



**HAL**  
open science

# Improvement of algebraic attacks for solving superdetermined MinRank instances

Magali Bardet, Manon Bertin

► **To cite this version:**

Magali Bardet, Manon Bertin. Improvement of algebraic attacks for solving superdetermined MinRank instances. PQCrypto 2022, Sep 2022, virtual, France. hal-03740766

**HAL Id: hal-03740766**

**<https://hal.science/hal-03740766>**

Submitted on 29 Jul 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Improvement of algebraic attacks for solving superdetermined MinRank instances

Magali Bardet <sup>\*1</sup> and Manon Bertin <sup>†1</sup>

<sup>1</sup>LITIS, University of Rouen Normandie, France.

July 29, 2022

## Abstract

The MinRank (MR) problem is a computational problem that arises in many cryptographic applications. In Verbel et al. [24], the authors introduced a new way to solve superdetermined instances of the MinRank problem, starting from the bilinear Kipnis-Shamir (KS) modeling. They use linear algebra on specific Macaulay matrices, considering only multiples of the initial equations by one block of variables, the so called “kernel” variables. Later, Bardet et al. [7] introduced a new Support Minors modeling (SM), that consider the Plücker coordinates associated to the kernel variables, *i.e.* the maximal minors of the Kernel matrix in the KS modeling.

In this paper, we give a complete algebraic explanation of the link between the (KS) and (SM) modelings (for any instance). We then show that superdetermined MinRank instances can be seen as easy instances of the SM modeling. In particular, we show that performing computation at the smallest possible degree (the “first degree fall”) and the smallest possible number of variables is not always the best strategy. We give complexity estimates of the attack for generic random instances.

We apply those results to the DAGS cryptosystem, that was submitted to the first round of the NIST standardization process. We show

---

\*magali.bardet@univ-rouen.fr

†manon.bertin@etu.univ-rouen.fr

that the algebraic attack from Barelli and Couvreur [8], improved in Bardet et al. [5], is a particular superdetermined MinRank instance. Here, the instances are not generic, but we show that it is possible to analyse the particular instances from DAGS and provide a way to select the optimal parameters (number of shortened positions) to solve a particular instance.

**keywords** Post-quantum cryptography – MinRank problem – algebraic attack – DAGS cryptosystem.

# 1 Introduction

## The MinRank Problem

The MinRank problem was first mentioned in [12] where its NP-completeness was also proven. It is a central problem in algebraic cryptanalysis, starting with the Kipnis and Shamir modeling [18] for the HFE encryption scheme. The MinRank problem is very simple to state:

**Problem 1** (Homogeneous MinRank problem).

Input: a target rank  $r \in \mathbb{N}$  and  $K$  matrices  $\mathbf{M}_1, \dots, \mathbf{M}_K \in \mathbb{F}_q^{m \times n}$ .

Output: field elements  $x_1, x_2, \dots, x_K \in \mathbb{F}_q$ , not all zero, such that

$$\text{Rank} \left( \mathbf{M}_{\mathbf{x}} \stackrel{\text{def}}{=} \sum_{i=1}^K x_i \mathbf{M}_i \right) \leq r.$$

It plays a central role in public key cryptography. Many multivariate schemes are strongly related to the hardness of this problem, as in [20, 18, 21, 22]. The 3rd round NIST post-quantum competition finalist Rainbow [14], or alternate GeMSS [13] suffered attacks based on the MinRank problem [9, 10, 23, 3].

In code-based cryptography, the MinRank problem is exactly the decoding problem for matrix codes in rank metric. The two submissions ROLLO and RQC [2, 1], from the 2nd round of the NIST post-quantum competition, have been attacked using algebraic cryptanalysis in [6, 7]. Their security analysis relies on the decoding problem for  $\mathbb{F}_{q^m}$ -linear rank-metric codes, which can actually be reduced to the MinRank problem.

This is of great importance for cryptographic purposes to design algorithms that solve efficiently algebraic modeling for the MinRank problem, and to understand their complexity.

## Algebraic modeling

There has been a lot of recent progress in the algebraic modeling and solving of the MinRank problem. We start by recalling the first modeling, namely the Kipnis-Shamir (KS) modeling. Note that it implicitly assumes that the  $n-r$  first columns of the small-rank matrix  $\mathbf{M}_{\mathbf{x}}$  we are looking for are linearly dependent from the last  $r$  ones. In this paper, we will assume that we are looking for a matrix  $\mathbf{M}_{\mathbf{x}}$  of rank *exactly*  $r$  (this can be achieved by looking for increasing ranks, starting from  $r = 1$ ), and that the last  $r$  columns of  $\mathbf{M}_{\mathbf{x}}$  are linearly independent (it is true up to a permutation of the columns, and for random matrices it is true with high probability). We will see later that this last assumption is not mandatory.

**Modeling 1** (Kipnis-Shamir Modeling [18]). *Consider a MinRank instance  $(\mathbf{M}_1, \dots, \mathbf{M}_K) \in \mathbb{F}_q^{m \times n}$  with target rank  $r$ . Then, the MR problem can be solved by finding  $x_1, \dots, x_K \in \mathbb{F}_q^K$ , and  $\mathbf{C} = (c_{i,j}) \in \mathbb{F}_q^{r \times (n-r)}$  such that*

$$\left( \sum_{i=1}^K x_i \mathbf{M}_i \right) \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathbf{C} \end{pmatrix} = \mathbf{0}_{m \times (n-r)}. \quad (\text{KS})$$

The  $m(n-r)$  equations are bilinear in the  $K$  linear variables  $\mathbf{x} = (x_1, \dots, x_K)$  and the  $r(n-r)$  entries of the formal matrix  $\mathbf{C} = (c_{i,j})_{i,j}$ , referred to as the kernel variables.

It is clear that a matrix has rank  $\leq r$  if and only if its right kernel has dimension at least  $n-r$ , so that any solution of the MinRank problem is a solution of (KS).

The complexity of solving a generic bilinear system has been studied in [17, 15], and gives an upper bound for the KS system, but this estimate wildly overestimates the experimental results.

The matrix  $\mathbf{M}_{\mathbf{x}}$  has rank  $\leq r$  if and only if all its minors of size  $r+1$  are zero. This modeling has been presented and analysed in [16, 17]. Under the assumption that the last  $r$  columns of  $\mathbf{M}_{\mathbf{x}}$  are linearly independent, it is sufficient to consider minors involving columns in sets  $T = \{t\} \cup \{n-r+1..n\}$

with  $1 \leq t \leq n - r$ , as it means that the last  $r$  columns generate the column vector space. The notation  $|\mathbf{M}|_{J,T}$  represents the determinant of the submatrix of  $\mathbf{M}$  where we keep only rows in  $J$  and columns in  $T$ .

**Modeling 2** (Minors Modeling). *Let  $(\mathbf{M}_1, \dots, \mathbf{M}_K) \in \mathbb{F}_q^{m \times n}$  be a MinRank instance with target rank  $r$ . Then, the MR problem can be solved by finding  $x_1, \dots, x_K \in \mathbb{F}_q^K$  such that*

$$\left\{ |\mathbf{M}_x|_{J,T} = 0, \forall J \subset \{1..m\}, \#J = r + 1, T = \{t\} \cup \{n - r + 1..n\} \subset \{1..n\} \right\} \quad (\text{Minors})$$

Recently, a new modeling has been introduced in [7], that is at the moment the most efficient one from the complexity point of view. It uses two ideas, that we will separate in two modelings: the first idea is that (KS) means that the vector space with generator matrix  $\mathbf{M}_x$  is orthogonal to the one with generator matrix  $(\mathbf{I}_{n-r} \ \mathbf{C}^\top)$ . It is then straightforward to see that any row of  $\mathbf{M}_x$  belongs to the dual space with generator matrix  $(-\mathbf{C} \ \mathbf{I}_r)$ . This leads to:

**Modeling 3** (Support Minor Modeling-C [7]). *Let  $(\mathbf{M}_1, \dots, \mathbf{M}_K) \in \mathbb{F}_q^{m \times n}$  be a MinRank instance with target rank  $r$ . Then, the MR problem can be solved by finding  $x_1, \dots, x_K \in \mathbb{F}_q^K$ , and  $\mathbf{C} = (c_{i,j})_{1 \leq i \leq r, 1 \leq j \leq n-r} \in \mathbb{F}_q^{r \times (n-r)}$  such that*

$$\left\{ \left| \begin{pmatrix} \mathbf{r}_i \\ -\mathbf{C} \ \mathbf{I}_r \end{pmatrix} \right|_{*,T} = 0, \forall T \subset \{1..n\}, \#T = r + 1, \text{ and } \mathbf{r}_i \text{ row of } \mathbf{M}_x \right\}. \quad (\text{SM-C})$$

The  $m \binom{n}{r+1}$  equations are bi-homogeneous with bi-degree  $(1, r)$  in the  $K$  linear variables  $\mathbf{x} = (x_1, \dots, x_K)$  and the  $r(n - r)$  entries of the formal matrix  $\mathbf{C} = (c_{i,j})_{i,j}$ , referred to as the kernel variables.

Note that in (SM-C), the entries of  $\mathbf{C}$  appear only as maximal minors of  $(-\mathbf{C} \ \mathbf{I}_r)$ . This leads to the second idea from [7], which consists in using the Plücker coordinates: we replace each  $|(-\mathbf{C} \ \mathbf{I}_r)|_{*,T}$ , that is a polynomial of degree  $\#T = r$  in the entries of  $\mathbf{C}$ , by a new variable  $c_T$  using the injective Plücker map, see [11, p.6].

$$p : \{ \mathcal{W} \subset \mathbb{F}_q^n : \dim(\mathcal{W}) = r \} \rightarrow \mathbb{P}^N(\mathbb{F}_q) \quad (N = \binom{n}{r} - 1)$$

$$\mathbf{C} \mapsto (c_T)_{T \subset \{1..n\}, \#T=r}.$$

**Modeling 4** (Support Minor Modeling- $c_T$ [7]). *Let  $(\mathbf{M}_1, \dots, \mathbf{M}_K) \in \mathbb{F}_q^{m \times n}$  be a MinRank instance with target rank  $r$ . Then, the MR problem can be solved by finding  $x_1, \dots, x_K \in \mathbb{F}_q^K$ , and  $(c_T)_{T \subset \{1..n\}, \#T=r} \subset \mathbb{F}_q^{\binom{n}{r}}$  such that*

$$\left\{ \sum_{t \in T} (\mathbf{M}_{\mathbf{x}})_{i,t} c_{T \setminus \{t\}} = 0, \forall T \subset \{1..n\}, \#T = r + 1, \text{ and } i \in \{1..m\} \right\}. \quad (\text{SM})$$

*The  $m \binom{n}{r+1}$  equations are bilinear in the  $K$  linear variables  $\mathbf{x} = (x_1, \dots, x_K)$  and the  $\binom{n}{r}$  minor variables  $c_T$ , for all  $T \subset \{1..n\}, \#T = r$ .*

The benefit of introducing such coordinates to describe a vector space rather than a matrix describing a basis is that contrarily to the matrix representation, a vector space  $W$  has unique Plücker coordinates associated to it. This is not the case of the matrix representation of a vector space: if the rows of a matrix  $\mathbf{C}$  generate the vector space, then the rows of  $\mathbf{A}\mathbf{C}$  generate the same vector space for any invertible  $\mathbf{A} \in GL(r, \mathbb{F}_q)$ . For our algebraic system, this brings the benefit of reducing the number of solutions of the system: there are several solutions  $\mathbf{C}$  to the algebraic system (SM- $\mathbf{C}$ ), that correspond to one unique solution to (SM). As already pointed out in [7], it is also extremely beneficial for the computation to replace polynomials  $|(-\mathbf{C} \mathbf{I}_r|_{*,T})|$  with  $r!$  terms of degree  $r$  in the entries of  $\mathbf{C}$  by single variables  $c_T$ 's in  $\mathbb{F}_q$ . We will use (SM- $\mathbf{C}$ ) for the theoretical analysis of the link between the various modelings, and (SM) for the computational solving.

## Contributions

As a first contribution, we show that the first three systems are related, more precisely

**Proposition 1.** *The set of equations (KS) is included in the set of equations (SM- $\mathbf{C}$ ), and the ideals generated by the SM- $\mathbf{C}$  and KS equations are equal. Equations Minors (Minors modeling) are included in the ideal generated by KS.*

This proposition applies to any instance, without particular hypothesis.

Note that Eq. (Minors) contain only the linear variables, hence the ideals cannot be equal.

This proposition is not only interesting on the theoretical point of view, but it also allows to understand different computational strategies and to select the best one. A discussion is provided in Remark 1.

In [24], Verbel et al. analyse degree falls occurring during a Gröbner basis computation of (KS), and show that for overdetermined systems this can occur before degree  $r+2$ , which is the general case. As a second contribution, we show that these degree falls are in fact equations from SM-C. Using the Plücker coordinates in (SM) allows to drastically reduce the size of the considered matrices. Moreover, we give example to show that minimising the degree and number of equations is not always the best strategy for optimising the solving complexity.

Finally, we revisit the DAGS cryptosystem [4], that was a 1st round candidate to the NIST post-quantum standardization process, and was attacked by Barelli and Couvreur [8]. We show that the attack is in fact a MinRank attack, and describe the structure of this non-random superdetermined Min-Rank instance. This precise understanding of the problem makes it possible to choose the right parameters for an optimal attack.

## 2 Notation and preliminaries

Vectors are denoted by lower case boldface letters such as  $\mathbf{x}$ ,  $\mathbf{e}$  and matrices by upper case letters  $\mathbf{C}$ ,  $\mathbf{M}$ . The all-zero vector of length  $\ell$  is denoted by  $\mathbf{0}_\ell$ . The  $j$ -th coordinate of a vector  $\mathbf{x}$  is denoted by  $x_j$  and the submatrix of a matrix  $\mathbf{C}$  formed from the rows in  $I$  and columns in  $J$  by  $\mathbf{C}_{I,J}$ . When  $I$  (*resp.*  $J$ ) represents all the rows (*resp.* columns), we may use the notation  $\mathbf{C}_{*,J}$  (*resp.*  $\mathbf{C}_{I,*}$ ). We simplify  $\mathbf{C}_{i,*} = \mathbf{C}_{\{i\},*}$  (*resp.*  $\mathbf{C}_{*,j} = \mathbf{C}_{*,\{j\}}$ ) for the  $i$ -th row of  $\mathbf{C}$  (*resp.*  $j$ -th column of  $\mathbf{C}$ ) and  $c_{i,j} = \mathbf{C}_{\{i\},\{j\}}$  for the entry in row  $i$  and column  $j$ . Finally,  $|\mathbf{C}|$  is the determinant of a matrix  $\mathbf{C}$ ,  $|\mathbf{C}|_{I,J}$  is the determinant of the submatrix  $\mathbf{C}_{I,J}$  and  $|\mathbf{C}|_{*,J}$  the one of  $\mathbf{C}_{*,J}$ . The transpose of a matrix  $\mathbf{C}$  is  $\mathbf{C}^\top$ .

The all-one vector of size  $n$  is denoted by  $\mathbf{1}_n = (1, \dots, 1)$ .

To simplify the presentation, we restrict ourselves to a field of characteristic 2, but the results are valid for any characteristic, the only difference being the occurrence of a  $\pm$  sign before each formula.

For any matrix  $\mathbf{A}$  of size  $q \times r$  with  $r \leq q$ , and any set  $J \subset \{1..q\}$  of size  $r+1$ , we define the vector  $\mathbf{V}_J(\mathbf{A})$  of length  $q$  whose  $j$ st entry is 0 if  $j \notin J$  and  $|\mathbf{A}|_{J \setminus \{j\},*}$  for  $j \in J$ . For  $\mathbf{A}$  of size  $r \times q$  with  $r \leq q$  we define  $\mathbf{V}_J(\mathbf{A}) \stackrel{\text{def}}{=} \mathbf{V}_J(\mathbf{A}^\top)$ .

Using Laplace expansion along a column, it is clear that for any vector  $\mathbf{a}$

of length  $q$  we have

$$\mathbf{V}_J(\mathbf{A})\mathbf{a}^\top = |(\mathbf{a}^\top \ \mathbf{A})|_{J,*}. \quad (1)$$

We denote by  $vec_{row}(\mathbf{A})$  (*resp.*  $vec_{col}(\mathbf{A})$ ) the vertical vector formed by concatenating successive rows (*resp.* cols) of  $\mathbf{A}$ . We have the formula

$$\begin{aligned} vec_{row}(\mathbf{A}\mathbf{X}\mathbf{Y}) &= (\mathbf{A} \otimes \mathbf{Y}^\top)vec_{row}(\mathbf{X}) \\ vec_{col}(\mathbf{A}\mathbf{X}\mathbf{Y}) &= (\mathbf{Y}^\top \otimes \mathbf{A})vec_{col}(\mathbf{X}) \end{aligned} \quad (2)$$

where  $\mathbf{A} \otimes \mathbf{B} \stackrel{\text{def}}{=} (a_{i,j}\mathbf{B})_{i,j}$  is the Kronecker product of two matrices  $\mathbf{A} = (a_{i,j})_{i,j}$  and  $\mathbf{B}$ .

For a system  $\mathcal{F} = \{f_1, \dots, f_M\}$  of bilinear equations in two sets of variables  $\mathbf{x} = (x_j)_{1 \leq j \leq n_x}$  and  $\mathbf{y} = (y_\ell)_{1 \leq \ell \leq n_y}$ , it is usual to consider the associated Jacobian matrices:

$$Jac_{\mathbf{x}}(\mathcal{F}) = \left( \frac{\partial f_i}{\partial x_j} \right)_{i=1..M, j=1..n_x}, \quad Jac_{\mathbf{y}}(\mathcal{F}) = \left( \frac{\partial f_i}{\partial y_\ell} \right)_{i=1..M, \ell=1..n_y}.$$

For homogeneous bilinear systems they satisfy the particular relation:

$$Jac_{\mathbf{x}}(\mathcal{F})\mathbf{x}^\top = (f_1 \ \dots \ f_M)^\top$$

and any vector in the left kernel of a Jacobian matrix is a syzygy of the system. Moreover,  $Jac_{\mathbf{x}}$  is a matrix whose entries are linear form in the variables  $\mathbf{y}$ , and Cramer's rule show that the left kernel of  $Jac_{\mathbf{x}}$  contains vectors  $\mathbf{V}_T(Jac_{\mathbf{x}}(\mathcal{F})_{T,*})$  using the notation from (1) for all  $T \subset \{1..M\}$  of size  $n_y + 1$ . Generically those vectors generate the left kernel. For affine systems, we consider the jacobian matrix associated to the homogeneous part of highest degree of the system, and any syzygy for this part, that is not a syzygy of the entire system leads to a degree fall.

### 3 Relations between the various modelings

This section applies to any MinRank instance without any specific hypothesis.

The KS modeling consists in bilinear equations in two blocks of variables  $\mathbf{x}$  and  $\mathbf{C}$ , whereas the SM-C modeling contains equations of degree 1 in  $\mathbf{x}$  and  $r$  in  $\mathbf{C}$ , the variables  $\mathbf{C}$  appearing only as maximal minors of  $(-\mathbf{C} \ \mathbf{I}_r)$ .

In the case of the KS modeling, it has been noticed in [24], and later [7, Lemma1] that the jacobian matrices have a very particular shape: if we write  $\mathbf{M}_x = \begin{pmatrix} \mathbf{M}_x^{(1)} & \mathbf{M}_x^{(2)} \end{pmatrix}$  with  $\mathbf{M}_x^{(1)}$  of size  $m \times (n-r)$  and  $\mathbf{M}_x^{(2)}$  of size  $m \times r$ , and in the same way we write each  $\mathbf{M}_i = \begin{pmatrix} \mathbf{M}_i^{(1)} & \mathbf{M}_i^{(2)} \end{pmatrix}$ , then the homogeneous part of highest degree of the system is  $\mathbf{M}_x^{(2)}\mathbf{C}$ , and its Jacobian matrices are, if we take the variables and equations in row/column order:

$$\begin{aligned} Jac_{x_i} \left( vec_{row} \left( x_i \mathbf{M}_i^{(2)} \mathbf{C} \right) \right) &= vec_{row} \left( \mathbf{M}_i^{(2)} \mathbf{C} \right) = (\mathbf{I}_m \otimes \mathbf{C}^\top) vec_{row} \left( \mathbf{M}_i^{(2)} \right) \\ Jac_x \left( vec_{row} \left( \mathbf{M}_x^{(2)} \mathbf{C} \right) \right) &= (\mathbf{I}_m \otimes \mathbf{C}^\top) \left( vec_{row} \left( \mathbf{M}_1^{(2)} \right) \quad \dots \quad vec_{row} \left( \mathbf{M}_K^{(2)} \right) \right) \\ Jac_{vec_{col}(\mathbf{C})} \left( vec_{col} \left( \mathbf{M}_x^{(2)} \mathbf{C} \right) \right) &= \mathbf{I}_{n-r} \otimes \mathbf{M}_x^{(2)} \end{aligned} \quad (3)$$

The jacobian matrix in  $\mathbf{C}$  admits a left kernel that contains the following vectors:

$$\mathbf{e}_i \otimes \mathbf{V}_J(\mathbf{M}_x^{(2)}) \text{ for any } J \subset \{1..p\}, \#J = r+1, 1 \leq i \leq n-r, \quad (4)$$

where  $\mathbf{e}_i$  is the  $i$ th row of  $\mathbf{I}_{n-r}$ . As a consequence, the ideal generated by the (KS) equations contains the equations

$$\begin{aligned} (\mathbf{e}_i \otimes \mathbf{V}_J(\mathbf{M}_x^{(2)})) vec_{col}(\mathbf{M}_x \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathbf{C} \end{pmatrix}) &= (\mathbf{e}_i \otimes \mathbf{V}_J(\mathbf{M}_x^{(2)})) vec_{col}(\mathbf{M}_x \mathbf{1}) \\ &= \mathbf{V}_J(\mathbf{M}_x^{(2)}) \mathbf{M}_x^{(1)} \mathbf{e}_i^\top \text{ ( thanks to (2) )} \\ &= |\mathbf{M}_x|_{J, \{i\} \cup \{n-r+1..n\}} \text{ ( thanks to (1) )}. \end{aligned}$$

Those are precisely the (Minors) equations.

The jacobian matrix in  $\mathbf{x}$  admits a left kernel that contains the vectors

$$\mathbf{e}_\ell \otimes \mathbf{V}_J(\mathbf{C}) \text{ for any } J \subset \{1..n-r\}, \#J = r+1, 1 \leq \ell \leq m. \quad (5)$$

The ideal generated by the (KS) equations contains the equations

$$\begin{aligned} (\mathbf{e}_\ell \otimes \mathbf{V}_J(\mathbf{C})) vec_{row}(\mathbf{M}_x^{(1)}) &= \mathbf{e}_\ell \mathbf{M}_x^{(1)} \mathbf{V}_J(\mathbf{C})^\top = \mathbf{V}_J(\mathbf{C}) (\mathbf{M}_x^{(1)}|_{\ell,*})^\top \\ &= \left| \left( (\mathbf{M}_x^{(1)}|_{\ell,*})^\top \quad \mathbf{C}^\top \right) \right|_{J,*} = \left| \begin{pmatrix} \mathbf{M}_x^{(1)}|_{\ell,*} \\ \mathbf{C} \end{pmatrix} \right|_{*,J}. \end{aligned}$$

They are exactly the (SM- $\mathbf{C}$ ) equations for  $J \subset \{1..n-r\}$ . They have a degree  $r$  in the kernel variables  $c_{i,j}$ .

In [24], the authors propose to solve (any) instances of KS by considering particular elements in the left kernel of the Jacobian matrix in  $\mathbf{x}$  for some degree  $1 \leq d \leq r - 1$ . This is done by considering all combination of the polynomials with coefficients

$$\mathbf{e}_\ell \otimes \mathbf{V}_J(\mathbf{C}_{T,*}) \quad (6)$$

for any  $d \in \{1..r\}$ ,  $J \subset \{1..n - r\}$ ,  $\#J = d + 1$ ,  $T \subset \{1..r\}$ ,  $\#T = d$  and  $\ell \in \{1..m\}$ . The authors in [24, Theorem 2] construct a matrix  $\mathbf{B}_J$  whose left kernel contains elements related to the left kernel of the Jacobian matrix in  $\mathbf{x}$ . The key remark is that the equations they consider are

$$\begin{aligned} (\mathbf{e}_\ell \otimes \mathbf{V}_J(\mathbf{C}_{T,*}))_{\text{vec}_{\text{row}}(\mathbf{M}_\mathbf{x} \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathbf{C} \end{pmatrix})} &= \mathbf{e}_\ell \mathbf{M}_\mathbf{x} \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathbf{C} \end{pmatrix} \mathbf{V}_J(\mathbf{C}_{T,*})^\top \\ &= \left| \begin{pmatrix} \mathbf{r}_\ell \\ -\mathbf{C} \mathbf{I}_r \end{pmatrix} \right|_{*,T'} \end{aligned}$$

for  $T' = J \cup (\{n - r + 1..n\} \setminus (T + n - r)) \subset \{1..n\}$  of size  $r + 1$ . The equations have indeed a degree  $d$  in the kernel variables  $c_{i,j}$ .

Note that for  $d = 0$ , for  $T' = \{\ell\} \cup \{n - r + 1..n\}$  the equation  $\left| \begin{pmatrix} \mathbf{r}_\ell \\ -\mathbf{C} \mathbf{I}_r \end{pmatrix} \right|_{*,T'}$  is the  $\ell$ th KS equation, and we get all SM equations. As a consequence, we have proven Proposition 1.

**Remark 1.** *In the light of the previous results, we can understand more precisely the behavior of a generic Gröbner basis (GB) algorithm with a graded monomial ordering and a Normal selection strategy run on (KS) or (SM-C). As (SM-C) contains (KS) directly into the system, computing a GB on (SM-C) will also compute all equations that would be computed by (KS). On the other hand, when computing a GB for (KS), the algorithm will produce all equations (SM-C) by multiplying by monomials in  $\mathbf{C}$ , hence we can expect many syzygies during a GB computation on (SM-C).*

*This encourages to compute with (SM-C), but to look only at multiple of the equations by the  $x_i$ 's variables, which is the strategy proposed in [7]. Adding to this the change of variable that consider any minor of  $\mathbf{C}$  as a variable removes the hardness of computing with high degree polynomials (as the new variables have degree 1 instead of a polynomial of degree  $d$  with  $d!$  coefficients for the minor).*

## 4 Complexity of solving superdetermined systems

Superdetermined MinRank instances are defined in [24] as MinRank instances where  $K < rm$ . In the light of the previous section, it is now clear that [24] considers for any  $0 \leq d \leq r$  the equations

$$\mathcal{E}(d) \stackrel{\text{def}}{=} \left\{ E_{J,T,\ell} \stackrel{\text{def}}{=} \mathbf{e}_\ell \mathbf{M}_x \begin{pmatrix} \mathbf{I}_{n-r} \\ \mathbf{C} \end{pmatrix} \mathbf{V}_J(\mathbf{C}_{T,*})^\top : \begin{array}{l} \forall J \subset \{1..n-r\}, \#J=d+1, \\ \forall T \subset \{1..r\}, \#T=d, \\ \forall \ell \in \{1..m\} \end{array} \right\}. \quad (7)$$

and search for linear combination that will produce degree falls. We can rewrite the equations

$$\begin{aligned} E_{J,T,\ell} &= \left| \begin{pmatrix} \mathbf{r}_\ell \\ \mathbf{C} \mathbf{I}_r \end{pmatrix} \right|_{*,T'} \quad \text{with } T' = J \cup (\{n-r+1..n\} \setminus (T+n-r)) \subset \{1..n\} \\ &= \sum_{i=1}^K \sum_{j \in J} (\mathbf{M}_i^{(1)})_{\ell,j} x_i |\mathbf{C}|_{T,J \setminus \{j\}} + \sum_{i=1}^K \sum_{s \notin T} (\mathbf{M}_i^{(2)})_{\ell,s} x_i |\mathbf{C}|_{T \cup \{s\}, J}. \end{aligned}$$

We have a total of  $\sum_{d=0}^r m \binom{n-r}{d+1} \binom{r}{d} = m \binom{n}{r+1}$  equations in  $\sum_{d=0}^r K \binom{n-r}{d} \binom{r}{d} = K \binom{n}{r}$  variables described by

$$\mathcal{V}(d) \stackrel{\text{def}}{=} \{x_i |\mathbf{C}|_{T,J}\}_{i=1..K, \#J=d, \#T=d}, \quad \mathcal{V}(r+1) = \emptyset.$$

The system can be solved by linearization by constructing the associated Macaulay matrix: its rows are indexed by  $J, T, \ell$  (with  $\#J = d+1, \#T = d$ ) and its columns by  $i, J', T'$  (with  $\#J' = d = \#T'$ ), and the coefficient in row  $(J, T, \ell)$  and column  $(i, J', T')$  corresponds to the coefficient of  $E_{J,T,\ell}$  in the monomial  $x_i |\mathbf{C}|_{T',J'}$ . We can sort the columns by decreasing degree, *i.e.* consider first monomials in  $\mathcal{V}(r)$ , up to  $\mathcal{V}(0)$  that are the  $K$  variables  $x_i$ , see Fig. 1. Then finding linear combination of the equations that produce degree falls can be done by computing the echelon form of the Macaulay matrix. For a set of rows in  $\mathcal{E}(d)$ , we have  $m \binom{n-r}{d+1} \binom{r}{d}$  equations in  $K \binom{n-r}{d+1} \binom{r}{r+1} + K \binom{n-r}{d} \binom{r}{d}$  monomials, and we get generically a degree fall under the condition

$$m \binom{n-r}{d+1} \binom{r}{d} \geq K \binom{n-r}{d+1} \binom{r}{d+1}$$

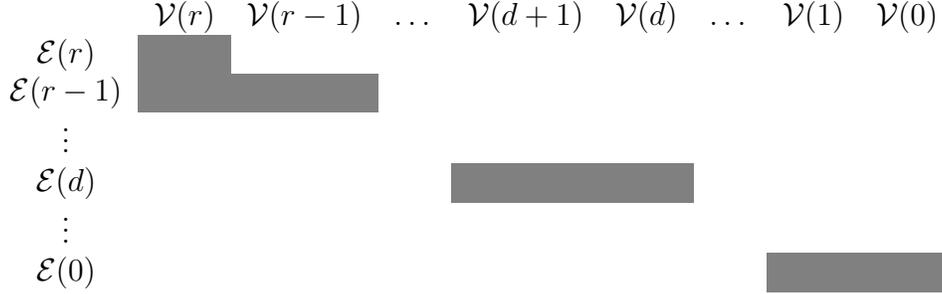


Figure 1: Shape of the Macaulay matrix associated to Eq. (7). The columns correspond to equations in  $\mathcal{V}(d)$ , the rows to equations  $\mathcal{E}(d)$ . Gray cells correspond to non-zero part of the matrix.

which is Corollary 5 in [24], and the first part of the Macaulay matrix with columns in  $\mathcal{V}(d+1)$  is, up to a good choice of the ordering of rows and columns, a block of diagonal matrices  $\mathbf{B}_J$  as described in [24].

The best complexity estimates comes from the (SM) modeling, when considering the minors  $|\mathbf{C}|_{T,J}$  as new variables. Eq. (7) contains  $m \binom{n}{r+1}$  equations in  $K$  variables  $x_i$  and  $\binom{n}{r}$  variables that are minors of  $(-\mathbf{C} \mathbf{I}_r)$ . Hence the system can be solved whenever  $m \binom{n}{r+1} \geq K \binom{n}{r}$  by linearization, *i.e.*  $m(n-r) \geq K(r+1)$ . After linearization, we get with overwhelming probability  $\#\mathcal{V}(0) - 1 = m - 1$  linear equations in the  $x_i$ 's only.

As always, it is possible to improve computation by puncturing the matrix  $\mathbf{M}_x$  (taking only sufficiently many columns so that we keep an overdetermined system), or by hybrid approach (performing an exhaustive search on some columns of  $\mathbf{C}$ , at the cost of  $q^{ar}$  operation in  $\mathbb{F}_q$  for  $a$  columns). It is also possible, as in [7, Eq. (23) p. 19], to compute equations at higher degree  $b$  in  $x_i$ . For instance, at  $b = 2$  we can multiply all equations in (SM) by  $x_i$ 's variables, and we get for each set  $\mathcal{E}(d)$  of equations:

$$m \binom{n}{r+1} K - \binom{n}{r+2} \binom{m+1}{2} \text{ equations, } \binom{n}{r} \binom{K+1}{2} \text{ monomials.} \quad (8)$$

For instance, Table 1 compares the (SM) system with previous results from [24]. For  $r = 5$ , the ratio between equations and monomials in (SM) is smaller than 1, so that we cannot expect to solve by linearization directly. Computing at  $b = 2$  would produce 14400 equations in 13860 variables of

$m$	$n$	$K$	$r$	$\frac{m(n-r)}{K(r+1)}$	$n_{eq}$	$n_{vars}$	$n_{rows}$ in [24]
10	10	10	2	2.6	1200	450	1530
10	5	10	2	1	100	100	
10	10	10	3	1.75	2100	1200	20240
10	7	10	3	1	350	350	
10	10	10	4	1.2	2520	2100	38586
10	9	10	4	1	1260	1260	
10	10	10	5	<b>0.8</b>	2100	2520	341495

Table 1: Size of matrices on (SM) for a MinRank instance with  $K = 10$  matrices of size  $m \times n$ , for various  $r$ .  $n$  can be decreased by puncturing the matrices to get a speedup. The results have been verified experimentally on random instances.

degree less than or equal to 7 in  $(\mathbf{x}, \mathbf{C})$ . Note that the last entry for  $r = 6$  would theoretically require to go up to  $b = 5$  with matrices of size 427350.

However, [24] suggest that we can have a closer look at the shape of the equations and maybe find a better complexity for some very overdetermined instances.

Hence, for a fixed  $d \in \{0..r\}$ , the set  $\mathcal{E}(d)$  contains  $m \binom{n-r}{d+1} \binom{r}{d}$  equations with  $K \binom{n-r}{d} \binom{r}{d}$  monomials  $\mathcal{V}(d)$  of bidegree  $(1, d)$  and  $K \binom{n-r}{d+1} \binom{r}{d+1}$  variables  $\mathcal{V}(d+1)$  of bidegree  $(1, d+1)$ . In [24], the authors determine the first degree fall in KS by looking for the smallest  $d$  for which we have more equations in  $\mathcal{E}(d)$  than variables in  $\mathcal{V}(d+1)$ . This produces  $\text{Rank}(\mathcal{E}(d)) - \#\mathcal{V}(d+1)$  degree fall, but it is not clear how to end the computation. If there is more equations than variables, *i.e.*

$$m \binom{n-r}{d+1} \binom{r}{d} \geq K \binom{n-r}{d+1} \binom{r}{d+1} + K \binom{n-r}{d} \binom{r}{d} - 1$$

then with overwhelming probability, the linear system is full rank (its kernel has dimension 1 as the system is homogeneous in  $\mathbf{x}$ ) and a non-zero element in the kernel of the Macaulay matrix gives a value for each variable  $x_i | \mathbf{C}|_{T,J}$ . It is then straightforward to deduce  $x_i/x_{i_0}$  from two values  $x_i | \mathbf{C}|_{T,J}$  and

$x_{i_0} |\mathbf{C}|_{T,J}$ . On the other hand, if

$$K \binom{n-r}{d+1} \binom{r}{d+1} \leq m \binom{n-r}{d+1} \binom{r}{d}$$

and

$$m \binom{n-r}{d+1} \binom{r}{d} < K \binom{n-r}{d+1} \binom{r}{d+1} + K \binom{n-r}{d} \binom{r}{d} - 1$$

then it is necessary to add new equations to end the computation. It can be done by consider equations of “higher degree”. If each minor of  $\mathbf{C}$  is taken as a variable, it doesn’t add a computational burden. We can solve as soon as we can get sufficiently many blocks of equations  $\mathcal{E}(d)$  such that we get more equations than columns. Experimental results are presented Table 2 on the same parameters as [24, Table 2]. For instance for a MinRank problem with  $12 \times 12$  matrices and a target rank  $r = 4$ , the authors in [24] solve at degree  $d = 4$  in 58s, whereas it is more interesting to consider equations in  $\mathcal{E}(3..4)$  that have degree up to 5, without considering equations of degree 2..3. Note that if we puncture too much the matrices, for instance by taking only  $n = 8$  columns, then we do not have any more an overdetermined SM system, and solving it now require to produce more equations, for instance by considering  $b = 2$ . In this case, we get a system of 5880 equations in 5460 and we can solve, but this will be more costly than solving with  $n = 9$ .

**Remark 2.** *There is an asymetry between  $m$  and  $n$  in the modelings. It is always possible to exchange  $m$  and  $n$  by considering the transpose of the matrices, but it is not clear in general which problem will be easier ( $m > n$  or  $m < n$ ). For instance, for  $K = 10$ ,  $r = 2$  we can have the following behaviors: for  $m = 6, n = 7$  the Macaulay matrix up to  $d = r$  has size  $210 \times 210$ , whereas for  $m = 7, n = 6$  it is not possible to solve at  $b = 1$  (the Macaulay matrix has dimension  $140 \times 150$ ). We need to go to  $b = 2$  and solve a matrix of size  $980 \times 825$ . On the contrary, for  $m = 10, n = 6$ , the Macaulay matrix has dimension  $160 \times 140$  ( $d = 1..2$ ), whereas for  $m = 6, n = 10$ , the Macaulay matrix for  $d = r$  has size  $336 \times 280$ .*

*However, as the number of equations is a multiple of  $m$ , the best solution is often with  $m \geq n$ .*

## 5 Application to DAGS

DAGS scheme [4] is a key encapsulation mechanism (KEM) based on quasi-dyadic alternant codes that was submitted to the first round of the NIST

$r$	$\kappa$	$d$	size	time	[24]
4	8	0..4	$9504 \times 5940$	5.6 s	58 s
		3..4	$4032 \times 3528$	2.4 s	
	7	0..4	$5544 \times 3960$	2.1 s	38 s
	6	0..4	$3024 \times 2520$	0.74 s	21 s
		2..4	$2232 \times 2220$	0.52 s	
	5	0..4	$1512 \times 1512$	0.23 s	13 s
5	7	0..5	$11088 \times 9504$	11 s	756 s
		2..4	$9660 \times 9072$	9.6 s	
	6	0..5	$5544 \times 5544$	3.1 s	367s

Table 2: Experimental size of matrices on SM for a MinRank instance with  $K = 12$  matrices of size  $12 \times 12$ , for various  $r$ . It is possible to puncture the codes, by considering only  $n = \kappa + r$  columns of the matrices. We consider only systems for which SM solves at  $b = 1$ . The second row gives the size of a submatrix of blocks  $\mathcal{E}(d)$  for some  $d$  that solves the problem faster.

standardization process for a quantum resistant public key algorithm. It suffered from an algebraic attack [8] that efficiently recovers the private key, and was improved in [5]. Here, we show that the DAGS algebraic modeling is in fact a MinRank problem. However, the previous complexity results do not apply, as those MinRank instances have a structure, that can be used to understand more precisely the complexity.

## 5.1 Principle of the attack

We recall some elements of the scheme. DAGS is based on the McEliece scheme and uses Quasi-Dyadic Generalized Srivastava codes, which are a subfamily of alternant codes. The structure of such codes is what allowed DAGS to be attacked [8, 5].

The idea of the key-recovery attack leading to the modeling presented here is to find a subcode of the public code. The attack was proposed in two versions: a combinatorial one that uses brute force to find the subcode, and an algebraic one that relies on solving a polynomial system. The complexity of the combinatorial version is easy to compute, however the numbers of calculations remains too high to be done in practice. On the contrary, the algebraic attack is more efficient but its complexity is harder to estimate.

We focus on the second version and explain the principle. We begin by

computing the invariant subcode of the public code of the scheme. Then, we search for a subcode of this invariant code by solving a bilinear system built from public parts of the scheme. Finally, we can recover the support and multiplier of the original alternant code.

In the next subsection, we explain how the system we want to solve is built.

## 5.2 Original Modeling

Let  $\mathcal{C}_{\text{pub}}$  be the DAGS public code,  $\mathbf{H}_{\text{pub}}$  be the public key of the scheme, which is a parity-check matrix of  $\mathcal{C}_{\text{pub}}$ , and let  $\mathbf{G}_{\text{pub}}$  be its generator matrix.

We refer to [19, Chap. 12] for the definition of alternant codes. DAGS codes are quasi-dyadic alternant codes over  $\mathbb{F}_{q^m}$ , with  $q$  a power of 2 and  $m = 2$ . To build the system we need to understand the construction of quasi-dyadic alternant codes, that are alternant codes for which the support  $\mathbf{x}$  and the multiplier  $\mathbf{y}$  have a particular structure.

**Definition 1.** *Let  $\gamma \geq 1$  and  $n = 2^\gamma n_0$ . The support  $\mathbf{x} \in \mathbb{F}_{q^m}^n$  of a quasi-dyadic alternant code of order  $2^\gamma$  is constructed from  $(b_1, \dots, b_\gamma) \in \mathbb{F}_{q^m}^\gamma$  that are linearly independent over  $\mathbb{F}_2$ , and  $\tau = (\tau_1, \dots, \tau_{n_0}) \in \mathbb{F}_{q^{n_0}}^{n_0}$  as*

$$\mathbf{x} \stackrel{\text{def}}{=} \tau \otimes \mathbf{1}_{2^\gamma} + \mathbf{1}_{n_0} \otimes \mathbf{g},$$

where  $\mathbf{g} \stackrel{\text{def}}{=} (g)_{g \in \mathbb{G}}$  is a vector of all  $2^\gamma$  elements of the group  $\mathbb{G} = \langle b_1, \dots, b_\gamma \rangle_{\mathbb{F}_2}$  which is the vector space generated by the elements  $(b_i)$  over  $\mathbb{F}_2$ .

The elements  $\tau_i$  are randomly drawn from  $\mathbb{F}_{q^m}$  such that the cosets  $\tau_i + \mathbb{G}$  are pairwise disjoint.

For instance for  $\gamma = 2$ , we can choose  $\mathbf{g} = (0, b_1, b_2, b_1 + b_2) = b_1(0, 1, 0, 1) + b_2(0, 0, 1, 1)$ . For  $\gamma = 3$  we take  $\mathbf{g} = (0, b_1, b_2, b_1 + b_2, b_3, b_1 + b_3, b_2 + b_3, b_1 + b_2 + b_3) = b_1(0, 1, 0, 1, 0, 1, 0, 1) + b_2(0, 0, 1, 1, 0, 0, 1, 1) + b_3(0, 0, 0, 0, 1, 1, 1, 1)$ . In general, one possible order for  $\mathbf{g}$  is given by  $\mathbf{g} = \sum_{i=1}^\gamma b_i \mathbf{e}_i$  where

$$\mathbf{e}_i \stackrel{\text{def}}{=} (\mathbf{0}_{2^{i-1}}, \mathbf{1}_{2^{i-1}}, \mathbf{0}_{2^{i-1}}, \mathbf{1}_{2^{i-1}}, \dots) = \mathbf{1}_{2^{\gamma-i}} \otimes (0, 1) \otimes \mathbf{1}_{2^{i-1}}.$$

The group  $\mathbb{G}$  acts by translation on  $\mathbb{F}_{q^m}$ , and its action induces a permutation of the code  $\mathcal{C}_{\text{pub}}$ . This is what allows the DAGS system to have reduced public keys: the public matrix  $\mathbf{G}_{\text{pub}}$  is formed by blocks of size  $2^\gamma$

where each row of the block is deduced from the first row by one of the permutation induced by  $\mathbb{G}$ .

The attack in [8] introduces the *invariant subcode*  $\mathcal{C}_{\text{pub}}^{\mathbb{G}}$  with respect to  $\mathbb{G}$  of  $\mathcal{C}_{\text{pub}}$ , which is defined as

**Definition 2.** *The invariant code of  $\mathcal{C}_{\text{pub}}$  is defined by:*

$$\mathcal{C}_{\text{pub}}^{\mathbb{G}} = \{\mathbf{c} \in \mathcal{C}_{\text{pub}} \mid \forall (i, j) \in \{0..n_0 - 1\} \times \{1..2^\gamma\}, c_{i2^\gamma+j} = c_{i2^\gamma+1}\}.$$

The invariant subcode has dimension  $k_0 = k/2^\gamma$  where  $k$  is the dimension of  $\mathcal{C}_{\text{pub}}$ . Its generator matrix  $\mathbf{G}_{\text{inv}}$  is easy to compute from  $\mathbf{G}_{\text{pub}}$ : each block of  $2^\gamma$  rows of  $\mathbf{G}_{\text{pub}}$  gives one row of  $\mathbf{G}_{\text{inv}}$  by summation. The entries of  $\mathbf{G}_{\text{pub}}$  are then repeated by blocks of size  $2^\gamma$ , so that we can define a matrix  $\tilde{\mathbf{G}} \in \mathbb{F}_{q^m}^{k_0 \times n_0}$  satisfying  $\mathbf{G}_{\text{inv}} = \tilde{\mathbf{G}} \otimes \mathbf{1}_{2^\gamma}$ .

We introduce the component-wise product called Schur product:

**Definition 3.** *The Schur product of two codes  $\mathcal{A}$  and  $\mathcal{B} \subseteq \mathbb{F}_q^n$  corresponds to the code generated by all the component-wise products of one codeword from  $\mathcal{A}$  and one codeword of  $\mathcal{B}$ :*

$$\mathcal{A} \star \mathcal{B} = \langle \mathbf{a} \star \mathbf{b} \mid \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B} \rangle_{\mathbb{F}_q}$$

The attack in [8] amounts to find  $\mathcal{D}$ , an unknown subcode of  $\mathcal{C}_{\text{pub}}^{\mathbb{G}}$  such that  $\mathbf{x}$  is orthogonal to  $\mathcal{D} \star \mathcal{C}_{\text{pub}}^\perp$ . This leads to following system with 2 unknowns,  $\mathcal{D}$  and  $\mathbf{x}$ :

$$\mathbf{G}_{\mathcal{D} \star \mathcal{C}_{\text{pub}}^\perp} \cdot \mathbf{x}^\top = 0 \quad (9)$$

Algebraically, a generator matrix for  $\mathcal{D} \star \mathcal{C}_{\text{pub}}^\perp$  can be written with high probability as

$$((\mathbf{I}_{k_0-c} \quad \mathbf{U}) \cdot \mathbf{G}_{\text{inv}}) \star \mathbf{H}_{\text{pub}} \quad (10)$$

with  $c$  the codimension of  $\mathcal{D}$  in the invariant subcode  $\mathcal{C}_{\text{pub}}^{\mathbb{G}}$ . If we can not express the system like that, we just need to take another generator matrix for the invariant subcode of  $\mathcal{C}_{\text{pub}}$ . This finally leads to the original modeling:

$$((\mathbf{I}_{k_0-c} \quad \mathbf{U}) \mathbf{G}_{\text{inv}} \star \mathbf{H}_{\text{pub}}) \mathbf{x}^\top = 0 \quad (11)$$

where  $\mathbf{U}$  is a matrix of unknowns of size  $(k_0 - c) \times c$ ,  $\mathbf{G}_{\text{inv}} = \tilde{\mathbf{G}} \otimes \mathbf{1}_{2^\gamma}$  and  $\tilde{\mathbf{G}}$  is a public invariant matrix of size  $k_0 \times n_0$ ,  $\mathbf{H}_{\text{pub}}$  is the public parity-check matrix, and  $\mathbf{x} = \tau \otimes \mathbf{1}_{2^\gamma} + \sum_{i=1}^\gamma b_i \mathbf{1}_{n_0} \otimes \mathbf{e}_i \in \mathbb{F}_{q^m}^n$  is a vector of unknowns  $\tau = (\tau_1, \dots, \tau_{n_0})$  and  $(b_1, \dots, b_\gamma)$ .

**Remark 3.** As explained in [8], any affine map  $\mathbf{x} \rightarrow a\mathbf{x} + b$  for  $a \in \mathbb{F}_q^*$ ,  $b \in \mathbb{F}_q$  preserves the quasi-dyadic structure of the code, and leaves the code invariant, so that it is always possible to search among all possible  $\mathbf{x}$  for the ones that satisfy  $b_1 = 1$  and  $\tau_{n_0} = 0$ . Moreover, the vector  $\mathbf{x}^q$ , hence  $\text{Tr}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{x} + \mathbf{x}^q$  are also solution of the system (11), so that  $\text{Tr}(b_2)^{-1} \text{Tr}(\mathbf{x})$  is a solution with  $\tau_{n_0} = 0$ ,  $b_1 = 0$  and  $b_2 = 1$  (as  $\text{Tr}(a) = 0$  for  $a \in \mathbb{F}_q$  when  $m = 2$ ).

**Remark 4.** As explained in [5], there is a lot of redundancy among the equations. We avoid that by considering only one out of every  $2^\gamma$  rows in  $\mathbf{H}_{\text{pub}}$ .

### 5.3 Modeling Update

A simple (but fastidious, see Appendix A) computation allows to write the system as a MinRank instance with matrices of size  $(n_0 - k_0) \times k_0$ , when  $\tilde{\mathbf{G}} = (\mathbf{I}_{k_0} \mathbf{G})$  is taken in systematic form:

$$\left( \sum_{i=1}^{k_0} \tau_i \mathbf{M}_i + \sum_{j=1}^{n_0-k_0} \tau_{j+k_0} \mathbf{M}_{j+k_0} + \sum_{b=2}^{\gamma} b_i \mathbf{H}_i \right) \begin{pmatrix} \mathbf{I}_{k_0-c} \\ \mathbf{U}^\top \end{pmatrix} = \mathbf{0} \quad (12)$$

with  $\mathbf{M}_i = \begin{pmatrix} \mathbf{0}_{i-1} & (\mathbf{G}_{\{i\},*})^\top & \mathbf{0}_{k_0-i} \end{pmatrix} \forall 1 \leq i \leq k_0$

$\mathbf{M}_{j+k_0} = \begin{pmatrix} \mathbf{0}_{j-1} \\ (\mathbf{G}_{*,\{j\}})^\top \\ \mathbf{0}_{n_0-k_0-j} \end{pmatrix} \forall 1 \leq j \leq n_0 - k_0$

$\mathbf{H}_i = \mathbf{H}_{\text{pub}} (\mathbf{I}_{n_0} \otimes \mathbf{e}_i^\top)_{*,\{1..k_0\}} \forall 2 \leq i \leq \gamma$

It is clear that the matrices  $\mathbf{M}_i$  from DAGS instances are not random, and in practice we have more degree falls than expected. On the other hand, the part concerning the variables  $b_i$  with matrices  $\mathbf{H}_i$  seems to behave like a random system. Note also that experimentally we find that the system always produces 3 solutions. However, this is small enough to be able to recover the good one from the kernel of the Macaulay matrix, as only one belong to the finite field  $\mathbb{F}_q$ .

**Proposition 2.** For the DAGS modeling, the Macaulay matrix associated to the set of equations  $\mathcal{E}(d)$  has size  $N_{\text{rows}} \times N_{\text{cols}} = (n_0 - k_0) \binom{k_0-c}{d+1} \binom{c}{d} \times (n_0 -$

Security Level	$q$	$n_0$	$k_0$	$\gamma$	$c$	$k_0 - a_0 - c$	Matrix size	Rank	Time
DAGS_1 (128)	$2^5$	52	26	4	4	4	$1456 \times 2520$	1322	3.5s
DAGS_3 (192)	$2^6$	38	16	4	4	5	$2772 \times 4284$	2540	8.8s
DAGS_5 (256)	$2^6$	33	11	2	2	3	$220 \times 310$	194	0.0s

Table 3: DAGS original sets of parameters

$k_0 - 1 + c + \gamma - 1) \binom{k_0 - c}{d+1} \binom{c}{d+1}$ , but its rank is

$$\text{Rank}(\mathcal{E}(d)) = \min \left( N_{rows}, \binom{k_0 - c}{d+1} \left( (n_0 - k_0) \binom{c-1}{d} + \binom{c}{d+1} d \right) \right)$$

Note that it is always possible to use shortened codes on  $a_0$  positions, that amounts to consider codes with parameters  $(n_0 - a_0, k_0 - a_0)$ .

The first sets of parameters were given in the specifications of the scheme. They are shown in Table3. Experimental results in [5] give a solution of the system DAGS\_3 in degree 4 with linear algebra on a matrix of size  $725,895 \times 671,071$ . It is improved by shortening the system up to  $k_0 - a_0 - c = 4$  with a matrix of size  $103,973 \times 97,980$  and a computation lasting 70 seconds. All results presented here allows to choose to shorten the system to  $k_0 - a_0 - c = 5$  instead of 4, as for 4 the system does not leads directly to linear equations, and it reduces the computation to linear algebra on a matrix of size  $2772$  by  $4284$  that last only few seconds.

## Conclusion

We have presented the link between the different modelings for the MinRank problem. This allows a more accurate understanding of the best strategy to solve MinRank instances.

We have shown that superdetermined MinRank instances are instances for which (SM) solves at  $b = 1$ , and that the maximal degree in the computations is not the best parameter to use to optimize the computation.

We have also presented the DAGS attack as a particular superdetermined MinRank one, and how the accurate study of the involved matrices allows to find the best strategy.

### 5.3.1 Acknowledgements

This work has been supported by the French ANR project CBCRYPT (ANR-17-CE39-0007).

## A Appendix

We want to reduce (11) to a MinRank problem (12). We start from (11):

$$\left( (\mathbf{I}_{k_0-c} \ \mathbf{U}) (\tilde{\mathbf{G}} \otimes \mathbf{1}_{2\gamma}) \star \mathbf{H}_{pub} \right) \mathbf{x}^\top = 0.$$

Using the fact that  $(\mathbf{A} \star \mathbf{B})\mathbf{a}^\top = 0$  is equivalent to  $(\mathbf{A} \star \mathbf{a})\mathbf{B}^\top = 0$ , that  $\mathbf{A} \otimes \mathbf{a} = \mathbf{A}(\mathbf{I} \otimes \mathbf{a})$  and  $(\mathbf{A}\mathbf{B}) \star \mathbf{a} = \mathbf{A}(\mathbf{B} \star \mathbf{a})$ , it can be rewritten

$$(\mathbf{I}_{k_0-c} \ \mathbf{U}) \tilde{\mathbf{G}}((\mathbf{I}_{n_0} \otimes \mathbf{1}_{2\gamma}) \star \mathbf{x})\mathbf{H}_{pub}^\top = 0$$

Now we can use the relations  $(\mathbf{A} \otimes \mathbf{a}) \star (\mathbf{b} \otimes \mathbf{x}) = (\mathbf{A} \star \mathbf{b}) \otimes (\mathbf{a} \star \mathbf{x})$ ,  $\tau \star \mathbf{I} = \text{Diag}(\tau)$  and  $\mathbf{A} \otimes \mathbf{a} = \mathbf{A}(\mathbf{I} \otimes \mathbf{a})$  to simplify

$$\begin{aligned} (\mathbf{I}_{n_0} \otimes \mathbf{1}_{2\gamma}) \star \mathbf{x} &= (\mathbf{I}_{n_0} \otimes \mathbf{1}_{2\gamma}) \star \left( \tau \otimes \mathbf{1}_{2\gamma} + \sum_{i=1}^{\gamma} b_i \mathbf{1}_{n_0} \otimes \mathbf{e}_i \right) \\ &= (\tau \star \mathbf{I}_{n_0}) \otimes \mathbf{1}_{2\gamma} + \sum_{i=1}^{\gamma} b_i (\mathbf{I}_{n_0} \otimes \mathbf{e}_i) \\ &= \text{Diag}(\tau)(\mathbf{I}_{n_0} \otimes \mathbf{1}_{2\gamma}) + \sum_{i=1}^{\gamma} b_i (\mathbf{I}_{n_0} \otimes \mathbf{e}_i) \end{aligned}$$

We can now define  $\tilde{\mathbf{H}}_i = \mathbf{H}_{pub}(\mathbf{I}_{n_0} \otimes \mathbf{e}_i^\top)$  and  $\tilde{\mathbf{H}} = \mathbf{H}_{pub}(\mathbf{I}_{n_0} \otimes \mathbf{1}_{2\gamma}^\top)$  and we get the system

$$\begin{aligned} (\mathbf{I}_{k_0-c} \ \mathbf{U}) \tilde{\mathbf{G}}(\text{Diag}(\tau)\tilde{\mathbf{H}}^\top + \sum_{i=1}^{\gamma} b_i \tilde{\mathbf{H}}_i^\top) &= 0, \\ (\tilde{\mathbf{H}} \text{Diag}(\tau)\tilde{\mathbf{G}}^\top + \sum_{i=1}^{\gamma} b_i \tilde{\mathbf{H}}_i \tilde{\mathbf{G}}^\top) \begin{pmatrix} \mathbf{I}_{k_0-c} \\ \mathbf{U}^\top \end{pmatrix} &= 0 \end{aligned}$$

We now simplify the products using the remark that  $\tilde{\mathbf{H}}$  is the parity-check matrix corresponding to  $\tilde{\mathbf{G}} = (\mathbf{I}_{k_0} \ \mathbf{G})$ :  $\tilde{\mathbf{H}} = (\mathbf{G}^\top \ \mathbf{I}_{n_0-k_0})$ , and that  $\tilde{\mathbf{H}}_i = (\mathbf{H}_i \ \mathbf{0}_{n_0-k_0})$  contains columns of zeros on the last  $n_0 - k_0$  positions. This gives (12).

## References

- [1] Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bros, M., Couvreur, A., Deneuville, J.C., Gaborit, P., Zémor, G., Hauteville, A.: Rank quasi cyclic (RQC). Second Round submission to NIST Post-Quantum Cryptography call (Apr 2020), <https://pqc-rqc.org>
- [2] 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>
- [3] Baena, J., Briaud, P., Cabarcas, D., Perlner, R.A., Smith-Tone, D., Verbel, J.A.: Improving support-minors rank attacks: applications to GeMSS and Rainbow. IACR Cryptol. ePrint Arch., accepted for publication in CRYPTO 2022 p. 1677 (2021), <https://eprint.iacr.org/2021/1677>
- [4] Banegas, G., Barreto, P.S., Boidje, B.O., Cayrel, P.L., Dione, G.N., Gaj, K., Gueye, C.T., Haeussler, R., Klamti, J.B., N'diaye, O., Nguyen, D.T., Persichetti, E., Ricardini, J.E.: DAGS : Key encapsulation for dyadic GS codes. <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/> (Nov 2017), first round submission to the NIST post-quantum cryptography call
- [5] Bardet, M., Bertin, M., Couvreur, A., Otmani, A.: Practical algebraic attack on DAGS. In: Baldi, M., Persichetti, E., Santini, P. (eds.) Code-Based Cryptography - 7th International Workshop, CBC 2019, Darmstadt, Germany, May 18-19, 2019, Revised Selected Papers. LNCS, vol. 11666, pp. 86–101. Springer (2019). [https://doi.org/10.1007/978-3-030-25922-8\\_5](https://doi.org/10.1007/978-3-030-25922-8_5)
- [6] Bardet, M., Briaud, P., Bros, M., Gaborit, P., Neiger, V., Ruatta, O., Tillich, J.: An algebraic attack on rank metric code-based cryptosystems. In: Advances in Cryptology - EUROCRYPT 2020 - 30th Annual International Conference on the Theory and Applications of Cryptography

- tographic Techniques, Zagreb, Croatia, May 10-14, 2020. Proceedings (2020), <http://arxiv.org/abs/1910.00810>
- [7] Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Improvements of algebraic attacks for solving the rank decoding and minrank problems. In: Advances in Cryptology - ASIACRYPT 2020, International Conference on the Theory and Application of Cryptology and Information Security, 2020. Proceedings. pp. 507–536 (2020), <https://arxiv.org/abs/2002.08322v4>
- [8] Barelli, E., Couvreur, A.: An efficient structural attack on NIST submission DAGS. In: Peyrin, T., Galbraith, S. (eds.) Advances in Cryptology - ASIACRYPT’18. LNCS, vol. 11272, pp. 93–118. Springer (Dec 2018)
- [9] Beullens, W.: Improved cryptanalysis of UOV and Rainbow. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. Lecture Notes in Computer Science, vol. 12696, pp. 348–373. Springer International Publishing (2021). [https://doi.org/10.1007/978-3-030-77870-5\\_13](https://doi.org/10.1007/978-3-030-77870-5_13)
- [10] Beullens, W.: Breaking Rainbow takes a weekend on a laptop. IACR Cryptol. ePrint Arch. (2022), <https://eprint.iacr.org/2022/214>
- [11] Bruns, W., Vetter, U.: Determinantal Rings, lncs, vol. 1327. Springer (1988)
- [12] 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)
- [13] Casanova, A., Faugère, J., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: GeMSS: A Great Multivariate Short Signature. Second round submission to the NIST post-quantum cryptography call (Apr 2019), <https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions/>
- [14] Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow. Second round submission to the NIST post-quantum cryptography call (Apr 2019), <https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions/>

- [15] Faugère, J.C., Safey El Din, M., Spacodecrenlehauer, 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)
- [16] Faugère, J.C., Safey El Din, M., Spacodecrenlehauer, P.J.: On the complexity of the generalized minrank problem. *J. Symbolic Comput.* **55**, 30–58 (2013). <https://doi.org/https://doi.org/10.1016/j.jsc.2013.03.004>
- [17] 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>
- [18] 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>
- [19] MacWilliams, F.J., Sloane, N.J.A.: *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, fifth edn. (1986)
- [20] Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*. LNCS, vol. 1070, pp. 33–48. Springer (1996). [https://doi.org/10.1007/3-540-68339-9\\_4](https://doi.org/10.1007/3-540-68339-9_4), [https://doi.org/10.1007/3-540-68339-9\\_4](https://doi.org/10.1007/3-540-68339-9_4)
- [21] Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for HFEv- based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*. LNCS, vol. 9452, pp. 311–334. Springer (2015). [https://doi.org/10.1007/978-3-662-48797-6\\_14](https://doi.org/10.1007/978-3-662-48797-6_14), [https://doi.org/10.1007/978-3-662-48797-6\\_14](https://doi.org/10.1007/978-3-662-48797-6_14)

- [22] Porras, J., Baena, J., Ding, J.: ZHFE, a new multivariate public key encryption scheme. In: Mosca, M. (ed.) Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. LNCS, vol. 8772, pp. 229–245. Springer (2014). [https://doi.org/10.1007/978-3-319-11659-4\\_14](https://doi.org/10.1007/978-3-319-11659-4_14), [https://doi.org/10.1007/978-3-319-11659-4\\_14](https://doi.org/10.1007/978-3-319-11659-4_14)
- [23] Tao, C., Petzoldt, A., Ding, J.: Efficient key recovery for all HFE signature variants. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 70–93. Springer (2021). [https://doi.org/10.1007/978-3-030-84242-0\\_4](https://doi.org/10.1007/978-3-030-84242-0_4), [https://doi.org/10.1007/978-3-030-84242-0\\_4](https://doi.org/10.1007/978-3-030-84242-0_4)
- [24] 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), [https://doi.org/10.1007/978-3-030-25510-7\\_10](https://doi.org/10.1007/978-3-030-25510-7_10)