u_{i,j}) = d_2 - 1$, $deg_k(v_{i,j}) = d_2 - 2$ and $deg_k(w_{i,j}) = d_2 - 3$ and $u_{i,m} < t'$.

The number of columns of $\mathcal{M}^k_{\leq D,>d,m}(\mathcal{F})$ is given by $t = M\!\leq D\!> d$. As explained in Section 2.2.1, in order to generate the new polynomials in the Crossbred algorithm it suffices to partially echelonize the first t rows in $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ and extract the sub-matrix corresponding to rows that only have zero coefficients on these t columns. If the row labeled (t', f_m) yields a new polynomial, then this polynomial is given by:

$$p = t' f_m + \sum_{i=1}^{m} u_{i,j} f_j.$$

Using Equation (3), we have :

$$\begin{aligned} p &= \sum_{j=1}^{m-1} \left(\sum_i v_{i,j} + w_{i,j} \right) f_j - \left(\sum_i v_{i,m} + w_{i,m} \right) f_m \\ &= \sum_{j=1}^{m} \left(\sum_i v_{i,j} + w_{i,j} \right) f_j. \end{aligned}$$

After specification, we have :

$$p^* = \sum_{j=1}^{m} \left(\sum_i v^*_{i,j} + w^*_{i,j} \right) f^*_j.$$

Note that $v^*_{i,j} f^*_j$ and $w^*_{i,j} f^*_j$ are rows of $Mac_{\leq d,m}(\mathcal{F}^*)$. We conclude that the polynomial generated with the row labeled (t', f_m) is a linear combination of rows of $Mac_{\leq d_2,m}(\mathcal{F}^*)$. $\square$

As explained in Section 2.1, for polynomial systems defined over $\mathbb{F}_2$ another set of linear dependencies appear due to the fact that for any polynomial f we have $f^2 = f$. The following proposition is an adaption of the Frobenius criterion to the matrices used in Crossbred.

Proposition 2. *(The Frobenius Criterion) Let $d \leq d_2 \leq d_1 \leq D$. For any row labeled by (t, f_j) in $\widetilde{Mac}^k_{d_1-2,d_2-2,m}(\mathcal{F})$ having the monomial t' as a leading term, the row labeled by (t', f_m) is a linear combination of previous rows in $Mac_{d_1,m}(\mathcal{F})$ and the polynomial generated with this row in the preprocessing step of the Crossbred algorithm is a linear combination of rows of $Mac_{\leq d_2,m}(\mathcal{F}^*)$.*

Proof. Let $t' = LT(f)$, where $f = \sum_{j=1}^m h_j f_j$ with h_j polynomials with $\deg h_j f_j = d_1 - 2$ and $\deg_k h_j = d_2 - 3$. We may then write :

$$\begin{aligned} t' f_m &= f f_m - (f - LT(f)) f_m \\ &= \sum_{j=1}^{m} (h_j f_m) f_j - (f - LT(f)) f_m \\ &= \sum_{j=1}^{m-1} (h_j f_m) f_j + h_m f_m^2 - (f - LT(f)) f_m \\ &= \sum_{j=1}^{m-1} (h_j f_m) f_j + h_m f_m - (f - LT(f)) f_m. \end{aligned}$$

Hence $t' f_m$ can be written as a sum of polynomials $u f_j$ with either $j \leq m$ and $\deg_k u = \{d_2 - 3, d_2 - 2, d_2 - 1\}$ or $j = m$ and $u < t'$ with respect to the chosen monomial ordering. The rest of the proof is similar to that of the General Criterion (Proposition 1). ☐

We may now present a simplified version of Algorithm 1, which takes in a homogenous system of polynomials $\mathcal{F}$ and is based on linear algebra on the matrix $Mac^k_{D,d,m}(\mathcal{F})$ (instead of $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ as in Crossbred). We recursively construct matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$, for $d \leq d_2 \leq d_1 \leq D$, and apply the criteria in Propositions 1 and 2 in the process. The pseudocode of our algorithm, that we call Block Crossbred , is given in Algorithm 2. We will see in Section 5 that this algorithm has the advantage that it is easier to analyze than the original Crossbred.

How to generate $Mac^k_{d_1,d_2,m'}(\mathcal{F})$. The **GenMat** method computes the matrix $Mac^k_{d_1,d_2,m'}(\mathcal{F})$ by adding to $Mac^k_{d_1,d_2,m'-1}(\mathcal{F})$ all rows labeled $(u, f_{m'})$ where u is a monomial with $\deg u = d_1 - 2$ and $deg_k(u) = d_2 - 1$ that does not satisfy the conditions in the General and Frobenius criteria (i.e. u is not a leading term of a row in $\widetilde{Mac}^k_{d_1-2,d_2-2,m'-1}(\mathcal{F})$.)

GenPoly. The **GenPoly** method takes in the matrix $\mathcal{M}^k_{d_1,d_2,m}$ to generate new polynomials that will be added to the initial system $\mathcal{F}$. To do that, we first compute the left kernel LK of the matrix $\mathcal{M}^k_{D,d,m}(\mathcal{F})$ is a null matrix. We then obtain new polynomials thanks to the operation $LK \cdot Mac^k_{D,d,m}(\mathcal{F})$. This computation is similar to the preprocessing step of the Crossbred algorithm, each row of the resulting matrix represents a polynomial of total degree D and of degree d over the first k variables.

4 Semi-regular sequences and Block Crossbred

A first step towards understanding the complexity of the Block Crossbred , and eventually that of Crossbred, is to evaluate the cost of its preprocessing step. In order to compute

Algorithm 2: Block Crossbred

Data: A polynomial system $\mathcal{F}$ of m homogeneous polynomials of n variables over $\mathbb{F}_2$, three parameters D, d, k
Result: A solution of the system (if it exists)

```
for d_1 from 2 to D do
    for d_2 from 1 to d - 1 do
        for m' from 1 to m do
            Mac^k_{d_1,d_2,m'}(F) <- GenMat(Mac^k_{d_1,d_2,m'-1}(F))
            Compute the echelon form ~Mac^k_{d_1,d_2,m'}(F)
        end
    end
end
Generate M^k_{D,d,m}(F) from Mac^k_{D,d,m}(F)
F' <- GenPoly(M^k_{D,d,m}(F))
for (i_1, i_2, ..., i_{n-k}) in F_2^{n-k} do
    Partially evaluate each polynomial f in F at (i_1, i_2, ..., i_{n-k})
    Compute Mac_{<=d,m}(F*)
    Partially evaluate the polynomials in F' at (i_1, i_2, ..., i_{n-k}) and append them
     to Mac_{d,m}(F*)
    if the system is consistent then
        return the solution
    end
end
```

the number of new polynomials generated by the **GenPoly** procedure, which depends on the dimensions of the kernels of the matrices $Mac^k_{D,d,m}(\mathcal{F})$ and $\mathcal{M}^k_{D,d,m}(\mathcal{F})$, we need to account for reductions to zero while constructing these matrices. We will use here a standard assumption in the literature, that of semi-regularity.
We define

$$R^k_{d_1,d_2} = \mathbb{F}_p[x_{k+1}, \ldots, x_n]_{d_1-d_2}[x_1, \ldots, x_k]_{d_2},$$

and

$$I^k_{d_1,d_2} = I_{d_1} \cap R^k_{d_1,d_2},$$

regarded as $\mathbb{F}_p$-vector spaces.

Proposition 3. *There exists a tuple* (D_1, D_2) *such that*

$$\dim_K R^k_{D_1,D_2} = \dim_K I^k_{D_1,D_2}.$$

Proof. It is well known that for all $D \geq D_{reg}$ we have $I_D = K[x_1, \ldots, x_n]_D$. Fix $D = D_{reg}$. Then for any (d_1, d_2) such that $d_1 \geq D_{reg}$ we have that

$$R^k_{d_1,d_2} \subseteq I_{d_1} = K[x_1, \ldots, x_n]_{d_1}.$$

It follows that $I^k_{d_1,d_2} = I_{d_1} \cap R^k_{d_1,d_2} = R^k_{d_1,d_2}$. □

This proposition suggests that there exists a set of parameters (d, D) with $d < D \leq D_{reg}$ such that the left kernel of the matrix $\mathcal{M}^k_{D,d,k}(\mathcal{F})$ has positive dimension, i.e. the **GenPoly** procedure generates a positive number of new polynomials. Note that $d < D \leq D_{reg}$ is the only interesting case for Crossbred anyway, since when $D = D_{reg}$ the cost of the linear algebra in the preprocessing is asymptotically close to that of linear algebra in the F5 algorithm.

Whenever the sequence of polynomials $\mathcal{F}$ is γ-strong semi-regular, we will show that there are no reductions to zero in the **GenMat** procedure of the Block Crossbred algorithm.

Proposition 4. *Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be a homogeneous system of polynomials such that the ideal $I = \langle f_1, \dots, f_m\rangle$ has dimension 0. Let $0 < k < n$ and $0 < \gamma \leq n$ such that $k = (1-\gamma)n$. Assume that the sequence $\mathcal{F}$ is γ-strong semi-regular and denote by $d_{wit}(k)$ the witness degree of $\mathcal{F}^*$ for all $(a_{k+1}, \dots a_n) \in \mathbb{F}_2^{n-k} \backslash \mathcal{S}(I)$. Then there are no reductions to zero in the matrix $Mac^k_{d_1,d_2,k}(\mathcal{F})$ constructed by the* ***GenMat*** *procedure of the Block Crossbred algorithm with $d_2 \leq d < d_{wit}(k)$ and $d_1 \leq D < D_{reg}$.*

Proof. Assume that there is a reduction to 0 in the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$, with $d_1 < D_{reg}$. Then there exist g_i and h_j, $j \in \{1, \dots i-1\}$, such that $g_i f_i = \sum_{j=1}^{i-1} h_j f_j$, with $\deg g_i f_i = d_1$ and $g_i, h_j \in R^k_{d_1-2,d_2-1}$. From the semi-regularity hypothesis, it follows that $g_i = \sum_{j=1}^{i-1} h'_j f_j$. Since the sequence is γ-strong semi-regular, there is no fall of the degree over the first k variables for the specialized system, for $d_2 < d_{wit}(k)$. We deduce that $\deg_k(h'_j) = d_2 - 3$, hence $LT(g_i)$ is the leading term of a row in $Mac^k_{d_1-2,d_2-2,m-1}(\mathcal{F})$. These are exactly the rows that are removed when applying the General and the Frobenius Criteria. As such, there is no reduction to zero in the Block Crossbred up to the degree D_{reg}. □

5 A bivariate generating series for Block Crossbred

In this section, we investigate the complexity of the Block Crossbred algorithm for solving a system of polynomials. To this purpose, we have to estimate first the number of new polynomials obtained when running the **GenPoly** procedure in Algorithm 2.

Let $\mathcal{F} = \{f_1, \dots f_m\}$ be a system of homogenous quadratic polynomials in R and denote by $U^k_{d_1,d_2,m}$, $d_1, d_2 \geq 0$, the number of rows of the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$, and thus of $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$. The number of columns of $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$ is given by $M^k_{d_1,d_2+1}$, which corresponds to the number of monomials v of total degree d_1 such that $\deg_k v = d_2 + 1$.

We define the following sequence:

$$h^k_{d_1,d_2,m} = \begin{cases} U^k_{d_1,d_2,m} - M^k_{d_1,d_2+1}, & \text{si } d_1 \geq d_2 \geq 0, \\ -M^k_{d_1,0}, & \text{si } d_1 > 0, d_2 = -1, \\ 0 & \text{in all other cases.} \end{cases} \tag{4}$$

The dimension of this space gives the number of new "independent" polynomials generated in the preprocessing step of Algorithm 2.

Proposition 5. *If $h^k_{d_1,d_2,m} > 0$ and there are no reductions to zero in $Mac^k_{d_1,d_2,m}(\mathcal{F})$, then then the number of polynomials computed with the* ***GenPoly*** *procedure is $h^k_{d_1,d_2,m}$.*

Proof. Since $h^k_{d_1,d_2,m} > 0$ and the matrix $\mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$ has full rank, we have that $h^k_{d_1,d_2,m} = \operatorname{corank} \mathcal{M}^k_{d_1,d_2,m}(\mathcal{F})$. □

Proposition 6. *Assume that there are no reductions to zero in the Block Crossbred algorithm. Then the sequence $h^k_{d_1,d_2,m}$ satisfies the following recurrence relation:*

$$h^k_{d_1,d_2,m} = h^k_{d_1,d_2,m-1} - h^k_{d_1-2,d_2-2,m}, \tag{5}$$

with the initial condition $h^k_{d_1,d_2,0} = -M^k_{d_1,d_2+1}$, for all $d_1, d_2 \in \mathbb{Z}$.

Proof. The number of rows added to $Mac^k_{d_1,d_2,m-1}(\mathcal{F})$ to get $Mac^k_{d_1,d_2,m}(\mathcal{F})$ is equal to the number of monomials u with $\deg(u) = d_1 - deg(f_m) = d_1 - 2$ and $\deg_k(u) = d_2 - 1$. From this number we subtract the number of monomials which satisfy the General and the Frobenius Criterion. As such, the number of rows of the matrix $Mac^k_{d_1,d_2,m}(\mathcal{F})$ verifies the following equation :

$$U^k_{d_1,d_2,m} - U^k_{d_1,d_2,m-1} = M^k_{d_1-2,d_2-1} - U^k_{d_1-2,d_2-2,m}.$$

By using this formula and Equation (4) we get:

$$h^k_{d_1,d_2,m} - h^k_{d_1,d_2,m-1} = -h^k_{d_1-2,d_2-2,m},$$

which concludes the proof. □

Using the recurrence relation in Equation (5) we may now compute the generating bivariate series which will allow us to determine admissible parameters for Algorithm 2.

Theorem 7. *Let* $H^k_{m,n}(X,Y) = \sum_{d_1\geq 0, d_2\geq 0} h^k_{d_1,d_2,m} X^{d_1} Y^{d_2}$ *be the bivariate series with coefficients defined by Equation* (4)*. This series is given by:*

$$H^k_{m,n}(X,Y) \quad = \quad \frac{1}{Y}\left[(1+X)^{n-k} - \frac{(1+XY)^k(1+X)^{n-k}}{(1+X^2Y^2)^m}\right].$$

Proof. Since the values of k and n are fixed and let $H_m(X,Y) = H^k_{m,n}(X,Y)$ and write $h_{d_1,d_1,m}$ instead of $h^k_{d_1,d_2,m}$. Moreover, we define

$$\hat{H}_m(X,Y) = \sum_{d_1\geq 0, d_2\geq 0} h_{d_1-2,d_2-2,m} X^{d_1} Y^{d_2}.$$

Then, we have

$$\begin{aligned}
\hat{H}_m(X,Y) &= \sum_{d_1\geq 0, d_2\geq 0} U^k_{d_1-2,d_2-2,m} X^{d_1} Y^{d_2} - \sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1-2,d_2-1} X^{d_1} Y^{d_2} \\
&= X^2Y^2 \sum_{d_1\geq 0, d_2\geq 0} U^k_{d_1,d_2,m} X^{d_1} Y^{d_2} \\
&- X^2Y^2 \sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1,d_2+1} X^{d_1} Y^{d_2} - X^2Y \sum_{d_1\geq 0} M^k_{d_1,0} X^{d_1} \\
&= X^2Y^2 \sum_{d_1\geq 0, d_2\geq 0} h_{d_1,d_2,m} X^{d_1} Y^{d_2} - X^2Y \sum_{d_1\geq 0} M^k_{d_1,0} X^{d_1} \\
&= X^2Y^2 H_m(X,Y) - X^2Y \sum_{d_1\geq 0} M^k_{d_1,0} X^{d_1}.
\end{aligned}$$

Using the recurrence relation obtained in Equation (5) we obtain:

$$\begin{aligned}
H_m(X,Y) &= \sum_{d_1\geq 0, d_2\geq 0} h_{d_1,d_2,m} X^{d_1} Y^{d_2} \\
&= \sum_{d_1\geq 0, d_2\geq 0} h_{d_1,d_2,m-1} X^{d_1} Y^{d_2} - \sum_{d_1\geq 0, d_2\geq 0} h_{d_1-2,d_2-2,m} X^{d_1} Y^{d_2} \\
&= H_{m-1}(X,Y) - \hat{H}_m(X,Y) \\
&= H_{m-1}(X,Y) - X^2Y^2 H_m(X,Y) + X^2Y \sum_{d_1\geq 0} M^k_{d_1,0} X^{d_1}.
\end{aligned}$$

Hence we get

$$\begin{aligned} H_m(X,Y) &= (1+X^2Y^2)^{-1}H_{m-1}(X,Y) + \frac{X^2Y}{1+X^2Y^2}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1} \\ &= (1+X^2Y^2)^{-m}H_0(X,Y) - \frac{1-(1+X^2Y^2)^m}{Y(1+X^2Y^2)^m}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1}. \end{aligned}$$

By Equation (4), for a fixed value fo k, we have that $h_{d_1,d_2,0} = -M^k_{d_1,d_2+1}$. Hence we get :

$$H_0(X,Y) = -\sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1,d_2+1}X^{d_1}Y^{d_2}.$$

Since $M^k_{d_1,d_2+1} = \binom{k}{d_2+1}\binom{n-k}{d_1-d_2-1}$ (with the convention that $\binom{n-k}{d_1-d_2-1} = 0$ for $d_2 \geq d_1$) we get:

$$\begin{aligned} \sum_{d_1\geq 0, d_2\geq 0} M^k_{d_1,d_2+1}X^{d_1}Y^{d_2} &= \sum_{d_1\geq 0, d_2\geq 0} \binom{k}{d_2+1}\binom{n-k}{d_1-d_2-1}X^{d_1}Y^{d_2} \\ &= \sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2}\sum_{d_1\geq 0}\binom{n-k}{d_1-d_2-1}X^{d_1} \\ &= \sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2}X^{d_2+1}\sum_{d_1\geq 0}\binom{n-k}{d_1}X^{d_1} \\ &= \sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2}X^{d_2+1}(1+X)^{n-k} \\ &= \frac{(1+X)^{n-k}}{Y}\sum_{d_2\geq 0}\binom{k}{d_2+1}Y^{d_2+1}X^{d_2+1} \\ &= \frac{(1+X)^{n-k}}{Y}\left(\sum_{d_2\geq 0}\binom{k}{d_2}Y^{d_2}X^{d_2} - 1\right) \\ &= \frac{(1+X)^{n-k}}{Y}\left((1+XY)^k - 1\right). \end{aligned}$$

In conclusion, we have :

$$\begin{aligned} H_m(X,Y) &= -\frac{(1+X)^{n-k}}{Y(1+X^2Y^2)^m}\left((1+XY)^k-1\right) - \frac{1-(1+X^2Y^2)^m}{Y(1+X^2Y^2)^m}\sum_{d_1\geq 0} M^k_{d_1,0}X^{d_1} \\ &= \frac{1}{Y(1+X^2Y^2)^m}\left[-(1+XY)^k(1+X)^{n-k} + (1+X^2Y^2)^m(1+X)^{n-k}\right] \\ &= \frac{1}{Y}\left[(1+X)^{n-k} - \frac{(1+XY)^k(1+X)^{n-k}}{(1+X^2Y^2)^m}\right]. \end{aligned}$$

□

6 From Block Crossbred to Joux-Vitse's Crossbred

The matrix $Mac^k_{\leq D,\geq d,m}(\mathcal{F})$ is constructed by concatenating the matrices $Mac^k_{d_1,d_2,m}(\mathcal{F})$, $d \leq d_2 \leq d_1 \leq D$, constructed in Algorithm 2. This observation allows us to computed the corank of the matrix $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ in terms of the coefficients of the generating series $H(X,Y)$ examined in Proposition 7.

Proposition 8. *Assuming that there are no reductions to zero in the preprocessing step of the Crossbred algorithm, the corank of the matrix $\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})$ is given by the following formula:*

$$corank(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) = \sum_{d_1\leq D, d_2\geq d,m} h^k_{d_1,d_2,m},$$

where the sequence $h^k_{d_1,d_2,m}$ is defined by Equation (4).

Proof. Indeed, we have that:

$$\begin{aligned} corank(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) &= \#Rows(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) - \#Col(\mathcal{M}^k_{\leq D,\geq d,m}(\mathcal{F})) = \\ &= \sum_{d_1\leq D,d_2\geq d} U^k_{d_1,d_2} - \sum_{d_1\leq D,d_2\geq d} M^k_{d_1,d_2+1} = \sum_{d_1\leq D,d_2\geq d,m} h^k_{d_1,d_2,m}. \end{aligned}$$

□

We are now in position to compute the generating bivariate series which will eventually allow us to determine admissible parameters for Algorithm 1.

Proposition 9. *For fixed values of m, n and k the bivariate series $G^k_{m,n}(X,Y) = \sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_1'\leq d_1,d_2'\geq d_2} h^k_{d_1',d_2',m}\right) X^{d_1}Y^{d_2}$ is given by the formula:*

$$G^k_{m,n}(X,Y) = -\frac{YH^k_{m,n}(X,Y)}{(1-X)(1-Y)}. \tag{6}$$

Proof. Since m, n and k are fixed, we compute the bivariate series $G(X,Y) = G^k_{m,n}(X,Y)$ as follows:

$$G(X,Y) = \sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_1'\leq d_1,d_2'\geq d_2} h_{d_1',d_2',m}\right) X^{d_1}Y^{d_2}.$$

First note that

$$\begin{aligned} \sum_{d_1'\leq d_1,d_2'\geq d_2} h_{d_1',d_2',m}X^{d_1}Y^{d_2} =& h_{d_1,d_2,m}X^{d_1}Y^{d_2} + \sum_{d_1'\leq d_1-1} h_{d_1',d_2,m}X^{d_1}Y^{d_2} \\ &+ \sum_{d_2'\geq d_2+1} h_{d_1',d_2,m}X^{d_1}Y^{d_2} + \sum_{d_1'\leq d_1-1,d_2'\geq d_2+1} h_{d_1',d_2',m}X^{d_1}Y^{d_2}. \end{aligned}$$

It follows that

$$\begin{aligned} G(X,Y) = \sum_{d_1,d_2} h_{d_1,d_2,m}X^{d_1}Y^{d_2} + \frac{X}{Y}\sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_1'\leq d_1-1,d_2'\geq d_2+1} h_{d_1',d_2',m}\right) X^{d_1-1}Y^{d_2+1} \\ + \sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_1'\leq d_1-1} h_{d_1',d_2,m}\right) X^{d_1}Y^{d_2} + \sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_2'\geq d_2+1} h_{d_1,d_2',m}\right) X^{d_1}Y^{d_2}. \end{aligned} \tag{7}$$

We denote by S the sequence

$$S(X,Y) = \sum_{d_1\geq 0,d_2\geq 0}\left(\sum_{d_1'\leq d_1-1} h_{d_1',d_2,m}\right) X^{d_1-1}Y^{d_2},$$

and by T the sequence

$$T(X,Y) = \sum_{d_1 \geq 0, d_2 \geq 0} \left(\sum_{d_2' \geq d_2+1} h_{d_1,d_2',m} \right) X^{d_1} Y^{d_2+1}.$$

Then we write

$$\begin{aligned} S(X,Y) &= \sum_{d_1 \geq 1, d_2 \geq 0} h_{d_1-1,d_2} X^{d_1-1} Y^{d_2} + \sum_{d_1 \geq 1, d_2 \geq 0} \sum_{d_1' \leq d_1-2} h_{d_1',d_2} X^{d_1-1} Y^{d_2} \\ &= \sum_{d_1 \geq 0, d_2 \geq 0} h_{d_1,d_2} X^{d_1} Y^{d_2} + X \sum_{d_1 \geq 0, d_2 \geq 0} \sum_{d_1' \leq d_1-2} h_{d_1',d_2} X^{d_1-2} Y^{d_2} \\ &= H(X,Y) + XS(X,Y), \end{aligned}$$

and get that

$$S(X,Y) = \frac{H(X,Y)}{1-X}. \tag{8}$$

To compute $T(X,Y)$ we follow a similar approach:

$$\begin{aligned} T(X,Y) = \sum_{d_1 \geq 0, d_2 \geq 0} \sum_{d_2' \geq d_2+1} h^k_{d_1,d_2',m} X^{d_1} Y^{d_2+1} &= \sum_{d_1 \geq 0, d_2 \geq 1} \sum_{d_2' \geq d_2} h^k_{d_1,d_2',m} X^{d_1} Y^{d_2} \\ = \sum_{d_1 \geq 0, d_2 \geq 0} \sum_{d_2' \geq d_2} h^k_{d_1,d_2',m} X^{d_1} Y^{d_2} &- \sum_{d_1 \geq 0} \sum_{d_2' \geq 0} h^k_{d_1,d_2',m} X^{d_1} \\ = \sum_{d_1 \geq 0, d_2 \geq 0} \sum_{d_2' \geq d_2+1} h^k_{d_1,d_2',m} X^{d_1} Y^{d_2} &+ \sum_{d_1 \geq 0, d_2 \geq 0} \sum_{d_2' = d_2} h^k_{d_1,d_2',m} X^{d_1} Y^{d_2} \\ - \sum_{d_1 \geq 0} \sum_{d_2' \geq 0} h^k_{d_1,d_2',m} X^{d_1} &= \frac{1}{Y} T(X,Y) + H(X,Y) - H(X,1). \end{aligned}$$

We conclude that :

$$T(X,Y) = \frac{Y}{Y-1} \left[H(X,Y) - H(X,1) \right].$$

Now let us focus on the series

$$\hat{G} = \sum_{d_1 \geq 0, d_2 \geq 0} \left(\sum_{d_1' \leq d_1-1, d_2' \geq d_2+1} h_{d_1',d_2',m} \right) X^{d_1-1} Y^{d_2+1},$$

which appears in the second term of the sum in Equation (7). We have that

$$\begin{aligned} \hat{G} &= \sum_{d_1 \geq 0, d_2 \geq 0} \left(\sum_{d_1' \leq d_1, d_2' \geq d_2+1} h_{d_1',d_2',m} X^{d_1} Y^{d_2+1} \right) \\ &= \sum_{d_1 \geq 0, d_2 \geq 0} \left(\sum_{d_1' \leq d_1-1, d_2' \geq d_2+1} h_{d_1',d_2',m} \right) X^{d_1} Y^{d_2+1} + \sum_{d_1 \geq 0, d_2 \geq 0} \left(\sum_{d_2' \geq d_2+1} h_{d_1,d_2',m} \right) X^{d_1} Y^{d_2+1}. \end{aligned}$$

We have

$$\hat{G}(X,Y) = X\hat{G}(X,Y) + T(X,Y),$$

hence we compute

$$\hat{G}(X,Y) = \frac{1}{1-X}T(X,Y).$$

Finally, we obtain

$$\begin{aligned} G(X,Y) &= H(X,Y) + \frac{X}{Y(1-X)}T(X,Y) + XS(X,Y) + \frac{1}{Y}T(X,Y) \\ &= H(X,Y) + XS(X,Y) + \frac{1}{(1-X)Y}T(X,Y). \end{aligned} \tag{9}$$

We plug in the expressions obtained in Equations (8) and (6) in the last equality in Equation (9) and conclude that

$$G(X,Y) = \frac{H(X,1)}{(1-X)(1-Y)} - \frac{YH(X,Y)}{(1-X)(1-Y)}.$$

□

For a fixed value of k, the non-zero coefficients of $G^k_{m,n}$ give us values of (d_1, d_2) for which the left kernel of $Mac^k_{\leq d_1, \geq d_2, k}(\mathcal{F})$ is non-trivial. Consequently, for these pairs (d_1, d_2) the number of polynomial generated during the pre-processing step, taking into account the criteria, is given by the coefficient of $X^{d_1}Y^{d_2}$.

Example 6.1. We are interested in solving polynomial systems with $m = 166$ polynomials and $n = 83$ variables, which is the set of parameters for the current record of a polynomial system that was solved over $\mathbb{F}_2$ in the Fukuoka Type I MQ challenge [BS23]. By choosing $k = 30$, we get the following series :

$$\begin{aligned} G^{30}_{166,83}(X,Y) = \;& - \; 30X - 1889X^2 - 269X^2Y - 56566X^3 - 13606X^3Y + 920X^3Y^2 \\ & - \; 1050324X^4 - 304584X^4Y + 80624X^4Y^2 \\ & + \; 30944X^4Y^3 + \mathcal{O}(X^5). \end{aligned}$$

With this series, we know the number of polynomials generated by the pre-processing of the Crossbred algorithm (when applying criterion) for parameters $(4, 3, 30)$ is 30944.

6.1 Admissible parameters

Now that we know how many polynomials are generated by the pre-processing step of the algorithm, we need to check if it generate enough polynomials for the algorithm to terminate. To tackle the question of determining admissible parameters for Crossbred, let us look at a toy example.

Example 6.2. For a system of $m = 49$ polynomials and $n = 23$ variables, if we choose $k = 18$, the corresponding series is :

$$G^{18}_{49,23}(X,Y) = -18X - 212X^2 - 104X^2Y - 846X^3 - 558X^3Y + 66X^3Y^2 + \mathcal{O}(X^4).$$

Choosing parameters $(D, d) = (3, 2)$, we will generate 66 polynomials. Assuming these polynomials are linearly independent after specification of the last $n-k$ variables, we claim that this is not enough to linearize the specified system. Indeed, since $d = 2$, the Macaulay matrix $Mac_{\leq 2,49}(\mathcal{F}^*)$ has $M^{18}_{\leq 2} = 172$ columns. After adding the 66 new polynomials, this matrix has 115 rows. We conclude that $(D, d) = (3, 2)$ and $k = 18$ are not admissible parameters for this polynomial system.

Notation Let $R' = \mathbb{F}_p[x_1, \ldots, x_k]$. Let $\mathcal{F} = \{f_1, \ldots f_m\}$ be a system of polynomials in R and denote as usual by I the ideal generated by $f_1, \ldots, f_m$. Then we denote by $I^* = \langle f_1^*, \ldots, f_m^* \rangle$, where f_i^* are obtained by specifying the variables $x_{k+1}, \ldots, x_n$ at any values $(a_1, \ldots, a_{n-k}) \in \mathbb{F}_2^{n-k}$.

As explained in Section 2, the witness degree of an affine semi-regular polynomial system with k variables and m equations is given by the index of the first non-positive coefficient of the generating series in Lemma 1. We denote the value of this index by $d_{wit}(k)$. Obviously, if $\mathcal{F}$ is γ-strong semi-regular, this implies that for any $d < d_{wit}(k)$, all rows of $Mac_{\leq d,m}(\mathcal{F}^*)$ are linearly independent.

When $d \geq d_{wit}(k)$, the matrix $Mac_{\leq d,m}(\mathcal{F}^*)$ has more rows than columns and it has full rank. In this case, d is not interesting as input parameter for the Crossbred algorithm since we do not need any new polynomials generated in the pre-processing step. Indeed, in this case it suffices to perform exhaustive search, assign the last $n-k$ variables in the system and solve it (for instance by linearisation on the $Mac_{\leq d,m}(\mathcal{F}^*)$ matrix).

This leads to the following definition.

Definition 6. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a sequence of polynomials in $\mathbb{F}_2[x_1, \ldots, x_n]$ and k and γ are such that $0 \leq k = (1-\gamma)n \leq n$ and $\mathcal{F}$ is γ-strong semi-regular. The parameters D, d and k are called potentially admissible for the Crossbred algorithm if the following hold:

(1) $d < d_{wit}(k)$,

(2) $D < D_{reg}$,

(3) For all $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k} \backslash \mathcal{S}(I)$ and the ideal $I^* = \langle f_1^*, \ldots, f_m^* \rangle$ obtained by evaluating $f_1, \ldots, f_m$ at $(a_{k+1}, \ldots, a_n)$ we have that:

$$\sum_{d_1 \leq D, d_2 \geq d} h^k_{d_1,d_2,m} + \dim I^*_{\leq d} \geq \dim R'_{\leq d}.$$

We now show that if the system $\mathcal{F}$ is γ-strong semi-regular, we compute the generating series which determines potentially admissible parameters for the Crossbred algorithm.

Theorem 10. *Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be a γ-strong semi-regular sequence of polynomials in $\mathbb{F}_2[x_1, \ldots x_n]$. Then k, D and d are potentially admissible parameters for the Crossbred algorithm if the coefficient corresponding to $X^D Y^d$ of the following bivariate series*

$$J^k_{m,n}(X,Y) = \frac{1}{(1-X)(1-Y)} \left[\frac{(1+X)^{n-k}(1+XY)^k}{(1+X^2Y^2)^m} - \frac{(1+X)^n}{(1+X^2)^m} \right. \\ \left. -(\frac{1}{Y} - 1)(1+X)^{n-k} - \frac{(1+Y)^k}{(1+Y^2)^m} \right] \tag{10}$$

is non-negative.

Proof. Using Definition 6, to determine potentially admissible parameters we look at the coefficients of the following bivariate series :

$$G^k_{m,n}(X,Y) - \sum_{D,d \geq 0} \dim(R'_{\leq d}/I^*_{\leq d}) X^D Y^d, \tag{11}$$

where $I^* = \langle f_1^*, \ldots, f_m^* \rangle$ is the ideal obtained by specializing $f_1, \ldots, f_m$ at any $(a_{k+1}, \ldots, a_n) \in \mathbb{F}_2^{n-k} \backslash \mathcal{S}(I)$. Using Lemma 1 we have that:

$$\sum_{0 \leq d} \left(\dim R'_{\leq d}/J_{\leq d} \right) Y^d = \frac{(1+Y)^k}{(1-Y)(1+Y^2)^m}. \tag{12}$$

Then by replacing $G^k_{m,n}(X,Y)$ with its expression computed in Proposition 7, we get the series claimed in the statement of the theorem. $\square$

Example 6.3. Let's take the same polynomials system as in example 6.1. We compute the degree of regularity of $\mathcal{F}$ and the witness degree $d_{wit}(k)$ of $\mathcal{F}^*$ and get $D_{reg} = 9$ and $d_{wit}(30) = 3$. We compute $J^k_{m,n}(X,Y)$ for this sytem :

$$\begin{aligned} J^{30}_{166,83}(X,Y) = & - \quad 30X - 1889X^2 - 300X^2Y - 56566X^3 - 13637X^3Y + 620X^3Y^2 \\ & - \quad 1050324X^4 - 304615X^4Y + 80324X^4Y^2 + 31564X^4Y^3 + \mathcal{O}(X^5) \end{aligned}$$

Following the condition given in definition 6, we know that $(D, d, k) = (4, 3, 30)$ is not a potentially admissible parameter as $d = d_{wit}(30) = 3$. On the other hand, parameter $(D, d) = (3, 2)$ is potentially admissible as it satisfies every condition of definition 6.

In Section 7 we show experimental evidence supporting the conjecture that potentially admissible parameters are indeed admissible (see Table 3).

7 Experiments

7.1 γ-strong semi-regularity

In this subsection, we show experimental evidence supporting the claim that random polynomials system are γ-strong semi-regular. In particular, we will see that a random polynomial system, which we know to be semi-regular [BFS03], is still semi-regular after specification with high probability. Furthermore, we will also try to give an upper bound for γ.

To test the semi-regularity of the specialised system $\mathcal{F}^*$, we look at the rank of the associated Macaulay matrix for each degree up to the witness degree $d_{wit}(k)$. In Table 1 we show that experimental result confirms our assumption. In the fourth column of this Table, we give the value of the witness degree $d_{wit}(k)$, computed using the series in Lemma 1. In the fifth column, we give the value of γ, rounded with three decimals. In the seventh column, we computed the numbers of rows and columns of $Mac_{\leq d,m}(\mathcal{F}^*)$ and its rank, for successive values of d. Since the matrix has less rows than columns and it has full rank for $d < d_{wit}(k)$, we conclude that $\mathcal{F}^*$ is semi-regular and $\mathcal{F}$ is γ-strong semi-regular.

Note that this does not hold for every value of γ. Indeed, if the witness degree $d_{wit}(k)$ is small enough ($d_{wit}(k) \leq 2$), then linear dependencies will appear in degree 2 in the specialised system. That lead us to compute a lower bound on k (which is equivalent to an upper bound on γ). To find it, we search k such that the number of columns of $Mac_{\leq 2,m}(\mathcal{F}^*)$ is less that the number of rows, which gives the following inequality :

$$k^2 + k + 2(1-m) \quad < \quad 0 \tag{13}$$

The polynomial on the left hand-side of Equation (13) has two roots:

$$k_{1,2} \quad = \quad \frac{-1 \pm \sqrt{8m-7}}{2}$$

We ignore k_2 since it is negative and get that k_1 yields a lower bound on the values of k for which $\mathcal{F}$ is γ-strong semi-regular series. For $m = 49$, this is equal to $k_1 \approx 9.31$. As such, for a system $\mathcal{F}$ of 49 polynomials, if $k \leq 9$ (which corresponds to $\gamma \approx 0.609$), then $\mathcal{F}$ is not γ-strong semi-regular as the specialised system is not semi-regular.

Table 1: Experimental data for γ-strong semi-regularity

m	n	k	$d_{wit}(k)$	γ	d	(# rows, # columns)	Rank of $Mac_{\leq d,m}(\mathcal{F}^*)$
49	23	18	4	0.217	2	$(49, 172)$	49
					3	$(931, 988)$	931
					4	$(8428, 4048)$	≈ 4048
49	23	17	3	0.261	2	$(49, 154)$	49
					3	$(882, 834)$	≈ 834
49	23	12	3	0.478	2	$(49, 79)$	49
					3	$(637, 299)$	≈ 299
53	25	19	4	0.24	2	$(53, 191)$	53
					3	$(1060, 1160)$	1060
					4	$(10123, 5036)$	≈ 5036

7.2 Admissible parameters for Crossbred

We implemented Algorithm 1 in Magma including the General and the Frobenius criterion when constructing Macaulay matrices (see Prop. 1 and 2) and ran experiments over pseudo-random polynomial systems. Polynomials systems used in the experiments are obtained by using Sedlacek's implementation [Sed22] of Beullens's differential attack on Rainbow instances [DS05, Beu22]. We also experimented with polynomials systems from the Fukuoka Type I MQ challenge [YDH+15, Yas15].

Data in Tables 2, 3 and 4 is obtained by generating polynomials in the preprocessing step of Algorithm 1 for different choices of parameters. In these Tables we use the following notation :

- As usual D, d and k are the input parameters for the algorithm and m and n denote the number of polynomials and the number of variables of the system, respectively.
- r corresponds to the number of polynomial generated by the precomputation step of the Crossbred algorithm.
- $M^k_{\leq d}$ denotes the number of monomials of degree $\leq d$ over k variables. This is the number of polynomials needed to successfully solve the degree d system obtained after assigning the last $n-k$ variables, by linearization.

Recall that during the exhaustive search step of the algorithm we evaluate the newly generated polynomials in the last $n-k$ variables and add them to the degree d Macaulay matrix of the specialized system $Mac_{\leq d,m}(\mathcal{F}^*)$. Then we count how many independent polynomials there are for each iteration of the exhaustive search. Each of the couples for an entry in the last column of Table 2 stands for the number of independent polynomials and the number of iterations of the exhaustive search for which we obtained this value. In Table 2, we experimented using 5 polynomial systems obtained using the generator in [Sed22], using a different seed each time to ensure that these systems are distinct.

Table 2: Example with 5 polynomial systems

seed	(D, d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)
261	$(4, 1)$	59	28	20	108	21	256	$(20, 1)$ $(21, 255)$
262	$(4, 1)$	59	28	20	108	21	256	$(20, 1)$ $(21, 255)$
263	$(4, 1)$	59	28	20	108	21	256	$(20, 1)$ $(21, 255)$
264	$(4, 1)$	59	28	20	108	21	256	$(20, 1)$ $(21, 255)$
265	$(4, 1)$	59	28	20	108	21	256	$(20, 2)$ $(21, 254)$
-	$(4, 1)$	59	28	20	108	21	256	$(20, 1.2)$ $(21, 254.8)$

As expected, for each of the 5 polynomial systems in the Table the preprocessing step of Algorithm 1 outputs exactly the same numbers of polynomials, which is 108. Since

$n = 28$ and $k = 20$, we search through $2^{n-k} = 256$ different values for the last $n - k$ variables. We see that for all possible values, except for one or two, the maximal number of independent polynomial after specification is $21 = k + 1$. This is similar to the test of consistency done in [BFSS13] in the sense that if the ideal I^* has no solution, then $corank(Mac_{\leq 1,59}(\mathcal{F}^*) = 0$, which means that the matrix has full rank. Otherwise, if the ideal I^* has a solution, then $corank(Mac_{\leq 1,59}(\mathcal{F}^*) \neq 0$ which implies the matrix will not reach full rank. We see that each seed has one solution except for seed 265 which has two solutions.

The last row in Table 2 computes the average number of independent polynomials obtained after specification, for this set of 5 polynomial systems.

In Table 3 we re-do the experiment and compute the same average, for different sets of admissible parameters. Whenever $m = 2n$, the data is obtained with polynomials from the Fukuoka MQ challenge. To obtain this data for the Fukuoka MQ challenge, we took the five available polynomial systems available in [Yas15] for any n and m, and computed the average of the result for each system. Every polynomial system gave the same result in the experiment. When $m \neq 2n$, the data is obtained with polynomials systems generated by Sedlacek's implementation. For that, we generated distinct polynomials systems with different seeds. The number of generated polynomials is the same for each seed, which was expected, and the number of solution in each system varies between one or two depending on the seed.

Finally, Table 4 shows similar experiments for non-admissible parameters, i.e. when $m + r < M^k_{\leq d}$. In this case, we see that the $m + r$ polynomials of degree d are independent after specification, which confirms our γ-strong semi-regularity hypothesis.

Table 3: Experimental data on the Crossbred algorithm for admissible parameters

(D, d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)	D_{reg}	$d_{wit}(k)$
$(4, 2)$	49	23	18	3608	172	32	$(171, 1.3)$ $(172, 30.7)$	4	4
$(4, 2)$	49	23	17	4130	154	64	$(153, 1.1)$ $(154, 62.9)$	4	3
$(4, 2)$	40	20	17	2240	154	8	$(153, 1)$ $(154, 7)$	4	4
$(4, 1)$	49	23	18	1944	19	32	$(18, 1.3)$ $(19, 30.7)$	4	4
$(4, 1)$	49	23	17	2216	18	64	$(17, 1.1)$ $(18, 62.9)$	4	3
$(4, 1)$	40	20	17	1568	18	8	$(17, 1)$ $(18, 7)$	4	4
$(3, 1)$	47	22	11	256	12	2048	$(11, 1.2)$ $(12, 2046.8)$	4	3

References

[Bar04] Magali Bardet. *Etude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Paris 6, 2004.

[BCC+10] Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In *Cryptographic Hardware and Embedded Systems, CHES 2010*, pages 203–218. Springer Berlin Heidelberg, 2010.

Table 4: Experimental data on the Crossbred algorithm for non-admissible parameters

(D, d)	m	n	k	r	$M^k_{\leq d}$	2^{n-k}	(#Ind. pol., #Iteration)	D_{reg}	$d_{wit}(k)$
$(3, 2)$	49	23	18	66	172	32	$(115, 32)$	4	4
$(3, 2)$	53	25	19	38	191	64	$(91, 64)$	4	4
$(3, 2)$	55	26	19	76	191	128	$(131, 128)$	5	4
$(3, 2)$	57	27	19	114	191	256	$(171, 256)$	5	4

[BCT+24] John Baena, Daniel Cabarcas, Sharwan K. Tiwari, Javier Verbel, and Luis Villota. Admissible parameters for the Crossbred algorithm and semi-regular sequences over finite fields. https://eprint.iacr.org/2024/758, 2024.

[Beu22] Ward Beullens. Breaking Rainbow takes a weekend on a laptop. Cryptology ePrint Archive, Paper 2022/214, 2022. https://eprint.iacr.org/2022/214. URL: https://eprint.iacr.org/2022/214.

[BFS03] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over F_2 with solutions in F_2. Research Report RR-5049, INRIA, 2003. URL: https://inria.hal.science/inria-00071534.

[BFSS13] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic Boolean systems. *Journal of Complexity*, 29(1):53–75, 2013.

[BS23] Charles Bouillaguet and Julia Sauvage. High-performance xbred. https://gitlab.lip6.fr/almasty/hpXbred, 2023.

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

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 392–407. Springer Berlin Heidelberg, 2000.

[DCO97] J. Little D. Cox and D. O'Shea. *Ideals, Varieties and Algorithms*. 1997.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 164–175, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Dua23] João Diogo Duarte. On the Complexity of the Crossbred Algorithm. https://eprint.iacr.org/2023/1664.pdf, 2023.

[Fau99] Jean-Charles Faugére. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1):61–88, 1999. URL: https://www.sciencedirect.com/science/article/pii/S0022404999000055, doi:https://doi.org/10.1016/S0022-4049(99)00005-5.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner basis without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, ISSAC '02, pages 75–83, New York, NY, USA, 2002. ACM. URL: http://doi.acm.org/10.1145/780506.780516.

[FY79] A.S. Fraenkel and Y. Yesha. Complexity of problems in games, graphs and algebraic equations. *Discrete Applied Mathematics*, 1(1):15–30, 1979. URL: https://www.sciencedirect.com/science/article/pii/0166218X7990012X, doi:https://doi.org/10.1016/0166-218X(79)90012-X.

[JV17] Antoine Joux and Vanessa Vitse. A crossbred algorithm for solving boolean polynomial systems. In *Number-Theoretic Methods in Cryptology - NuTMiC 2017*, volume 10737 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2017. URL: https://doi.org/10.1007/978-3-319-76620-1_1.

[Laz] Daniel Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations, editor =J.A. van Hulzen, booktitle=EUROCAL 1983, series=Lecture Notes in Computer Science, volume=162, pages=146–156, publisher=Springer, Heidelberg, year=1983,.

[Nak24] Shuhei Nakamura. Admissible parameter sets and complexity estimation of Crossbred algorithm. https://eprint.iacr.org/2023/1687.pdf, 2024.

[NNY17] Ruben Niederhagen, Kai-Chun Ning, and Bo-Yin Yang. https://github.com/kcning/mqsolver, 2017.

[NNY18] Ruben Niederhagen, Kai-Chun Ning, and Bo-Yin Yang. Implementing joux-vitse's crossbred algorithm for solving MQ systems over GF(2) on gpus. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 121–141. Springer, 2018.

[Sed22] Vladimír Sedláček. mq-comparaison-suite. https://github.com/VladaSedlacek/mq-comparison-suite, 2022.

[TW12] Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, pages 156–171. Springer Berlin Heidelberg, 2012.

[Yas15] Takanori Yasuda. Fukuoka MQ Challenge. https://www.mqchallenge.org, 2015.

[YDH+15] Takanori Yasuda, Xavier Dahan, Yun-Ju Huang, Tsuyoshi Takagi, and Kouichi Sakurai. MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems. https://eprint.iacr.org/2015/275, 2015.