An Algebraic Attack on Rank Metric Code-Based Cryptosystems

Magali Bardet[1,2], Pierre Briaud[2], Maxime Bros[3], Philippe Gaborit[3], Vincent Neiger[3], Olivier Ruatta[3], and Jean-Pierre Tillich[2]

[1] LITIS, University of Rouen Normandie, France
[2] Inria, 2 rue Simone Iff, 75012 Paris, France
[3] Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France

Abstract. The Rank metric decoding problem is the main problem considered in cryptography based on codes in the rank metric. Very efficient schemes based on this problem or quasi-cyclic versions of it have been proposed recently, such as those in the submissions ROLLO and RQC currently at the second round of the NIST Post-Quantum Cryptography Standardization Process. While combinatorial attacks on this problem have been extensively studied and seem now well understood, the situation is not as satisfactory for algebraic attacks, for which previous work essentially suggested that they were ineffective for cryptographic parameters. In this paper, starting from Ourivski and Johansson's algebraic modelling of the problem into a system of polynomial equations, we show how to augment this system with easily computed equations so that the augmented system is solved much faster via Gröbner bases. This happens because the augmented system has solving degree r, $r+1$ or $r+2$ depending on the parameters, where r is the rank weight, which we show by extending results from Verbel *et al.* (PQCrypto 2019) on systems arising from the MinRank problem; with target rank r, Verbel *et al.* lower the solving degree to $r+2$, and even less for some favorable instances that they call "superdetermined". We give complexity bounds for this approach as well as practical timings of an implementation using `magma`. This improves upon the previously known complexity estimates for both Gröbner basis and (non-quantum) combinatorial approaches, and for example leads to an attack in 200 bits on ROLLO-I-256 whose claimed security was 256 bits.

Keywords: Post-quantum cryptography · NIST-PQC candidates · rank metric code-based cryptography · Gröbner basis.

1 Introduction

Rank metric code-based cryptography. In the last decade, rank metric code-based cryptography has proved to be a powerful alternative to more traditional code-based cryptography based on the Hamming metric. This thread of research started with the GPT cryptosystem [37] based on Gabidulin codes [36], which are rank metric analogues of Reed-Solomon codes. However, the strong

algebraic structure of those codes was successfully exploited for attacking the original GPT cryptosystem and its variants with the Overbeck attack [53] (see for example [51] for one of the latest related developments). This has to be traced back to the algebraic structure of Gabidulin codes that makes masking extremely difficult; one can draw a parallel with the situation in the Hamming metric where essentially all McEliece cryptosystems based on Reed-Solomon codes or variants of them have been broken. However, recently a rank metric analogue of the NTRU cryptosystem from [44] has been designed and studied, starting with the pioneering paper [38]. Roughly speaking, the NTRU cryptosystem relies on a lattice that has vectors of rather small Euclidean norm. It is precisely those vectors that allow an efficient decoding/deciphering process. The decryption of the cryptosystem proposed in [38] relies on LRPC codes that have rather short vectors in the dual code, but this time for the rank metric. These vectors are used for decoding in the rank metric. This cryptosystem can also be viewed as the rank metric analogue of the MDPC cryptosystem [50] that relies on short vectors in the dual code for the Hamming metric.

This new way of building rank metric code-based cryptosystems has led to a sequence of proposals [38,40,5,6], culminating in submissions to the NIST post-quantum competition [1,2], whose security relies solely on the decoding problem in rank metric codes with a ring structure similar to the ones encountered right now in lattice-based cryptography. Interestingly enough, one can also build signature schemes using the rank metric; even though early attempts which relied on masking the structure of a code [41,9] have been broken [24], a promising recent approach [8] only considers random matrices without structural masking.

Decoding in rank metric. In other words, in rank metric code-based cryptography we are now only left with assessing the difficulty of the decoding problem for the rank metric. The rank metric over $\mathbb{F}_q^N$, where $\mathbb{F}_q$ is the finite field of cardinality q and $N = mn$ is a composite integer, consists in viewing elements in this ambient space as $m \times n$ matrices over $\mathbb{F}_q$ and considering the distance $d(\boldsymbol{X}, \boldsymbol{Y})$ between two such matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ as

$$d(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{Rank}\,(\boldsymbol{Y} - \boldsymbol{X})\,.$$

A (linear matrix) code $\mathcal{C}$ in $\mathbb{F}_q^{m\times n}$ is simply a $\mathbb{F}_q$-linear subspace in $\mathbb{F}_q^{m\times n}$, generated by K matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K$. The decoding problem for the rank metric at distance r is as follows: given a matrix $\boldsymbol{Y}$ in $\mathbb{F}_q^{m\times n}$ at distance $\leq r$ from $\mathcal{C}$, recover an element $\boldsymbol{M}$ in $\mathcal{C}$ at distance $\leq r$ from $\boldsymbol{Y}$. This is precisely the MinRank problem given as input $\boldsymbol{Y}$ and $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K$:

Problem 1 (MinRank).
Input: an integer $r \in \mathbb{N}$ and $K + 1$ matrices $\boldsymbol{Y}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_K \in \mathbb{F}_q^{m\times n}$.
Output: field elements $x_1, x_2, \ldots, x_K \in \mathbb{F}_q$ such that

$$\mathrm{Rank}\left(\boldsymbol{Y} - \sum_{i=1}^{K} x_i \boldsymbol{M}_i\right) \leq r.$$

As observed in [20], the MinRank problem is NP-complete and the best known algorithms solving it have exponential complexity bounds.

Matrix codes specified as $\mathbb{F}_{q^m}$-linear codes. However, the trend in rank metric code-based cryptography has been to consider a particular form of linear matrix codes: they are linear codes of length n over an extension $\mathbb{F}_{q^m}$ of degree m of $\mathbb{F}_q$, that is, $\mathbb{F}_{q^m}$-linear subspaces of $\mathbb{F}_{q^m}^n$. In the rest of this section, we fix a basis $(\beta_1, \ldots, \beta_m)$ of $\mathbb{F}_{q^m}$ as a $\mathbb{F}_q$-vector space. Then such codes can be interpreted as matrix codes over $\mathbb{F}_q^{m\times n}$ by viewing a vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ as a matrix $\mathrm{Mat}(\boldsymbol{x}) = (X_{ij})_{i,j}$ in $\mathbb{F}_q^{m\times n}$, where $(X_{ij})_{1\le i\le m}$ is the column vector formed by the coordinates of x_j in the basis $(\beta_1, \ldots, \beta_m)$, that is, $x_j = X_{1j}\beta_1 + \cdots + X_{mj}\beta_m$.

Then the "rank" metric d on $\mathbb{F}_{q^m}^n$ is the rank metric on the associated matrix space, namely

$$d(\boldsymbol{x}, \boldsymbol{y}) := |\boldsymbol{y} - \boldsymbol{x}|, \quad \text{where we define } |\boldsymbol{x}| := \mathrm{Rank}\left(\mathrm{Mat}(\boldsymbol{x})\right).$$

An $\mathbb{F}_{q^m}$-linear code $\mathcal{C}$ of length n and dimension k over $\mathbb{F}_{q^m}$ specifies a matrix code $\mathrm{Mat}(\mathcal{C}) := \{\mathrm{Mat}(\boldsymbol{c}) : \boldsymbol{c} \in \mathcal{C}\}$ in $\mathbb{F}_q^{m\times n}$ of dimension $K := mk$ over $\mathbb{F}_q$: it is readily verified that a basis of this $\mathbb{F}_q$-subspace is given by $(\mathrm{Mat}(\beta_i \boldsymbol{c}_j))_{1\le i\le m, 1\le j\le k}$ where $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$ is a basis of $\mathcal{C}$ over $\mathbb{F}_{q^m}$.

There are several reasons for this trend. On the one hand, the families of matrix codes for which an efficient decoding algorithm is known are families of $\mathbb{F}_{q^m}$-linear codes. On the other hand, $\mathbb{F}_{q^m}$-linear codes have a much shorter description than general matrix codes. Indeed, a matrix code in $\mathbb{F}_q^{m\times n}$ of dimension $K = km$ can be specified by a basis of it, which uses $Kmn\log(q) = km^2n\log(q)$ bits, whereas a matrix code obtained from an $\mathbb{F}_{q^m}$-linear code of dimension k over $\mathbb{F}_{q^m}$ can be specified by a basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$ of it, which uses $kmn\log(q)$ bits and thus saves a factor m.

Progress in the design of efficient algorithms for decoding $\mathbb{F}_{q^m}$-linear codes suggests that their additional structure may not have a significant impact on the difficulty of solving the decoding problem. For instance, a generic matrix code over $\mathbb{F}_q^{m\times n}$ of dimension $K = mk$ can be decoded using the information set decoder of [39] within a complexity of the order of q^{kr} when the errors have rank at most r and $m \ge n$, compared to q^{kr-m} for the decoding of a linear code over $\mathbb{F}_{q^m}^n$ in the same regime, using a similar decoder [10]. Moreover, even if the decoding problem is not known to be NP-complete for these $\mathbb{F}_{q^m}$-linear codes, there is a randomised reduction to an NP-complete problem [42] (namely to decoding in the Hamming metric). Hereafter, we will use the following terminology.

Problem 2 ((m, n, k, r)-decoding problem).
Input: an $\mathbb{F}_{q^m}$-basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$ of a subspace $\mathcal{C}$ of $\mathbb{F}_{q^m}^n$, an integer $r \in \mathbb{N}$, a vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ at distance at most r of $\mathcal{C}$ (i.e. $|\boldsymbol{y} - \boldsymbol{c}| \le r$ for some $\boldsymbol{c} \in \mathcal{C}$).
Output: $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ and $|\boldsymbol{e}| \le r$.

The region of parameters which is of interest for the NIST submissions corresponds to $m = \Theta(n)$, $k = \Theta(n)$ and $r = \Theta(\sqrt{n})$.

Gröbner basis techniques for decoding in the rank metric. The aforementioned algorithm from [10] for solving the decoding problem follows a combinatorial approach pioneered in [52], which is related to decoding techniques for the Hamming metric. Another approach consists in viewing the decoding problem as a particular case of MinRank and using the algebraic techniques designed for this problem; namely these techniques use a suitable algebraic modelling of a MinRank instance into a system of multivariate polynomial equations, and then solve this system with Gröbner basis techniques. Several modellings have been considered, such as the Kipnis-Shamir modelling [45] and the minors modelling (described for example in [34]); the complexity of solving MinRank using these modellings has been investigated in [33,34]. The Kipnis-Shamir modelling boils down to a polynomial system which is *affine bilinear*. This means that each equation has degree at most 2 and the set of variables can be partitioned into two sets $\{x_1, \ldots, x_s\} \cup \{y_1, \ldots, y_t\}$ such that all monomials of degree 2 involved in the equations are of the form $x_i y_j$; in other words, the equations are formed by a quadratic part which is bilinear plus an affine part. Although the complexity of solving this system can be bounded by that of solving bilinear systems, which is studied in [35], the complexity estimates thus obtained are very pessimistic, as observed experimentally in [21]. A theoretical explanation of why Gröbner basis techniques perform much better on the Kipnis-Shamir modelling than on generic bilinear systems was later given in [56]. It was also demonstrated there that the Kipnis-Shamir approach is more efficient than the minors approach on several multivariable encryption or signature schemes relying on the MinRank problem. However, the speed-up obtained for the Kipnis-Shamir modelling in the latter reference mostly comes from the "superdetermined" case considered therein. When applied to the (m, n, k, r)-decoding problem, this corresponds to the case where $m = n$ and $km < nr$; this condition is not met in the decoding problem instances we are interested in.

Another algebraic approach to solve the (m, n, k, r)-decoding problem was suggested in [39, §V.]. It is based on a new modelling specific to $\mathbb{F}_{q^m}$-linear codes which fundamentally relies on the underlying $\mathbb{F}_{q^m}$-linear structure and on q-polynomials. Also, it results in a system of polynomial equations that are sparse and have large degree. This approach seems to be efficient only if rk is not much larger than n.

Our contribution. If one compares the best known complexity estimates, the algebraic techniques appear to be less efficient than the combinatorial ones, such as [52], [39], and [10] for the parameters of the rank metric schemes proposed to the NIST [7,3] or of other rank metric code-based cryptosystems [49]. In [55], Levy-dit-Vehel and Perret pioneered the use of Gröbner basis techniques to solve the polynomial system arising in the Ourivski-Johansson algebraic modelling [52], with promising practical timings. In this paper, we follow on from this approach and show how this polynomial system can be augmented with additional equations that are easy to compute and bring on a substantial speed-up in the Gröbner basis computation for solving the system. This new algebraic

algorithm results in the best practical efficiency and complexity bounds that are currently known for the decoding problem; in particular, it significantly improves upon the above-mentioned combinatorial approaches.

There are several reasons why the Ourivski-Johansson algebraic modelling improves upon the Kipnis-Shamir one. First, it has the same affine bilinear structure and a similar number of equations, but it involves much fewer variables. Indeed, for the case of interest to us where m and k are in $\Theta(n)$ and r is in $\Theta(n^{1/2})$, the Kipnis-Shamir modelling involves $\Theta(n^2)$ equations and variables, while the Ourivski-Johansson one involves $\Theta(n^2)$ equations and $\Theta(n^{3/2})$ variables. Second, this modelling naturally leads to what corresponds to reducing by one the value of r, as explained in Section 3. Third, and most importantly, the main properties that ensure that the Kipnis-Shamir modelling behaves much better with respect to Gröbner basis techniques than generic bilinear systems also hold for the Ourivski-Johansson modelling. In essence, this is due to a *solving degree* which is remarkably low: at most $r+2$ for the former modelling and at most $r+1$ for the latter. Recall that the solving degree indicates the maximum degree reached during a Gröbner basis computation; it is known to be a strong predictor of the complexity of the most expensive step in a Gröbner basis computation and has been widely used for this purpose with confirmations via numerical experiments, see for instance [43,29,26,27,28,56].

To prove the third point, we start from the result about degree falls at the core of [56], which is based on work from [35], and we extend it to a more general setting which includes the Ourivski-Johansson modelling. In our case, these degree falls mean that from the initial system of quadratic equations $f_i = 0$ of the Ourivski-Johansson modelling, we are able to build many new equations of degree r that are combinations $\sum_i f_i g_{ij} = 0$ where the g_{ij}'s are polynomials of degree $r-1$ involved in the j-th new equation. We also prove that, when the parameters satisfy the condition

$$m\binom{n-k-1}{r} \geq \binom{n}{r} - 1, \tag{1}$$

by using that these polynomials $\sum_i f_i g_{ij}$ can be expressed as linear combinations of only a few other polynomials, we can perform suitable linear combinations of the equations $\sum_i f_i g_{ij} = 0$'s giving $\binom{n-1}{r-1} - 1$ equations of degree $r-1$. All these polynomial combinations are easily computed from the initial quadratic equations. By adding these equations and then performing Gröbner basis computations on the augmented system, we observe that experimentally the Gröbner basis algorithm behaves as expected from the degree fall heuristic:

- if (1) holds, this degree is r and the overall complexity is $\mathcal{O}\left(\left(\frac{((m+n)r)^r}{r!}\right)^\omega\right)$ operations in $\mathbb{F}_q$.
- if (1) does not hold, the maximum degree reached in the Gröbner basis computation is $r+1$ (in some intermediate cases), or $r+2$, leading to an overall complexity of at most $\mathcal{O}\left(\left(\frac{((m+n)r)^{r+1}}{(r+1)!}\right)^\omega\right)$ (resp. $\mathcal{O}\left(\left(\frac{((m+n)r)^{r+2}}{(r+2)!}\right)^\omega\right)$) operations in $\mathbb{F}_q$, where ω is the exponent of matrix multiplication;

Note that for a majority of parameters proposed in [7,3], the condition (1) holds. Taking for ω the smallest value currently achievable in practice, which is $\omega \approx 2.8$ via Strassen's algorithm, this leads to an attack on the cryptosystems proposed in the aforementioned NIST submissions which is in all cases below the claimed classical security levels.

2 Notation

In the whole paper, we use the following notation and definitions:

- Matrices and vectors are written in boldface font $\boldsymbol{M}$.
- For a matrix $\boldsymbol{M}$ its entry in row i and column j is denoted by $\boldsymbol{M}[i,j]$.
- The transpose of a matrix $\boldsymbol{M}$ is denoted by $\boldsymbol{M}^{\mathsf{T}}$.
- For a given ring $\mathcal{R}$, the space of matrices with m rows and n columns and coefficients in $\mathcal{R}$ is denoted by $\mathcal{R}^{m\times n}$.
- For $\boldsymbol{M} \in \mathcal{R}^{m\times n}$, we denote by $vec_{row}(\boldsymbol{M})$ the column vector formed by concatenating the rows of $\boldsymbol{M}$, i.e. $vec_{row}(\boldsymbol{M}) = \left(\boldsymbol{M}_{\{1\},*} \ldots \boldsymbol{M}_{\{n\},*}\right)^{\mathsf{T}}$.
- For $\boldsymbol{M} \in \mathcal{R}^{m\times n}$, we denote by $vec_{col}(\boldsymbol{M})$ the column vector formed by concatenating the columns of $\boldsymbol{M}$, i.e. $vec_{col}(\boldsymbol{M}) = vec_{row}(\boldsymbol{M}^{\mathsf{T}})$.
- $\{1..n\}$ stands for the set of integers from 1 to n, and for any subset $J \subset \{k+1..n\}$, we denote by $J-k$ the set $J-k = \{j-k : j \in J\} \subset \{1..n-k\}$.
- For two subsets $I \subset \{1..m\}$ and $J \subset \{1..n\}$, we write $\boldsymbol{M}_{I,J}$ for the submatrix of $\boldsymbol{M}$ formed by its rows (resp. columns) with index in I (resp. J).
- We use the shorthand notation $\boldsymbol{M}_{*,J} = \boldsymbol{M}_{\{1..m\},J}$ and $\boldsymbol{M}_{I,*} = \boldsymbol{M}_{I,\{1..n\}}$, where $\boldsymbol{M}$ has m rows and n columns.
- $\mathbb{F}_q$ is a finite field of size q, and $\alpha \in \mathbb{F}_{q^m}$ is a primitive element, so that $(1, \alpha, \ldots, \alpha^{m-1})$ is a basis of $\mathbb{F}_{q^m}$ as an $\mathbb{F}_q$-vector space. For $\beta \in \mathbb{F}_{q^m}$, we denote by $[\alpha^{i-1}]\beta$ its ith coordinate in this basis.
- For $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}_{q^m}^n$. The *support* of $\boldsymbol{v}$ is the $\mathbb{F}_q$-vector subspace of $\mathbb{F}_{q^m}$ spanned by the vectors $v_1, \ldots, v_n$. Thus this support is the column space of the matrix $\mathrm{Mat}(\boldsymbol{v})$ associated to $\boldsymbol{v}$ (for any choice of basis), and its dimension is precisely $\mathrm{Rank}(\mathrm{Mat}(\boldsymbol{v}))$.
- An $[n,k]$ $\mathbb{F}_{q^m}$-linear code is an $\mathbb{F}_{q^m}$-linear subspace of $\mathbb{F}_{q^m}^n$ of dimension k endowed with the rank metric.

3 Algebraic modellings of the decoding problem

In what follows, parameters are chosen in the cryptographically relevant region mentionned in the introduction, say $m = \Theta(n)$, $k = \Theta(n)$ and $r = \Theta(\sqrt{n})$. Decoding instances will then have a single solution $\boldsymbol{e}$. For simplicity, we assume that the rank of $\boldsymbol{e}$ is exactly r; in general one can run the algorithm for increasing values of the target rank up to r, until a solution is found, and the most expensive step will correspond to the largest considered rank. We consider here the (m,n,k,r)-decoding problem for the code $\mathcal{C}$ and assume we have received $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ at distance r from $\mathcal{C}$ and look for $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e}$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ and $|\boldsymbol{e}| = r$.

3.1 Solving the MinRank instance using Kipnis-Shamir's modelling

As explained in Section 1, a possible approach to perform the decoding is to solve the underlying MinRank instance with $km+1$ matrices in $\mathbb{F}_q^{m\times n}$; this is done by introducing $\boldsymbol{M}_0 := \mathrm{Mat}(\boldsymbol{y})$ and $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{km}$ which is an $\mathbb{F}_q$-basis of $\mathrm{Mat}(\mathcal{C})$. Several methods have been developed, and so far the Kipnis-Shamir modelling [45] seems to be the most efficient to solve this MinRank instance. We want to find $(z_0, \ldots, z_{km})$ in $\mathbb{F}_q^{mk+1}$ such that $\mathrm{Rank}(\sum_{i=0}^{km} z_i \boldsymbol{M}_i) = r$. $(z_0, z_1, \ldots, z_{km})$ is a solution to the MinRank problem if and only if the right kernel of $\sum_{i=0}^{km} z_i \boldsymbol{M}_i$ contains a subspace of dimension $n-r$ of $\mathbb{F}_q^n$. With high probability, a basis of such a space can be written in systematic form, that is, in the form $\begin{pmatrix} \boldsymbol{I}_{n-r} \\ \boldsymbol{K} \end{pmatrix}$. Thus we have to solve the system

$$\left(\sum_{i=0}^{km} z_i \boldsymbol{M}_i\right) \begin{pmatrix} \boldsymbol{I}_{n-r} \\ \boldsymbol{K} \end{pmatrix} = 0, \tag{2}$$

over $\mathbb{F}_q$, where $\boldsymbol{K}$ is an $r \times (n-r)$ matrix of indeterminates. This system is affine bilinear and has $m(n-r)$ equations and $km+1+r(n-r)$ variables, which are $z_0, z_1, \ldots, z_{km}$ and the $r(n-r)$ entries of $\boldsymbol{K}$; each equation has a bilinear part as well as a linear part which only involves the variables z_i.

3.2 Syndrome modelling

We recall here the modelling considered in [7,2]. Let $\boldsymbol{H}$ be a parity-check matrix of $\mathcal{C}$, i.e.

$$\mathcal{C} = \{\boldsymbol{c} \in \mathbb{F}_{q^m}^n : \boldsymbol{c}\boldsymbol{H}^\mathsf{T} = \boldsymbol{0}\}.$$

The (m,n,k,r)-decoding problem can be algebraically described by the system $\boldsymbol{e}\boldsymbol{H}^\mathsf{T} = \boldsymbol{s}$ where $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ has rank r and $\boldsymbol{s} \in \mathbb{F}_{q^m}^{(n-k)}$ is given by $\boldsymbol{s} := \boldsymbol{y}\boldsymbol{H}^\mathsf{T}$. Let $(S_1, \ldots, S_r) \in \mathbb{F}_{q^m}^r$ be a basis of the support of $\boldsymbol{e}$; then, $\boldsymbol{e} = (S_1 \ \cdots \ S_r)\boldsymbol{C}$, where $\boldsymbol{C} \in \mathbb{F}_q^{r\times n}$ is the matrix of the coordinates of $\boldsymbol{e}$ in the basis $(S_1, \ldots, S_r)$. Then expressing the elements S_i in the basis $(1, \alpha, \ldots, \alpha^{m-1})$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ yields $(S_1 \ \cdots \ S_r) = (1 \ \alpha \ \cdots \ \alpha^{m-1})\boldsymbol{S}$ for some matrix $\boldsymbol{S} \in \mathbb{F}_q^{m\times r}$. Thus, the system is rewritten as

$$\left(1 \ \alpha \cdots \ \alpha^{m-1}\right) \boldsymbol{S}\boldsymbol{C}\boldsymbol{H}^\mathsf{T} = \boldsymbol{s}, \ \text{ over } \mathbb{F}_{q^m} \text{ with solutions in } \mathbb{F}_q. \tag{3}$$

This polynomial system, that we refer to as the *syndrome modelling*, has $m(n-k)$ equations and $mr+nr$ variables when it is written over $\mathbb{F}_q$. It is affine bilinear (without terms of degree 1) with respect to the two sets of variables coming from the support and from the coordinates of the error. Besides, this system admits $(q^r-1)(q^r-q)\cdots(q^r-q^{r-1})$ solutions since this is the number of bases of the support. These solutions to the system all correspond to the same unique solution $\boldsymbol{e}$ of the initial decoding problem. We can easily impose a unique solution by fixing some of the unknowns as in the Kipnis-Shamir modelling, or as has been done in the Ourivski-Johansson modelling that we will present next. It

is worthwhile to note that this kind of modelling has, as the Kipnis-Shamir modelling, $\Theta\left(n^2\right)$ equations for our choice of parameters but significantly fewer variables since we now have only $\Theta\left(n^{3/2}\right)$ unknowns. The Ourivski-Johansson's modelling will be a related modelling that gives a further improvement.

3.3 Ourivski-Johansson's modelling

We now describe the algebraic modelling considered in the rest of this paper, which is basically Ourivski and Johansson's one [52]. It can be viewed as an homogenising trick. Instead of looking for $\boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{e}$ of rank r that satisfy $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, or what is the same for $\boldsymbol{c} \in \mathcal{C}$ such that $|\boldsymbol{c} + \boldsymbol{y}| = r$, we look for $\boldsymbol{c} \in \mathcal{C}$ and $\lambda \in \mathbb{F}_{q^m}$ such that

$$|\boldsymbol{c} + \lambda \boldsymbol{y}| = r. \tag{4}$$

It is precisely here that the $\mathbb{F}_{q^m}$-linearity of $\mathcal{C}$ is used in a crucial way. Once we have found such a $\boldsymbol{c}$ and λ, we have found a $\boldsymbol{c} + \lambda \boldsymbol{y}$ such that $\boldsymbol{c} + \lambda \boldsymbol{y} = \mu \boldsymbol{e}$ for some non-zero $\mu \in \mathbb{F}_{q^m}$ from which we deduce easily $\boldsymbol{e}$. The point of proceeding this way is that there are $q^m - 1$ solutions to (4) and that this allows us to fix more unknowns in the algebraic system. Another point of view [52, Sec. 2] is to say that we introduce the code $\tilde{\mathcal{C}} := \mathcal{C} + \langle \boldsymbol{y} \rangle$ and that we look for a rank r word in $\tilde{\mathcal{C}}$, since all such words are precisely the multiples $\lambda \boldsymbol{e}$ for nonzero $\lambda \in \mathbb{F}_{q^m}$ of the error $\boldsymbol{e}$ we are looking for. Let $\tilde{\boldsymbol{G}} = \left(\boldsymbol{I}_{k+1} \ \boldsymbol{R}\right)$ (resp. $\tilde{\boldsymbol{H}} = \left(-\boldsymbol{R}^{\mathsf{T}} \ \boldsymbol{I}_{n-k-1}\right)$) be the generator matrix in systematic form (resp. a parity-check matrix) of the extended code $\tilde{\mathcal{C}}$; note that for a vector $\boldsymbol{v}$, we have $\boldsymbol{v} \in \tilde{\mathcal{C}}$ if and only if $\boldsymbol{v}\tilde{\boldsymbol{H}}^{\mathsf{T}} = 0$. Using the notation $\boldsymbol{e} = (1 \ \ \alpha \ \ \cdots \ \ \alpha^{m-1})\boldsymbol{S}\boldsymbol{C}$ as above, and writing $\boldsymbol{C} = (\boldsymbol{C}_1 \ \ \boldsymbol{C}_2)$ with $\boldsymbol{C}_1 \in \mathbb{F}_q^{r \times (k+1)}$ and $\boldsymbol{C}_2 \in \mathbb{F}_q^{r \times (n-k-1)}$, the fact that $\boldsymbol{e} \in \tilde{\mathcal{C}}$ yields the system

$$\left(1 \ \alpha \ \cdots \ \alpha^{m-1}\right) \boldsymbol{S} \left(\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{R}\right) = 0, \ \text{ over } \mathbb{F}_{q^m} \text{ with solutions in } \mathbb{F}_q. \tag{5}$$

Since all multiples $\lambda \boldsymbol{e}$ are solutions of this system, we can specify the first column of $\boldsymbol{C}$ to $(1 \ 0 \ \cdots \ 0)^{\mathsf{T}}$. In this way, there is a single $\lambda \boldsymbol{e}$ satisfying these constraints: the one where λ is the inverse of the first coordinate of $\boldsymbol{e}$ (assuming it is nonzero, see below). The system still admits several solutions which correspond to different bases of the support of $\lambda \boldsymbol{e}$. To fix one basis of this support, similarly to what is done in [52, Sec. 3], we can specify $S_1 = 1$, or equivalently, set the first column of $\boldsymbol{S}$ to be $(1 \ 0 \ \cdots \ 0)^{\mathsf{T}}$, and take an $r \times r$ invertible submatrix of $\boldsymbol{S}$ and specify it to be the identity matrix; thus the system has a single solution.

Doing so, the resulting system is affine bilinear (without constant term), with $(n - k - 1)m$ equations and $(m - 1)r + nr$ variables, and has a unique solution.

For the sake of presentation, in Section 5 we present our results assuming that the first coordinate of $\boldsymbol{e}$ is nonzero and that the top $r \times r$ block of $\boldsymbol{S}$ is invertible; these results are easily extended to the general case. Under these assumptions, our system can be rewritten as follows:

$$\mathcal{F} = \left\{ \left(1 \ \alpha \ \cdots \ \alpha^{m-1}\right) \left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0} & \boldsymbol{S}' \end{array}\right) \left(\boldsymbol{C}_2 - \left(\begin{array}{c|c} 1 & \\ \boldsymbol{0} & \boldsymbol{C}_1' \end{array}\right) \boldsymbol{R}\right) \right\}, \tag{6}$$

where $\boldsymbol{S}'$ is the $(m-r)\times(r-1)$ submatrix $\boldsymbol{S}_{\{r+1..m\},\{2..r\}}$ and $\boldsymbol{C}'_1$ is the $r\times k$ submatrix $\boldsymbol{C}_{*,\{2..k+1\}}$. We call the entries of $\boldsymbol{S}'$ the *support variables* whereas the entries of $\boldsymbol{C}'_1$ and $\boldsymbol{C}_2$ are called the *coefficient variables*. In Section 6.2 we give a procedure to handle the general case, by making several attempts to find the invertible block of $\boldsymbol{S}$ and a nonzero component of $\boldsymbol{e}$.

4 Gröbner bases and degree falls

We refer to [23] for basic definitions and properties of monomial orderings and Gröbner bases.

Field equations and monomial ordering Since we are looking for solutions in $\mathbb{F}_q$, we augment the polynomial system we want to solve with the field equations, that is, the equation $x_i^q - x_i = 0$ for each variable x_i arising in the system. In our case, as the system we consider in practice has mainly only one solution in $\mathbb{F}_q$ (see Section 6), the ideal of the system with the field equations is radical, and for any monomial ordering the reduced Gröbner basis is the set of linear polynomials $\{x_i - a_i\}_i$, where $\{x_i\}_i$ are the variables and $a_i \in \mathbb{F}_q$ is the i-th coordinate of the solution. The classical approach consists in computing the Gröbner basis with respect to a degree-reverse lexicographic order (grevlex), that will keep the degree of the polynomials as small as possible during the computation, and behaves usually better than other monomial orderings in terms of complexity.

Generic Gröbner bases algorithms and their link with linear algebra Since the first descriptions of algorithms to compute Gröbner bases [18], far more efficient algorithms have been developed. On the one hand, substantial practical speed-ups were achieved by incorporating and accelerating fast linear algebra operations such as Gaussian elimination on the Macaulay matrices, which are sparse and structured (see Faugère's F4 algorithm [31], variants of the XL algorithm [22], and for instance GBLA [17]). We recall that the Macaulay matrix in degree d of a homogeneous system $(f_i)_i$ is the matrix whose columns correspond to the monomials of degree d sorted in descending order w.r.t. a chosen monomial ordering, whose rows correspond to the polynomials tf_i for all i where t is a monomial of degree $d - \deg(f_i)$, and whose entry in row tf_i and column u is the coefficient of the monomial u in the polynomial tf_i. In the case of a system containing field equations, we consider compact Macaulay matrices, where all monomials are reduced w.r.t. the field equations. For an affine system, the Macaulay matrix in degree d contains all polynomials $\{tf_i\}$ for $\deg(tf_i) \leq d$ and the columns are the monomials of degree less than or equal to d.

The approaches from F4 or XL are similar in that they both compute row echelon forms of some submatrices of Macaulay matrices for some given degree; in fact, it was proven in [11] that the XL algorithm computes a so-called d-Gröbner basis, which is a basis of the initial system where all computations in degree larger than d are ignored, and that one can rephrase the original XL algorithm in terms of the original F4 algorithm.

Now, many variants of these algorithms have been designed to tackle specific families of polynomial systems, and it seems that none of them performs always better than the others. In our experimental considerations, we rely on the implementation of the F4 algorithm which is available in `magma V2.22-2` and is recognised for its efficiency.

On the other hand, improvements have been obtained by refining criteria which allow one to avoid useless computations (avoiding to consider monomials that cannot appear, a priori detection of reductions to zero as in the F5 algorithm [32] and other signature-based algorithms that followed, see [30] for a survey).

Complexity analysis for homogeneous systems For *homogeneous* systems, and for a graded monomial ordering, the complexity of these algorithms in terms of arithmetic operations is dominated by the cost of the row echelon forms on all Macaulay matrices up to degree d, where d is the largest degree of a polynomial in the reduced Gröbner basis[4]. This degree d is called the *index of regularity*, or *degree of regularity*, and it only depends on the ideal generated by the system, not on the specific generators forming the system. Some algorithms may need to go beyond degree d to check that no new polynomials will be produced, like the XL Algorithm or the F4 Algorithm without the F5 criteria, but those computations may be avoided if one knows in advance the degree of regularity of the system. This parameter can be precisely estimated for different families of generic systems, using the notions of regularity, of semi-regularity in the overdetermined case, and of bi-regularity in the bilinear case [12,15,14,35]. However, those bounds may be very pessimistic for other specific (sub-)families of systems, and deriving estimations in this situation is difficult a priori, in particular for affine systems.

Definition 1. *Let $(f_i)_i$ be (non necessarily homogeneous) polynomials in a polynomial ring $\mathcal{R}$. A syzygy is a vector $(s_i)_i$, $s_i \in \mathcal{R}$ such that $\sum_i s_i f_i = 0$. The degree of the syzygy is defined as $\max_i(\deg(f_i) + \deg(s_i))$. The set of all syzygies of $(f_i)_i$ is an $\mathcal{R}$-module called the syzygy module of $(f_i)_i$.*

For a given family of systems, there are syzygies that occur for any system in the family. For instance, for any system $(f_i)_i$, the syzygy module contains the $\mathcal{R}$-module spanned by the so-called *trivial syzygies* $(e_j f_i - e_i f_j)_{i,j}$, where e_i is the coordinate vector with 1 at index i. A system is called *regular* if its syzygy module is generated by these trivial syzygies.

Let us consider the particular case of a zero-dimensional system $(f_i)_i$ of homogeneous polynomials, generating an ideal I. As the system is homogenous and has a finite number of solution, then it must have only 0 as a solution (with maybe some multiplicities). In this case, the degree of regularity of the system is the lowest integer d_{reg} such that all monomials of degree d_{reg} are in

[4] If the system contains redundant polynomials of degree larger than d, additional operations are needed to check that those polynomials reduce to zero w.r.t. the Gröbner basis, but this has usually a negligible cost.

the ideal of leading terms of I (see [12,15]). Such a system is called *semi-regular* if the set of its syzygies of degree less than $d_{\mathrm{reg}}(I)$ is exactly the set of trivial syzygies of degree less than $d_{\mathrm{reg}}(I)$. Note that there may be non-trivial syzygies in degree $d_{\mathrm{reg}}(I)$, which may be different for each system. As a consequence, all polynomials occurring in the computation of a Gröbner basis have degree $\leq d_{\mathrm{reg}}$ and the arithmetic complexity is bounded by the cost of the row echelon form on the Macaulay matrices in degree $\leq d_{\mathrm{reg}}$.

Complexity analysis for affine systems For affine systems, things are different. The degree of regularity can be defined in the same way w.r.t. the Gröbner basis for a grevlex ordering. But is not any more related to the complexity of the computation: for instance, a system with only one solution will have a degree of regularity equal to 1. We need another parameter to control the complexity of the computation.

Let $(f_i)_i$ be a system of affine polynomials, and f_i^h the homogeneous part of highest degree of f_i. Let $I = \langle \{f_i\}_i \rangle$ and $I^h = \langle \{f_i^h\}_i \rangle$, and let d_{reg}^h be the degree of regularity of I^h. What may happen is that, during the computation of the basis in some degree d, some polynomials of degree less than d may be added to the basis. This will happen any time a syzygy $(s_i^h)_i$ for $(f_i^h)_i$ of degree d is such that there exists no syzygy $(s_i)_i$ for $(f_i)_i$ where s_i^h is the homogeneous part of highest degree of s_i. In that case, $\sum_i s_i^h f_i$ is a polynomial of degree less than d (the homogeneous part of highest degree cancels), that will not be reduced to zero during the Gröbner basis computation since this would give a syzygy $(s_i)_i$ for $(f_i)_i$ with homogeneous part $(s_i^h)_i$. This phenomenon is called a *degree fall* in degree d, and we will call such syzygies (s_i^h) that cannot be extended to syzygies for $(f_i)_i$ in the same degree *partial syzygies*; the corresponding polynomial $\sum_i s_i^h f_i$ is called the *residue*.

In cryptographic applications, the *first degree fall* d_{ff} has been widely used as a parameter controlling the complexity in algebraic cryptanalysis, for instance in the study of some HFE-type systems [29,43,25] and Kipnis-Shamir systems [56]. This first degree fall is simply the smallest d such that there exists a degree fall in degree d on $(f_i)_i$, and this quantity does depend on $(f_i)_i$: it might be different for another set of generators of the same ideal. Still, this notion takes on its full meaning while computing a Gröbner basis for a graded ordering, if we admit that the algorithm terminates shortly after reaching the first degree fall and without considering polynomials of higher degree. This can happen for some families of systems, as explained in the next paragraph, but there are examples of systems where the first degree fall d_{ff} is not the maximal degree reached during the computation, in which case it is not related to the complexity of the computation.

If the system $(f_i^h)_i$ is semi-regular, then the computation in degree less than d_{reg}^h will act as if the polynomials where homogeneous: there cannot be degree falls, as they would correspond to syzygies for the system $(f_i^h)_i$ that is assumed to be semi-regular. In degree d_{reg}^h, degree falls will occur for the first time, but at this point the remainder of the computation is negligible compared to the previous ones: by definition of d_{reg}^h, all monomials of degree d_{reg}^h are leading terms of

polynomials in the basis, and the remaining steps in the computation will necessarily deal with polynomials of degree at most d_{reg}^h. Hence, the computations are almost the same as the ones for $(f_i^h)_i$, and the complexity is controlled by d_{reg}^h, which is here the *first degree fall* for the system $(f_i)_i$.

The behavior of the computation may be very different if degree falls occur in a much smaller degree. A good example of what may happen for particular families of systems is the affine bilinear case. It is proven in [35, Prop. 5] that a generic affine bilinear system of m equations $(f_1, \ldots, f_m) \in \mathbb{K}[x_1, \ldots, x_{n_x}, y_1, \ldots, y_{n_y}]$ in $n_x + n_y \geq m$ variables is regular. In particular, the Macaulay bound $d_{\text{reg}} \leq n_x + n_y + 1$ applies [46]. However, it was also proven in [35, Thm. 6] that for a zero-dimensional affine bilinear system ($m = n_x + n_y$), d_{reg} satisfies a much sharper inequality $d_{\text{reg}} \leq \min(n_x + 1, n_y + 1)$. The reason is that (homogeneous) bilinear systems are not regular, but the syzygy module of those systems is well understood [35]. In particular, there are syzygies for $(f_i^h)_i$ coming from Jacobian matrices, that are partial syzygies for $(f_i)_i$ and produce degree falls.

For affine systems, that are mainly encountered in cryptographic applications, and in particular for systems coming from a product of matrices whose coefficients are the variables of the system, the Jacobian matrices have a very particular shape that is easily described, and leads to a series of degree falls that reduces the degree of regularity of those systems. This is explained in detail in Section 5.

5 Degree falls and low degree equations

5.1 Degree falls from the kernel of the Jacobian

Fundamental results from [35,56]. It has been realized in [56] that the first degree fall in the Kipnis and Shamir modelling can be traced back to partial syzygies obtained from low degree vectors in the kernel of the Jacobian of the bilinear part of a system either with respect to the kernel variables or the linear variables. This argument can also be adapted to our case and Jacobians with respect to the support variables are relevant here. To understand the relevance of the Jacobians for bilinear affine systems over some field $\mathbb{K}$ in general, consider a bilinear affine system $\mathcal{F} = \{f_1, \ldots, f_M\} \subset \mathbb{K}[s_1, \ldots, s_{t_s}, c_1, \ldots, c_{t_c}]$ of M equations in t_s variables s and t_c variables c. We denote by $\mathcal{F}^h := \{f_1^h, \ldots, f_M^h\}$ the bilinear part of these equations. In other words each f_i can be written as

$$f_i = f_i^h + r_i,$$

where each r_i is affine and ${f_i}^h$ is bilinear with respect to $\{s_1, \ldots, s_{t_s}\} \cup \{c_1, \ldots, c_{t_c}\}$. We define the Jacobian matrices associated to $\mathcal{F}^h$ as

$$\text{Jac}_{\boldsymbol{S}}(\mathcal{F}^h) = \begin{pmatrix} \frac{\partial f_1^h}{\partial s_1} & \cdots & \frac{\partial f_1^h}{\partial s_{t_s}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_M^h}{\partial s_1} & \cdots & \frac{\partial f_M^h}{\partial s_{t_s}} \end{pmatrix} \quad \text{and} \quad \text{Jac}_{\boldsymbol{C}}(\mathcal{F}^h) = \begin{pmatrix} \frac{\partial f_1^h}{\partial c_1} & \cdots & \frac{\partial f_1^h}{\partial c_{t_c}} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_M^h}{\partial c_1} & \cdots & \frac{\partial f_M^h}{\partial c_{t_c}} \end{pmatrix}.$$

Note that $\mathrm{Jac}_{\boldsymbol{S}}(\mathcal{F}^h)$ is a matrix with linear entries in $\mathbb{K}[c_1, \ldots, c_{t_c}]$ whereas $\mathrm{Jac}_{\boldsymbol{C}}(\mathcal{F}^h)$ is a matrix with linear entries in $\mathbb{K}[s_1, \ldots, s_{t_s}]$. As shown in [56][Prop. 1 & 2] vectors in the left kernel of these Jacobians yield partial syzygies. This is essentially a consequence of the following identities that are easily verified:

$$\mathrm{Jac}_{\boldsymbol{S}}(\mathcal{F}^h) \begin{pmatrix} s_1 \\ \vdots \\ s_{t_s} \end{pmatrix} = \begin{pmatrix} f_1^h \\ \vdots \\ f_M^h \end{pmatrix} \quad \text{and} \quad \mathrm{Jac}_{\boldsymbol{C}}(\mathcal{F}^h) \begin{pmatrix} c_1 \\ \vdots \\ c_{t_c} \end{pmatrix} = \begin{pmatrix} f_1^h \\ \vdots \\ f_M^h \end{pmatrix}.$$

For instance, a vector $(g_1, \ldots, g_M)$ in the left kernel of $\mathrm{Jac}_{\boldsymbol{C}}(\mathcal{F}^h)$ is a syzygy for $\mathcal{F}^h$, as it satisfies

$$\sum_{i=1}^{M} g_i f_i^h = (g_1 \ \cdots \ g_M) \begin{pmatrix} f_1^h \\ \vdots \\ f_M^h \end{pmatrix} = (g_1 \ \cdots \ g_M) \, \mathrm{Jac}_{\boldsymbol{C}}(\mathcal{F}^h) \begin{pmatrix} c_1 \\ \vdots \\ c_{t_c} \end{pmatrix} = 0.$$

This gives typically a degree fall for $\mathcal{F}$ at degree $2 + \max(\deg g_i)$, with the corresponding residue given by

$$\sum_{i=1}^{M} g_i f_i = \sum_{i=1}^{M} g_i f_i^h + \sum_{i=1}^{M} g_i r_i = \sum_{i=1}^{M} g_i r_i.$$

These Jacobians are matrices with entries that are linear forms. The kernel of such matrices is well understood as shown by the next result.

Theorem 1 ([35]). *Let $\boldsymbol{M}$ be an $M \times t$ matrix of linear forms in $\mathbb{K}[s_1, \ldots, s_{t_s}]$. If $t < M$, then generically the left kernel of $\boldsymbol{M}$ is generated by vectors whose coefficients are maximal minors of $\boldsymbol{M}$, specifically vectors of the form*

$$\boldsymbol{V}_J = (\ldots, \underbrace{0}_{j \notin J}, \ldots, \underbrace{(-1)^{l+1} \det(\boldsymbol{M}_{J \setminus \{j\}, *})}_{j \in J, j = j_l}, \ldots)_{1 \leq j \leq M}$$

where $J = \{j_1 < j_2 < \cdots < j_{t+1}\} \subset \{1, \ldots, M\}, \#J = t + 1$.

A direct use of this result however yields degree falls that occur for very large degrees, namely at degrees $t_s + 2$ or $t_c + 2$. In the case of the Kipnis-Shamir modelling, the syndrome modelling or the Ourivski-Johansson modelling, due to the particular form of the systems, degree falls occur at much smaller degrees than for generic bilinear affine systems. Roughly speaking, the reason is that the Jacobian of a system coming from a matrix product splits as a tensor product, as we now explain. This has been realized in [56] for the Kipnis-Shamir modelling, and here we slightly generalize this result in order to use it for more general modellings, and in particular for the Ourivski-Johansson modelling.

Jacobian matrices of systems coming from matrix products. Consider a system $\boldsymbol{AXY} = \boldsymbol{0}$ where $\boldsymbol{A} = (a_{i,s})_{1\leq i\leq m, 1\leq s\leq p}$, $\boldsymbol{X} = (x_{s,t})_{1\leq s\leq p, 1\leq t\leq r}$ and $\boldsymbol{Y} = (y_{t,j})_{1\leq t\leq r, 1\leq j\leq n}$. The variables considered for this Jacobian matrix are the $x_{s,t}$. The matrices $\boldsymbol{A}$ and $\boldsymbol{Y}$ may have polynomial coefficients, but they do not involve the $x_{s,t}$ variables. Below, we use the Kronecker product of two matrices, for example $\boldsymbol{A} \otimes \boldsymbol{Y}^\mathsf{T} = \left(a_{i,s}\boldsymbol{Y}^\mathsf{T}\right)_{1\leq i\leq m, 1\leq s\leq p}$. We use the notations $vec_{row}(\boldsymbol{A}) = \left(\boldsymbol{A}_{\{1\},*} \ \ldots \ \boldsymbol{A}_{\{n\},*}\right)^\mathsf{T}$ and $vec_{col}(\boldsymbol{A}) = vec_{row}(\boldsymbol{A}^\mathsf{T})$.

Lemma 1. *The Jacobian matrix of the system $\boldsymbol{AXY} = \boldsymbol{0}_{m\times n}$ with respect to the variables $\boldsymbol{X}$ can be written, depending on the order of the equations and variables:*

$$\mathrm{Jac}_{vec_{col}(\boldsymbol{X})}(vec_{col}(\boldsymbol{AXY})) = \boldsymbol{Y}^\mathsf{T} \otimes \boldsymbol{A} \in \mathbb{K}[\boldsymbol{A}, \boldsymbol{Y}]^{nm\times rp}$$
$$\mathrm{Jac}_{vec_{row}(\boldsymbol{X})}(vec_{row}(\boldsymbol{AXY})) = \boldsymbol{A} \otimes \boldsymbol{Y}^\mathsf{T} \in \mathbb{K}[\boldsymbol{A}, \boldsymbol{Y}]^{nm\times rp}.$$

Proof. For $1 \leq i \leq m$, $1 \leq j \leq n$, the equation in row i and column j of $\boldsymbol{AXY}$ is

$$f_{i,j} = \sum_{s=1}^{p}\sum_{t=1}^{r} a_{i,s}x_{s,t}y_{t,j}.$$

We then have, for $1 \leq s \leq p$ and $1 \leq t \leq r$, $\frac{\partial f_{i,j}}{\partial x_{s,t}} = a_{i,s}y_{t,j}$ so that in row order,

$$\mathrm{Jac}_{x_{s,1},\ldots,x_{s,r}}(\{f_{i,1},\ldots,f_{i,n}\}) = \left(\frac{\partial f_{i,j}}{\partial x_{s,t}}\right)_{\substack{1\leq j\leq n\\ 1\leq t\leq r}} = a_{i,s}\left(y_{t,j}\right)_{\substack{1\leq j\leq n\\ 1\leq t\leq r}} = a_{i,s}\boldsymbol{Y}^\mathsf{T}.$$

The result follows from the definition of the Kronecker product of matrices. The proof when the equations and variables are in column order is similar. □

Application to the Kipnis-Shamir modelling. Recall the system:

$$\left(\sum_{i=1}^{km} x_i\boldsymbol{M}_i\right)\begin{pmatrix}\boldsymbol{I}_{n-r}\\ \boldsymbol{K}\end{pmatrix} = \boldsymbol{0}_{m,n-r}, \tag{7}$$

where $\boldsymbol{M}_i \in \mathbb{F}_q^{m\times n}$ and $\boldsymbol{K}$ is an $r \times (n-r)$ matrix of indeterminates. If we write each $\boldsymbol{M}_i = (\boldsymbol{M}'_i \ \ \boldsymbol{M}''_i)$ with $\boldsymbol{M}'_i \in \mathbb{F}_q^{m\times(n-r)}$ and $\boldsymbol{M}''_i \in \mathbb{F}_q^{m\times r}$, then we have

$$\sum_{i=1}^{km} x_i\left(\boldsymbol{M}'_i + \boldsymbol{M}''_i\boldsymbol{K}\right) = \boldsymbol{0}_{m,n-r} \tag{KS}$$

The bilinear and linear parts of the system are respectively $\sum_{i=1}^{km} x_i\boldsymbol{M}''_i\boldsymbol{K}$ and $\sum_{i=1}^{km} x_i\boldsymbol{M}'_i$. Using Lemma 1 (with equations in column order), when we compute the Jacobian with respect to the entries of $\boldsymbol{K}$ (the so-called kernel variables in [56]), we obtain

$$\mathrm{Jac}_{vec_{col}(\boldsymbol{K})}(vec_{col}(\sum_{i=1}^{km} x_i\boldsymbol{M}''_i\boldsymbol{K})) = \sum_{i=1}^{km} x_i(\boldsymbol{I}_{n-r}\otimes\boldsymbol{M}''_i) = \boldsymbol{I}_{n-r}\otimes\left(\sum_{i=1}^{km} x_i\boldsymbol{M}''_i\right).$$

The kernel of $\mathrm{Jac}_{vec_{col}(\boldsymbol{K})}$ is generated by the vectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n-r})$ with $\boldsymbol{v}_l$ in the left kernel of $\boldsymbol{M} = \sum_{i=1}^{km} x_i \boldsymbol{M}''_i$, that should be generated by $\binom{m}{r+1}$ vectors of minors, according to Theorem 1. Hence the kernel of $\mathrm{Jac}_{vec_{col}(\boldsymbol{K})}$ is generated by $\binom{m}{r+1}(n-r)$ vectors. It is here that we see the point of having this tensor product form. These kernel vectors have entries that are polynomials of degree r by using Theorem 1. This gives degree falls at degree $r+2$ and yields partial syzygies that have degree $r+1$. These considerations are a slightly different way of understanding the results given in [56, §3]. The syndrome modelling displays a similar behavior, i.e. a degree fall at $r+2$ for the very same reason as can be readily verified. Let us apply now Lemma 1 to the Ourivski-Johansson modelling.

Application to the Ourivski-Johansson modelling. The system here is

$$\mathcal{F} = \left\{ \left(1 \ \alpha \ \cdots \ \alpha^{m-1}\right) \left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0} & \boldsymbol{S}' \end{array}\right) \left(\boldsymbol{C}_2 - \left(\begin{array}{c|c} 1 & \\ \boldsymbol{0} & \boldsymbol{C}'_1 \end{array}\right) \boldsymbol{R}\right) \right\}, \tag{8}$$

where $\boldsymbol{S}'$ is the $(m-r) \times (r-1)$ matrix $\boldsymbol{S}_{\{r+1..m\},\{2..r\}}$ and $\boldsymbol{C}'_1$ is the $r \times k$ matrix $\boldsymbol{C}_{*,\{2..k+1\}}$. We add to $\mathcal{F}$ the field equations $\mathcal{F}_q = \{s_{i,j}^q - s_{i,j}, r+1 \leq i \leq m, 2 \leq j \leq r, c_{i,j}^q - c_{i,j}, 1 \leq i \leq r, 2 \leq j \leq n\}$.

With high probability, this system has a unique solution. As we used the field equations, the ideal $\langle \mathcal{F}, \mathcal{F}_q \rangle$ is radical. The system has $n_{\boldsymbol{S}} = (m-r)(r-1)$ variables $\boldsymbol{S}$, $n_{\boldsymbol{C}} = (n-1)r$ variables $\boldsymbol{C}$, and $n-k-1$ equations over $\mathbb{F}_{q^m}$, hence $n_{eq} = (n-k-1)m$ equations over $\mathbb{F}_q$, plus the field equations.

Consider the system $\mathcal{F}^h$ formed by the bilinear parts of the equations in $\mathcal{F}$. A simple computation shows that

$$\mathcal{F}^h = \left\{ \alpha^r \left(1 \ \alpha \ \cdots \ \alpha^{m-r-1}\right) \boldsymbol{S}' (\boldsymbol{C}''_2 - \boldsymbol{C}''_1 \boldsymbol{R}') \right\},$$

where $\boldsymbol{C}''_2 = \boldsymbol{C}_{\{2..r\},\{k+2..n\}}$, $\boldsymbol{C}''_1 = \boldsymbol{C}_{\{2..r\},\{2..k+1\}}$ and $\boldsymbol{R}' = \boldsymbol{R}_{\{2..k+1\},*}$.

If we take the equations and variables in row order, and use Lemma 1, then

$$\mathrm{Jac}_{vec_{row}(\boldsymbol{S})}(vec_{row}(\mathcal{F}^h)) = \alpha^r \left(1 \ \alpha \ \cdots \ \alpha^{m-r-1}\right) \otimes \left(\boldsymbol{C}''_2 - \boldsymbol{C}''_1 \boldsymbol{R}'\right)^{\mathsf{T}} \tag{9}$$

The elements in the left kernel of $Jac_{vec_{row}(\boldsymbol{S})}(vec_{row}(\mathcal{F}^h))$ are those in the right kernel of $\boldsymbol{C}''_2 - \boldsymbol{C}''_1 \boldsymbol{R}'$, and applying Theorem 1, they belong to the vector space generated by the vectors $\boldsymbol{V}_J$ for any $J = \{j_1 < j_2 < \cdots < j_r\} \subset \{1, \ldots, n-k-1\}$ of size r defined by

$$\boldsymbol{V}_J = (\ldots, \underbrace{0}_{j \notin J}, \ldots, \underbrace{(-1)^{l+1} \det(\boldsymbol{C}''_2 - \boldsymbol{C}''_1 \boldsymbol{R}'_{*,J \setminus \{j\}})}_{j = j_l \in J}, \ldots)_{1 \leq j \leq n-k-1}.$$

Each V_J gives a syzygy for $\mathcal{F}^h$ and when applying it to $\mathcal{F}$ it yields a degree fall in degree $r+1$ because the entries of $\boldsymbol{V}_J$ are homogeneous polynomials of degree $r-1$. The inner product of $\boldsymbol{V}_J$ with the vector of the equations gives an equation of degree $\leq r$ since the homogeneous part of highest degree cancels, as has been observed at the beginning of this section. Now the affine part of the equations $\mathcal{F}$ is $\left(1 \ \alpha \ \cdots \ \alpha^{r-1}\right) (\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{R})$.

Writting $\tilde{\boldsymbol{H}} = \left(-\boldsymbol{R}^{\mathsf{T}}\ \boldsymbol{I}_{n-k-1}\right)$, then

$$\det(\boldsymbol{C}_2'' - \boldsymbol{C}_1''\boldsymbol{R}'_{*,J\setminus\{j\}}) = \det((\boldsymbol{C}\tilde{\boldsymbol{H}}^{\mathsf{T}})_{\{2..r\},J\setminus\{j\}}).$$

Using the reverse of Laplace's formula expressing a determinant in terms of minors, we can compute the inner product of the vector $\boldsymbol{V}_J$ with the ith row of $\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R} = \boldsymbol{C}\tilde{\boldsymbol{H}}^{\mathsf{T}}$, that is 0 for $2 \leq i$ and $\det((\boldsymbol{C}\tilde{\boldsymbol{H}}^{\mathsf{T}})_{*,J})$ for $i = 1$.

The product gives

$$\begin{aligned}\boldsymbol{V}_J\left(\left(1\ \alpha\ \cdots\ \alpha^{r-1}\right)(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R})\right)^{\mathsf{T}} &= \boldsymbol{V}_J\,(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R})^{\mathsf{T}}\left(1\ \alpha\ \cdots\ \alpha^{r-1}\right)^{\mathsf{T}} \\ &= \det(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R})_{*,J}. \end{aligned} \tag{10}$$

This yields a corresponding equation that will be reduced to zero by a degree-$(r+1)$ Gröbner basis of $\mathcal{F}$. Hence the partial syzygies of degree r coming from the degree fall in the $(r+1)$-Macaulay matrix are exactly the maximal minors of $\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}$. We have thus proven that

Theorem 2. *The equations* $\mathrm{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}) = 0$*, that are the maximal minors of the matrix* $\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}$*, belong to the ideal* $\langle \mathcal{F}, \mathcal{F}_q\rangle$*. Moreover, they are reduced to zero by a degree* $(r+1)$*-Gröbner basis of* $\{\mathcal{F}, \mathcal{F}_q\}$*.*

Remark 1. If we are only interested in the first part of the theorem about the maximal minors, then there is a simple and direct proof which is another illustration of the role of the matrix form of the system. Indeed, let $(\boldsymbol{S}^*, \boldsymbol{C}^*)$ be a solution of $\{\mathcal{F}, \mathcal{F}_q\}$, then the nonzero vector $\left(1\ S_2^*\ \cdots S_m^*\right) = \left(1\ \alpha\ \cdots\ \alpha^{m-1}\right)\boldsymbol{S}^*$ belongs to the left kernel of the matrix $\boldsymbol{C}_2^* - \boldsymbol{C}_1^*\boldsymbol{R}$. Hence this matrix has rank less than r, and the equations $\mathrm{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}) = 0$ are satisfied for any solution of the system $\{\mathcal{F}, \mathcal{F}_q\}$, which means that the equations belong to the ideal $\langle \mathcal{F}, \mathcal{F}_q\rangle$ as this ideal is radical.

5.2 Analysis of the ideal $\mathrm{MaxMinors}(\boldsymbol{C_2 - C_1 R})$

The previous theorem allows us to obtain directly degree r equations without having to compute first the Macaulay matrix of degree $r+1$. This is a significant saving when performing the Gröbner basis computation. A nice feature of these equations is that they only involve one part of the unknowns, namely the coefficient variables.

Moreover all these equations can be expressed by using a limited number of polynomials as we now show. Some of them will be of degree r, some of them will be of degree $r-1$. If we perform Gaussian elimination on these equations by treating these polynomials as variables and trying to eliminate the ones corresponding to the polynomials of degree r first, then if the number of equations we had was greater than the number of polynomials of degree r, we expect to find equations of degree $r-1$. Roughly speaking, when this phenomenon happens we just have to add all the equations of degree $r-1$ we obtain in this way to the Ourivski-Johansson modelling and the Gröbner basis calculation will not go beyond degree r.

Let us analyse precisely the behavior we just sketched. The shape of the equations $\mathrm{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}) = 0$ is given by the following proposition, where by convention $\det(\boldsymbol{M}_{\emptyset,\emptyset}) = 1$ and the columns of $\boldsymbol{R}$ are indexed by $\{k+2..n\}$:

Proposition 1. $\mathrm{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R})$ *is a set of* $\binom{n-k-1}{r}$ *polynomials* P_J*, indexed by* $J \subset \{k+2..n\}$ *of size* r*:*

$$P_J = \sum_{\substack{T_1 \subset \{1..k+1\}, T_2 \subset J \\ \text{such that } T = T_1 \cup T_2 \text{ has size } \#T = r}} (-1)^{\sigma_J(T_2)} \det(\boldsymbol{R}_{T_1, J\setminus T_2}) \det(\boldsymbol{C}_{*,T}).$$

where $\sigma_J(T_2)$ *is an integer depending on* T_2 *and* J*.*

If $1 \notin T$*, the polynomial* $\det(\boldsymbol{C}_{*,T})$ *is homogeneous of degree* r *and contains* $r!$ *monomials; if* $1 \in T$*,* $\det(\boldsymbol{C}_{*,T})$ *is homogeneous of degree* $r-1$ *and contains* $(r-1)!$ *monomials.*

Proof. The matrix $\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}$ has size $r \times (n-k-1)$, hence there are $\binom{n-k-1}{r}$ different minors $P_J = \det(\boldsymbol{C}(\begin{smallmatrix} -\boldsymbol{R} \\ \boldsymbol{I}_{n-k-1} \end{smallmatrix})_{*,J})$. To compute them, we use the Cauchy-Binet formula for the determinant of a product of non-square matrices:

$$\det(\boldsymbol{A}\boldsymbol{B}) = \sum_{T \subset \{1..p\}, \#T = r} \det(\boldsymbol{A}_{*,T}) \det(\boldsymbol{B}_{T,*})$$

where $\boldsymbol{A} \in \mathbb{K}^{r \times p}$, $\boldsymbol{B} \in \mathbb{K}^{p \times r}$, and $p \geq r$. We apply this formula to P_J, and use the fact that, for $T = T_1 \cup T_2$ with $T_1 \subset \{1..k+1\}$ and $T_2 \subset \{k+2..n\}$,

$$\begin{aligned} \det\left(\begin{pmatrix} -\boldsymbol{R} \\ \boldsymbol{I}_{n-k-1} \end{pmatrix}_{T_1 \cup T_2, J}\right) &= 0 \text{ if } T_2 \not\subset J \\ &= (-1)^{\sigma_J(T_2)} \det(\boldsymbol{R}_{T_1, J\setminus T_2}) \text{ if } T_2 \subset J, \end{aligned}$$

using the Laplace expansion of this determinant along the last rows, with $\sigma_J(T_2) = d(k+r) + (d-1)d/2 + \sum_{t \in T_2} Pos(t, J)$ where $Pos(t, J)$ is the position of t in J, and $d = \#J - \#T_2$. □

Each polynomial P_J can be expanded into m equations over $\mathbb{F}_q$, the polynomial $P_J[i]$ being the coefficient of P_J in α^{i-1}. When computing a grevlex Gröbner basis of the system of the $P_J[i]$'s over $\mathbb{F}_q$, with an algorithm like F4 using linear algebra, the first step consists in computing a basis of the $P_J[i]$'s over $\mathbb{F}_q$.

It appears that there may be a fall of degree in this first step, in degree r, that produces equations of degree $r-1$. The following heuristic explains when this fall of degree occurs.

Heuristic 1 – Overdetermined case: *when* $m\binom{n-k-1}{r} \geq \binom{n}{r} - 1$*, generically, a degree-*r *Gröbner basis of the projected system* $\mathrm{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R}) = 0$ *of* $m\binom{n-k-1}{r}$ *equations over* $\mathbb{F}_q$ *contains* $\binom{n-1}{r-1} - 1$ *equations of degree* $r-1$*, that are obtained by linear combinations of the initial equations.*

– Intermediate case: *when* $\binom{n}{r}-1 > m\binom{n-k-1}{r} > \binom{n-1}{r}$*, generically a degree-*r *Gröbner basis of the projected system* $MaxMinors(\boldsymbol{C}_2-\boldsymbol{C}_1\boldsymbol{R})=0$ *contains* $m\binom{n-k-1}{r}-\binom{n-1}{r}$ *equations of degree* $r-1$*, that are obtained by linear combinations of the initial equations.*
– Underdetermined case: *When* $m\binom{n-k-1}{r} \leq \binom{n-1}{r}$*, then generically a degree-*r *Gröbner basis of the system contains* $m\binom{n-k-1}{r}$ *polynomials that are all of degree* r*.*

Remark 2. Here overdetermined/underdetermined refers to the system of maximal minors given by the set of equations $\text{MaxMinors}(\boldsymbol{C}_2-\boldsymbol{C}_1\boldsymbol{R})=0$

Remark 3. The degree-r Gröbner bases also contain polynomials of degree r in the overdetermined and intermediate cases, but we will not compute them, as experimentally they bring no speed-up to the computation, see Section 6.1.

Proposition 2. *Computing the polynomials in a degree-*r *Gröbner basis of the projected equations* MaxMinors *amounts to solving a linear system with* $\nu = m\binom{n-k-1}{r}$ *equations in* $\mu = \binom{n}{r}$ *variables, which costs* $O(\min(\mu,\nu)^{\omega-2}\mu\nu)$ *operations in the base field, where* ω *is the exponent of matrix multiplication (see Section 6.2).*

Proof. It is possible to view the system $\text{MaxMinors}(\boldsymbol{C}_2-\boldsymbol{C}_1\boldsymbol{R})$ projected over $\mathbb{F}_q$ as a linear system of $\mu = m\binom{n-k-1}{r}$ equations, whose variables are the $\nu = \binom{n}{r}$ unknowns $x_T = \det(\boldsymbol{C}_{*,T})$ for all $T \subset \{1..n\}$ of size r. The matrix associated to this linear system is a matrix $\boldsymbol{M}$ of size $\mu \times \nu$ whose coefficient in row (i,J) : $i \in \{1..m\}, J \subset \{k+2..n\}, \#J = r$, and column x_T is, with $T_2 = T \cap \{k+2..n\}$:

$$\boldsymbol{M}[(i,J),x_T] = \begin{cases} [\alpha^{i-1}](-1)^{\sigma_J(T_2)}\det(\boldsymbol{R}_{T\cap\{1..k+1\},J\setminus T_2}) & \text{if } T_2 \subset J, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

where $[\alpha^{i-1}]\beta$ is the ist component of $\beta \in \mathbb{F}_{q^m}$ viewed in the vector space $\mathbb{F}_q^m$ with generator basis $\begin{pmatrix}1 & \alpha & \dots & \alpha^{m-1}\end{pmatrix}$.

A basis of the vector space generated by the equations $\text{MaxMinors}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{R})=0$ is given by $\tilde{\boldsymbol{M}}\cdot\boldsymbol{T}$ where $\tilde{\boldsymbol{M}}$ is the row echelon form of $\boldsymbol{M}$ and $\boldsymbol{T}$ is the column vector formed by the polynomials $\det(\boldsymbol{C}_{*,T}) : \#T = r$. As we are searching for equations of degree $r-1$, we order the variables x_T such that the ones with $1 \in T$ that correspond to polynomials $\det(\boldsymbol{C}_{*,T})$ of degree $r-1$ are the rightmost entries of the matrix. □

Heuristic 1 can be stated in terms of the matrix $\boldsymbol{M}$. In the overdetermined case, that is when $m\binom{n-k-1}{r} \geq \binom{n}{r}-1$, we expect matrix $\boldsymbol{M}$ to have rank $\binom{n}{r}-1$ with high probability. This rank can not be larger, as the (left) kernel space of the matrix has dimension 1 (this comes from the fact that the equations are homogeneous). Hence, $\tilde{\boldsymbol{M}}\cdot\boldsymbol{T}$ produces $\binom{n-1}{r}$ equations of degree r, and $\binom{n-1}{r-1}-1$ equations of degree $r-1$, that have all the shape $\det(\boldsymbol{C}_{*,T})$ or $\det(\boldsymbol{C}_{*,T}) - \det(\boldsymbol{C}_{*,T_0})$ where T_0 corresponds to the free variable x_{T_0} of the linear system, $1 \in T_0$. In the intermediate and underdetermined cases, we also expect matrix $\boldsymbol{M}$ to be full rank in general, and to be also full rank on the columns corresponding to the $\boldsymbol{c}_T$'s of degree r.

6 Experimental results, complexity bounds, and security

6.1 Experimental results

We did various computations for different values of the parameters (m, n, k, r). We got our best complexity results by doing the following steps:

1. compute the set of equations $\mathcal{F}$ which comes from $\left(1\ \alpha\ \cdots\ \alpha^{m-1}\right) \boldsymbol{S} \left(\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{R}\right)$ specialised as in (6),
2. compute the system MaxMinors$(\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{R})$,
3. compute the matrix $\boldsymbol{M}$ from (11) and its echelon form $\tilde{\boldsymbol{M}}$, let $\mathcal{J}$ be the set of the resulting equations of degree $r-1$ in the $\boldsymbol{C}$ variables,
4. if $\mathcal{J}$ is empty, then let $\mathcal{J}$ be the set of equations coming from $\tilde{\boldsymbol{M}}$ of degree r in the $\boldsymbol{C}$ variables,
5. compute $\boldsymbol{G}$ a reduced degree-d Gröbner basis of the system $\{\mathcal{F}, \mathcal{J}, \mathcal{F}_q\}$, where
$$d = \begin{cases} r & \text{in the overdetermined case,} \\ r \text{ or } r+1 & \text{in the intermediate case,} \\ r+2 & \text{in the underdetermined case.} \end{cases}$$

The computations are done using `magma v2.22-2` on a machine with an Intel® Xeon® 2.00GHz processor. Here are the notation used in all tables:

- $n_S = (r-1)(m-r)$: the number of variables in $\boldsymbol{S}$
- $n_C = r(n-1)$: the number of variables in $\boldsymbol{C}$
- $n_{eq} = m(n-k-1)$: the number of equations in $\mathcal{F}$
- $d : n_{syz}$: the number of equations in $\mathcal{J}$, where d denotes the degree of the equations and n_{syz} the number of them:
  - $r-1 : \binom{n-1}{r-1} - 1$ in the overdetermined case
  - $r-1 : m\binom{n-k-1}{r} - \binom{n-1}{r}$ in the intermediate case
  - $r : m\binom{n-k-1}{r}$ in the underdetermined case
- $T_{syz.}$: time of computing the n_{syz} equations of degree $r-1$ or r in $\mathcal{J}$
- T_{Gbsyz}: time of the Gröbner basis computation of $\{\mathcal{J}, \mathcal{F}_q\}$
- T_{Gb}: time of the Gröbner basis computation of $\{\mathcal{F}, \mathcal{J}, \mathcal{F}_q\}$
- d_{ff}: the degree where we observe the first fall of degree
- d_{max}: the maximal degree where some new polynomial is produced by the F4 algorithm
- "Max Matrix size": the size of the largest matrix reduced during the F4 computation, given by `magma`. We did not take into account the useless steps (the matrices giving no new polynomials)

Table 1 page 20 gives our timings on the parameters proposed in [55]. For each set of parameters, the first row of the table gives the timing for the direct computation of a Gröbner basis of $\{\mathcal{F}, \mathcal{F}_q\}$ whereas the second row gives the timings for the Gröbner basis of $\{\mathcal{F}, \mathcal{J}, \mathcal{F}_q\}$. We can see that, apart from very small parameters, the computation of the equations MaxMinors$(\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{R})$ is negligible compared to the time of the Gröbner basis computation.

Among the proposed parameters, only the $(15, 15, 8, 3)$ was in the case where the system MaxMinors is underdetermined. In that case, the most consuming part of the computation is the Gröbner basis of the system MaxMinors, that depends only on the $\boldsymbol{C}$ variables. Once this computation is done, the remaining Gröbner basis of $\{\mathcal{F}, \mathcal{J}, \mathcal{F}_q\}$ has a negligible cost.

Table 2 page 21 gives timing for different values of k and r, with $m = 14$ and $n = 18$ fixed. For $r = 2$, the values $k \in \{1..11\}$ correspond to the overdetermined case, the value $k = 12$ to the intermediate one, and $k = 13$ to the underdetermined case. The values $k \in \{1..11\}$ behave all like $k = 11$. As for the parameters from [55], the hardest cases are the ones when the system MaxMinors is underdetermined, where the maximal degree reached during the computation is $r+2$. For the overdetermined cases, the maximal degree is r, and for the intermediate cases, it may be r or $r + 1$.

For $r = 3$, the overdetermined cases are $k \in \{1..8\}$, $k = 9$ is intermediate and $k \in \{10..11\}$ are underdetermined. Values of $k \geq 12$ do not allow a unique decoding for $r = 3$, the Gilbert-Varshamov bound being 2 for those values.

For $r = 4$ the tradeoffs are $1 \leq k \leq 6$, $k = 7$ and $8 \leq k \leq 9$ for the three cases, and for $r = 5$, $1 \leq k \leq 5$, $k = 6$ and $7 \leq k \leq 8$. We could not perform the computations for the intermediate and underdetermined cases, due to a lack of memory. We also observe that the first fall of degree (d_{ff}) does not always predict the complexity of the computation.

Table 3 page 21 gives the timings for a fixed $r = 3$, a ratio $n = 2k$ and various values of k. Again, we can observe that for defavorable cases ($k = 6, 7$) the maximal degree is $r + 2$ or $r + 1$ rather than r, making the computation harder for small values of k than for larger.

Table 1. We compare the behavior of the Gröbner basis computation for the parameters considered in [48], with and without adding to the system the equations $\mathcal{J}$.

m	n	k	r	n_S	n_C	n_{eq}	n_{syz}	T_{syz}	T_{Gbsyz}	T_{Gb}	d_{ff}	d_{max}	Max Mat Size
25	30	15	2	23	58	350				0.4 s	3	3	18550 ×19338
							1:28	0.4 s		0.02 s	2	2	1075 × 749
30	30	16	2	28	58	390				0.5 s	3	3	22620 × 25288
							1:28	0.4 s		0.02 s	2	2	1260 × 899
30	50	20	2	28	98	870				2.2 s	3	3	67860 × 57898
							1:48	3.8 s		0.07 s	2	2	2324 × 1499
50	50	26	2	48	98	1150				7.4 s	3	3	112700 × 120148
							1:48	3.5 s		0.2 s	2	2	3589×2499
15	15	7	3	24	42	105				60.1 s	4	4	77439 × 153532
							2:90	0.2 s		0.06 s	3	3	8860 × 13658
15	15	8	3	24	42	90			–		4	≥5	–
							3:300	0.3 s	162 s	0.2 s	4	5	191515 × 457141
20	20	10	3	34	57	180				450 s	4	4	233672 × 543755
							2:170	1.0 s		0.2 s	3	3	22124 × 35087

Table 2. $m = 14$ and $n = 18$.

k	r	n_{syz}	n_S	n_C	n_{eq}	$T_{Syz.}$	T_{Gbsyz}	T_{Gb}	d_{ff}	d_{max}	Max Matrix size	Mem.
11	2	1:16	12	34	84	< 0.1s		< 0.1s	2	2	322×251	34 Mo
12	2	1:4	12	34	70	< 0.1s		< 0.1s	3	3	1820×2496	34 Mo
13	2	2:84	12	34	56	< 0.1s	32 s	0 s	3	4	231187×141064	621 Mo
8	3	2:135	22	51	126	0.6 s		0.1 s	3	3	13179×18604	34 Mo
9	3	2:104	22	51	112	0.5 s		0.7 s	3	3	10907×18743	67 Mo
4	4	3:679	30	68	182	12.1 s		53.7 s	2	4	314350×650610	1.3 Go
5	4	3:679	30	68	168	9.4 s		59.3 s	4	4	314350×650610	2.0 Go
6	4	3:679	30	68	154	7.1 s		69.4 s	4	4	281911×679173	3.6 Go
2	5	4:2379	36	85	210	138.8 s		27.5 s	2	4	416433×669713	1.1 Go
5	5	4:2379	36	85	196	44.8 s		5h08	2	5	7642564×30467163	253.6 Go

Table 3. The parameters are $r = 3$, $m = n$, $k = \frac{n}{2}$.

k	n_{syz}	n_S	n_C	n_{eq}	T_{syz}	T_{Gbsyz}	T_{Gb}	d_{max}	Memory
6	3:120	18	33	60	0.2s	117 s	0.02s	5	4.9 Go
7	3:280	22	39	84	0.1s	9.7 s	0.1s	4	0.3 Go
8	2:104	26	45	112	0.2s		0.1s	3	.04 Go
17	2:527	62	99	544	34.3s		4.7s	3	0.3 Go
27	2:1377	102	159	1404	650.2s		161.3s	3	2.7 Go
37	2:2627	142	219	2664	5603.6s		3709.4s	3	15.0 Go
47	2:4277	182	279	4324	26503.9s		26022.6s	3	83.0 Go

6.2 Complexity analysis and security over $\mathbb{F}_2$

Now, we give an upper bound on the complexity of our algebraic approach to solve the (m, n, k, r)-decoding problem using the modelling of Section 3.3. The complexity is estimated in terms of the number of operations in $\mathbb{F}_2$ that the algorithm uses. This allows us to update the number of bits of security for several cryptosystems, as showed in Table 4: Loidreau's one [49], ROLLO [7], and RQC [3]. Note that the restriction to $\mathbb{F}_2$ is only there because we want to derive security values. If one works over a larger field $\mathbb{F}_q$, a similar analysis can be derived. The only change in this case is to consider the relevant number of monomials. Note also that even if Algorithm 1 works over any field, its success probability given in Proposition 3 depends on q.

Remark that, in Table 4, for the sets of parameters which do not satisfy Eq. (1), which correspond to underdetermined instances, we assume that the system can be solved at $d = r + 1$. It is a conservative choice: in the experiments of Section 6.1, the maximal degree is often r for the underdetermined cases.

The complexity bound follows from the fact that the Gröbner basis algorithm works with Macaulay matrices of degree δ for increasing values of δ up to d, the degree for which the Gröbner basis is found (see Section 4 for a more detailed description). At each of these steps, the algorithm performs a Gaussian elimination algorithm on a Macaulay matrix which has at most $\binom{(m-r)(r-1)+(n-1)r}{\delta}$

Cryptosystem	**Parameters** (m, n, k, r)	$d = r$	$d = r + 1$	**Previous**
Loidreau	$(128, 120, 80, 4)$	**96.3**	117.1	256
ROLLO-I-128	$(79, 94, 47, 5)$	**114.9**	134.5	128
ROLLO-I-192	$(89, 106, 53, 6)$	**142.2**	162.5	192
ROLLO-I-256	$(113, 134, 67, 7)$	174.0	**195.3**	256
ROLLO-II-128	$(83, 298, 149, 5)$	**132.3**	155.4	128
ROLLO-II-192	$(107, 302, 151, 6)$	**161.5**	185.0	192
ROLLO-II-256	$(127, 314, 157, 7)$	191.6	**215.4**	256
ROLLO-III-128	$(101, 94, 47, 5)$	**117.1**	137.2	128
ROLLO-III-192	$(107, 118, 59, 6)$	**145.7**	166.6	192
ROLLO-III-256	$(131, 134, 67, 7)$	175.9	**197.5**	256
RQC-I	$(97, 134, 67, 5)$	**121.1**	142.0	128
RQC-II	$(107, 202, 101, 6)$	**154.2**	176.5	192
RQC-III	$(137, 262, 131, 7)$	188.4	**211.9**	256

Table 4. Security in bits for several cryptosystems with respect to our attack, computed using Eq. (12) with $\omega = 2.807$, $d = r$ or $d = r+1$. The values in bold correspond the most likely maximal degree, i.e. r if Eq. (1) holds and $r + 1$ otherwise. The last column gives the previous best known security values, based on the attack in [10].

columns and fewer rows than columns. The number of columns is the number of squarefree monomials of degree δ in $(m - r)(r - 1) + (n - 1)r$ variables.

In general, Gaussian elimination of a $\mu \times \nu$ matrix of rank ρ over a field has a complexity of $O(\rho^{\omega-2}\mu\nu) \subseteq O(\max(\mu, \nu)^\omega)$ operations in that field [19,54]. Here, ω is the exponent of matrix multiplication, with naive bounds $2 \leq \omega \leq 3$. The best currently known value for ω is $\omega \approx 2.37$ [47], by an improvement of Coppersmith-Winograd's algorithm. In terms of practical performances, the best known method is based on Strassen's algorithm, which allows one to take $\omega \approx 2.807$, and when the base field is a finite field, this exponent is indeed observed in practice for matrices with more than a few hundreds rows and columns.

The Macaulay matrices encountered in the Gröbner basis computations we consider are usually very sparse and exhibit some structure. Some Gaussian elimination algorithms have been designed specifically for matrices over $\mathbb{F}_2$ [4], also for sparse matrices [16], and even to take advantage of the specific structure of Macaulay matrices (see [17]; we expect `Magma`'s closed-source implementation of F_4 to use similar techniques). However, none of these optimized algorithms has been proven to reach a complexity which is asymptotically better than the one mentioned above, apart from speed-ups by constant factors.

As a result, we bound the complexity of the step of degree δ in the Gröbner basis computation by that of performing Gaussian elimination on a $\mu \times \nu$ matrix over $\mathbb{F}_2$, with $\mu \leq \nu = \binom{(m-r)(r-1)+(n-1)r}{\delta}$; the overall computation then costs

$$\mathcal{O}\left(\left(\sum_{\delta=0}^{d} \binom{(m-r)(r-1)+(n-1)r}{\delta}\right)^{\omega}\right) \tag{12}$$

operations in $\mathbb{F}_2$. Let us now focus on the case $m = n = 2k$ and $r \approx \sqrt{n}$. Then the complexity of our approach is as in Eq. (12) with $d = r$. Using a similar analysis, the approach based on Kipnis-Shamir's modelling has a complexity of

$$\mathcal{O}\left(\left(\sum_{\delta=0}^{r+2}\binom{km + r(n-r)}{\delta}\right)^{\omega}\right)$$

operations. Asymptotically, the dominant term in the former bound is of the order of $2^{\frac{3}{2}\omega r \log_2(n)}$, to be compared to $2^{2\omega r \log_2(n)}$ in the Kipnis-Shamir bound. Also, the aforementioned combinatorial attacks ([10]) would have a complexity of the order of $2^{\frac{1}{2}rn}$ when $m = n = 2k$.

Finally, note that the complexity bound stated above was derived under assumptions: in Section 3.3, we presented the modelling along with some assumptions which allowed us to specialize variables a priori and still ensure that the algorithm of Section 5 yields the solution $\lambda\boldsymbol{e}$. In general, the assumption might not hold, that is, the specific specialization made in Section 3.3 could be wrong. We use Algorithm 1 in order to specialize more variables: it first uses the specialization detailed in Section 3.3, and if that one fails, follows on with other similar specializations. This algorithm uses the subroutine $\texttt{Solve}(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{R})$, which augments the system as explained in Section 5 and returns a solution to Eq. (5) if one is found and $\emptyset$ otherwise.

Input: Matrix $\boldsymbol{R}$
Output: A solution to the system in Eq. (5) or $\emptyset$
$\boldsymbol{S} = m \times r$ matrix of variables
$\boldsymbol{C} = r \times n$ matrix of variables
Set the first column and the first row of $\boldsymbol{S}$ to $[1\ \ 0\ \ \cdots\ \ 0]$
Set a randomly selected column of $\boldsymbol{C}$ to $[1\ \ 0\ \ \cdots\ \ 0]^{\mathsf{T}}$
Choose at random $\lfloor \frac{m-1}{r-1} \rfloor$ disjoint subsets $T_i \subseteq \{2, \ldots, m\}$ of cardinality $r-1$
for $i \leftarrow 1$ **to** $\lfloor \frac{m-1}{r-1} \rfloor$ **do**
  Set the $(r-1) \times (r-1)$ submatrix $\boldsymbol{S}_{T_i, \{2,\ldots,r\}}$ to $\boldsymbol{I}_{r-1}$
  sol = $\texttt{Solve}(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{R})$
  if sol $\neq \emptyset$ **then return** sol
return $\emptyset$

Algorithm 1: (m, n, k, r)-Decoding

For positive integers a and b with $a \leq b$, we denote by $p_{q,a,b} := \prod_{i=0}^{a-1}\left(1 - q^{i-b}\right)$ the probability that a uniformly random matrix in $\mathbb{F}_q^{a \times b}$ has full rank.

Proposition 3. *Fix integers m, n, k, r, and let $c \in \{1, \ldots, \lfloor \frac{m-1}{r-1} \rfloor\}$. Suppose that a (m, n, k, r)-rank decoding instance is chosen uniformly at random, and that the input matrix $\boldsymbol{R}$ is built from this instance. Then, the probability that Algorithm 1 makes at most c calls to $\texttt{Solve}(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{R})$ before finding a solution is greater than*

$$\frac{1 - q^{-r}}{1 - q^{-n}}\left(1 - \frac{\left(1 - p_{q,r-1,r-1}\right)^c}{p_{q,r-1,m-1}}\right).$$

The proof is differed to Appendix.

If one applies this proposition to the cryptosystems mentioned in Table 4, with at most 5 calls to $\texttt{Solve}(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{R})$, Algorithm 1 will return a solution with a probability always greater than 0.8; note that for these instances the quantity $\lfloor \frac{m-1}{r-1} \rfloor$ is greater than 15, and around 20 for most of them.

In the event where Algorithm 1 returns $\emptyset$ after $\lfloor \frac{m-1}{r-1} \rfloor$ calls to $\texttt{Solve}(\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{R})$, one can run it again until a solution is found. The probabilities mentioned in the previous paragraph show that for parameters of interest a second run of the algorithm is very rarely needed.

7 Conclusion

In this paper we introduce a new approach for solving the Rank Metric Decoding problem with Gröbner basis techniques. Our approach is based on adding partial syzygies to a newer version of a modelling due to Ourivski and Johansson.

Overall our analysis shows that our attack, for which we give a general estimation, clearly outperforms all previous attacks in rank metric for a classical (non quantum) attacker. In particular we obtain an attack below the claimed security level for all rank-based schemes proposed to the NIST Post-Quantum Cryptography Standardization Process. Note that there has been some very recent progress [13] on the modelling and the attack proposed here. This results in even less complex attacks and in the removal of the Gröbner basis computation step: it is replaced by solving a linear system. Although our attack and its recent improvement really improve on previous attacks for rank metric, they meanwhile suffer from two limitations.

First these attacks do not benefit from a direct Grover quantum speed-up, unlike combinatorial attacks. For the NIST parameters (with the exception of Rollo-I-192 for the latest attack [13]) the best quantum attacks still remain quantum attacks based on combinatorial attacks, because of the Grover speed-up. Second, these attacks need an important amount of memory for large parameters.

Acknowledgements

This work has been supported by the French ANR projects CBCRYPT (ANR-17-CE39-0007) and the MOUSTIC project with the support from the European Regional Development Fund (ERDF) and the Regional Council of Normandie. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, as well as Ray Perlner and Daniel Smith for useful discussions.

References

1. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Zémor, G.: Ouroboros-R. First round submission

to the NIST post-quantum cryptography call (Nov 2017), https://pqc-ouroborosr.org
2. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Zémor, G.: Rank quasi cyclic (RQC). First round submission to the NIST post-quantum cryptography call (Nov 2017), https://pqc-rqc.org
3. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Zémor, G., Couvreur, A., Hauteville, A.: Rank quasi cyclic (RQC). Second round submission to the NIST post-quantum cryptography call (Apr 2019), https://pqc-rqc.org
4. Albrecht, M., Bard, G.: The M4RI Library – Version 20140914. The M4RI Team (2014), http://m4ri.sagemath.org
5. Aragon, N., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.P., Zémor, G.: LAKE – Low rAnk parity check codes Key Exchange. First round submission to the NIST post-quantum cryptography call (Nov 2017), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LAKE.zip
6. Aragon, N., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.P., Zémor, G.: LOCKER – LOw rank parity ChecK codes EncRyption. First round submission to the NIST post-quantum cryptography call (Nov 2017), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LOCKER.zip
7. Aragon, N., Blazy, O., Deneuville, J.C., Gaborit, P., Hauteville, A., Ruatta, O., Tillich, J.P., Zémor, G., Aguilar Melchor, C., Bettaieb, S., Bidoux, L., Magali, B., Otmani, A.: ROLLO (merger of Rank-Ouroboros, LAKE and LOCKER). Second round submission to the NIST post-quantum cryptography call (Mar 2019), https://pqc-rollo.org
8. Aragon, N., Blazy, O., Gaborit, P., Hauteville, A., Zémor, G.: Durandal: a rank metric based signature scheme. In: Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III. LNCS, vol. 11478, pp. 728–758. Springer (2019). https://doi.org/10.1007/978-3-030-17659-4_25, https://doi.org/10.1007/978-3-030-17659-4_25
9. Aragon, N., Gaborit, P., Hauteville, A., Ruatta, O., Zémor, G.: Ranksign – a signature proposal for the NIST's call. First round submission to the NIST post-quantum cryptography call (Nov 2017), https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/RankSign.zip
10. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.P.: A new algorithm for solving the rank syndrome decoding problem. In: 2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018. pp. 2421–2425. IEEE (2018). https://doi.org/10.1109/ISIT.2018.8437464
11. Ars, G., Faugère, J.C., Imai, H., Kawazoe, M., Sugita, M.: Comparison between XL and Gröbner basis algorithms. In: ASIACRYPT (2004)
12. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. Ph.D. thesis, Université Paris VI (Dec 2004), http://tel.archives-ouvertes.fr/tel-00449609/en/
13. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Algebraic attacks for solving the Rank Decoding and MinRank problems without Gröbner basis. arXiv e-prints arXiv:2002.08322 (Feb 2020)
14. Bardet, M., Faugère, J.C., Salvy, B.: On the complexity of the F_5 Gröbner basis algorithm. J. Symbolic Comput. **70**, 49–70 (2015)

15. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.Y.: Asymptotic expansion of the degree of regularity for semi-regular systems of equations. In: MEGA'05 – Effective Methods in Algebraic Geometry. pp. 1–14 (2005)
16. Bouillaguet, C., Delaplace, C.: Sparse Gaussian elimination modulo p: An update. In: Proceedings CASC 2016 – Computer Algebra in Scientific Computing. pp. 101–116. Springer International Publishing (2016)
17. Boyer, B., Eder, C., Faugère, J., Lachartre, S., Martani, F.: GBLA: Gröbner basis linear algebra package. In: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016. pp. 135–142 (2016). https://doi.org/10.1145/2930889.2930914
18. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph.D. thesis, Universitat Innsbruck (1965)
19. Bunch, J.R., Hopcroft, J.E.: Triangular factorization and inversion by fast matrix multiplication. Mathematics of Computation **28**(125), 231–236 (1974)
20. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The computational complexity of some problems of linear algebra. J. Comput. System Sci. **58**(3), 572–596 (Jun 1999)
21. Cabarcas, D., Smith-Tone, D., Verbel, J.: Key recovery attack for ZHFE. In: Post-Quantum Cryptography 2017. LNCS, vol. 10346, pp. 289–308. Utrecht, The Netherlands (Jun 2017). https://doi.org/10.1007/978-3-319-59879-6_17, https://doi.org/10.1007/978-3-319-59879-6_17
22. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000. pp. 392–407. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
23. Cox, D., Little, J., O'Shea, D.: Ideals, Varieties, and algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra. Undergraduate Texts in Mathematics, Springer-Verlag, New York. (2001)
24. Debris-Alazard, T., Tillich, J.P.: Two attacks on rank metric code-based schemes: Ranksign and an identity-based-encryption scheme. In: Advances in Cryptology - ASIACRYPT 2018. LNCS, vol. 11272, pp. 62–92. Springer, Brisbane, Australia (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_3, https://doi.org/10.1007/978-3-030-03326-2_3
25. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. In: Advances in Cryptology - CRYPTO (2011)
26. Ding, J., Kleinjung, T.: Degree of regularity for HFE-. Cryptology ePrint Archive, Report 2011/570 (2011), http://eprint.iacr.org/2011/570, https://eprint.iacr.org/2011/570
27. Ding, J., Schmidt, D.: Solving degree and degree of regularity for polynomial systems over a finite fields. In: Number Theory and Cryptography - Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday. LNCS, vol. 8260, pp. 34–49. Springer (2013). https://doi.org/10.1007/978-3-642-42001-6_4, https://doi.org/10.1007/978-3-642-42001-6_4
28. Ding, J., Yang, B.Y.: Degree of regularity for HFEv and HFEv-. In: Post-Quantum Cryptography 2013. pp. 52–66. Limoges, France (Jun 2013). https://doi.org/10.1007/978-3-642-38616-9_4, https://doi.org/10.1007/978-3-642-38616-9_4
29. Dubois, V., Gama, N.: The degree of regularity of HFE systems. In: Advances in Cryptology - ASIACRYPT 2010. LNCS, vol. 6477, pp. 557–576. Springer, Singapore (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_32, https://doi.org/10.1007/978-3-642-17373-8_32

30. Eder, C., Faugère, J.C.: A survey on signature-based algorithms for computing gröbner bases. Journal of Symbolic Computation **80**, 719 – 784 (2017). https://doi.org/https://doi.org/10.1016/j.jsc.2016.07.031, http://www.sciencedirect.com/science/article/pii/S0747717116300785
31. Faugère, J.C.: A new efficient algorithm for computing gröbner bases (F4). J. Pure Appl. Algebra **139**(1-3), 61–88 (1999)
32. Faugère, J.C.: A new efficient algorithm for computing gröbner bases without reduction to zero: F5. In: Proceedings ISSAC'02. pp. 75–83. ACM press (2002)
33. Faugère, J.C., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of Minrank. In: Wagner, D. (ed.) Advances in Cryptology - CRYPTO 2008. LNCS, vol. 5157, pp. 280–296 (2008)
34. Faugère, J., Safey El Din, M., Spaenlehauer, P.: Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. In: International Symposium on Symbolic and Algebraic Computation, ISSAC 2010, Munich, Germany, July 25-28, 2010. pp. 257–264 (2010). https://doi.org/10.1145/1837934.1837984
35. Faugère, J.C., Safey El Din, M., Spaenlehauer, P.J.: Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1,1): Algorithms and complexity. J. Symbolic Comput. **46**(4), 406–437 (2011)
36. Gabidulin, E.M.: Theory of codes with maximum rank distance. Problemy Peredachi Informatsii **21**(1), 3–16 (1985)
37. Gabidulin, E.M., Paramonov, A.V., Tretjakov, O.V.: Ideals over a non-commutative ring and their applications to cryptography. In: Advances in Cryptology - EUROCRYPT'91. pp. 482–489. No. 547 in LNCS, Brighton (Apr 1991)
38. Gaborit, P., Murat, G., Ruatta, O., Zémor, G.: Low rank parity check codes and their application to cryptography. In: Proceedings of the Workshop on Coding and Cryptography WCC'2013. Bergen, Norway (2013), www.selmer.uib.no/WCC2013/pdfs/Gaborit.pdf
39. Gaborit, P., Ruatta, O., Schrek, J.: On the complexity of the rank syndrome decoding problem. IEEE Trans. Information Theory **62**(2), 1006–1019 (2016)
40. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: New results for rank-based cryptography. In: Progress in Cryptology - AFRICACRYPT 2014. LNCS, vol. 8469, pp. 1–12 (2014)
41. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: Ranksign: An efficient signature algorithm based on the rank metric (extended version on arxiv). In: Post-Quantum Cryptography 2014. LNCS, vol. 8772, pp. 88–107. Springer (2014), https://arxiv.org/pdf/1606.00629.pdf
42. Gaborit, P., Zémor, G.: On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Trans. Information Theory **62(12)**, 7245–7252 (2016)
43. Granboulan, L., Joux, A., Stern, J.: Inverting HFE is quasipolynomial. In: Advances in Cryptology - CRYPTO 2006. LNCS, vol. 4117, pp. 345–356. Springer, Santa Barbara, California, USA (Aug 2006). https://doi.org/10.1007/11818175_20, https://doi.org/10.1007/11818175_20
44. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. LNCS, vol. 1423, pp. 267–288. Springer (1998)
45. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Advances in Cryptology - CRYPTO'99. LNCS, vol. 1666, pp. 19–30.

Springer, Santa Barbara, California, USA (Aug 1999). https://doi.org/10.1007/3-540-48405-1
46. Lazard, D.: Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In: Computer algebra. LNCS, vol. 162, pp. 146–156. Springer, Berlin (1983), proceedings Eurocal'83, London, 1983
47. Le Gall, F.: Powers of tensors and fast matrix multiplication. In: Proceedings ISSAC'14. pp. 296–303. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2608628.2608664, http://doi.acm.org/10.1145/2608628.2608664
48. Levy-dit-Vehel, F., Perret, L.: Algebraic decoding of rank metric codes. Talk at the Special Semester on Gröbner Bases - Workshop D1 pp. 1–19 (2006), https://ricamwww.ricam.oeaw.ac.at/specsem/srs/groeb/download/Levy.pdf
49. Loidreau, P.: A new rank metric codes based encryption scheme. In: Post-Quantum Cryptography 2017. LNCS, vol. 10346, pp. 3–17. Springer (2017)
50. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.L.M.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes (2012), http://eprint.iacr.org/2012/409
51. Otmani, A., Talé-Kalachi, H., Ndjeya, S.: Improved cryptanalysis of rank metric schemes based on Gabidulin codes. Des. Codes Cryptogr. **86**(9), 1983–1996 (2018). https://doi.org/10.1007/s10623-017-0434-5, https://doi.org/10.1007/s10623-017-0434-5
52. Ourivski, A.V., Johansson, T.: New technique for decoding codes in the rank metric and its cryptography applications. Problems of Information Transmission **38**(3), 237–246 (2002). https://doi.org/10.1023/A:1020369320078
53. Overbeck, R.: A new structural attack for GPT and variants. In: Mycrypt. LNCS, vol. 3715, pp. 50–63 (2005)
54. Storjohann, A.: Algorithms for Matrix Canonical Forms. Ph.D. thesis, Swiss Federal Institute of Technology – ETH (2000)
55. Lévy-dit Vehel, F., Perret, L.: Algebraic decoding of codes in rank metric (Jun 2006), communication at YACC06, Porquerolles, France
56. Verbel, J., Baena, J., Cabarcas, D., Perlner, R., Smith-Tone, D.: On the complexity of "superdetermined" Minrank instances. In: Post-Quantum Cryptography 2019. LNCS, vol. 11505, pp. 167–186. Springer, Chongqing, China (May 2019). https://doi.org/10.1007/978-3-030-25510-7_10, https://doi.org/10.1007/978-3-030-25510-7_10

Appendix: Proof of Proposition 3

Let n, m, k, r be positive integers such that n and m are both greater than r. Let E be a $\mathbb{F}_q$-vector space of $\mathbb{F}_{q^m}$ of dimension r spanned by $\{E_1, E_2, \ldots, E_r\}$ and let $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ whose components generate E. By definition, there exists a non-zero coordinate e_j of $\boldsymbol{e}$, and hereafter one focuses on the vector space $\lambda E = \langle \lambda E_1, \lambda E_2, \ldots, \lambda E_r \rangle$ where $\lambda = e_j^{-1}$.

Given a basis $(1, \alpha, \ldots, \alpha^{m-1})$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, one can write a basis of λE as a matrix $\boldsymbol{S} \in \mathbb{F}_q^{m \times r}$. By construction, $1 \in \lambda E$, so that we can set the first column and the first row of $\boldsymbol{S}$ to the vectors $[1 \ 0 \ \cdots \ 0]^\mathsf{T}$ and $[1 \ 0 \ \cdots \ 0]$. We write $\widehat{\boldsymbol{S}}$ for the remaining $(m-1) \times (r-1)$ block of $\boldsymbol{S}$. One can also express the coordinates of the components of $\lambda \boldsymbol{e}$ (with respect to the basis $\{\lambda E_1, \lambda E_2, \ldots, \lambda E_r\}$)

as a matrix $\boldsymbol{C} \in \mathbb{F}_q^{r\times n}$. By construction, the j-th column of $\boldsymbol{C}$ is the vector $[1\ 0\ \cdots\ 0]^\mathsf{T}$.

Lemma 2 estimates the probability to come across an index j such that e_j is non-zero. Once such an index is found, Lemma 3 computes the probability that Algorithm 1 succeeds in finding a non-singular block in $\widehat{\boldsymbol{S}}$.

Lemma 2. *With the same notation and hypotheses as above, if an index j is chosen uniformly at random in $\{1,\dots,n\}$, then e_j will be non-zero with probability $(1-q^{-r})/(1-q^{-n})$.*

Proof. A component e_j of $\boldsymbol{e}$ will be non-zero if and only if its corresponding column of coordinates in the matrix $\boldsymbol{C}$ is non-zero. If the components of $\boldsymbol{e}$ were chosen uniformly at random in the vector space E of dimension r, the probability for a random component to be equal to zero would be exactly q^{-r}. This is not the case since there is a constraint on $\boldsymbol{C}$, more precisely it has to be of rank r.

Taking this into account, we can count the number of full rank matrices in $\mathbb{F}_q^{r\times n}$ that have a zero column. The ratio between those matrices and all the full rank matrices in $\mathbb{F}_q^{r\times n}$ is exactly the probability for a column chosen at random in $\boldsymbol{C}$ to be zero:

$$\prod_{i=0}^{r-1}\frac{q^{n-1}-q^i}{q^n-q^i}=\frac{q^{n-r}-1}{q^n-1}.$$

One concludes the proof by taking the complementary event. □

Lemma 3. *Let $c\in\{1,\dots,\lfloor\frac{m-1}{r-1}\rfloor\}$; with the same notation and hypotheses as above, if E and $\boldsymbol{e}$ are chosen uniformly at random, and if the inverse of a non-zero coordinate of $\boldsymbol{e}$, λ, is given, then at least one of the c disjoint blocks B_i in $\widehat{\boldsymbol{S}}$ is not singular with probability greater than*

$$1-\frac{(1-p_{q,r-1,r-1})^c}{p_{q,r-1,m-1}}$$

Proof. Since λ is a fixed nonzero element in $\mathbb{F}_{q^m}$ and since E is uniformly random, the vector space λE is also uniformly random. Therefore $\widehat{\boldsymbol{S}}$ is a matrix chosen uniformly at random among all the full rank matrices in $\mathbb{F}_q^{(m-1)\times(r-1)}$. The probability that all the c blocks B_i in $\widehat{\boldsymbol{S}}$ are singular is then bounded from above by

$$\frac{\left(q^{(r-1)^2}-q^{(r-1)^2}p_{q,r-1,r-1}\right)^c\ q^{(r-1)(m-1-c(r-1))}}{q^{(m-1)(r-1)}p_{q,r-1,m-1}},\tag{13}$$

which is the ratio between the number of matrices in $\mathbb{F}_q^{(m-1)\times(r-1)}$ with c singular disjoint blocks and the total amount of full rank matrices in $\mathbb{F}_q^{(m-1)\times(r-1)}$. It is an upper bound since the number of matrices with c singular blocks includes matrices that are not of full rank.

The reader can check that the term (13) is equal to

$$\frac{(1-p_{q,r-1,r-1})^c}{p_{q,r-1,m-1}}.$$

The probability that at least one of the B_i's is non-singular is obtained using the complementary probability. □

In Algorithm 1, the first requirement not to return fail is to find an index j such that e_j is non-zero; Lemma 3 gives the probability of this event, that is to say $(1-q^{-r})/(1-q^{-n})$. Once this index is found, the associated vector space λE is distributed uniformly among all the vector spaces of $\mathbb{F}_{q^m}$ of dimension r since E is chosen at random. Using Lemma 2, one has a lower bound on the probability that at least one of the c block B_i's is non singular. Thus the probability of Proposition 3 is

$$\frac{1-q^{-r}}{1-q^{-n}}\left(1-\frac{(1-p_{q,r-1,r-1})^c}{p_{q,r-1,m-1}}\right).$$

□