On the Complexity and Admissible Parameters of the Crossbred Algorithm

João Diogo Duarte

Royal Holloway University of London*, United Kingdom

Abstract. The Joux–Vitse Crossbred algorithm's aim is to efficiently solve a system of semi-regular multivariate polynomials equations. The authors tested their algorithm for boolean polynomials in $\mathbb{F}_2$ and stated that this algorithm also works for other non-boolean finite fields. In addition, the algorithm is dependent on a set of parameters that control its course. Finding functional parameters for this algorithm is a non-trivial task, so the authors presented a bivariate generating series to test the admissibility of parameters in $\mathbb{F}_2$. However, due to restrictive page limits, the series itself and its derivation are not explained. In this work, the derivation of the bivariate generating series to test the admissibility of parameters in boolean $\mathbb{F}_2$ is explained and from this, a new series for other non-boolean fields, $\mathbb{F}_{q>2}$ is presented. In addition, a complexity estimate of the algorithm is given for both $\mathbb{F}_2$ and $\mathbb{F}_{q>2}$. By obtaining optimal parameters using the previous results, the cost of applying Crossbred to polynomial systems of various sizes, numbers of variables and values of q was plotted. Overall, it was determined that the Crossbred algorithm provides an improved complexity over FES (Fast Exhaustive Search) for larger overdetermined systems, but for any overdetermined system, it does not improve the complexity when compared to state-of-the-art algorithms, Hybrid-F_5 and FXL.

Keywords: Crossbred Algorithm · Post-Quantum Cryptography · Multivariate Equations

1 Introduction

Currently, public key cryptosystems such as RSA rely on the difficulty of factoring a large number into two prime factors. Furthermore, with the development of Shor's Algorithm for quantum computers, the integer factorisation and discrete logarithm problems are solvable in polynomial time with a quantum computer of sufficient size, rather than exponential time with classical algorithms. Henceforth, when quantum computers become a feasible tool for computation, cryptosystems that rely on these aforementioned problems will be considered insecure.

Research into cryptography that is secure against an attack by a quantum computer (post-quantum cryptography) has heavily contributed to the continued development of multivariate cryptography. Its security relies on the hardness of solving a set of multivariate polynomial equations. There are no known polynomial time algorithms that solve such a problem for quantum computers and solving a set of random multivariate polynomials is an NP-complete problem [DY09, p.194]. Due to this, it is a fitting candidate for replacing current public-key cryptography.

However, it is believed that there may exist more efficient classical algorithms for solving multivariate polynomial system of equations than what currently exists. This

[*] This work is based on the author's Master's thesis while at Royal Holloway, supervised by Prof. Martin Albrecht. At the time of publication, the author is a PhD student at the University of Porto.

E-mail: joao@diogoduarte.pt (João Diogo Duarte)

Date of this document: 2024-01-18.

is why it is important to thoroughly study and understand new algorithms, such as the Joux–Vitse Crossbred algorithm. The Crossbred algorithm's aim is to efficiently solve a system of semi-regular multivariate polynomials equations.

1.1 Introduction to the Joux–Vitse Crossbred Algorithm

The Crossbred algorithm was created by [JV17]. Their purpose was to produce a scalable algorithm that solves random systems of multivariate polynomial equations by combining ideas from the BooleanSolve algorithm [BFSS13] and the Kipnis-Patarin-Goubin algorithm [KPG99], hence it being called 'Crossbred'. Specifically, given a multivariate system of polynomials equations, it will produce an equivalent smaller system which is more easily solved. Joux and Vitse only worked with systems in boolean $\mathbb{F}_2$, but stated that this algorithm also works for small finite fields $\mathbb{F}_{q>2}$. In addition, there are details to the algorithm which are not explicitly explained, such as the reasoning behind constructing some submatrices. This manuscript will fully explain the algorithm.

Some issues were found with the Joux–Vitse Crossbred algorithm, namely that upon performing asymptotic analysis, Joux and Vitse determined that it ultimately does not provide an asymptotic improvement in relation to BooleanSolve and FXL. Due to this disappointing result and restrictive page limits, they did not expand on this analysis, nor did they present a complexity estimate of their algorithm. Thus, there is room to investigate their algorithm's complexity for both boolean $\mathbb{F}_2$ and $\mathbb{F}_{q>2}$ and investigate if it provides any asymptotic improvement in relation to other algorithms, such as the similar state-of-the-art algorithm, Hybrid-F_5, and the Fast Exhaustive Search (FES).

In the supporting documentation of a NIST Post-Quantum Cryptography Round 2 candidate, MQDSS, Samardijska et al. provided a rough complexity estimate for the Crossbred algorithm, but they did not provide a clear explanation on how the estimate was obtained [SCH+19]. This allows us to see if their complexity estimate matches with the one derived in this work and compare any differences. In the end, it is determined that their estimate can be viewed as an upper-bound.

The Crossbred algorithm's course depends on three parameters. Selecting these parameters is essential as an 'incorrect' selection will cause the algorithm to either not function (*e.g.*, producing empty submatrices and the algorithm cannot continue its execution) or not produce to any solutions (if they exist). If the algorithm is functional (*e.g.*, does not produce any empty submatrices) and outputs a solution (if any exist), then the parameters are called admissible. Henceforth, selecting functional parameters is a non-trivial task. To address this issue, Joux and Vitse presented a bivariate generating series which determines if the parameters provided to the algorithm are admissible in $\mathbb{F}_2$ before execution. However, its derivation or why the series works is not explained. Hence, there is room to explain its derivation, why it works and present a bivariate generating series for $\mathbb{F}_{q>2}$.

2 Mathematical Recap

2.1 Monomial Ordering and Leading Terms

Definition 1. (Monomial ordering) A monomial ordering is a relation, $>$, on the set of monomials x^α whereby $\alpha \in \mathbb{Z}^n_{\geq 0}$ and satisfies the following properties:

1. $>$ is a total ordering on $\mathbb{Z}^n_{\geq 0}$ (follows the laws of reflexivity, antisymmetry, transivity and all items being ordered must be comparable).

2. Let $\beta \in \mathbb{Z}^n_{\geq 0}$. If $\alpha > \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$ then $\alpha + \gamma > \beta + \gamma$.

3. $>$ is a well ordering on $\mathbb{Z}^n_{\geq 0}$. This essentially means that there will always be a smallest element in the set of ordered monomials.

[CLO12, p.55]

Let us define an important, yet simple notion. Let f be an arbitrary polynomial in some polynomial ring, $F_q[X]$. We say that the leading monomial of f, $LM(f)$ is monomial with the largest total degree in f. For example, $LM(12x_1^5x_2^2 + 5x_2^{50}) = x_2^{50}$.

2.2 Homogeneous polynomials

A homogeneous polynomial is a polynomial whereby all its monomials are of the same degree [Bar04]. In the case of affine polynomials (i.e. polynomials with coefficients in an algebraically closed field $\mathbb{K}$ and have solutions in $\mathbb{K}$), we can make them homogeneous by including a homogenisation variable.

Lastly, we assume we are working with polynomials in $\mathbb{F}_q$ of degree q (i.e., for $\mathbb{F}_2$, all polynomials will be of degree 2).

2.3 Regular and semi-regular systems

Let $\mathbb{F}_q$ be a finite field of size q such that $\mathbb{F}_q[X = x_1 \ldots x_n]$ is a polynomial ring over n variables.

Definition 2. (Regular Systems) A regular system is a sequence of homogeneous polynomials, denoted as $f_1, \ldots, f_m \in \mathbb{F}_q[X]$, with degrees $\deg(f_i) = d_i$, such that if $i \in \{1, \ldots, m\}$ and $g \in \mathbb{F}_q[X]$,

$$gf_i \in \langle f_1, \ldots, f_{i-1} \rangle \tag{1}$$

then, g is also in the ideal generated by $f_1, \ldots, f_{i-1}$.

Regular systems, are at most, determined, meaning that they have, at most, $n = m$, whereby n is the number of variables in the system of polynomials and m is the number of equations [BFSY05].

Definition 3. (Semi-Regular Systems) A semi-regular system with degree of regularity $d_{reg} \in \mathbb{N}$ is a homogeneous overdetermined system $(m > n)$ in $\mathbb{F}_q[X]$ with degrees $\deg(f_i) = d_i$, such that if $i \in \{1, \ldots, m\}$ and $g \in \mathbb{F}_q[X]$,

$$gf_i \in \langle f_1, \ldots, f_{i-1} \rangle \text{ and } deg(gf_i) < d_{reg} \tag{2}$$

then, g is also in the ideal generated by $f_1, \ldots, f_{i-1}$ [BFSY05].

The degree of regularity and will be discussed in the following section.

Definition 4. (Boolean Semi-Regular Systems in $\mathbb{F}_2$) Let $R_b = \mathbb{F}_2[X]/(x_1^2, \ldots, x_n^2)$. A system with degree of regularity $d_{reg} \in \mathbb{N}$ is a semi-regular system in R_b if it is an homogeneous overdetermined system $(m > n)$ with degrees $\deg(f_i) = d_i$, such that if $i \in \{1, \ldots, m\}$ and $g_i \in \mathbb{F}_2[X]$, and,

$$\langle f_1, \ldots, f_m \rangle \subset R_h \tag{3}$$

and if,

$$g_i f_i = 0 \text{ and } deg(g_i f_i) < d_{reg} \tag{4}$$

then, $g_i = 0$ in $R_b/(f_1, \ldots, f_i)$ [Bar04, p.58].

Note how we include the field equations $x_i^2 = 0$, which will remove all squares. Hence, we assume that when working in $\mathbb{F}_2$, we include the field equations, which is why we refer to $\mathbb{F}_2$ as boolean.

2.4 Degree of regularity

Given that q is a power of a prime, let I be an ideal in $\mathbb{F}_q[X = x_1, \ldots, x_n]$ generated by a finite homogeneous system of m polynomials. Denote the set of all possible polynomials in a polynomial ring $\mathbb{F}_q[X]$ with a degree at most s as $\mathbb{F}_q[X]_{\leq s}$. We also denote $I_{\leq s}$ as the set of polynomials in I with degree at most s.

We now define the Hilbert Function of an ideal I as,

Definition 5. (Hilbert Function [CLO12, p.487])

$$HF_I(s) = \dim(\mathbb{F}_q[X]_{\leq s}) - \dim(I_{\leq s}) \tag{5}$$

Note that in this context, we define the dimension of the set $\mathbb{F}_q[X]_{\leq s}$ as a $\mathbb{F}_q[X]$ vector space. Equation 5 shows that the Hilbert Function is the dimension of polynomials in the polynomial ring that are not in the ideal. This means that when the dimension of the set of polynomials with degree at most s in ideal I is equal to the dimension of the set of polynomials with degree less or equal to s in the polynomial ring, the Hilbert Function will output 0.

Let $F = \{f_1, \ldots, f_m\}$ be a semi-regular system of equations with n variables and degrees $\deg(f_i) = d_i$ whereby $f_i \in F$. The Hilbert Series of this non-boolean (*i.e.*, in $\mathbb{F}_{q>2}$) semi-regular system of equations is the formal series,

Definition 6. (Hilbert Series of a non-boolean semi-regular system of equations [Bar04, p.66])

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1 - t^{d_i})}{(1 - t)^n} \tag{6}$$

For the rest of the manuscript, we define the degree of regularity of a non-boolean semi-regular system of equations as the power of the first non-negative coefficient of the series in Equation 6. Despite experimental evidence that semi-regular sequences are common, the problem lies on assessing the existence of such sequences [HMS14].

Let $F = \{f_1, \ldots, f_m\}$ be a semi-regular system of equations with n variables and degrees $\deg(f_i) = d_i$ whereby $f_i \in F$. Note that $d_i \leq 2$. The Hilbert Series of this boolean (*i.e.*, in $\mathbb{F}_2$) semi-regular system of equations is formal series,

Definition 7. (Hilbert Series of a boolean semi-regular system of equations [Bar04, p.68])

$$S_{m,n} = \frac{(1 + t)^n}{\prod_{i=1}^{m}(1 + t^{d_i})} \tag{7}$$

For the rest of the manuscript, we define the degree of regularity of a boolean semi-regular system of equations, as the power of the first non-negative coefficient of the series in Equation 7.

2.5 Macaulay matrix

Consider an ideal $I = \langle f_1, \ldots, f_m \rangle \in \mathbb{F}_q[X]$. Let $F \in \mathbb{F}_q[X]$ be a polynomial system of equations whereby the degree of each $f_i \in F$ is denoted as d_i.

Definition 8. (Macaulay Matrix of degree D) We define a Macaulay Matrix, $Mac_D(f)$ (or Mac_D for short), of degree D as a coefficient matrix of the set $\{u \cdot f_i \,|\, \deg(u) \leq D - d_i\}$ whereby $i \in \{1, \ldots, m\}$ and u is a monomial in $\mathbb{F}_q[X]$ and $f_i \in F$.

Another way of putting it is that a Macaulay Matrix of degree D is a coefficient matrix whose columns lists all monomials with degrees at most D from largest till smallest, as per some fixed ordering. We then multiply each f_i by all monomials of degree at most

$D - d_i$, whereby d_i is the degree of f_i. Each row of the Macaulay Matrix is indexed by the result of these multiplications.

[Laz83] proved that there exists a positive integer D whereby the rows of a row reduced Macaulay Matrix of degree D are a Gröbner Basis for the ideal $\langle f_1, \dots, f_m \rangle$. Hence, we can use Macaulay Matrices to aid us to solve polynomial systems of equations.

At times, when it is important to mention the number of equations and degrees in the system of equations generating ideal I, we denote the Macaulay Matrix as Mac_D^{m,d_f} for $d_f = (d_1, \dots, d_m)$, whereby each $d_i \in d_f$ is the degree of f_i. For homogeneous systems, we omit the d_f from the notation as all values in d_f are the identical.

2.6 General and Frobenius Criteria

In many algorithms that utilise Macaulay Matrices for computing the Gröbner Basis of a system of equations, some rows of the Macaulay Matrix are removed if they conform to some criteria.

The first criteria will remove any lines that will result in the trivial reduction $f^2 \to 0$ due to the field equations $x_i^2 = 0$.

Definition 9. (Frobenius Criteria [Bar04, Proposition 1.5.2]) In boolean homogeneous system of equations, f, if the row corresponding to (u, f_m) in the Macaulay Matrix Mac_{D-2}^m has leading term t, then the row (t, f_m) in Mac_D^m is a linearly dependent on the rows preceding it.

The second criteria will remove any lines that will result in the trivial reduction $f_i f_j = f_j f_i$ (and hence $f_i f_j - f_j f_i = 0$) for $ineqj$ due to the field equations $x_i^2 = 0$.

Definition 10. (General Criteria [Bar04, Proposition 1.5.1] [Fau02]) In system of potentially non-homogeneous equations, f, then for all $j < m$, if the row corresponding to (u, f_m) in the Macaulay Matrix $Mac_{D-d_m}^{m-1,d_f}$ with leading term t, then the row (t, f_m) in Mac_D^m is a linearly dependent on the rows preceding it.

2.6.1 Relationship with the degree of regularity

According to [Bar04, p.65-66], we can make a 1-to-1 correspondence between the Hilbert Function and a Macaulay Matrix for homogeneous semi-regular systems of equations. This is done by setting $\dim(\mathbb{F}_q[X]_{\leq s})$ as the number of columns of the Macaulay Matrix and $\dim(I_{\leq s})$ as the number of linearly independent rows (rank) of the Macaulay Matrix. Hence, another definition for the degree of regularity of a semi-regular polynomial system of equations with n variables and m equations is the smallest degree D whereby its Macaulay Matrix of this degree has as many (or more) linearly independent rows as columns. Because of this, it is possible to obtain linear equations (or low degree equations) from computing this Macaulay Matrix's row echelon form. It was first noted by [Laz83] that Gaussian Elimination of a degree $D = d_{reg}$ Macaulay Matrix is equivalent to performing the Buchberger Algorithm which outputs a Gröbner Basis.

Lastly, there exists a very close relationship between the corank of a Macaulay Matrix and the Hilbert Function for homogeneous semi-regular systems of equations. If we recall, we define the Hilbert Function as,

$$HF_I(s) = \dim(\mathbb{F}_q[X]_{\leq s}) - \dim(I_{\leq s})$$

which, according to the aforementioned relationship between a Macaulay Matrix and the Hilbert Function, is the number of columns of the Macaulay Matrix minus its linearly independent rows (its rank). This is, by definition, the corank of a Macaulay Matrix

3 State-of-the-Art for Solving Multivariate Polynomials

3.1 Exhaustive Search

This is the most basic way of solving a system of polynomial equations. If q is a power of a prime and n is the number of variables, we iterate over $\mathbb{F}_q^n$ and test if they produce a valid solution for our polynomials. Since there are q^n values to test, the complexity of this algorithm is $\mathcal{O}(mq^n)$ for m polynomials.

In $\mathbb{F}_2$, Fast Exhaustive Search (FES) was proposed by [BCC+10] and it efficiently enumerates over the search space of 2^n such that the complexity of exhaustive search is (hopefully) less than $\mathcal{O}(mq^n)$. In $\mathbb{F}_2$, the complexity of FES is $\mathcal{O}(\log_2(n) \cdot 2^n)$ and is independent of the number of polynomials, m. We can expand FES for larger fields using $q-$ary Gray codes codes with a complexity of $\log_q(n)q^n$.

3.2 Hybrid-F5

F_4 was created by [Fau99] and it outputs a Gröbner basis for a given set of polynomials. An improved version called F_5 was later developed by [Fau02].

Hybrid-F_5 [BFP08] combines exhaustive search and F_5 by specialising k variables and running the F_5 algorithm over the remaining $n-k$ variables. The most costly part in the complexity of F_5 is the row reduction of a matrix of size $\binom{n+d_{reg}}{d_{reg}}$ for $q > 2$ and n, d_{reg} hence once can simplify its complexity estimate to:

$$C_{F_5,q>2} = \mathcal{O}\left(\binom{n+d_{reg}}{d_{reg}}^{\omega}\right) \tag{8}$$

whereby $2 \leq \omega \leq 3$ is the exponent of matrix multiplication. We set $\omega = 2$. By combining the above complexity estimate and exhaustive search, we obtain the complexity estimate for Hybrid-F_5 [BPSV19]:

$$C_{Hybrid,q>2} = q^k \cdot \mathcal{O}\left(\binom{n-k-1+d_{reg}(n-k)}{d_{reg}(n-k)}^2\right) \tag{9}$$

whereby $d_{reg}(n-k)$ is the degree of regularity of the system after evaluating k variables and hence, having $n-k$ variables left. For $q = 2$, due to the square-free field, we obtain the complexity estimate

$$C_{Hybrid,q=2} = q^k \cdot \tilde{\mathcal{O}}\left(\left(\sum_{i=0}^{d_{reg}(n-k)} \binom{n-k}{i}\right)^2\right) \tag{10}$$

3.3 FXL/BooleanSolve

The BooleanSolve algorithm was developed by [BFSS13] before the Crossbred algorithm. This algorithm specialises the first $n-k$ variables in the polynomials by iterating it through $\mathbb{F}_2^{n-k}$. It then tests the consistency of the system via Macaulay Matrices of $d_{reg}(k)$, whereby $d_{reg}(k)$ is the degree of regularity of the system after specialisation. If this system is not consistent, iterate to the next value. Otherwise, exhaustive search is conducted over the first k variables [FHK+17]. The complexity of this algorithm is $\mathcal{O}(2^{0.841n})$ and a probabilistic variant called the Las Vegas variant has a conditional complexity of only $\mathcal{O}(2^{0.792n})$. Furthermore, a quantum version of the Las-Vegas variant was shown to only require the evaluation of $\mathcal{O}(2^{0.462n})$ quantum gates [FHK+17].

BooleanSolve can be seen as a specific instance of FXL [CKPS00], which does the same as BooleanSolve, with the exception that it supports finite fields larger than 2. The complexity of FXL is [YC05][BFSS13],

$$C_h = q^k \cdot \max\{\left(\binom{n-k+d_0}{n-k}\right)^2, m\left(\binom{n-k-q+d_0}{n-k}\right)\} \tag{11}$$

whereby d_0 is the first non-positive coefficient of the seroes expansion of $\frac{(1-t^q)^m}{(1-t)(1-t)^{n-k}}$.

4 Joux–Vitse Crossbred algorithm

Essentially, this algorithm involves the specialisation of variables and then solving the remaining ones. The advantage of this technique is that we can avoid solving the initial or the specialised system via, for example, the general version of F_5.

The main differentiating factor between this algorithm and BooleanSolve/FXL is that the manipulation of the Macaulay Matrix is done before specialising any variables. This is an advantage since linear algebra in a Macaulay Matrix is the most costly step of the BooleanSolve algorithm since it is performed 2^{n-k} times [JV17]. The Crossbred algorithm attempts to limit the size of the Macaulay Matrix to speed up the computation of a polynomial system's Gröbner Basis. Note that this algorithm assumes semi-regularity. Let $F \in \mathbb{F}_q[X = x_1, \dots, x_n]$ be the polynomial system of m equations over n variables which we want to solve via the Crossbred algorithm. The algorithm accepts three parameters,

1. D, the degree of the Macaulay Matrix of F. $D \geq 2$ and must be at least the degree of regularity of a system with k variables over m equations in $\mathbb{F}_q[X]$.

2. k, the number of variables we want our specialised system of equations to have, $1 \leq k \leq n$.

3. d, the desired degree of the system of equations after we specialise the last $n-k$ variables, $1 \leq d < D$.

The total degree in the first k variables of a polynomial p is labelled as $deg_k p$.

If we wish for our system of equations to reduce down to a linear system, then we set $d = 1$. However, to extend this to larger values of d, we must construct new equations with $deg_k p \leq d$. According to [JV17], this allows us to select smaller values of D since we do not need to produce a Macaulay Matrix with a lot of polynomials in order to 'break them down' to a system of degree d. This is desirable since D must be large enough for any reduced equations to exist but also small enough to make the Macaulay Matrix manageable.

As previously mentioned, selecting values for D, d and k is a non-trivial task.

4.1 Description of the algorithm

To better understand the algorithm, let us divide it into two main steps, the pre-processing and then the actual algorithm. Furthermore, let us also assume we are working with a boolean system of polynomial equations. The pre-processing goes as following,

1. Construct the Macaulay Matrix of degree D of a polynomial system of equations F with its columns sorted in reverse graded lex. Label this matrix as Mac_D

2. Let $Mac^k_{D,d}(F)$ be a submatrix of Mac_D whereby each row $u_{ij} f_i$ represents a polynomial with the property that $deg_k u_{ij} \geq d-1$.

3. Let $M_{D,d}^k(F)$ be a submatrix of $Mac_{D,d}^k$, whereby each column i represents the monomial M_i whereby $deg_k M_i > d$.

The actual algorithm does the following,

1. Construct the cokernel of $M_{D,d}^k$ with elements $v_1, \ldots, v_r$, whereby r is the number of elements in the cokernel.
2. Compute new polynomials $p_i = v_i Mac_{D,d}^k(F)$. This forms a system of polynomial equations, $P = p_1, \ldots, p_r$, whereby they have a total degree at most D and at most d in $x_1, \ldots, x_k$.
3. For all $a = \{a_{k+1}, \ldots, a_n\} \in \mathbb{F}_2^{n-k}$,
   (a) Partially evaluate the last $n-k$ variables of F at a. Let F^* represent this new system.
   (b) Construct a new Macaulay Matrix of degree d of F^*. Now, let $Mac_d(F^*)$ represent this new matrix.
   (c) Partially evaluate the last $n-k$ variables of polynomial system P at a. Let P^* represent this new system as a coefficient matrix.
   (d) Append $Mac_d(F^*)$ to P^*. Let PM^* represent this new polynomial system of equations.
   (e) Check if this system is consistent using dense linear algebra. If it is, extract variables $x_1 \ldots x_k$ and test the solution. Output any valid solutions.

According to [JV17], the number of equations of P must be at least the number of monomials in k variables of degree d, which is $\binom{k+d-1}{d}$ for $\mathbb{F}_{q>2}$ and $\sum_{d'=0}^{d} \binom{k}{d'}$ for $\mathbb{F}_2$. Thus, for $d = 1$ we need to simply check whether $|P| > k$. This is to ensure enough independent relations to finally solve the system.

Note how we are using field equations for polynomial systems in $\mathbb{F}_2$. This is equivalent of working within a quotient ring of the form $\mathbb{F}_{2}/\langle x_1^2, \ldots, x_n^2\rangle$ [BFSY05]. However, multiplication in quotient rings involves reducing the polynomials by the field equations to ensure that they are in the quotient ring. This means that including field equations for large fields may not be computationally feasible.

4.2 Understanding various matrices

4.2.1 First submatix

By constructing $Mac_{D,d}^k(F)$, we obtain polynomials of the form $u_{i.j} \cdot f_i$ whereby the total degree of $u_{i.j}$ in the first k variables is at least $d-1$. For example, for $\mathbb{F}_2$, consider $n = 5$, $D = 4$, $d = 2$ and $k = 3$. That means that the first k variables are x_1, x_2, x_3 and the last $n-k$ are x_4 and x_5. Let $f = x_1x_2 + 1$. Consider the following row in Mac_D,

$$x_4x_5 \cdot f = x_1x_2x_4x_5 + x_4x_5$$

Clearly, since $Mac_{D,d}^k$ only contains rows whereby the multiplier has at least total degree $d-1 = 1$ in the first k variables, we would not include this row. This is because upon specialisation, such as $x_4 = x_5 = 1$, we would obtain $1 \cdot f = x_1x_2 + 1$, which is simply f. If we set $x_4 = x_3 = 0$ or any variation whereby at least one of the variables is 0, we would obtain, $0 \cdot f = 0$.

Hence, if we include these rows, we would simply obtain 0 or our initial f. None of these add any new information to our system because they will result in a linearly dependent row and thus, produce the trivial solution $0 = 0$.

4.2.2 Cokernel

Since multiplying a matrix by the elements of the cokernel is equal to 0, multiplying the elements of the cokernel of $M^k_{D,d}(F)$ by $Mac^k_{D,d}(F)$, we simply 'remove' all monomials in $Mac^k_{D,d}(F)$ that do not have a degree at most d in the first k variables as we remove columns whereby their total degree in the first k variables is greater than d. This will allow us to achieve a system of degree at most d after specialising the last $n-k$ variables.

All of this boils down to the fact that we want to obtain an equivalent system of F that has a degree at most D and a degree of at most d in the first k variables. As mentioned before, we want this because after we specialise the last $n-k$ variables, we obtain a smaller system with a total degree of at most d.

4.2.3 Appending our initial equations

For $d = 1$, $Mac^k_{D,d}(F) = Mac_D(F)$, which means that we have selected all of the equations to be used when constructing P. The reason we create $Mac_d(F^*)$ is to include more equations since it reduces the amount of consistent systems that are obtained and therefore, you have less systems to test whether they are also consistent with F. This may seem like a disadvantage but the consistent systems we have avoided would also not be consistent with F and hence, we evaluate less systems that are bound to be incorrect Clearly, when $d = 1$, we are not producing any new information since all of the rows in $Mac_d(F^*)$ would be linearly dependent with P. Henceforth, we consider $Mac_d(F^*)$ to be empty when $d = 1$.

4.3 Columns of second submatrix

As we will see in Section 7.1, we need to have an equation that outputs the number of columns of $M^k_{D,d}$ to construct the complexity estimate of the Crossbred algorithm. Firstly, let us establish that the number of monomials in n variables from degree 0 up to, and including, d is given by,

$$\mathcal{M}_{n,d} = \binom{n+d}{d} = \sum_{d'=0}^{d} \binom{n+d'-1}{d'} \tag{12}$$

However, in $\mathbb{F}_2$, we are working in a quotient field with field equations, so we adapt the above definition for boolean $\mathbb{F}_2$,

$$\mathcal{M}_{n,d} = \sum_{d'=0}^{d} \binom{n}{d'} \tag{13}$$

Since the columns of the Macaulay Matrix index monomials in n variables from degree 0 to d, we can use the above equations to calculate its number of columns. Let us continue with this example and let us construct $M^k_{D,d}$, which requires us to get rid of any monomials in the columns of $Mac^k_{D,d}$ whereby their degree in the first k variables is smaller or equal to d. If we use $d = 1$ and $k = 2$, all monomials that have a degree lesser or equal to d in the first k variables are removed. In this case, we have x_1x_2.

However, we would also include, for example, x_1x_2 multiplied by any monomial that is comprised of the last $n-k$ variables such as x_3, such that the result is $x_1x_2x_3$. This multiplication would have to result in a monomial whose total degree is at most D, hence, if we let d_k represent our initial monomial's degree in the first k variables (in this case, the initial monomial is x_1x_2), the degree of the monomial with variables from the last $n-k$ variables must be at most $D-d_k$. Henceforth, this is the same as saying that we also want all monomials comprised of the last $n-k$ variables of degree 0 till $D-d_k$. This leads us to the following equation for the number of columns for $\mathbb{F}_{q>2}$,

$$n_{cols}(M^k_{D,d}) = \sum_{d_k=d+1}^{D} \sum_{d'=0}^{D-d_k} \binom{k+d_k-1}{d_k}\binom{n-k+d'-1}{d'} \tag{14}$$

In $\mathbb{F}_2$ [JV17], the number of columns would be,

$$n_{cols}(M^k_{D,d}) = \sum_{d_k=d+1}^{D} \sum_{d'=0}^{D-d_k} \binom{k}{d_k}\binom{n-k}{d'} \tag{15}$$

5 Selecting parameters

5.1 Incorrect parameter selection

Let us show how incorrectly choosing parameters can lead to the algorithm not functioning. Consider the columns of $M^k_{D,d}$, whose columns contain monomials whereby their total degree in the first k variables is larger than d. If we have $k = 2$, the first k variables are x_1 and x_2. In $\mathbb{F}_2$, the maximum degree obtainable with 2 variables is with the monomial x_1x_2 as we cannot have squares. Henceforth, $M^k_{D,d}$ is empty as it is impossible for any monomials in the first k variables to be larger than 2. Since it is an empty matrix, it does not have a cokernel.

Therefore, if the number of degree d monomials over k is lesser than d, the algorithm will not function properly. For $\mathbb{F}_2$, this is equivalent of saying $\sum_{d'=0}^{d}\binom{k}{d'} < d$ and for $\mathbb{F}_{q>2}$, $\binom{k+d-1}{d} < d$.

5.2 Admissibility of parameters

For $d > 1$, to determine the admissibility of the parameters, [JV17] derived a bivariate generating series. Firstly, let us define,

$$S^k_{D,d} = \frac{(1+X)^{n-k}}{(1-X)(1-Y)}\left(\frac{(1+XY)^k}{(1+X^2Y^2)^m} - \frac{(1+X)^k}{(1+X^2)^m}\right) \tag{16}$$

The coefficient of X^DY^d of $S^k_{D,d}$ represents the number of new independent polynomials after the reduction of $Mac^k_{D,d}$, which is equivalent to corank of $M^k_{D,d}$. The reason for this will be explained in the next section.

If the coefficient of X^DY^d is non-negative in the following equation, then the parameters (D, d, k) for the algorithm are admissible,

$$A^k_{D,d} = S^k_{D,d} - \frac{(1+Y)^k}{(1-X)(1-Y)(1+Y^2)^m} \tag{17}$$

The reason why will be explained further down this section.

For example, for a system with 3 variables and 4 equations, let $k = 2$. The admissibility series will produce,

$$2X - X^2 + 10X^3 + 3X^2Y + \ldots + 8X^3Y^2 + \ldots \tag{18}$$

Since the coefficient of X^3Y^2 is non-negative (*i.e.* 8), $D = 3$, $d = 2$ and $k = 2$ are admissible parameters for this polynomial system.

6 Understanding and deriving the generating series

6.1 Informal analysis of the generating series

Let us now break down the various parts of the bivariate generating series. By expanding out the multiplication that occurs in $S^k_{D,d}$ and ignoring the $\frac{1}{1-X}$ and $\frac{1}{1-Y}$ parts as they will be explained later, we obtain,

$$\frac{(1+XY)^k(1+X)^{n-k}}{(1+X^2Y^2)^m} - \frac{(1+X)^n}{(1+X^2)^m} \tag{19}$$

Informally, the leftmost term represents the formal power series of the corank of $M^k_{D,d}$ and the rightmost term represents the formal power series of the corank of Mac_D. This allows for us to obtain the corank of these matrices for various values of parameters. We have already discussed how the evaluation of a Hilbert Function $HF'_I(d)$ is by definition the corank of a Macaulay Matrix of degree d in Section 2.6.1. Proof of this will be provided in the following section.

As stated before, finding the dimension of the left kernel of $M^k_{D,d}$ will tell us the number of polynomials produced in P. Furthermore, we need to know how many of these polynomials are new in relation to our initial system, F, as we are going to be including our initial equations alongside P since $d > 1$. This is why we subtract the Hilbert Series of our initial Macaulay Matrix of degree D, which is the rightmost term of Equation 19.

Consider the subtraction that occurs on the left hand side of the admissibility series (once again, omitting the $\frac{1}{1-X}$ and $\frac{1}{1-Y}$ parts),

$$A^k_{D,d} = S^k_{D,d} - \frac{(1+Y)^k}{(1+Y^2)^m} \tag{20}$$

This subtracts the corank of the new polynomials after evaluation, which removes the number of polynomials that reduce to 0 after evaluation. Let us refer to this term of pure Y as S^k_0.

Let $S'^k_{D,d}$ and S'^k_0 refer to $S^k_{D,d}$ and S^k_0 without the $\frac{1}{1-X}$ and $\frac{1}{1-Y}$ parts, respectively. Hence,

$$S'^k_{D,d} = (1+X)^{n-k}\left(\frac{(1+XY)^k}{(1+X^2Y^2)^m} - \frac{(1+X)^k}{(1+X^2)^m}\right)$$

$$S'^k_0 = \frac{(1+Y)^k}{(1+Y^2)^m}$$

The use of $\frac{1}{1-X}$ and $\frac{1}{1-Y}$ is to copy $S'^k_{D,d}$ and S'^k_0 to all degrees of X and Y. What this means is that since the expansion of $\frac{1}{1-X} = 1 + X + X^2 + \ldots$, we obtain,

$$\frac{(S^k_{D,d} - S^k_0)}{(1-X)(1-Y)} = X^0Y^0(S^k_{D,d} - S^k_0) + X^1Y^1(S^k_{D,d} - S^k_0) + X^2Y^2(S^k_{D,d} - S^k_0)\ldots$$

This means that all possible combinations of X^DY^d are included in the series. Hence, if the coefficient of X^DY^d is a non-negative number, it means that the set of parameters will produce a non-negative amount of new polynomials, which is why we consider the parameters admissible. Note how we can produce 0 new polynomials, but this is allowed since we append our initial system.

In conclusion, the proof that $\frac{(1+Y)^k}{(1+Y^2)^m}$ is the formal power series for the corank the polynomials that reduce to 0 and that $\frac{(1+X)^n}{(1+X^2)^m}$ is the formal power series for the corank of the Macaulay Matrix will not be presented as they follow trivially from [Bar04, p.65-66]. However, to complete the proof, we must demonstrate a non-trivial proof of the formal power series of $M^k_{D,d}$.

6.2 Deriving the admissibility series for boolean finite fields

6.2.1 Number of monomials in second submatrix

Let us now derive the admissibility series from the Hilbert Function as to provide clarity and mathematical proof of its correctness.

Proposition 1. *The number of number of monomials with degree less than or equal to D and to d in the first k variables can be obtained by the following generating series in a boolean finite field,*

$$\mathcal{M}_{\leq D, \leq d, m} = \frac{(1+X)^k(1+XY)^{n-k}}{(1-X)(1-Y)} \tag{21}$$

Proof. From Equation 15, we get the equation for the number columns of $M^k_{D,d}$,

$$\mathcal{M}(M^k_{D,d}) = \sum_{d_k=d+1}^{D} \sum_{d'=0}^{D-d_k} \binom{k}{d_k}\binom{n-k}{d'} \tag{22}$$

From this, we can deduce the number of monomials with degree less than or equal to D and to d in the first k variables,

$$\mathcal{M}_{\leq D, \leq d, m} = \sum_{d_k=0}^{d} \sum_{d'=0}^{D-d_k} \binom{k}{d_k}\binom{n-k}{d'} \tag{23}$$

and hence, given that $\sum_{d\geq 0}\binom{n}{d}z^d = (1+z)^n$, we obtain we want to copy this to all degrees of X and Y,

$$\mathcal{M}_{\leq D, \leq d, m} = \frac{(1+XY)^k(1+X)^{n-k}}{(1-X)(1-Y)} \tag{24}$$

□

6.2.2 Hilbert Function and Hilbert Series for second submatrix

Let us now introduce a slightly altered definition of the Hilbert Function to allow us to derive the generating series for $M^k_{D,d}$. For the readability of the following theorem and proof, we will denote $M^k_{D,d}$ and $M^{f,k}_{D,d,m}$ whereby m is the number of equations in our initial system, f. If we use a subset of the first $v \leq m$ equations of our initial system, we write $M^{f,k}_{D,d,v}$.

Definition 11. (Hilbert Function for ideal generated by equations in $M^{f,k}_{D,d,m}$) Given m equations in $\mathbb{F}_2[X]$, an ideal I generated by said equations (i.e $I = \langle f = f_1, \ldots f_m \rangle$), we define the Hilbert Function for $M^{f,k}_{D,d,m-1}$ as,

$$HF'_{I,m}(D,d,k) = \dim(\mathbb{F}_2[X]_{D,d,k}) - \dim(I_{D,d,k}) \tag{25}$$

whereby

- $\mathbb{F}_2[X]_{D,d,k}$ is the set of all monomials in $\mathbb{F}_2[X]$ of degree at least $d+1$ in its first k variables.
- $I_{D,d,k}$ is the set of all $f \in \mathbb{F}_2[X]$ in the form mg, whereby m is a monomial of degree at least $d+1$ in its first k variables and f has total degree of at most $D+d-1$.

We can calculate this altered Hilbert Function from the standard Hilbert Function by simply removing the polynomials of form uf for a monomial u and polynomial f whereby $deg_k u \geq d-1$ and then all monomials in uf whereby $deg_k M_i > d$.

Theorem 1. *Given m homogeneous linearly independent equations in $\mathbb{F}_2[X]$, $f_1 \dots f_m$, then we can recursively define the Hilbert Function for $M^{f,k}_{D,d,m}$ as,*

$$HF'_{I,m}(D,d,k) = HF'_{I,m-1}(D,d,k) - HF'_{I,m}(D-2,d,k,m) \tag{26}$$

Proof. This proof closely follows the proof of [Bar04, Lemma 3.3.6], with the exception that we now use the altered definition of the Hilbert Function for the ideal generated by the equations in $M^{f,k}_{D,d,m}$.

To do this, we start with Mac^{m-1}_D simply add equations with the form $u \cdot f_m$. However, we will need to remove some equations due to the General Criteria and the Frobenius Criteria, which in this case is equal to the number of columns in Mac^m_{D-2}. However, as we are want to use the submatrix of Mac^m_D and Mac^m_{D-2}, we simply remove all the polynomials of form uf for a monomial u and polynomial f whereby $deg_k u \geq d-1$ and then all monomials in uf whereby $deg_k M_i > d$. If we then remove these polynomials and monomials and denote the number of rows of $M^{f,k}_{D,d,m}$ as $U_{D,d,m}$, then we have,

$$U_{D,d,m} = U_{D,d,m-1} + \mathcal{M}_{\leq D-2,\leq d,m} - U_{D-2,d,m} \tag{27}$$

with $U_{D,d,0} = 0$ or D is smaller than the smallest total degree of equations $f_1, \dots, f_m$. We can rewrite this as $U_{D,d,m} - U_{D,d,m-1} = \mathcal{M}_{\leq D-2,\leq d,m} - U_{D-2,d,m}$. Since $HF'_{I,m}(D,d,k) = \mathcal{M}_{\leq D,\leq d,m} - U_{D,d,m}$, we obtain $HF'_{I,m}(D,d,k) - HF'_{I,m-1}(D-2,d,k) = U_{D,d,m-1} - U_{D,d,m}$. □

Theorem 2. *Given m linearly independent equations in $\mathbb{F}_2[X]$, $f_1 \dots f_m$ with degrees d_i for $i \in \{0, \dots, m\}$, then we can recursively define the Hilbert Series for the ideal generated by the equations in $M^{f,k}_{D,d,m}$ as,*

$$S_{m,n,D,d,k}(X,Y) = \sum_{D,d\geq 0} HF'_{I,m}(D,d,k) X^D Y^d = \frac{(1+XY)^k (1+X)^{n-k}}{(1+X^2Y^2)^m (1-X)(1-Y)} \tag{28}$$

Proof. Once again, this closely follows the proof of [Bar04, Proposition 3.3.7]. Consider the following generating series adapted from the recursive definition of the Hilbert Function in Theorem 26, we get,

$$\begin{aligned} &\sum_{D,d\geq 0} HF'_{I,m}(D,d,k) X^D Y^d \\ &\quad = \sum_{D,d\geq 0} HF'_{I,m}(D,d,k) X^D Y^d - \sum_{D,d\geq 0} HF'_{I,m-1}(D-2,d,k) X^D Y^d \\ &\quad = \sum_{D,d\geq 0} HF'_{I,m-1}(D,d,k) X^D Y^d - X^2Y^2 \sum_{D,d\geq 0} HF'_{I,m}(D,d,k) X^D Y^d \end{aligned} \tag{29}$$

and hence,

$$\begin{aligned} \sum_{D,d\geq 0} HF'_{I,m}(D,d,k) X^D Y^d &= \frac{1}{1+X^2Y^2} \sum_{D,d\geq 0} HF'_{I,m-1}(D,d,k) X^D Y^d \\ &= \frac{1}{(1+X^2Y^2)^{m-1}} \sum_{D,d\geq 0} HF'_{I,1}(D,d,k) X^D Y^d \end{aligned} \tag{30}$$

and since $HF'_{I,1}(D,d,k) = \mathcal{M}_{\leq D,\leq d,m} - HF'_{I,1}(D-2,d,k)$, we can deduce that,

$$\begin{aligned} \sum_{D,d\geq 0} HF'_{I,1}(D,d,k) X^D Y^d &= \frac{1}{1+X^2Y^2} \sum_{D,d\geq 0} \mathcal{M}_{\leq D,\leq d,1} X^D Y^d \\ &= \frac{(1+X)^{n-k}(1+XY)^k}{(1+X^2Y^2)(1-X)(1-Y)} \end{aligned} \tag{31}$$

and plugging this into Equation 30, we obtain,

$$S_{m,n,D,d,k}(X,Y) = \sum_{D,d\geq 0} HF'_{I,m}(D,d,k)X^D Y^d = \frac{(1+X)^{n-k}(1+XY)^k}{(1+X^2Y^2)^m(1-X)(1-Y)} \tag{32}$$

□

6.3 Admissibility series for non-boolean finite fields

Recall the definition of the Hilbert Series for homogenous semi-regular system of m equations in $\mathbb{F}_{q>2}$,

$$S_{m,n} = \frac{(1-t^q)^m}{(1-t)^n}$$

The only difference for deriving the bivariate generating series for $\mathbb{F}_{q>2}[X]$ and Theorem 2 is that we do not use the Frobenius Criteria as squares do not reduce to 0 and the number of monomials in the columns of $M^k_{D,d}$ is different.

In terms of the number of monomials in the in the columns of $M^k_{D,d}$ is,

$$\sum_{d_k=d+1}^{D} \sum_{d'=0}^{D-d_k} \binom{k+d_k-1}{d_k}\binom{n-k+d'-1}{d'}$$

and hence, due to $\sum_{d\geq 0}\binom{n+d-1}{d}z^d = \frac{1}{(1-z)^n}$, the generating series for the number monomials with degree less than or equal to D and to d in the first k variables is,

$$\mathcal{M}_{\leq D,\leq d,m} = \frac{1}{(1-X)^{n-k}(1-XY)^k(1-X)(1-Y)}$$

Furthermore, due to the fact we do not use the Frobenius Criteria, we obtain,

$$U_{D,d,m} = U_{D,d,m-1} + \mathcal{M}_{\leq D-q,\leq d,m} - U_{D-q,d,m-1} \tag{33}$$

and following a similar logic for the proof of Theorem 1, we get the recursive relationship,

$$HF'_{I,m}(D,d,k) = HF'_{I,m-1}(D,d,k) - HF'_{I,m-1}(D-q,d,k) \tag{34}$$

From this, we can construct the Hilbert Series following the exact same logic as before.

Theorem 3. *Given m linearly independent equations in $\mathbb{F}_{q>2}[X]$, $f_1 \dots f_m$ with degrees d_i for $i \in \{0,\dots,m\}$, then we can recursively define the Hilbert Series for the ideal generated by the equations in $M^{f,k}_{D,d,m}$ as,*

$$S_{m,n,D,d,k}(X,Y) = \sum_{D,d\geq 0} HF'_{I,m}(D,d,k)X^D Y^d = \frac{(1-X^qY^q)^m}{(1-X)^{n-k}(1-XY)^k(1-X)(1-Y)} \tag{35}$$

Therefore, if we follow the same idea (omitting the Frobenius Criteria) for Equation 17, the admissibility series for homogeneous system of equations in $\mathbb{F}_{q>2}$ will be,

$$A^k_{D,d,q} = \frac{1}{(1-X)(1-Y)}\left(\frac{(1-X^qY^q)^m}{(1-XY)^k(1-X)^{n-k}} - \frac{(1-X^q)^m}{(1-X)^n} - \frac{(1-Y^q)^m}{(1-Y)^k}\right) \tag{36}$$

In $\mathbb{F}_{q>2}$, the parameters (D,d,k) are admissible if the coefficient of X^DY^d in $A^k_{D,d,q}$ is non-negative.

7 Complexity of the Crossbred algorithm

7.1 Complexity estimate

The complexity of the algorithm for any homogeneous polynomial system in $\mathbb{F}_q$ has the following form,

$$\mathcal{C}_{cross_q} = \mathcal{O}(\texttt{kernel}(M^k_{D,d})) + q^{n-k} \cdot \mathcal{O}((\texttt{solving}(P^* \cup Mac_d(F^*))))$$

Recall that P^* is the system P upon evaluation of the last $n-k$ variables. Block Wiedemann or Lanczös algorithms can be used to calculate the cokernel of a sparse matrix. The complexity of finding cokernel vectors of a sparse matrix is $\tilde{\mathcal{O}}(n^2_{cols})$, whereby n_{cols} the number of columns in our matrix [GLS98]. We have already established how to calculate the number of columns of the cokernel $M^k_{D,d}$ in the previous section using either Equation 14 or Equation 15.

We can also use the block Wiedemann or Lanczös to probabilistically test the consistency and find a small number of solutions (if any) of a set of polynomials [JV17], which in our case, is $P^* \cup Mac_d(F^*)$ for $d > 1$ and just P for $d = 1$. The complexity of doing so is $\tilde{\mathcal{O}}(n^\omega_{cols})$ whereby ω is the exponent of matrix multiplication. Let us set $\omega = 2$.[1]

The number of columns of both $P^* \cup Mac_d(F^*)$ and P is equal to the number of monomials in k variables from degrees 0 to d. We have also already established how to calculate this in the previous section using either Equation 12 or Equation 13. Therefore, we can write the complexity estimate for the Crossbred Algorithm as,

$$\mathcal{C}_{c_{q>2}} = \tilde{\mathcal{O}}\left(\left(\sum_{d_k=d+1}^{D}\sum_{d'=0}^{D-d_k}\binom{k+d_k-1}{d_k}\binom{n-k+d'-1}{d'}\right)^2\right)+q^{n-k}\cdot\tilde{\mathcal{O}}\left(\binom{k+d-1}{d}^2\right) \tag{37}$$

And in $\mathbb{F}_2$,

$$\mathcal{C}_{cq=2} = \tilde{\mathcal{O}}\left(\left(\sum_{d_k=d+1}^{D}\sum_{d'=0}^{D-d_k}\binom{k}{d_k}\binom{n-k}{d'}\right)^2\right) + 2^{n-k}\cdot\tilde{\mathcal{O}}\left(\left(\sum_{i=0}^{d}\binom{k}{i}\right)^2\right) \tag{38}$$

For the rest of the manuscript, when discussing cases where the value of q could be ≥ 2 we refer to the complexity of the Crossbred algorithm as $\mathcal{C}_{cross_q}$, which could either be $\mathcal{C}_{cross_{q=2}}$ or $\mathcal{C}_{cross_{q>2}}$.

7.2 Comparison with MQDSS's estimate

The supporting documentation for a NIST Post Quantum Cryptography Round 2 candidate based on multivariate cryptography, MQDSS by [SCH+19], provides a complexity for the Crossbred algorithm for $\mathbb{F}_{q>2}$, which has a very similar form,

$$C_{cross_{q>2}} = \mathcal{O}\left(\binom{n+D-1}{D}^2\right) + \log(n-k)\cdot q^{n-k}\cdot\mathcal{O}\left(\binom{k+d-1}{d}^\omega\right) \tag{39}$$

Note that Samardijska et al. included $\log(n-k)$. This is because $\log(n-k)$ is the amount of field operations necessary to specialise $n-k$ variables [SCH+19].

The $\mathcal{O}\left(\binom{n+D-1}{D}^2\right)$ part represents the complexity of finding kernel vectors in $M^k_{D,d}$ using, for example, the block Wiedemann algorithm. The only difference between the complexity estimate of this step in relation to the estimate provided in this work is the

[1] In reality, at the time of writing, the smallest practical value of ω is 2.807, but we consider the worse-case scenario.

number of columns of $M^k_{D,d}$. Samardijska et al. assumes that $M^k_{D,d}$ has $\binom{n+D-1}{D}$ columns, which is equivalent to the number of monomials in n variables of degree up to D. Clearly, $M^k_{D,d}$ will not have the same number of columns as the the initial Macaulay Matrix, making this estimate inaccurate. However, this may be interpreted as an upper bound.

The last $\mathcal{O}\left(\binom{k+d-1}{d}^\omega\right)$ represents solving an overdetermined system of multivariate polynomials with k variables of degree d via computing the Gaussian Elimination of a large matrix, which represents solving the system. This is the same in our complexity estimate. Hence, the only real difference between the estimate provided in this manuscript is that the estimate presented by Samardijska et al. may be interpreted as an upper bound.

8 Methodology

To analyse the overall performance of the Crossbred Algorithm, n was incremented from 1 until 200 for $\mathbb{F}_2$, $\mathbb{F}_3$ and $\mathbb{F}_7$ for various ratios of m. We assume that all polynomial system of equations are homogeneous, of the highest possible degree and semi-regular.

In terms of the field sizes, $\mathbb{F}_2$ was chosen as a baseline to compare the other results to. $\mathbb{F}_3$ was chosen since it represents a very small field, but still larger than $\mathbb{F}_2$, and $\mathbb{F}_7$ represents larger field, hence, it represents a corner case. $\mathbb{F}_{127}$ was also considered, by when taking into account that we assume that the system is homogeneous with degree $q = 127$, even finding parameters for $n = 11$ was not feasible to due a very large degree of regularity.

The results were obtained by finding the optimal admissible parameters, which was done by iterating through all possible values of D, d and k, similar to the Algorithm 3.5 named FindTradeOff in [BFP08, p.10]. The referenced FindTradeOff algorithm was used to obtain optimal parameters for Hybrid-F_5.

9 Results

9.1 Results for m = 2n

The results for $m = 2n$ are shown in Figure 1.

In $\mathbb{F}_2$, the Crossbred algorithm clearly provides a better concrete complexity than FES for all fields, which falls in line with the results presented by [JV17]. However, In general, Crossbred provided no complexity improvement over Hybrid-F_5 and FXL. However, it seems that FXL may outperform Crossbred for larger fields.

9.2 Results for m = n + 1

The results for $m = n + 1$ are shown in Figure 2.

As the ratio of m and n decreases, Crossbred's complexity improvement over FES becomes small and may even be attributed to miscellanious missing terms in either complexity estimates. Once again, Crossbred does not provide a complexity improvement over Hybrid-F_5 and FXL.

10 Conclusion

In conclusion, the bivariate generating function to test the admissibility of parameters to the algorithm in $\mathbb{F}_2$ was explained and derived, and a new series for $\mathbb{F}_{q>2}$ was presented. A complexity estimate was given for the Crossbred algorithm for both $\mathbb{F}_2$ and $\mathbb{F}_{q>2}$. By plotting the best-case complexities of applying the Crossbred, FES, FXL and Hybrid-F_5 to polynomial systems of equations with various sizes of n, m and q, it was determined

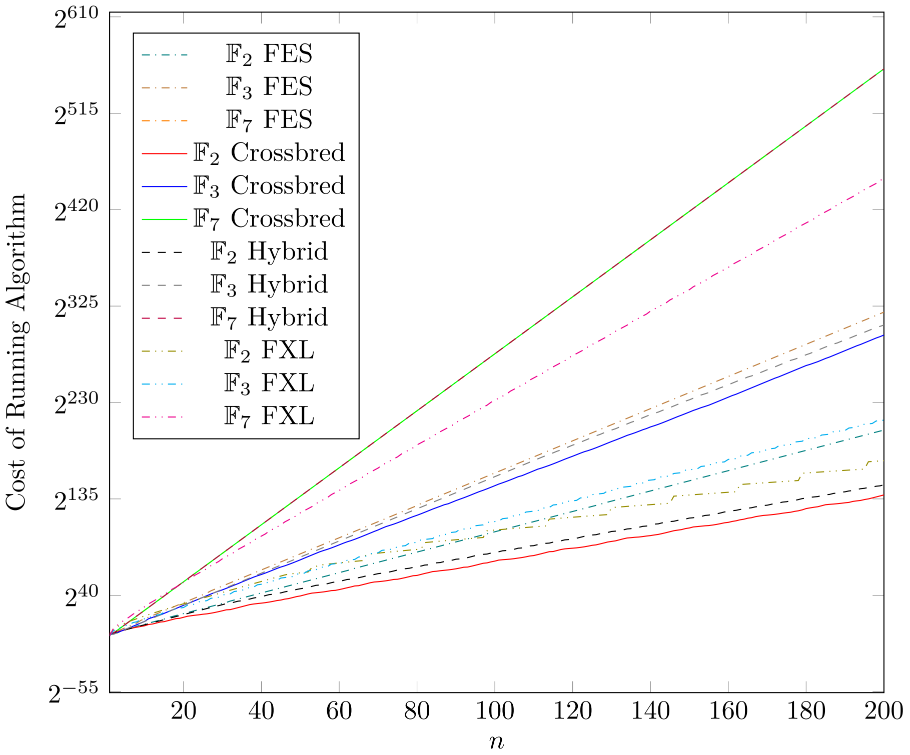

Figure 1: Optimal Cost of Running Crossbred, FES, FXL and Hybrid-F_5 as $m = 2n$ increases.

that for larger fields, the Crossbred algorithm does not provide an improved asymptotic complexity over FES, FXL or Hybrid-F_5.

Whilst this question about whether the Crossbred attack poses any improvement over the state of the art algorithms, namely FES, FXL and Hybrid-F_5, is answered, there is still much research to be done. This is because the reason why this topic was investigated to begin with is because it is believed that there probably exists a better way of solving polynomial systems than Hybrid-F_5, FXL or FES and by scrutinising these sort of algorithms, we are able to fully understand them. Hence, we are able to make an informed decision of whether an algorithm is an improvement on previous work and if so, how much of an improvement. In the end, even a slight improvement may be the difference between breaking an algorithm in the real-world or not.

Acknowledgements

This work is based on the author's Master's thesis while at Royal Holloway, supervised by Prof. Martin Albrecht. Special thanks to Dr. Simona Samardijska and Dr. Thomas Attema for reviewing the manuscript before submission.

References

[Bar04] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* Theses, Université Pierre et Marie Curie - Paris VI, December 2004. URL: https://tel.archives-ouvertes.fr/tel-00449609.

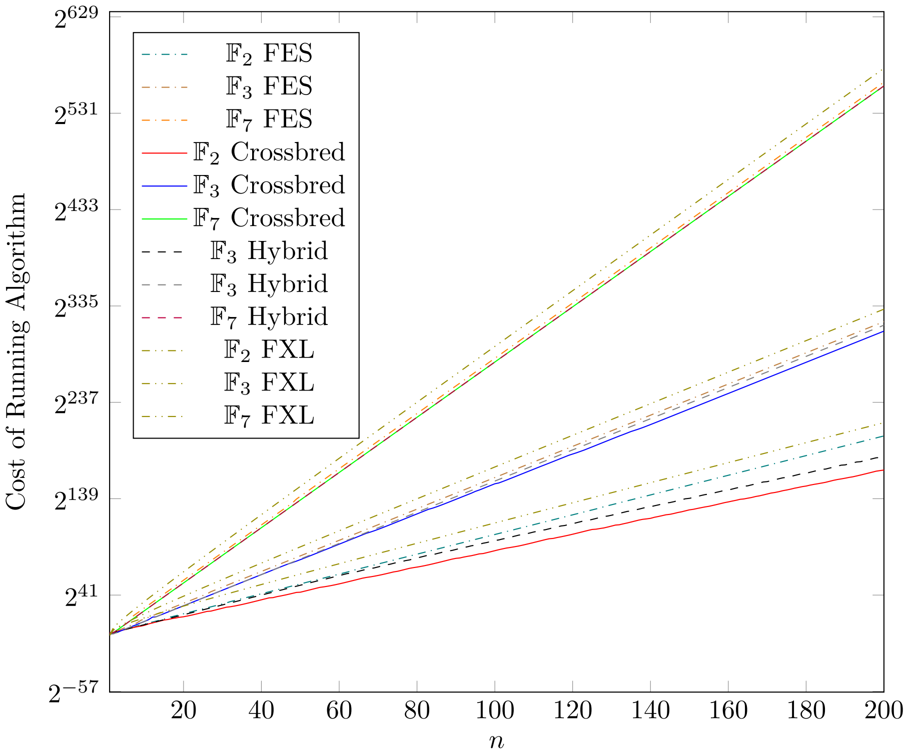

Figure 2: Optimal Cost of Running Crossbred, FES, FXL and Hybrid-F_5 as $m = n + 1$ increases.

[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 Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 203–218, Santa Barbara, CA, USA, August 17–20, 2010. Springer, Heidelberg, Germany. doi:10.1007/978-3-642-15031-9_14.

[BFP08] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid Approach for Solving Multivariate Systems over Finite Fields. *Journal of Mathematical Cryptology*, 3:177–197, 2008. doi:10.1515/JMC.2009.009.

[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, 02 2013. URL: http://dx.doi.org/10.1016/j.jco.2012.07.001, doi:10.1016/j.jco.2012.07.001.

[BFSY05] M Bardet, Jean-Charles Faugère, B Salvy, and Bo-Yin Yang. Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In *MEGA'05*, Sardinia, 2005. URL: https://www.iis.sinica.edu.tw/papers/byyang/2396-F.pdf.

[BPSV19] Ward Beullens, Bart Preneel, Alan Szepieniec, and Frederik Vercauteren. LUOV. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, page 392–407. Springer Berlin Heidelberg, 2000. URL: http://dx.doi.org/10.1007/3-540-45539-6_27, doi:10.1007/3-540-45539-6_27.

[CLO12] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties and Algorithms*. Springer International Publishing, 4 edition, 2012. doi:10.1007/978-3-319-16721-3.

[DY09] Jintai Ding and Bo-Yin Yang. *Multivariate Public Key Cryptography*, pages 193–241. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-88702-7_6.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 6 1999. URL: https://www.sciencedirect.com/science/ARTICLE/pii/S0022404999000055, doi:10.1016/S0022-4049(99)00005-5.

[Fau02] Jean Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 75–83, 01 2002. doi:10.1145/780506.780516.

[FHK^{+}17] Jean-Charles Faugère, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi, and Ludovic Perret. Fast quantum algorithm for solving multivariate quadratic equations. Cryptology ePrint Archive, Report 2017/1236, 2017. https://eprint.iacr.org/2017/1236.

[GLS98] M Giesbrecht, A Lobo, and B D Saunders. Certifying Inconsistency of Sparse Linear Systems. *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, pages 113–119, 1998. URL: http://doi.acm.org/10.1145/281508.281591, doi:10.1145/281508.281591.

[HMS14] T. J. Hodges, S. D. Molina, and J. Schlather. On the existence of semi-regular sequences, 2014. arXiv:1412.7865.

[JV17] Antoine Joux and Vanessa Vitse. A Crossbred Algorithm for Solving Boolean Polynomial Systems. In Jerzy Kaczorowski, Josef Pieprzyk, and Jacek Pomykała, editors, *Number-Theoretic Methods in Cryptology*, pages 3–21. Springer International Publishing, 2017. doi:10.1007/978-3-319-76620-1_1.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. *Advances in Cryptology — EUROCRYPT '99*, pages 206–222, 1999. URL: http://link.springer.com/10.1007/3-540-48910-X_15, doi:10.1007/3-540-48910-X_15.

[Laz83] Daniel Lazard. Gröbner-Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations. In *Proceedings of the European Computer Algebra Conference on Computer Algebra*, EUROCAL 8́3, pages 146–156, London, UK, 1983. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646657.700393.

[SCH^{+}19] Simona Samardjiska, Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. MQDSS. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions.

[YC05] Bo-Yin Yang and Jiun-Ming Chen. All in the XL family: Theory and practice. In Choonsik Park and Seongtaek Chee, editors, *ICISC 04: 7th International Conference on Information Security and Cryptology*, volume 3506 of *Lecture Notes in Computer Science*, pages 67–86, Seoul, Korea, December 2–3, 2005. Springer, Heidelberg, Germany.