Cryptanalysis of MinRank 2.0

Nicolas Aragon[1], Maxime Bros[2,3], Ray Perlner[5], Daniel Smith-Tone[5,6], and Javier Verbel[7]

[1] XLIM, CNRS UMR 7252, University of Limoges, France
nicolas.aragon@unilim.fr
[2] Associate, National Institute of Standards and Technology (NIST), Maryland, USA
maxime.bros@nist.gov
[3] Contractor, Izum Inc., Potomac, Maryland, USA
[4] National Institute of Standards and Technology (NIST), Maryland, USA
{ray.perlner,daniel.smith}@nist.gov
[5] University of Louisville, Louisville KY, USA
dcsmit11@louisville.edu
[6] Technology Innovation Institute, Abu Dhabi, UAE
javier.verbel@tii.ae

Abstract. MinRank is an NP-complete problem underlying the security of several post-quantum schemes. It is the hardness assumption of the NIST second-round signature candidate Mirath and of the encryption schemes of Chatterjee et al. and Debris-Alazard et al. at Eurocrypt 2026. Moreover, it is ubiquitous in the cryptanalysis of multivariate schemes, among them the third-round candidates UOV, MAYO, QR-UOV, and SNOVA.

In this work, we revisit the attacks on MinRank. On the combinatorial side, we improve the polynomial part of the kernel-search attack by sharing its linear algebra across guesses, which lowers its cost by 3 to 6 bits on the Mirath parameters and the Courtois challenges, and by up to 3 bits in combination with the hybrid approach. With the improved attack, we solved Courtois challenge D, open since 2001: an instance over $\mathbb{F}_2$ with 19×19 matrices, 81 unknowns, and target rank 10, solved in 11434 CPU days, an estimated 2^{61} bit opetations. Our multi-threaded C++ code is publicly available.

On the algebraic side, we show that the complexity of the Kipnis–Shamir (KS) attack is less understood than the literature suggests: the existing estimates rely on a generic Hilbert series that does not predict the rank of the Macaulay matrices of the KS equations. We prove a lower bound on the corank that explains the discrepancy, show that the attack requires a reduced instance with $k \leq m - r$, and derive corrected complexity estimates, which are 1 to 24 bits above the best other attack on every Mirath parameter set. We also extend the Support-Minors and Minors modelings with the field equations, which lowers their estimates over small fields: for $q = 2$ and $m = n = 22$, by 9 to 205 bits for Support-Minors and by 17 to 278 bits for Minors. The rank predictions behind the algebraic estimates are stated as assumptions and verified experimentally.

Keywords: MinRank · Courtois challenge · Kernel search · Kipnis–Shamir · Support-Minors

1 Introduction

The MinRank problem was introduced and proven NP-complete by Buss, Frandsen, and Shallit [12], and was further explored by Courtois [18]. Given matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k$ over a finite field and a target rank r, MinRank asks for a linear combination $\mathbf{M}_0 + x_1\mathbf{M}_1 + \cdots + x_k\mathbf{M}_k$ of rank at most r. Early on, MinRank was used both in the design of cryptographic schemes [17] and in cryptanalysis, starting with the attacks on the multivariate HFE cryptosystem [38,35]: for many such schemes, the secret key can be recovered from a low-rank combination of matrices derived from its public key, so solving a MinRank instance breaks the scheme, and the cost of the best MinRank algorithm bounds its security. Both uses continue today. MinRank is the hardness assumption of the second-round NIST PQC signature candidate Mirath [1] and of two public-key encryption schemes presented at Eurocrypt 2026, by Chatterjee, Mu, and Vasudevan [15] and by Debris-Alazard, Gaborit, Neveu, and Ruatta [19]. It is also ubiquitous in the cryptanalysis of multivariate and code-based schemes, since Rank Decoding reduces to MinRank [5].

Many multivariate schemes have been broken with a MinRank attack, most notably GeMSS [3] and the NIST finalist Rainbow [21]. All multivariate candidates in the third round of the NIST PQC standardization process (UOV [10], MAYO [9], QR-UOV [29], SNOVA [44]) feature MinRank in their security analysis, so any improvement in the complexity of solving MinRank directly affects their security estimates.

Progress on MinRank is naturally measured by comparing the complexity estimates of the available algorithms, but solving concrete challenges shows what is feasible in practice. Courtois proposed six challenges in 2001 [17], denoted A to F, whose parameters are given in Table 1. Challenges A and B were solved by

Table 1: Courtois MinRank challenges proposed in 2001 [17].

Challenge	MinRank parameters				
	q	m	n	k	r
A	65521	6	6	10	3
B	65521	7	7	10	4
C	65521	11	11	10	8
D	2	19	19	81	10
E	2	21	21	121	10
F	2	29	29	190	15

Faugère, Levy-dit-Vehel, and Perret in 2008 [28], who also refined the complexity estimates for challenge C. Challenge C was solved in 2010 by Faugère, Safey El Din, and Spaenlehauer [27], and again in 2015 by Steel [41] with an improved implementation. Challenges D, E, and F have remained unsolved for 25 years.

The algorithms for solving MinRank, also called attacks, fall into two categories: combinatorial attacks, which contain a guessing step of exponential cost, and algebraic attacks, which model the problem as a system of polynomial equations and then solve it. The main combinatorial attacks are brute-force, enumeration, and kernel search [17,31], and the main algebraic attacks are Minors [26], Kipnis–Shamir (KS) [35], and Support-Minors [6]. The cost of an algebraic attack is governed by the rank of a large structured matrix, the Macaulay matrix of the system, and that rank is predicted by a Hilbert series of the system; these predictions are heuristic and must be verified experimentally. It is often beneficial to guess part of the solution combinatorially and to solve the remaining, smaller instance with any other attack; such combinations are called *hybrid* methods [5]. Which approach is the cheapest depends on the parameters.

Tools called *estimators* compute the complexity of the best known attacks on post-quantum hard problems, for instance [2] for the Learning With Errors problem, [22] for the Syndrome Decoding problem, and [8] for the Multivariate Quadratic problem. The CryptographicEstimators library [23] includes a MinRank estimator.

Our contributions. In this paper, we improve both combinatorial and algebraic attacks on MinRank, and we solve Courtois challenge D [17] in practice.

Improved kernel search. In Section 4.4, we improve the polynomial part of the kernel-search attack by sharing the row reduction of its linear system among the q^r guesses of the last kernel vector (Algorithm 5). On the Mirath parameters [1] and the Courtois challenges, the improved attack is 3 to 6 bits cheaper than the original on its own, and up to 3 bits cheaper in combination with the hybrid approach of [5] (Table 2).

Courtois challenge D. With the improved kernel search and the hybrid approach, guessing five entries of the solution vector, we solved Courtois challenge D ($q = 2$, $m = n = 19$, $k = 81$, $r = 10$) in 11434 CPU days; the estimated cost is 2^{61} $\mathbb{F}_2$-multiplications with $\omega = 3$ (Table 3). Our multi-threaded C++ code is publicly available (Section 4.5).

Field equations in Support-Minors and Minors. We extend the analysis of both modelings by adding the field equations of the linear variables to the ideals they generate. For Support-Minors (Section 5), this extends the analysis of Bardet and Gilard [7] to any $q \geq 2$: for $q = 2$, $m = n = 22$, and $k = (m-r)(n-r)-2$, our estimate is 9 to 205 bits below theirs, and the attack applies for all $1 \leq r \leq 20$, whereas the original Support-Minors attack [6] applies only for $r = 20$ (Figure 1). For Minors (Section 6), the same extension of the analysis of Gopalakrishnan, Neiger, and Safey El Din [30] gives a cheaper attack over small fields, $q \in \{2, 3, 4\}$. Assumptions 1 and 2 state the rank predictions behind both estimates and are tested for $q \in \{2, 3, 4\}$.

Kipnis–Shamir. In Section 7, we revisit the KS attack, in which the KS equations are solved by multi-homogeneous XL. We show that the generic multi-graded Hilbert series used by Ning, Ran, and Samardjiska [37] to predict the rank of the Macaulay matrices of the KS equations does not apply to KS systems (Table 4), because of the syzygies of Verbel et al. [43], which generic systems do not

have. We prove a lower bound on the corank of these matrices (Theorem 1), estimate it by a Hilbert series, exact in all our experiments (Table 5), and derive a corrected selection of the multi-degree (Assumption 3). The multi-degrees selected by the generic prediction fail on three Mirath parameter sets; with the corrected selection, its estimate is 1 to 27 bits above that of the best other attack (Table 6).

In addition, we incorporated the new complexity estimates in the MinRank Estimator of the CryptographicEstimators library [23]. Overall, all the results in this paper provide theoretical tools for evaluating the security of MinRank-based schemes.

Organization of the paper. Section 2 fixes the notation, defines the MinRank problem, and declares the complexity model. Section 3 recalls the hybrid approach. Section 4 describes the combinatorial attacks and presents the improved kernel search, and Section 4.5 describes our implementation and the solution of Courtois challenge D. Section 5 analyzes the Support-Minors attack with field equations. Section 6 does the same for the Minors attack. Section 7 revisits the Kipnis–Shamir attack. Finally, Section 8 concludes with open problems.

2 Preliminaries

2.1 Notation

Throughout, $\mathbb{F}_q$ denotes the field with q elements; bold uppercase and lowercase letters denote matrices and vectors, respectively. We write $\mathbf{I}_t$ for the $t \times t$ identity matrix and $\mathbf{0}^{u\times v}$ for the $u \times v$ zero matrix. The symbols a, l_v, n', b, c, $\mathbf{D}$, and $\mathbf{d}$ are reserved as specified below: a is the number of kernel vectors guessed by a hybrid attack, l_v is the number of solution-vector entries guessed, n' is the number of columns retained in a Support-Minors modeling, b is the degree in the linear variables in the Support-Minors Macaulay modeling, c is the degree in the minor variables in the Support-Minors Macaulay modeling or the number of kernel blocks in a multi-homogeneous XL modeling, $\mathbf{D}$ is a local reduced-row-echelon-form matrix in algorithm descriptions, and $\mathbf{d} = (d_0, \ldots, d_c)$ is a multi-degree. An internal MinRank solver returns either a vector satisfying the stated rank bound or $\perp$; thus, "success" means returning and verifying such a vector, whereas "failure" means returning $\perp$.

2.2 The MinRank Problem

We use the following affine and homogeneous conventions.

Definition 1 (MinRank problem (affine version)). *The* MinRank *problem with parameters* (q, m, n, k, r) *is defined by* $k+1$ *matrices* $\mathbf{M}_0, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{m\times n}$. *The goal is to find a vector* $\mathbf{s} := (x_1, \ldots, x_k) \in \mathbb{F}_q^k$, *such that* $\mathrm{Rank}\,(\mathbf{\Sigma_x}) \leq r$, *where* $\mathbf{\Sigma_x} := \mathbf{M}_0 + \sum_{i=1}^{k} x_i \mathbf{M}_i$.

Definition 2 (MinRank problem (homogeneous version)). *The homogeneous MinRank problem is the special case of the affine MinRank problem where* $\mathbf{M}_0 = \mathbf{0}^{m\times n}$ *and the vector* $\mathbf{s} := (x_1, \ldots, x_k) \in \mathbb{F}_q^k$ *has to be nonzero.*

The affine version can be transformed into an instance of the homogeneous version by adding one extra unknown $x_0 \in \mathbb{F}_q$, and discarding any solutions with $x_0 = 0$. Thus, any analysis of random instances of the homogeneous version applies to random instances of the affine version, with $k + 1$ variables x_i.

We may assume that the target matrix has rank exactly r: one can solve successively for ranks $1, 2, \ldots, r$, and the final step dominates the cost. In addition, cryptographic MinRank instances are typically generated with a unique solution of rank exactly r.

Remark 1 (Special structure of the kernel matrix). Throughout the paper, we assume that a rank-r solution matrix $\mathbf{\Sigma_x}$ is spanned by its last r columns. With high probability, any MinRank instance can be put in this form [11, Proposition 1]. Consequently, there is a matrix $\mathbf{K} \in \mathbb{F}_q^{r\times(n-r)}$ such that

$$\mathbf{\Sigma_x} \begin{bmatrix} \mathbf{I}_{n-r} \\ \mathbf{K} \end{bmatrix} = \mathbf{0}^{m\times(n-r)}.$$

2.3 Complexity Model

We work in the Random Access Machine (RAM) model and report complexities as $\log_2$ of the number of $\mathbb{F}_q$-multiplications unless stated otherwise; bit operations (gates) are obtained with the conversion factor below. The following expressions count $\mathbb{F}_q$-multiplications; nrows and ncols denote the numbers of rows and columns of the matrix, and $\mathsf{density}$ the average number of nonzero entries per row:

– *Wiedemann* [34,45]. Given a sparse singular square matrix of size ncols, the Wiedemann algorithm finds a nonzero vector in its right kernel. Its cost is

$$3 \cdot \mathsf{density} \cdot \mathsf{ncols}^2. \tag{1}$$

For a Macaulay matrix with more rows than columns, we first select ncols of its rows; for the extended linearization (XL) algorithm [16], a random selection suffices. For the algorithms presented in this work, we assume that such a selection algorithm exists.

– *Row echelon form* [33]. The reduced row echelon form of a matrix of rank ρ, together with its rank profile, can be computed with $O(\mathsf{nrows} \cdot \mathsf{ncols} \cdot \rho^{\omega-2})$ field operations [33]. For full-rank matrices, where $\rho = \min(\mathsf{nrows}, \mathsf{ncols})$, we use the cost

$$\max(\mathsf{nrows}, \mathsf{ncols}) \cdot \min(\mathsf{nrows}, \mathsf{ncols})^{\omega-1}. \tag{2}$$

– *Strassen* [42]. Gaussian elimination on a dense square matrix of size ncols based on Strassen's matrix multiplication, with $\omega = \log_2 7 \approx 2.81$, costs

$$7 \cdot \mathsf{ncols}^{\omega}. \tag{3}$$

To convert every displayed $\mathbb{F}_q$-multiplication count to bit operations (gates), we multiply it by $2(\log_2 q)^2 + \log_2 q$, as in the third-round Rainbow submission [20, Section 8.2]: a schoolbook multiplication in $\mathbb{F}_q$ costs $(\log_2 q)^2$ AND gates and as many XOR gates, and the addition that follows it costs $\log_2 q$ XOR gates. This is the conversion factor used by most schemes in the third round of the NIST standardization process.

3 Hybrid Approach

The hybrid approach enumerates l_v entries of the affine solution vector $\mathbf{s}$ and a kernel vectors of $\mathbf{\Sigma_s}$. For each guess, it derives a smaller affine MinRank instance with parameters $(q, m, n-a, k-am-l_v, r)$, which is solved using a MinRank solver denoted MRSolver. This reduction is correct provided that the guess agrees with a solution and that MRSolver returns only verified solutions.

Algorithm 1 Hybrid Approach

HybridMRSolver$(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, l_v, a)$

1 : **for** $c_1, \ldots, c_{l_v} \in \mathbb{F}_q^{l_v}$
2 : $\quad \mathbf{M}'_0 = \mathbf{M}_0 + c_1\mathbf{M}_{k-l_v+1} + \cdots + c_{l_v}\mathbf{M}_k$
3 : $\quad$ **for** $\mathbf{k}_1, \ldots, \mathbf{k}_a \in \mathbb{F}_q^r$
4 : $\quad\quad \mathbf{M} \leftarrow$ ProcessGuess$(\mathbf{M}'_0, \ldots, \mathbf{M}_{k-l_v}, \mathbf{k}_1, \ldots, \mathbf{k}_a)$
5 : $\quad\quad$ **if** $\mathbf{M} \neq \perp$
6 : $\quad\quad\quad (\mathbf{N}_0, \ldots, \mathbf{N}_{k-l_v-am}) \leftarrow \mathbf{M}$
7 : $\quad\quad\quad \mathbf{s}' \leftarrow$ MRSolver$(\mathbf{N}_0, \ldots, \mathbf{N}_{k-l_v-am}, r)$
8 : $\quad\quad\quad$ **if** $\mathbf{s}' \neq \perp$
9 : $\quad\quad\quad\quad \mathbf{s} \leftarrow$ ExpandSolution$(\mathbf{s}', \mathbf{k}_1, \ldots, \mathbf{k}_a) \parallel (c_1, \ldots, c_{l_v})$
10 : $\quad\quad\quad\quad$ **return** $\mathbf{s}$
11 : **return** $\perp$

Algorithm 1 describes the MinRank hybrid approach named HybridMRSolver. It internally uses an algorithm MRSolver, which can be any algorithm that solves the MinRank problem on an instance $\mathbf{N}_0, \ldots, \mathbf{N}_{k-l_v-am}, r$. HybridMRSolver applies the enumeration of linear variables at step 2, while the enumerated kernel vectors are applied by using the ProcessGuess algorithm at step 4. The ProcessGuess algorithm is provided in Algorithm 2, which, on input $\mathbf{k}_1, \ldots, \mathbf{k}_a$ with $a < \lceil (k-l_v)/m \rceil$, outputs either $\perp$ or a set of $k-am-l_v+1$ matrices in $\mathbb{F}_q^{m \times (n-a)}$. These matrices, along with the input integer r, define a MinRank problem to be solved subsequently with the MRSolver algorithm. If a solution $\mathbf{s}'$ is found, then it is expanded to a solution to the input problem using the currently enumerated coefficients and vectors $c_1, \ldots, c_{l_v}, \mathbf{k}_1, \ldots, \mathbf{k}_a$. ExpandSolution performs this last step efficiently by

recomputing the polynomials ℓ_i defined by $\mathbf{k}_1, \ldots, \mathbf{k}_a$ and appending the enumerated coefficients $c_1, \ldots, c_{l_v}$ as the last l_v coordinates of $\mathbf{s}$.

Algorithm 2 Process input guess of kernel vectors

ProcessGuess($\mathbf{M}_0, \ldots, \mathbf{M}_{k'}, \mathbf{k}_1, \ldots, \mathbf{k}_a$)

$\mathbf{\Sigma_x} = \mathbf{M}_0 + x_1\mathbf{M}_1 + \cdots + x_{k'}\mathbf{M}_{k'}$

// Let $\mathbf{e}_j \in \mathbb{F}_q^{n-r}$ be the j-th canonical vector.

$\mathcal{L} = \left[\mathbf{\Sigma_x} \cdot \begin{bmatrix}\mathbf{e}_1\\ \mathbf{k}_1\end{bmatrix} | \cdots | \mathbf{\Sigma_x} \cdot \begin{bmatrix}\mathbf{e}_a\\ \mathbf{k}_a\end{bmatrix}\right] \in (\mathbb{F}_q[x_1, \ldots, x_{k'}])^{m \times a}$

// Let $\mathbf{C} \in \mathbb{F}_q^{am \times (k'+1)}$ be the coefficient matrix of the entries of $\mathcal{L}$ in lexicographic order.

$\mathbf{D} = \mathrm{RREF}(\mathbf{C})$ // The reduced row echelon form; $\widehat{\mathbf{D}}$ denotes $\mathbf{D}$ without its last column.

if $a = \lceil k'/m \rceil$ **then** // Only applies to the kernel-search attack.

  if $\mathsf{rank}(\widehat{\mathbf{D}}) \neq \mathsf{rank}(\mathbf{D})$ **then return** $\perp$ // Inconsistent system: wrong guess.

  for $(s_1, \ldots, s_{k'}) \in \mathbb{F}_q^{k'}$ such that $\mathbf{D} \cdot (s_1, \ldots, s_{k'}, 1)^\top = 0$

    if $\mathsf{rank}\big(\mathbf{M}_0 + \sum_{i=1}^{k'} s_i\mathbf{M}_i\big) \leq r$ **then**

      return $(s_1, \ldots, s_{k'})$

  return $\perp$

if $a < \lceil k'/m \rceil$ **then** // Only applies to the hybrid approach.

  if $\mathsf{rank}(\widehat{\mathbf{D}}) \neq am$ **then return** $\perp$

// Let $P = \{p_1, \ldots, p_{am}\}$ be the pivots of $\mathbf{D}$, and set $\{j_1, \ldots, j_{k'-am}\} = [k'] \setminus P$.

// Without loss of generality, assume $P = \{1, \ldots, am\}$.

// Let $x_i + \ell_i(x_{am+1}, \ldots, x_{k'})$ be the polynomial corresponding to the i-th row of $\mathbf{D}$.

  $\mathbf{\Sigma_x} = \mathbf{M}_0 - \sum_{i=1}^{am} \ell_i(x_{am+1}, \ldots, x_{k'})\mathbf{M}_i + \sum_{j=1}^{k'-am} x_{am+j}\mathbf{M}_{am+j}$

// At this point the first a columns of $\mathbf{\Sigma_x}$ lie in the span of its last r columns,

// so dropping them preserves the rank (as $a \leq n - r$).

// Let $\mathbf{\Sigma}'_\mathbf{x}$ be the submatrix of $\mathbf{\Sigma_x}$ formed by its last $n - a$ columns .

// Let $\mathbf{M}'_0, \ldots, \mathbf{M}'_{k'-am} \in \mathbb{F}_q^{m \times (n-a)}$ be such that $\mathbf{\Sigma}'_\mathbf{x} = \mathbf{M}'_0 + x_{am+1}\mathbf{M}'_1 + \cdots + x_{k'}\mathbf{M}'_{k'-am}$.

  return $\mathbf{M}'_0, \ldots, \mathbf{M}'_{k'-am}$

The complexity of Algorithm 1 is given by

$$q^{a \cdot r + l_v}\Big(\mathcal{C}_{\mathrm{MR}}(q, m, n-a, k-am-l_v, r) + \mathtt{cpsi}\Big). \tag{4}$$

Here, $\mathcal{C}_{\mathrm{MR}}(q, m, n-a, k-am-l_v, r)$ denotes the complexity of the internal MRSolver on the reduced instance with parameters $(q, m, n-a, k-am-l_v, r)$, and `cpsi` is the cost to compute the input matrices of that reduced instance for each enumeration of a kernel vectors and l_v coefficients.

We highlight that for $a \approx k/m$, the complexity of solving the reduced MinRank problem may be smaller than the complexity of computing the reduced input matrices, i.e., $\mathcal{C}_{\mathrm{MR}}(q, m, n-a, k-am-l_v, r) < \mathtt{cpsi}$. Hence, a precise estimate of $\mathtt{cpsi}$ yields a precise estimate of the complexity of Algorithm 1.

The total cost of computing all q^{l_v+ar} reduced MinRank problems is

$$q^{l_v}\left(\mathtt{cpsi}_{lin} + q^{ar}\mathtt{cpsi}_{ker}\right),$$

where $\mathtt{cpsi}_{lin}$ is the cost of step 2 and $\mathtt{cpsi}_{ker}$ the cost of ProcessGuess at step 4. The cost of ProcessGuess for $a < \lceil k'/m \rceil$ is given by

$$\mathtt{cpsi}_{ker} := \underbrace{a \cdot (k'+1) \cdot (rm)}_{\text{step 2.}} + \underbrace{(k'+1) \cdot (am)^{\omega-1}}_{\text{step 3.}} + \underbrace{am \cdot (k'-am+1) \cdot m(n-a)}_{\text{step 12.}},$$

where $k' = k - l_v$. The case $a = \lceil k/m \rceil$, which only occurs in the kernel-search attack, is treated in Section 4.3. Since $\mathtt{cpsi}_{lin}$ is amortized over the q^{ar} kernel guesses, $\mathtt{cpsi}_{ker}$ is a good estimate of $\mathtt{cpsi}$, and we set $\mathtt{cpsi} = \mathtt{cpsi}_{ker}$ in (4). Our description of the hybrid approach is a reformulation of the approach introduced in [5].

4 Combinatorial Attacks

Combinatorial attacks are attacks in which the attacker enumerates all possibilities for a subset of the unknown values before trying to solve the input problem. These attacks are also called *enumeration* attacks or, when the remaining system is linear, *linear algebra search* attacks.

4.1 Brute Force over the Solution Vector

A naive way to solve the MinRank problem is to enumerate all vectors in $\mathbb{F}_q^k$. For each vector $\mathbf{s} = (s_1, \ldots, s_k)$ one checks whether the rank of the corresponding matrix $\mathbf{\Sigma_s}$ is at most r. This method is particularly relevant for the small values of k that appear in the hybrid approach when the hybridization parameter a is large (see Algorithm 1).

The BruteForce algorithm is presented in Algorithm 3. It applies an early-abort strategy by computing several minors of a candidate low-rank matrix $\mathbf{\Sigma_s}$. If one of the $(r+1)$-sized minors of $\mathbf{\Sigma_s}$ is nonzero, then $\mathsf{rank}(\mathbf{\Sigma_s}) > r$, so we discard $\mathbf{s}$ and continue with the next vector.

We estimate the average number of times BruteForce computes a minor of a matrix $\mathbf{\Sigma_s}$ at step 4. To this end, we model each square submatrix of $\mathbf{\Sigma_s}$ of size $r+1$ as a uniformly random matrix. We define X as the random variable that gives the number of times one has to compute a minor at step 4 to find a nonzero m. We obtain

$$\mathbb{E}(X) = \Big(\prod_{i=1}^{r+1}(1-q^{-i})\Big)^{-1}.$$

Algorithm 3 Brute-force the solution vector $\mathbf{s}$

BRUTEFORCE$(\mathbf{M}_0, \ldots, \mathbf{M}_k, r)$

1 : **for** $(s_1, \ldots, s_k) \in \mathbb{F}_q^k$
2 : $\quad \mathbf{\Sigma_s} = \mathbf{M}_0 + s_1\mathbf{M}_1 + \cdots + s_k\mathbf{M}_k$
3 : $\quad$ **for** $i = 1, \ldots,$ `max` // `max` is a small constant; see below.
4 : $\quad\quad$ m $=$ Minor$(\mathbf{\Sigma_s}, r+1, i)$ // The i-th minor of $\mathbf{\Sigma_s}$ of size $r+1$.
5 : $\quad\quad$ **if** m $\neq 0$
6 : $\quad\quad\quad$ go to step 1
7 : $\quad$ **if** rank$(\mathbf{\Sigma_s}) \leq r$
8 : $\quad\quad$ **return** $\mathbf{s}$
9 : $\quad$ **else**
10 : $\quad\quad$ go to step 1
11 : **return** $\perp$

Indeed, there are $\prod_{i=0}^{r}(q^{r+1} - q^i)$ invertible matrices in $\mathbb{F}_q^{(r+1)\times(r+1)}$, so a uniformly random one has a nonzero determinant with probability $p = \prod_{i=1}^{r+1}(1 - q^{-i})$. Under the heuristic that the successive minors behave as independent random determinants, X is geometric with parameter p, and $\mathbb{E}(X) = \sum_{j\geq 1} j\,p\,(1-p)^{j-1} = 1/p$. This is below 3.47 for $q = 2$ and tends to $q/(q-1)$ as q grows.

For each $\mathbf{s}$, we compute on average about 3.5 minors of size $r+1$ when $q = 2$ (fewer for larger q) before discarding it. That is, most of the time we need only a few $(r+1)$-sized square submatrices of $\mathbf{\Sigma_s}$, so we compute their entries at step 4 instead of the whole matrix at step 2. In addition, we assume that the cost of computing one of these submatrices can be amortized from $k(r+1)^2$ to $O(r^2)$ by reusing precomputed products of the form $s_i\mathbf{M}_i$. Therefore, for each $\mathbf{s}$, the expected cost of checking whether rank$(\mathbf{\Sigma_s}) \leq r$ is dominated by the computation of $\mathbb{E}(X)$ minors, each costing $(r+1)^\omega$. The amortized complexity of BRUTEFORCE is then

$$\mathcal{C}_{\mathrm{BF}} = \mathbb{E}(X) \cdot q^k \cdot (r+1)^\omega.$$

We take `max` to be a small constant, larger than $\mathbb{E}(X)$ and large enough that the expected cost of the full-rank computation at step 7, reached with probability $(1-p)^{\texttt{max}}$, is negligible.

4.2 Enumeration of the Entries of the Low-Rank Matrix (Big-k Attack)

Every correctly guessed or known entry of $\mathbf{\Sigma_x} = \mathbf{M}_0 + \sum_{i=1}^{k} x_i\mathbf{M}_i$ yields a linear equation in the entries of $\mathbf{s}$. This big-k attack (originally the big-m attack in [17]) is particularly effective when $k \geq m(n-r)$, in which case $n-r$ columns of $\mathbf{\Sigma_x}$ can be set to zero directly, or when k is only slightly smaller than $m(n-r)$. Its

heuristic complexity estimate is $q^{\max\{0,m(n-r)-k\}} \cdot (m(n-r))^{\omega}$. The $m(n-r)$ threshold counts entries whose guesses yield independent linear conditions; this independence is a generic-instance heuristic.

4.3 Original Kernel Search

The kernel-search attack introduced in [17,31] and described in Algorithm 4 can be seen as the corner case of the hybrid approach (Algorithm 1) where $a = \lceil k/m \rceil$ and $l_v = 0$. The cost estimate for the kernel-search attack is

$$\mathcal{C}_{\text{Kernel}} = q^{\lceil \frac{k}{m} \rceil \cdot r} \cdot \left(\texttt{cpsi}_{ker} + q^{-(am-k)} \cdot \mathcal{C}_{\text{check}} \right),$$

where $a = \lceil k/m \rceil$, $\texttt{cpsi}_{ker} = a \cdot (k+1) \cdot (rm) + \max(am, k+1) \cdot \min(am, k+1)^{\omega-1}$ is the cost of steps 2 and 3 of Algorithm 2, and $\mathcal{C}_{\text{check}} = kmn + \max(m,n) \cdot \min(m,n)^{\omega-1}$ is the cost of computing $\mathbf{M}_0 + \sum_{i=1}^{k} s_i \mathbf{M}_i$ and its rank (see (2)) at step 7. Under the heuristic that the coefficient matrix $\mathbf{C}$ of a wrong guess of $\mathbf{k}_1, \ldots, \mathbf{k}_a$ is uniformly random, the system $\mathbf{C} \cdot (x_1, \ldots, x_k, 1)^{\top} = 0$ has on average $q^{-(am-k)}$ solutions, each of which is rank-checked at step 7. The cost of these checks is negligible when $am - k$ is large, but not when $m \mid k$, in which case $am = k$ and a wrong guess yields on average one candidate solution.

Algorithm 4 Kernel Search

KernelSearch$(\mathbf{M}_0, \ldots, \mathbf{M}_k, r)$

1: **for** $\mathbf{k}_1, \ldots, \mathbf{k}_{\lceil k/m \rceil} \in \mathbb{F}_q^r$
2: $\quad \mathbf{s} \leftarrow$ ProcessGuess$(\mathbf{M}_0, \ldots, \mathbf{M}_k, \mathbf{k}_1, \ldots, \mathbf{k}_{\lceil k/m \rceil})$
3: $\quad$ **if** $\mathbf{s} \neq \perp$
4: $\quad\quad$ **return** $\mathbf{s}$
5: **return** $\perp$

4.4 Improved Kernel Search

Both the kernel-search attack and the hybrid approach try to find a vectors in the kernel of the secret low-rank matrix $\mathbf{\Sigma_s}$ defined by a MinRank problem with parameters (q, m, n, k, r). The former targets $a = \lceil \frac{k}{m} \rceil$ linearly independent vectors in the right kernel of $\mathbf{\Sigma_s}$, while the latter targets only $a < \lceil \frac{k}{m} \rceil$ of them, where a is an input. Here, we propose an improved kernel-search attack.

The kernel-search attack processes each guess $(\mathbf{k}_1, \ldots, \mathbf{k}_a)$ independently, solving from scratch the linear system $\mathcal{L} = \mathbf{C}\,(x_1, \ldots, x_k, 1)^{\top}$ of am equations in k unknowns. This system, however, consists of a blocks of m equations, and the j-th block depends only on $\mathbf{k}_j$. Our improvement exploits this by enumerating the guesses in a nested order: we fix $\mathbf{k}_1, \ldots, \mathbf{k}_{a-1}$, build the first $a-1$ blocks and

reduce them once (steps 2 and 3 of Algorithm 5); then, for each of the q^r choices of $\mathbf{k}_a$, only the m new equations are reduced against the echelon form already computed (step 6). The expensive reduction is thus shared by the q^r guesses of the last vector, and the work per guess reduces to processing a single block.

Algorithm 5 Improved processing of a guess of kernel vectors

$\textsc{ProcessGuess}_{\text{IMP}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, \mathbf{k}_1, \ldots, \mathbf{k}_{a-1})$

1 : $\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{M}_0 + x_1\mathbf{M}_1 + \cdots + x_k\mathbf{M}_k$

// Let $\mathbf{e}_j \in \mathbb{F}_q^{n-r}$ be the j-th canonical vector.

2 : $\mathcal{L} = \left[\mathbf{\Sigma}_{\mathbf{x}} \cdot \begin{bmatrix}\mathbf{e}_1\\ \mathbf{k}_1\end{bmatrix} | \cdots | \mathbf{\Sigma}_{\mathbf{x}} \cdot \begin{bmatrix}\mathbf{e}_{a-1}\\ \mathbf{k}_{a-1}\end{bmatrix}\right] \in (\mathbb{F}_q[x_1, \ldots, x_k])^{m\times(a-1)}$

// Set $\mathbf{C}' \in \mathbb{F}_q^{(a-1)m\times(k+1)}$ satisfying $\mathcal{L} = \mathbf{C}'(x_1, \ldots, x_k, 1)^\top$

3 : $\mathbf{D}' = \mathrm{RREF}(\mathbf{C}')$ // The reduced row echelon form.

4 : **for** $\mathbf{k}_a \in \mathbb{F}_q^r$

5 : $\quad \mathcal{L}_a = \mathbf{\Sigma}_{\mathbf{x}} \cdot \begin{bmatrix}\mathbf{e}_a\\ \mathbf{k}_a\end{bmatrix}$ // Set $\mathbf{C}_a \in \mathbb{F}_q^{m\times(k+1)}$ satisfying $\mathcal{L}_a = \mathbf{C}_a(x_1, \ldots, x_k, 1)^\top$.

6 : $\quad \mathbf{D} = \mathrm{RREF}\left(\begin{bmatrix}\mathbf{D}'\\ \mathbf{C}_a\end{bmatrix}\right)$ // $\widehat{\mathbf{D}}$ denotes $\mathbf{D}$ without its last column.

7 : $\quad$ **if** $\mathsf{rank}(\widehat{\mathbf{D}}) = \mathsf{rank}(\mathbf{D})$ **then**

8 : $\quad\quad$ **for** $(s_1, \ldots, s_k) \in \mathbb{F}_q^k$ such that $\mathbf{D} \cdot (s_1, \ldots, s_k, 1)^\top = 0$

9 : $\quad\quad\quad$ **if** $\mathsf{rank}\big(\mathbf{M}_0 + \sum_{i=1}^k s_i\mathbf{M}_i\big) \le r$ **then**

10 : $\quad\quad\quad\quad$ **return** $(s_1, \ldots, s_k)$

11 : **return** $\perp$

Algorithm 6 Improved Kernel Search

$\textsc{KernelSearch}_{\text{IMP}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, r)$

1 : **for** $\mathbf{k}_1, \ldots, \mathbf{k}_{\lceil k/m\rceil-1} \in \mathbb{F}_q^r$

2 : $\quad \mathbf{s} \leftarrow \textsc{ProcessGuess}_{\text{IMP}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, \mathbf{k}_1, \ldots, \mathbf{k}_{\lceil k/m\rceil-1})$

3 : $\quad$ **if** $\mathbf{s} \neq \perp$

4 : $\quad\quad$ **return** $\mathbf{s}$

5 : **return** $\perp$

$\textsc{ProcessGuess}_{\text{IMP}}$ is presented in Algorithm 5. Its complexity is given by

$$\mathcal{C}_{\text{PG}_{\text{IMP}}} = \mathcal{C}_{\text{step-2}} + \mathcal{C}_{\text{step-3}} + q^r \cdot \left(\mathcal{C}_{\text{step-5}} + \mathcal{C}_{\text{step-6}} + q^{-(am-k)} \cdot \mathcal{C}_{\text{step-9}}\right),$$

where $a = \lceil k/m \rceil$ and $\mathcal{C}_{\text{step-i}}$ denotes the complexity of one execution of the i-th step in Algorithm 5. We have

$$\begin{aligned}
\mathcal{C}_{\text{step-2}} &= (a-1)(k+1)rm \\
\mathcal{C}_{\text{step-3}} &= (k+1)((a-1)m)^{\omega-1} \\
\mathcal{C}_{\text{step-5}} &= (k+1)rm \\
\mathcal{C}_{\text{step-6}} &= O(\underbrace{m \cdot (a-1)m \cdot (k+1-m(a-1))}_{\text{Annihilate the first } (a-1)m \text{ columns of } \mathbf{C}_a}) \\
\mathcal{C}_{\text{step-9}} &= kmn + \max(m,n) \cdot \min(m,n)^{\omega-1}
\end{aligned}$$

The rank check at step 9 is executed once for each solution of the system $\mathbf{D} \cdot (x_1, \ldots, x_k, 1)^\top = 0$. Under the same heuristic as for $\mathcal{C}_{\text{Kernel}}$, a wrong guess of $\mathbf{k}_1, \ldots, \mathbf{k}_a$ yields on average $q^{-(am-k)}$ such solutions, hence the term $q^{-(am-k)}\,\mathcal{C}_{\text{step-9}}$ in $\mathcal{C}_{\text{PG}_{\text{IMP}}}$. This term is negligible when $am - k$ is large, but not when $m \mid k$, in which case $am = k$.

The cost estimate for the improved kernel-search attack is given by

$$\mathcal{C}_{\text{Kernel}_{\text{imp}}} = q^{\left(\lceil \frac{k}{m} \rceil - 1\right)r} \cdot \mathcal{C}_{\text{PG}_{\text{IMP}}}$$

For $\lceil k/m \rceil = 1$, there are no kernel vectors to fix in advance, and we use $\mathcal{C}_{\text{Kernel}}$ instead.

Table 2 compares the original and improved kernel-search attacks (Kernel and $\text{Kernel}_{\text{imp}}$), on their own and combined with the hybrid approach, with $\omega = 2.81$; the combination of the hybrid approach with Kernel is the strategy used to estimate the security of Mirath [1]. The value of a is given in parentheses; the optimum has $l_v = 0$ in all cases except Courtois challenge D ($q = 2$, $m = n = 19$, $k = 81$, $r = 10$), where both hybrid variants take $a = 3$ and $l_v = 5$, so that the reduced instance has $k - am - l_v = 19 = m$ variables and is solved with a single kernel vector. $\text{Kernel}_{\text{imp}}$ is consistently cheaper than Kernel: by 3 to 6 bits on its own, and by 2 to 3 bits in the hybrid combination for the Mirath parameters over $\mathbb{F}_2$, which gives the lowest estimates of this section; for the other parameter sets the hybrid cost is dominated by the guessing step and both variants coincide.

4.5 Implementation and Courtois Challenge D

We implemented a multi-threaded version of our improved kernel search as specified in Algorithm 6 and used it to solve an instance of Courtois challenge D [17]. This challenge had remained unsolved for more than two decades, whereas challenges A, B, and C were solved fairly quickly in [27,28]. We fixed the parameters $q = 2, m = n = 19, k = 81$ and solved instances with $r = 7, 8, 9, 10$ to observe the complexity growth from one value of r to another. Table 3 reports the timings: we solved Courtois challenge D in 11434 CPU days, that is about 22 days of wall-clock time on 512 cores for this instance. All the reported computations were performed on Intel® Xeon® Gold 6130 CPUs[7] at 2.1 GHz.

[7] Certain commercial software is identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National In-

q	m	n	k	r	Kernel	$\text{Kernel}_{\text{imp}}$	Hybrid	
							Kernel (a)	$\text{Kernel}_{\text{imp}}$ (a)
2	42	42	1443	4	169	165	159 (32)	157 (31)
2	50	50	2024	5	235	230	225 (38)	222 (38)
2	56	56	2499	6	301	295	290 (43)	288 (42)
16	16	16	143	4	164	159	155 (8)	155 (8)
16	19	19	195	5	241	235	230 (10)	230 (10)
16	22	22	255	6	310	304	300 (11)	300 (11)
2	19	19	81	10	68	64	59 (3)	59 (3)
2	21	21	121	10	79	76	73 (5)	73 (5)
2	29	29	190	15	126	122	118 (6)	118 (6)

Table 2: Complexity comparison between kernel-search variants for Mirath parameters and Courtois challenges D, E, and F. Complexities are shown as the $\log_2$ of the number of $\mathbb{F}_q$-multiplications with $\omega = 2.81$.

Together with the improved kernel search, we used the hybrid approach to solve these instances. More precisely, we enumerated $l_v = 5$ entries in $\mathbb{F}_2$ of the solution vector, which reduces $k = 81$ to $k' = 76$ and thus $\lceil \frac{k}{m} \rceil = 5$ to $\lceil \frac{k'}{m} \rceil = 4$. Then, we used the $\text{Kernel}_{\text{imp}}$ attack to solve each of the 2^5 reduced instances. Table 3 shows the complexity estimates of this approach using Kernel and $\text{Kernel}_{\text{imp}}$. We provide the actual instance we solved in the Github repository together with our code.

r	Kernel	$\text{Kernel}_{\text{imp}}$	Running time
7	2^{51}	2^{49}	2.7 core-days
8	2^{56}	2^{53}	45 core-days
9	2^{60}	2^{57}	576 core-days
10	2^{64}	2^{61}	11 434 core-days

Table 3: Comparison between estimated cost and actual running times. The cost is the number of $\mathbb{F}_2$-multiplications (with $\omega = 3$) needed to solve 2^5 MinRank instances with parameters $(q, m, n, k) = (2, 19, 19, 76)$ and the given r.

In $\text{KernelSearch}_{\text{IMP}}$ (Algorithm 6), we need to enumerate all $\mathbf{k}_1, \ldots, \mathbf{k}_{a-1} \in \mathbb{F}_2^r$ with $a = \lceil k'/m \rceil$. We implemented this enumeration by iterating over the integers from 0 to $2^{(\lceil k'/m \rceil - 1)r} - 1$ and splitting the binary representation of each integer into $\lceil k'/m \rceil - 1$ strings of r bits, each of which defines one vector $\mathbf{k}_i$. The same procedure is used to choose the last guess in the $\text{ProcessGuess}_{\text{IMP}}$ function.

stitute of Standards and Technology, nor does it imply that the software identified is necessarily the best available for the purpose.

This makes the enumeration of kernel vectors fully deterministic and easy to split across multiple cores: in addition to splitting the whole instance into 32 smaller instances that can be solved in parallel (recall that we enumerate 5 entries of the solution vector), we can assign a specific subset of the guesses to each core to further parallelize the workload.

Our code is written in C++ and is publicly available at

https://anonymous.4open.science/r/MinRank_Kernel_Search-0BF4/

5 The Support-Minors Attack

5.1 The Original Proposal

The Support-Minors (SM) attack for the homogeneous MinRank problem was proposed by Bardet et al. [6]. The folklore extension of SM to an affine MinRank problem, defined by matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k$, introduces one unknown $x_0 \in \mathbb{F}_q$ as the coefficient of $\mathbf{M}_0$, so that $\mathbf{\Sigma_x} = \sum_{i=0}^{k} x_i \mathbf{M}_i$ becomes an $m \times n$ matrix of homogeneous linear forms in the variables $\mathbf{x} = (x_0, x_1, \ldots, x_k)$. This section describes the SM attack for the affine MinRank problem following this folklore approach.

We first describe the SM modeling and then, in Algorithm 7, an algorithm to solve it. In Remark 1 we saw that, with high probability, there exists a matrix $\mathbf{K} \in \mathbb{F}_q^{r \times (n-r)}$ such that

$$\mathbf{\Sigma_x} \begin{bmatrix} \mathbf{I}_{n-r} \\ \mathbf{K} \end{bmatrix} = 0. \tag{5}$$

As a consequence, the matrices $\mathbf{\Sigma_x}$ and $\begin{bmatrix} -\mathbf{K} & \mathbf{I}_r \end{bmatrix}$ have the same row space. Hence, for any row $\boldsymbol{\rho}$ of $\mathbf{\Sigma_x}$, the matrix

$$\left[\begin{array}{c} \boldsymbol{\rho} \\ \hline -\mathbf{K} \;\; \mathbf{I}_r \end{array}\right] \in \mathbb{F}_q^{(r+1) \times n} \tag{6}$$

has rank r and all its maximal minors (minors of size $r+1$) are zero.

A *Support-Minors* equation is a minor of size $r+1$ of the matrix in (6), where $\boldsymbol{\rho}$ can be any row of the matrix of linear polynomials $\mathbf{\Sigma_x}$. More precisely, the set of Support-Minors equations is defined as follows.

Definition 3 (Support-Minors equations). *Let $\mathbf{\Sigma_x}$ be an $m \times n$ matrix of linear polynomials in the variables $\mathbf{x} = (x_0, \ldots, x_k)$. Given a row $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_n)$ of $\mathbf{\Sigma_x}$ and a subset $U = \{u_1, \ldots, u_{r+1}\}$ of $[n]$ of size $r+1$, the $(\boldsymbol{\rho}, U)$ Support-Minors equation $\mathsf{eq}_{\boldsymbol{\rho},U}$ is defined as*

$$\mathsf{eq}_{\boldsymbol{\rho},U} = \sum_{u_i \in U} (-1)^i \rho_{u_i} \cdot c_{U \setminus \{u_i\}},$$

where c_T denotes the maximal minor of $\begin{bmatrix} -\mathbf{K} & \mathbf{I}_r \end{bmatrix}$ from the columns indexed by an r-sized subset T of $[n]$. The Support-Minors modeling consists of the set of

equations

$$\mathrm{SM}_{eqs} = \left\{ \mathsf{eq}_{\boldsymbol{\rho},U} : U \in \binom{[n]}{r+1}, \boldsymbol{\rho} \in \mathsf{Rows}(\boldsymbol{\Sigma}_{\mathbf{x}}) \right\}. \tag{7}$$

The entries of $\mathbf{x}$ *are called the* linear variables *and the entries of* $\mathbf{c} = (c_T : T \subset [n]$ *of size* $r)$ *are called the* minor variables.

Algorithm 7 Support-Minors

SupportMinors$(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, b)$

1 : $\boldsymbol{\Sigma}_{\mathbf{x}} = x_0\mathbf{M}_0 + \sum_{i=1}^{k} x_i\mathbf{M}_i$
2 : Compute $\mathrm{SM}_{\mathrm{eqs}}$ using $\boldsymbol{\Sigma}_{\mathbf{x}}$ as in Definition 3
3 : $\mathrm{SM}_{\mathrm{eqs},b} = \{\mathrm{m} \cdot \mathsf{eq} : \mathsf{eq} \in \mathrm{SM}_{\mathrm{eqs}} \text{ and } \mathrm{m} \in \mathsf{Mon}_{b-1}(x_0, x_1, \ldots, x_k)\}$
4 : $I = \langle x_0^q - x_0, x_1^q - x_1, \ldots, x_k^q - x_k \rangle$
5 : $\overline{\mathrm{SM}}_{\mathrm{eqs},b} = \{f \mod I : f \in \mathrm{SM}_{\mathrm{eqs},b}\}$
6 : $\mathbf{A}_b = \textsc{Macaulay}(\overline{\mathrm{SM}}_{\mathrm{eqs},b})$
7 : $\mathbf{s} \leftarrow \textsc{ExtractSolution}(\mathbf{A}_b)$
8 : **return** $\mathbf{s}$

Algorithm 7 describes the folklore adaptation of the SM modeling of [6], proposed for the homogeneous case, to the affine MinRank problem. It starts by computing the matrix $\boldsymbol{\Sigma}_{\mathbf{x}}$ of homogeneous linear polynomials in $\mathbb{F}_q[x_0, x_1, \ldots, x_k]$, and the Support-Minors equations $\mathrm{SM}_{\mathrm{eqs}}$. The extended equations $\mathrm{SM}_{\mathrm{eqs},b}$ are computed using only the monomials from $\mathsf{Mon}_{b-1}(x_0, \ldots, x_k)$, where $\mathsf{Mon}_d(\mathbf{x})$ denotes the set of monomials of degree d in the entries of $\mathbf{x}$. Then, the field equations in the x_i variables are applied to $\mathrm{SM}_{\mathrm{eqs},b}$, i.e., $\overline{\mathrm{SM}}_{\mathrm{eqs},b} := \{f \mod I : f \in \mathrm{SM}_{\mathrm{eqs},b}\}$ with $I = \langle x_0^q - x_0, \ldots, x_k^q - x_k \rangle$. Finally, one builds the Macaulay matrix $\mathbf{A}_b$, finds at least one element in the kernel of $\mathbf{A}_b$ and extracts the MinRank solution from it. This last step is done by the algorithm ExtractSolution. This algorithm first computes an element $\mathbf{v}$ in the kernel of $\mathbf{A}_b$ by performing Gaussian elimination or using the Wiedemann algorithm. Then, it outputs a potential solution $\mathbf{s}$ by reading the coordinates of $\mathbf{v}$ that correspond to monomials of the form $x_i c_T$, where c_T is the minor variable corresponding to the r-sized minor of $\begin{bmatrix} -\mathbf{K} & \mathbf{I}_r \end{bmatrix}$ from its last r columns, which is 1.

Correctness. Let $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{m \times n}$ define an affine MinRank instance with target rank r. Clearly, any pair $(\mathbf{x}, \mathbf{K})$ that satisfies Equation (5) produces a $\mathbf{v}$ such that $\mathbf{A}_b \cdot \mathbf{v} = 0$. Therefore, when the underlying MinRank problem has a unique solution and the dimension of the kernel of $\mathbf{A}_b$ is 1, we can find a solution $\mathbf{x}$ by finding a kernel element $\mathbf{v}$ of $\mathbf{A}_b$.

Remark 2. Step 5 in Algorithm 7 is not explicitly mentioned in [6] for the case $q > 2$. We believe this is because, first, for $q > 2$ the condition $b < q$ is imposed

for the algorithm to behave as predicted, and second, the number of columns of the matrix $\mathbf{A}_b$ is estimated to be $\mathrm{N}^{\mathrm{cols}}_{b,n} = \binom{n}{r}\binom{K+b-1}{b}$, with $K = k+1$ and $q > 2$. Note that the field equations play a role only when $b \geq q$, that is, $\overline{\mathrm{SM}}_{\mathrm{eqs},b} = \mathrm{SM}_{\mathrm{eqs},b}$ for $b < q$. If the field equations were not applied, $\mathbf{A}_b$ would have $\mathrm{N}^{\mathrm{cols}}_{b,n}$ columns for any $b \geq 1$.

Remark 3 (The submatrix optimization). There is an optimization of the Support-Minors modeling in which an attacker uses the MinRank instance defined by the last n' columns of each $\mathbf{M}_i$, with $r + b \leq n' \leq n$ and $(m-r)(n'-r) \geq k+1$, and thus deals with a MinRank instance with parameters (q, m, n', k, r). The second condition is needed for this instance to keep a unique solution: otherwise, the instance is expected to have about $q^{k-(m-r)(n'-r)} \geq 1$ solutions besides the planted one.

Consider an attacker who uses only the last n' columns of the input matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k$, with $r + b \leq n' \leq n$ and $1 \leq b < r+2$. For an affine MinRank problem with a unique solution, in [6] it is heuristically shown that

$$\mathsf{rank}(\mathbf{A}_b) = \min(\mathrm{N}^{\mathrm{cols}}_{b,n'} - 1, \mathrm{R}^{\mathrm{exp}}_{b,n'}),$$

where the number of columns $\mathrm{N}^{\mathrm{cols}}_{b,n'}$ of $\mathbf{A}_b$ is given by

$$\mathrm{N}^{\mathrm{cols}}_{b,n'} = \begin{cases} \binom{n'}{r}\binom{k+b}{b} & \text{if } q > 2 \text{ and } b < q \\ \binom{n'}{r} \cdot \sum_{i=1}^{b} \binom{k+1}{i} & \text{if } q = 2 \end{cases}, \tag{8}$$

and its expected rank $\mathrm{R}^{\mathrm{exp}}_{b,n'}$ is given by

$$\mathrm{R}^{\mathrm{exp}}_{b,n'} = \begin{cases} \sum_{i=1}^{b} (-1)^{i+1} \binom{n'}{r+i}\binom{m+i-1}{i}\binom{k+b-i}{b-i} & \text{if } q > 2 \\ \sum_{j=1}^{b}\sum_{i=1}^{j} (-1)^{i+1} \binom{n'}{r+i}\binom{m+i-1}{i}\binom{k+1}{j-i} & \text{if } q = 2 \end{cases}. \tag{9}$$

Given integers (b, n') with $r+b \leq n' \leq n$, $(m-r)(n'-r) \geq k+1$, $1 \leq b < r+2$ and $b < q$ for $q > 2$, such that $\mathrm{N}^{\mathrm{cols}} - 1 \leq \mathrm{R}^{\mathrm{exp}}$, Support-Minors (Algorithm 7) executed with parameters (q, m, n', k, r) and target degree b has a complexity of

$$\mathcal{C}_{\mathrm{SM}} = \min\left(3 \cdot (k+1)(r+1) \cdot \left(\mathrm{N}^{\mathrm{cols}}_{b_{\mathrm{sol}},n'}\right)^2, 7 \cdot \left(\mathrm{N}^{\mathrm{cols}}_{b_{\mathrm{sol}},n'}\right)^{\omega}\right), \tag{10}$$

where $b_{\mathrm{sol}} := \min\{b : \mathrm{N}^{\mathrm{cols}}_{b,n'} - 1 \leq \mathrm{R}^{\mathrm{exp}}_{b,n'} \text{ and } 1 \leq b \leq \min(r+1, n'-r)\}$. Smith-Tone [40] provided bivariate series that can be used to determine b_{sol}.

Remark 4 (On the condition $b < q$ for $q > 2$). The number of columns $\mathrm{N}^{\mathrm{cols}}_{b,n'}$ of the Macaulay matrix $\mathbf{A}_b$ is the number of possible monomials that appear in the support of the polynomials in $\overline{\mathrm{SM}}_{\mathrm{eqs},b}$. For $q = 2$, the set of possible monomials in the support of polynomials in $\overline{\mathrm{SM}}_{\mathrm{eqs},b}$ is the set of all monomials of bi-degree $(i, 1)$ with $1 \leq i \leq b$ that are square-free in $\mathbf{x}$. Hence $\mathrm{N}^{\mathrm{cols}}_{b,n'}$ is as shown in (8) for $q = 2$. If $b < q$, the field equations have no impact on the equations in $\mathrm{SM}_{\mathrm{eqs},b}$, so that $\overline{\mathrm{SM}}_{\mathrm{eqs},b} = \mathrm{SM}_{\mathrm{eqs},b}$ and the set of possible monomials in the support of polynomials in $\overline{\mathrm{SM}}_{\mathrm{eqs},b}$ is the set of monomials of bi-degree $(b, 1)$. Thus, for $q > 2$, $\mathrm{N}^{\mathrm{cols}}_{b,n'}$ is as shown in (8) as long as $b < q$.

Remark 5 (On the condition $1 \leq b < r + 2$). The expected rank $\mathrm{R}^{\mathrm{exp}}_{b,n'}$ of the Macaulay matrix $\mathbf{A}_b$ is computed by counting the syzygies of $\overline{\mathrm{SM}}_{\mathrm{eqs},b}$ at bi-degree $(b, 1)$, i.e., polynomial relations among the polynomials in $\overline{\mathrm{SM}}_{\mathrm{eqs},b}$ whose coefficients are polynomials of degree $b - 1$ in the linear variables $\mathbf{x}$. For any $b \geq 1$, there are syzygies that come from considering the $(r + b)$-sized minors of matrices of the form

$$\left[\begin{array}{c|c} \boldsymbol{\rho}_{j_1} & \\ \hline \vdots & \\ \hline \boldsymbol{\rho}_{j_b} & \\ \hline -\mathbf{K} & \mathbf{I}_r \end{array}\right] \in \mathbb{F}_q^{(r+b)\times n'}.$$

Such syzygies have been verified (experimentally in [6] and theoretically in [13] for some particular cases) to be the only ones, with high probability, for a random MinRank problem as long as $b < r+2$. The condition $b < r+2$ is needed because at bi-degree $(r + 2, 1)$ there are other syzygies that are independent of those mentioned above, and the formula in (9) does not count them. We refer to [6, Section 5.4] for a detailed explanation of why such syzygies exist at degree $r + 2$ in the $\mathbf{x}$ variables.

5.2 Extended Analysis of Support-Minors

Bardet and Gilard [7] extended the complexity analysis of the Support-Minors attack for the homogeneous MinRank problem. They analyzed the complexity of an XL approach in which one multiplies the Support-Minors equations $\mathrm{SM}_{\mathrm{eqs}}$ by monomials of bi-degree $(b - 1, c - 1)$ in the linear variables $\mathbf{x}$ and the minor variables $\mathbf{c}$, with $c \geq 1$ and $b \geq 1$.

Algorithm 8 describes our interpretation of the approach proposed in [7]. Unlike the original proposal (Algorithm 7), Bardet and Gilard extend the initial set of equations $\mathrm{SM}_{\mathrm{eqs}}$ by multiplying by monomials of degree $c - 1 \geq 0$ in the minor variables $\mathbf{c}$. In this way, at step 3, we obtain a set of extended equations $\mathrm{SM}_{\mathrm{eqs},(b,c)}$ of bi-degree (b, c) in $(\mathbf{x}, \mathbf{c})$. Steps 4 and 5 work as in Algorithm 7.

Given an integer $c \geq 1$, for most MinRank problems with a unique solution, the corank of the Macaulay matrix $\mathbf{A}_{b,c}$ for any $b \geq 1$ is given by $\max\big(1, [t^b]\mathcal{H}_{\mathrm{SM},c}(t)\big)$, where $[t^b]\mathcal{H}_{\mathrm{SM},c}(t)$ denotes the coefficient of t^b in the series

$$\mathcal{H}_{\mathrm{SM},c}(t) = \left[\frac{\det(\mathbf{A}_c(t)) \cdot (1-t)^{(m-r)(n-r)-(k+1)}}{t^{\binom{r}{2}}}\right]_+, \tag{11}$$

Algorithm 8 Support-Minors (as proposed in [7])

$\text{SupportMinors}_{\text{bg}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, b, c)$

1 : $\mathbf{\Sigma_x} = x_0\mathbf{M}_0 + \sum_{i=1}^{k} x_i\mathbf{M}_i$
2 : Compute SM_{eqs} using $\mathbf{\Sigma_x}$ as in Definition 3
3 : $\text{SM}_{\text{eqs},(b,c)} = \{\text{m}_x\text{m}_c \cdot \mathsf{eq} : \mathsf{eq} \in \text{SM}_{\text{eqs}},\ \text{m}_x \in \mathsf{Mon}_{b-1}(\mathbf{x}) \text{ and } \text{m}_c \in \mathsf{Mon}_{c-1}(\mathbf{c})\}$
4 : $\mathbf{A}_{b,c} = \textsc{Macaulay}(\text{SM}_{\text{eqs},(b,c)})$
5 : $\mathbf{s} \leftarrow \textsc{ExtractSolution}(\mathbf{A}_{b,c})$
6 : **return s**

where $\mathbf{A}_c(t) \in (\mathbb{Z}[t])^{r\times r}$ is the matrix whose (i,j)-th entry is the polynomial

$$\sum_{l=0}^{\max(m-c-j,n-i)} \binom{m-c-j}{l}\binom{n+c-i}{l+c}t^l, \tag{12}$$

with $1 \le i, j \le r$ [7].

For $c = 1$, the complexity of $\textsc{SupportMinors}_{\text{BG}}$ following the approach given in [7] is

$$\mathcal{C}'_{\text{SM}} = \min\left(3\cdot(k+1)(r+1)\cdot\left(\binom{n'}{r}\binom{k+b'_{\text{sol}}}{b'_{\text{sol}}}\right)^2, 7\cdot\left(\binom{n'}{r}\binom{k+b'_{\text{sol}}}{b'_{\text{sol}}}\right)^{\omega}\right), \tag{13}$$

where $b'_{\text{sol}} := \deg(\mathcal{H}_{\text{SM},1}(t)) + 1$.

Comparison with the original algorithm. Clearly, for $c = 1$, we have $\text{SM}_{\text{eqs},b} = \text{SM}_{\text{eqs},(b,c)}$, and the series in (11) predicts the corank of $\mathbf{A}_b$ in Algorithm 7 for a given $b < q$ that can be greater than $r + 2$, which removes the condition $b < r+2$ of the original proposal. However, for $q = 2$ and $b \ge q$, the matrix $\mathbf{A}_{b,1}$ in Algorithm 8 has more columns than $\mathbf{A}_b$ in Algorithm 7. Thus, Algorithm 8 is less efficient than Algorithm 7 for $q = 2$ and $b'_{\text{sol}} \ge 2$, and both algorithms have the same efficiency for $q > 2$ and $b'_{\text{sol}} < \min(q, r + 2)$.

5.3 Extended Analysis with Field Equations

In this section, we extend the complexity analysis of the SM attack further by considering, for any $q \ge 2$, the field equations of the linear variables on top of the analysis proposed by Bardet and Gilard for the case $c = 1$. Our analysis applies to the affine MinRank problem directly without converting it into a homogeneous one, and hence without introducing an extra variable $x_0 \in \mathbb{F}_q$. Instead, we homogenize the polynomials of $\text{SM}_{\text{eqs},b}$ and apply the homogenized field equations to the linear variables, i.e., the equations of the form $x_i^q - h^{q-1}x_i$, where h is the homogenization variable.

The advantages of considering the homogenized Support-Minors equations over the folklore approach of adding an extra variable $x_0 \in \mathbb{F}_q$ (as the coefficient of $\mathbf{M}_0$) are as follows. First, in the homogenized approach, the homogenization variable h is not subject to a field equation, so the reduction modulo $I^{(h)}$ keeps the equations homogeneous of degree b. Second, when applying the homogenized field equations, the resulting polynomials are supported on a smaller set of monomials than in the folklore approach. This yields a Macaulay matrix with fewer columns and hence a faster attack. For instance, for $q = 2$ the homogenized approach produces a system with a total of $\binom{n}{r}\sum_{j=0}^{b}\binom{k}{j}$ monomials, while in the folklore approach that number is $\binom{n}{r}\sum_{j=1}^{b}\binom{k+1}{j}$ (see (8)).

Algorithm 9 Support-Minors with field equations

$\textsc{SupportMinors}_{\text{fe}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, b)$

1 : $\mathbf{\Sigma_x} = h\mathbf{M}_0 + \sum_{i=1}^{k} x_i\mathbf{M}_i$ // h is a homogenization variable.
2 : Compute SM_{eqs} using $\mathbf{\Sigma_x}$ as in Definition 3
3 : $\text{SM}_{\text{eqs},b} = \{\text{m} \cdot \mathsf{eq} : \mathsf{eq} \in \text{SM}_{\text{eqs}} \text{ and } \text{m} \in \mathsf{Mon}_{b-1}(h, x_1, \ldots, x_k)\}$
4 : $I^{(h)} = \langle x_1^q - h^{q-1}x_1, \ldots, x_k^q - h^{q-1}x_k\rangle$
5 : $\overline{\text{SM}}_{\text{eqs},b} = \{f \mod I^{(h)} : f \in \text{SM}_{\text{eqs},b}\}$
6 : $\mathbf{A}_b = \textsc{Macaulay}(\overline{\text{SM}}_{\text{eqs},b})$
7 : $\mathbf{s} \leftarrow \textsc{ExtractSolution}(\mathbf{A}_b)$
8 : **return s**

Algorithm 9 shows $\textsc{SupportMinors}_{\text{fe}}$. On inputs $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{m\times n'}$ and $b, r \in \mathbb{N}^+$, $\textsc{SupportMinors}_{\text{fe}}$ first computes $\mathbf{\Sigma_x} = h\mathbf{M}_0 + \sum_{i=1}^{k} x_i\mathbf{M}_i$ and uses it to compute the Support-Minors equations SM_{eqs}. Then, it computes the extended set $\text{SM}_{\text{eqs},b}$ by multiplying the elements in SM_{eqs} by all monomials in the variables $h, x_1, \ldots, x_k$ of degree $b-1$. Up to this point, the algorithm coincides with the original SM attack (Algorithm 7). Next, $\textsc{SupportMinors}_{\text{fe}}$ computes the set $\overline{\text{SM}}_{\text{eqs},b}$ formed by all the polynomials of the form $f \mod I^{(h)}$, where $I^{(h)}$ is the ideal generated by the set $\{x_1^q - h^{q-1}x_1, \ldots, x_k^q - h^{q-1}x_k\}$ (instead of $\{x_0^q - x_0, x_1^q - x_1, \ldots, x_k^q - x_k\}$ as in Algorithm 7). Finally, $\textsc{SupportMinors}_{\text{fe}}$ builds the Macaulay matrix $\mathbf{A}_b$ of $\overline{\text{SM}}_{\text{eqs},b}$ and attempts to extract a solution from one of its kernel vectors as in the original Support-Minors attack.

The Hilbert series that predicts the corank of the Macaulay matrix $\mathbf{A}_b$ at step 6 of Algorithm 9 is given in Assumption 1; this prediction is heuristic. Here, b is the degree of the equations in the linear variables, and the corresponding total degree is $b+1$. The idea is to multiply $\mathcal{H}_{\text{SM},1}(t)$ by the factor $(1-t^q)^k$, so that, instead of the series $(1-t)^{-(k+1)}$, which counts all monomials in the $k+1$ variables $h, x_1, \ldots, x_k$, we obtain the series

$$\text{M}_{q,k}(t) := (1-t^q)^k \cdot (1-t)^{-(k+1)}, \tag{14}$$

which counts monomials in $h, x_1, \ldots, x_k$ whose degree in each x_i is smaller than q, i.e., the monomials that survive the reduction modulo $I^{(h)}$.

Assumption 1 *For a random MinRank problem $\mathbf{M}_0, \ldots, \mathbf{M}_k$ with a unique planted solution and parameters (q, m, n, k, r), with high probability, the corank of the Macaulay matrix $\mathbf{A}_b$ at step 6 of Algorithm 9 is given by $\max\big(1, [t^b]\mathcal{H}^{fe}_{\mathrm{SM}}(t)\big)$, where $\mathcal{H}^{fe}_{\mathrm{SM}}(t)$ is defined as*

$$\mathcal{H}^{fe}_{\mathrm{SM}}(t) = \left[\mathcal{H}_{\mathrm{SM},1}(t) \cdot (1 - t^q)^k\right]_+, \tag{15}$$

where $\mathcal{H}_{\mathrm{SM},1}(t)$ is as given in (11).

The complexity of SupportMinors$_{\mathrm{fe}}$ is given by

$$\mathcal{C}''_{\mathrm{SM}} = \min\left(3 \cdot (k+1)(r+1) \cdot \left(\binom{n'}{r} \cdot [t^{b''_{\mathrm{sol}}}]\,\mathrm{M}_{q,k}(t)\right)^2, 7 \cdot \left(\binom{n'}{r} \cdot [t^{b''_{\mathrm{sol}}}]\,\mathrm{M}_{q,k}(t)\right)^\omega\right), \tag{16}$$

where $b''_{\mathrm{sol}} := \deg(\mathcal{H}^{\mathrm{fe}}_{\mathrm{SM}}(t)) + 1$ and $[t^b]\,\mathrm{M}_{q,k}(t)$ is the coefficient of t^b in $\mathrm{M}_{q,k}(t)$.

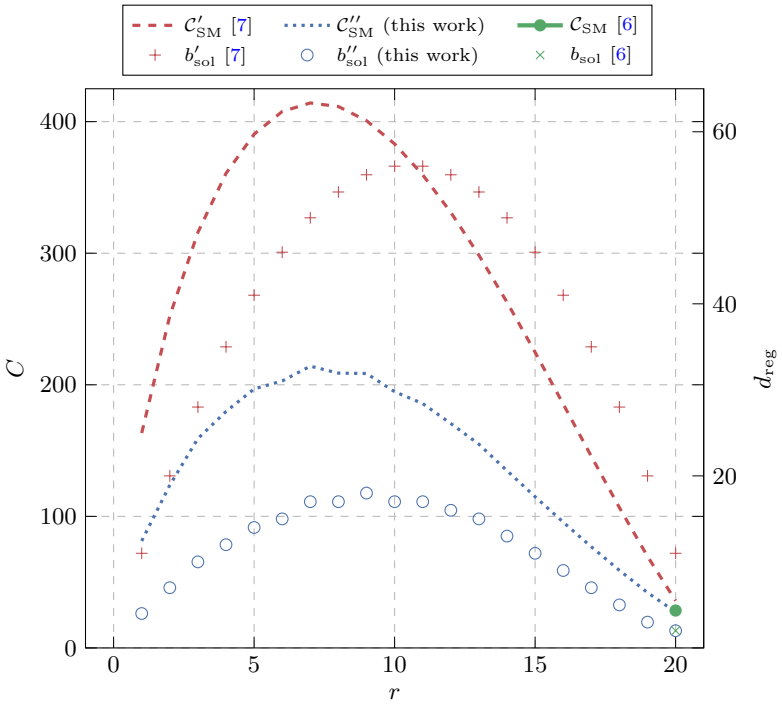


Fig. 1: Complexity comparison of the Support-Minors variants for $q = 2$, $n = m = n' = 22$, and $k = (n - r)(m - r) - 2$.

Figure 1 compares the complexity estimates of our variant of the SM attack against the state-of-the-art approaches for different values of r, with parameters $m = n = 22$ and $k = (n-r)(m-r) - 2$. The red dashed line corresponds to the version by Bardet and Gilard, the green dot at $r = 20$ represents the original SM formulation, and the blue dotted line corresponds to our proposed SM variant. For each value of r, we also report the associated solving degree b_{sol}.

The complexity estimates show that our variant improves upon the approach of Bardet and Gilard, while remaining comparable to the estimates of the original SM formulation where the latter applies. In Figure 1, our variant is 9 to 205 bits below the estimate of Bardet and Gilard, and 1.2 bits below the original SM formulation, which applies only for $r = 20$ (with $b_{\text{sol}} = 2$). The reason is that both our approach and the original SM formulation benefit from the low degree of the field equations. Our variant, however, applies to a broader range of instances than the original SM. For $1 \leq r \leq 19$, no pair (b, n') with $1 \leq b \leq r+1$ and $r + b \leq n' \leq n$ satisfies $\mathrm{N}^{\text{cols}}_{b,n'} - 1 \leq \mathrm{R}^{\text{exp}}_{b,n'}$, so the original SM formulation does not apply, whereas our formulation applies for all considered values of r. This is because our formulation requires neither $b''_{\text{sol}} \leq r+1$ nor $r + b''_{\text{sol}} \leq n'$.

Experimental verification of Assumption 1. We have verified that the series given in (15) correctly predicts the corank (and therefore the rank) of the Macaulay matrix $\mathbf{A}_b$ at step 6 of Algorithm 9 for $q \in \{2, 3, 4\}$, $m = 7$, $r = 2$, $n \in \{r+4, r+5\}$, $k \in \{10 \leq p \leq 26 : p+1 \leq (m-r)(n-r)\}$ and for each $b \geq 1$ such that the number of columns of $\mathbf{A}_b$ is smaller than $15 \cdot 10^4$.

6 The Minors Attack

6.1 The Original Proposal

Let $\mathbf{x} = (x_1, \ldots, x_k)$ be a vector of unknowns and set $\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{M}_0 + \sum_i x_i \mathbf{M}_i$. Let $\mathcal{M}_{r+1}(\mathbf{\Sigma}_{\mathbf{x}})$ denote the set of all $(r+1)$-sized minors of the matrix $\mathbf{\Sigma}_{\mathbf{x}}$. Each element in $\mathcal{M}_{r+1}(\mathbf{\Sigma}_{\mathbf{x}})$ is a polynomial in $\mathbb{F}_q[\mathbf{x}]$ of degree at most $r+1$. A common root $\mathbf{x} \in \mathbb{F}_q^k$ of the polynomials in $\mathcal{M}_{r+1}(\mathbf{\Sigma}_{\mathbf{x}})$ yields a solution of the MinRank problem defined by the $\mathbf{M}_i$ with target rank r.

The attack is described in Algorithm 10. If $r + 1 < D$, the initial minors are extended to polynomials of degree $D = \deg(\mathcal{H}_{\mathrm{M}}(t)) + 1$, where

$$\mathcal{H}_{\mathrm{M}}(t) = \left[\frac{\det(\mathbf{T}(t))(1-t)^{(m-r)(n-r)-(k+1)}}{t^{\binom{r}{2}}} \right]_+, \tag{17}$$

where $\mathbf{T}(t) \in (\mathbb{Z}[t])^{r \times r}$ is the matrix of polynomials given by

$$\mathbf{A}(t) = \left[\sum_{l=0}^{\max(m-i, n-j)} \binom{m-i}{l} \binom{n-j}{l} t^l \right]_{1 \leq i,j \leq r}. \tag{18}$$

The subroutine ExtractSolution can be instantiated with Gaussian elimination.

Algorithm 10 Minors attack

MINORS$(\mathbf{M}_0, \ldots, \mathbf{M}_k, r)$

1 : $\mathbf{\Sigma_x} = \mathbf{M}_0 + \sum_{i=1}^{k} x_i \mathbf{M}_i$
2 : $D = \deg(\mathcal{H}_{\mathrm{M}}(t)) + 1$ // $\mathcal{H}_{\mathrm{M}}$ as given in (17)
3 : $\mathcal{M} = \mathcal{M}_{r+1}(\mathbf{\Sigma_x})$
4 : $\mathrm{M}_{\mathrm{eqs}} = \{\mathrm{m} \cdot \mathsf{eq} : \mathsf{eq} \in \mathcal{M} \text{ and } \mathrm{m} \in \mathsf{Mon}_{\leq D-r-1}(x_1, \ldots, x_k)\}$
5 : $\mathbf{A} = \mathrm{MACAULAY}(\mathrm{M}_{\mathrm{eqs}})$
6 : $\mathbf{s} \leftarrow \mathrm{EXTRACTSOLUTION}(\mathbf{A})$
7 : **return s**

For $k < (m-r)(n-r)$ and random matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{m \times n}$, the Hilbert series of the ideal generated by $\mathcal{M}_{r+1}(\mathbf{\Sigma_x})$ is expected to be $\mathcal{H}_{\mathrm{M}}(t)$ [26,27].

As shown in [30], the F_5 criterion, which detects linearly dependent rows of the Macaulay matrix when solving multivariate equations, can be refined to construct a row submatrix of the Macaulay matrix $\mathbf{A}$ in step 5 with the same number of rows as the rank of $\mathbf{A}$. Hence, by using this refinement, the complexity of the Minors modeling is given by

$$\mathcal{C}_{\mathrm{M}} = \binom{k+D}{D}^{\omega}, \tag{19}$$

where $D = \deg(\mathcal{H}_{\mathrm{M}}(t)) + 1$.

6.2 Extended Analysis with Field Equations.

As for the Support-Minors attack, we extend the Minors attack by adding the field equations of the linear variables on top of the analysis proposed in [30]. This leads to a more efficient attack for MinRank problems over small fields, e.g., $q \in \{2, 3, 4\}$. The extended attack is given in Algorithm 11.

The following assumption gives the Hilbert series that predicts the corank of the Macaulay matrix $\mathbf{A}$ at step 7 of Algorithm 11; as for Support-Minors, the prediction is heuristic.

Assumption 2 *For a random MinRank problem $\mathbf{M}_0, \ldots, \mathbf{M}_k$ with a unique planted solution and parameters (q, m, n, k, r), with high probability, the corank of the Macaulay matrix $\mathbf{A}$ at step 7 of Algorithm 11 is given by $\max\big(1, [t^D]\mathcal{H}^{fe}_{\mathrm{M}}(t)\big)$, where $\mathcal{H}^{fe}_{\mathrm{M}}(t)$ is defined as*

$$\mathcal{H}^{fe}_{\mathrm{M}}(t) = \left[\mathcal{H}_{\mathrm{M}}(t) \cdot (1 - t^q)^k\right]_+, \tag{20}$$

where $\mathcal{H}_{\mathrm{M}}$ is given in (17).

The complexity of $\mathrm{MINORS}_{\mathrm{FE}}$ is dominated by the complexity of performing Gaussian elimination on the Macaulay matrix $\mathbf{A}$. Hence it is given by

$$\mathcal{C}'_{\mathrm{M}} = \left([t^D]\,\mathrm{M}_{q,k}(t)\right)^{\omega}, \tag{21}$$

Algorithm 11 Minors attack with field equations

$\textsc{Minors}_{\text{fe}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, r)$

1 : $\mathbf{\Sigma}_{\mathbf{x}} = h\mathbf{M}_0 + \sum_{i=1}^{k} x_i \mathbf{M}_i$ // h is a homogenization variable.
2 : $D = \deg(\mathcal{H}^{\text{fe}}_{\text{M}}(t)) + 1$ // $\mathcal{H}^{\text{fe}}_{\text{M}}$ as given in (20)
3 : $\mathcal{M} = \mathcal{M}_{r+1}(\mathbf{\Sigma}_{\mathbf{x}})$
4 : $\text{M}_{\text{eqs}} = \{\text{m} \cdot \mathsf{eq} : \mathsf{eq} \in \mathcal{M} \text{ and } \text{m} \in \mathsf{Mon}_{D-r-1}(h, x_1, \ldots, x_k)\}$
5 : $I^{(h)} = \langle x_1^q - h^{q-1}x_1, \ldots, x_k^q - h^{q-1}x_k \rangle$
6 : $\overline{\text{M}}_{\text{eqs},b} = \{f \mod I^{(h)} : f \in \text{M}_{\text{eqs}}\}$
7 : $\mathbf{A} = \textsc{Macaulay}(\overline{\text{M}}_{\text{eqs},b})$
8 : $\mathbf{s} \leftarrow \textsc{ExtractSolution}(\mathbf{A})$
9 : **return s**

where $\text{M}_{q,k}(t)$ is given in (14).

Figure 2 compares the complexity estimates for the Minors modeling with and without field equations for $q = 2$, $m = n = 22$, and $k = (n-r)(m-r) - 2$. The green solid curve gives $\mathcal{C}_{\text{M}}$ from [30], and the blue dotted curve gives $\mathcal{C}'_{\text{M}}$ for our variant, together with the corresponding solving degrees D. Our variant is 17 to 278 bits below the estimate of [30]. The gap is largest for intermediate values of r, where both curves attain their maxima, and smallest for $r = 20$.

Experimental verification of Assumption 2. We have verified that the series given in (20) correctly predicts the corank (and therefore the rank) of the Macaulay matrix $\mathbf{A}$ at step 7 of Algorithm 11 for $q \in \{2, 3, 4\}$, $m = 7$, $r \in \{2, 3\}$, $n \in \{r+4, r+5\}$, and $k \in \{3 \leq p \leq 19 : p + 1 \leq (m-r)(n-r)\}$. For each parameter set, we repeated the following experiment 100 times. First, we sampled random matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k$, then computed the set of equations $\overline{\text{M}}_{\text{eqs},b}$ as in step 6 of Algorithm 11, and finally computed the Hilbert series of the ideal generated by $\overline{\text{M}}_{\text{eqs},b}$. In all the tested cases, the empirical series matched the theoretical prediction $\mathcal{H}^{\text{fe}}_{\text{M}}(t)$ given in (20) up to $\deg(\mathcal{H}^{\text{fe}}_{\text{M}}(t))$. For parameters with $(m-r)(n-r) - (k+1) \in \{0, 1, 2\}$, we observed that, in some cases, the empirical series had terms of degree larger than $\deg(\mathcal{H}^{\text{fe}}_{\text{M}}(t))$. This is because, for $k + 1 \approx (m-r)(n-r)$, the random matrices $\mathbf{M}_0, \ldots, \mathbf{M}_k$ might have spurious MinRank solutions. In all cases with spurious MinRank solutions, the terms of the empirical series of degree larger than $\deg(\mathcal{H}^{\text{fe}}_{\text{M}}(t))$ had coefficients of at most 5, and in most cases the coefficient was 1.

7 The Kipnis–Shamir Attack

The Kipnis-Shamir (KS) attack aims at finding a common root over $\mathbb{F}_q$ of the bilinear polynomials given by the entries of the matrix

$$\mathbf{\Sigma}_{\mathbf{x}} \begin{bmatrix} \mathbf{I}_{n-r} \\ \mathbf{K} \end{bmatrix}, \tag{22}$$

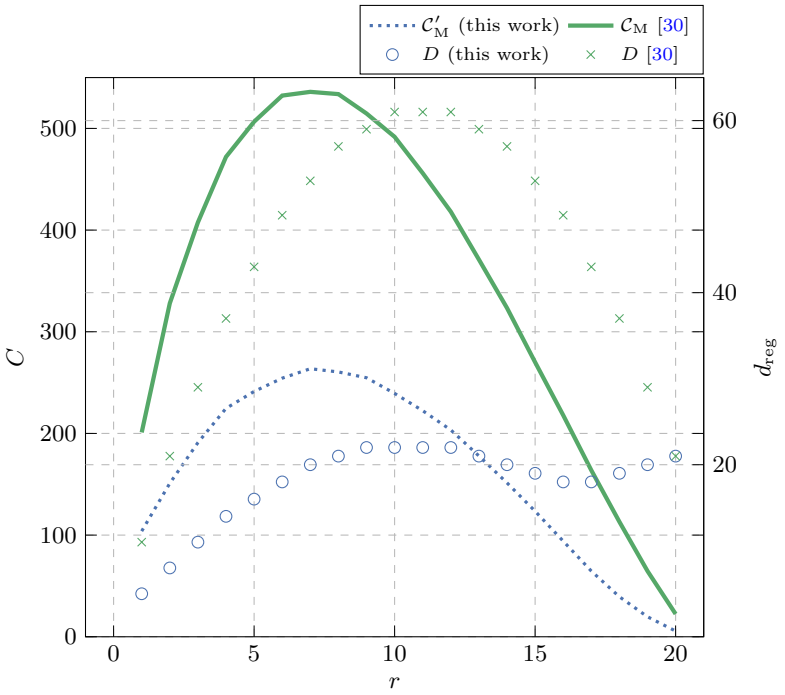


Fig. 2: Complexity comparison for the Minors modeling with and without field equations for $q = 2$, $n = m = 22$, and $k = (n - r)(m - r) - 2$.

where $\mathbf{I}_{n-r}$ is the identity matrix of size $n - r$, and $\mathbf{K} = [k_{i,j}]$ is a matrix of variables of size $r \times (n - r)$ [35]. These polynomials belong to the polynomial ring $\mathbb{F}_q[x_1, \ldots, x_k, k_{1,1}, \ldots, k_{r,n-r}]$, which, for simplicity, we denote by $\mathbb{F}_q[\mathbf{x}, \mathbf{K}]$. Clearly, a common root of the polynomials given by the entries of the matrix shown in (22) yields a solution to the underlying MinRank problem.

In this section we revisit the complexity of solving these equations with multi-homogeneous XL. We show that the Hilbert series used in the literature to predict the rank of the Macaulay matrices of the KS equations [36,37] does not apply to actual KS systems, we prove a lower bound on the corank of these matrices that explains the discrepancy, and we derive a corrected prediction, which we validate experimentally and use for our estimates.

7.1 Solving with Gröbner Bases

The KS equations can be solved using generic Gröbner basis algorithms, such as F_4 or F_5 [24,25]. However, analytically predicting the solving degree, that is, the maximum degree reached during a Gröbner basis computation under a graded monomial ordering, is an open problem. Consequently, the exact complexity of solving the KS equations via Gröbner basis algorithms remains unknown.

Some results are known regarding the complexity of generic Gröbner basis algorithms when applied to the KS equations. On the one hand, for certain MinRank parameters, empirical results indicate that the solving degree is smaller than $r+2$ [28, Section 5], [43, Section 6]. On the other hand, for parameters with $k \approx (n-r)(m-r)$, the solving degree can be significantly higher than $r+2$.

When viewed within the KS polynomial ring $\mathbb{F}_q[\mathbf{x}, \mathbf{K}]$, the ideal generated by the Support-Minors equations (with each c_T replaced by the corresponding minor of $[-\mathbf{K}\ \mathbf{I}_r]$) equals the KS ideal. Since any ideal in a finitely generated polynomial ring over a finite field is radical if it contains the field equations, Hilbert's Nullstellensatz implies that the ideal generated by the Minors equations along with the field equations is contained in the intersection of the ideal generated by the the KS equations and field equations with $\mathbb{F}_q[\mathbf{x}]$. Moreover, it has been shown that both the Minors and Support-Minors equations emerge as degree falls at degree $r+2$ during the Gröbner basis computation of the KS ideal [4,32].

7.2 Solving with Multi-homogeneous XL

In this section we recall the multi-homogeneous structure of the KS equations and describe the multi-homogeneous XL algorithm that exploits it; we first fix the notion of multi-degree.

Definition 4. *Let* $(\mathbf{x}_0, \ldots, \mathbf{x}_c)$ *be a partition of the variables of* $\mathbb{F}_q[\mathbf{x}_0, \ldots, \mathbf{x}_c]$. *The multi-degree of a monomial* m *is given by*

$$\mathsf{multi\text{-}deg}(\mathrm{m}) = (\deg_{\mathbf{x}_0}(\mathrm{m}), \ldots, \deg_{\mathbf{x}_c}(\mathrm{m})),$$

where $\deg_{\mathbf{x}_i}(\mathrm{m})$ *denotes the degree of* m *in the variables of* $\mathbf{x}_i$. *The multi-degree of a polynomial* f *is given by*

$$\mathsf{multi\text{-}deg}(f) = (\deg_{\mathbf{x}_0}(f), \ldots, \deg_{\mathbf{x}_c}(f)),$$

where $\deg_{\mathbf{x}_i}(f)$ *denotes the degree of* f *in the variables of* $\mathbf{x}_i$. *We say that* f *is multi-homogeneous if all the monomials in its support have the same multi-degree. Multi-degrees are compared componentwise.*

It has been known since 2008 that the quadratic components of the KS equations have a multi-homogeneous structure with respect to the partition $(\mathbf{x}, \mathbf{k}_1, \ldots, \mathbf{k}_{n-r})$, where $\mathbf{k}_j := (k_{1,j}, \ldots, k_{r,j})$ is the j-th column of $\mathbf{K}$ [28]. This structure is maintained if we only consider the equations coming from the first c columns of the kernel matrix, for any $1 \leq c \leq n-r$, that is, the entries of

$$\boldsymbol{\Sigma}_{\mathbf{x}} \begin{bmatrix} \mathbf{I}_c \\ \mathbf{0}^{(n-r-c)\times c} \\ \mathbf{K}_{[1..c]} \end{bmatrix}, \tag{23}$$

where $\mathbf{K}_{[1..c]}$ is the submatrix of $\mathbf{K}$ formed by its first c columns. A common root of these reduced KS polynomials is not guaranteed to give a solution to

the MinRank problem, but we expect this to be the case as long as the system is overdefined, which happens for $k < c(m-r)$; empirical evidence is given in [43, Table 2]. The equation in row ℓ and column j of (23) has multi-degree $\mathbf{e}_0 + \mathbf{e}_j$ with respect to $(\mathbf{x}, \mathbf{k}_1, \ldots, \mathbf{k}_c)$, where $\mathbf{e}_0, \ldots, \mathbf{e}_c$ are the canonical vectors of $\mathbb{Z}_{\geq 0}^{c+1}$. Its homogeneous component of highest degree is multi-homogeneous of multi-degree $\mathbf{e}_0 + \mathbf{e}_j$, while the linear part is an affine polynomial in the $\mathbf{x}$.

Multi-homogeneous XL. Multi-homogeneous XL is a variant of XL for systems whose highest-degree homogeneous components form a multi-homogeneous sequence. The term was coined by Ning, Ran, and Samardjiska [37], but such an algorithm was used in cryptanalysis before; see, for instance, [14, Section 2.5] and [39, Section 6]. Algorithm 12 describes Multi-$\mathrm{XL}_{\mathrm{KS}}$, a simplified version of the algorithm of [37] for the KS equations. Besides the MinRank problem, it takes as input the number c of kernel columns and a target multi-degree $\mathbf{d} = (d_0, d_1, \ldots, d_c)$; the Macaulay matrix $\mathbf{A}$ at step 4 has one row per product $\mathrm{m} \cdot \mathsf{eq}$ of multi-degree at most $\mathbf{d}$ and one column per monomial of multi-degree at most $\mathbf{d}$. As for Support-Minors and Minors, ExtractSolution computes a right kernel vector of $\mathbf{A}$ and reads from it a common root of the KS equations, and then a MinRank solution, which is assumed to be unique.

Algorithm 12 Multi-XL attack on the KS equations

Multi-$\mathrm{XL}_{\mathrm{KS}}(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, c, \mathbf{d})$

1: $\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{M}_0 + \sum_{i=1}^{k} x_i \mathbf{M}_i$
2: Compute $\mathrm{KS}_{\mathrm{eqs}} \subset \mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]$ as the entries of the matrix in (23)
3: $\mathrm{KS}_{\mathrm{eqs},\mathbf{d}} = \{\mathrm{m} \cdot \mathsf{eq} : \mathsf{eq} \in \mathrm{KS}_{\mathrm{eqs}} \text{ and } \mathsf{multi\text{-}deg}(\mathrm{m} \cdot \mathsf{eq}) \leq \mathbf{d}\}$
4: $\mathbf{A} = \textsc{Macaulay}(\mathrm{KS}_{\mathrm{eqs},\mathbf{d}})$
5: $\mathbf{s} \leftarrow \textsc{ExtractSolution}(\mathbf{A})$
6: **return** $\mathbf{s}$

The algorithm succeeds if the right kernel of $\mathbf{A}$ at $\mathbf{d}$ is spanned by the vector of all monomials evaluated at the solution, that is, if $\mathbf{A}$ has corank 1. The cost of the attack is therefore determined by the smallest multi-degree at which this happens.

7.3 The Generic Rank Prediction Fails

Ning, Ran, and Samardjiska predict the corank of $\mathbf{A}$ at multi-degree $\mathbf{d}$ using the series

$$\mathcal{H}(\mathbf{t}) = \sum_{\mathbf{d} \in \mathbb{Z}_{\geq 0}^{c+1}} a_{\mathbf{d}} \mathbf{t}^{\mathbf{d}} = \frac{\prod_{j=1}^{c} (1 - t_0 t_j)^m}{(1 - t_0)^{k+1} \prod_{j=1}^{c} (1 - t_j)^{r+1}}, \tag{24}$$

where $\mathbf{t}^{\mathbf{d}} = t_0^{d_0} t_1^{d_1} \cdots t_c^{d_c}$ [37, Example 5]. The series $\mathcal{H}$ is the multi-graded Hilbert series of a generic system of cm equations with the same multi-degrees as the

KS equations. Each positive coefficient $a_{\mathbf{d}}$ predicts the corank of the Macaulay matrix at multi-degree $\mathbf{d}$, and the multi-degrees with $a_{\mathbf{d}} \leq 0$ are those at which the matrix has corank 1, assuming that the system has a unique solution. The complexity analysis of [37] uses this prediction to select a multi-degree $\mathbf{d}$ at which the corank of $\mathbf{A}$ is expected to be 1, and thereby estimates the cost of MULTI-XL on KS systems with a unique solution.

Table 4 reports the corank of $\mathbf{A}$ at multi-degrees $\mathbf{d}_{\text{sol}}$ with $a_{\mathbf{d}_{\text{sol}}} \leq 0$ for KS systems of random MinRank instances with a planted solution and, for the same parameters, for random systems with the multi-degrees of the KS equations and a planted solution. The random systems have the multi-degrees of the KS equations but none of their structure, and their corank is the predicted one, $\max(1, a_{\mathbf{d}_{\text{sol}}})$, while for KS systems it is not.

q	m	n	k	r	c	$\mathbf{d}_{\text{sol}}$	$a_{\mathbf{d}_{\text{sol}}}$	Random system	KS system
16	10	5	10	2	3	(1,1,1,2)	-36	1	16
16	19	9	5	5	2	(1,1,1)	-12	1	1
16	12	8	8	3	3	(1,1,1,1)	0	1	19
16	8	5	4	3	2	(1,2,1)	-8	1	3
16	10	6	5	4	2	(1,3,1)	-50	1	11
16	7	7	3	4	3	(1,1,1,1)	-25	1	11
16	11	8	4	5	3	(1,1,1,1)	-108	1	11
2	42	8	15	4	2	(1,1,1)	-20	1	1–6
2	50	10	24	5	3	(1,1,1,1)	0	1–2	126–129

Table 4: Corank of the Macaulay matrix of MULTI-XL$_{\text{KS}}$ at a multi-degree $\mathbf{d}_{\text{sol}}$ with $a_{\mathbf{d}_{\text{sol}}} \leq 0$, for random systems with the multi-degrees of the KS equations and for KS systems, both with a planted solution.

The reason for this disparity is the structure of the KS equations. Write $\sigma_{\ell,t}(\mathbf{x})$ for the (ℓ, t) entry of $\mathbf{\Sigma}_{\mathbf{x}}$, an affine linear form in $\mathbf{x}$. The equation in row ℓ and column j of (23) is

$$\mathsf{eq}_{\ell,j} = \sigma_{\ell,j}(\mathbf{x}) + h_{\ell,j}, \qquad h_{\ell,j} := \sum_{i=1}^{r} k_{i,j}\, \sigma_{\ell,n-r+i}(\mathbf{x}), \tag{25}$$

where the quadratic part $h_{\ell,j}$ only involves the last r columns of $\mathbf{\Sigma}_{\mathbf{x}}$, which are the same for every j. Hence $h_{\ell,1}, \ldots, h_{\ell,c}$ are the same bilinear form evaluated at c different kernel columns, whereas the quadratic parts of a random system are unrelated. The series $\mathcal{H}$ does not account for this fact, and Section 7.4 shows that it forces a corank larger than one at many multi-degrees with $a_{\mathbf{d}} \leq 0$. Verbel et al. [43] derived syzygies of the KS equations from the same structure, through the Jacobians with respect to the linear and the kernel variables; conditions 2 and 3 of Section 7.4 avoid the multi-degrees at which they appear.

7.4 A Corrected Prediction

This section shows that identifying the c kernel columns yields a lower bound $B(\mathbf{d})$ on the corank of $\mathbf{A}$ (Theorem 1). We then estimate $B(\mathbf{d})$ with a Hilbert series and use it to select c and $\mathbf{d}_{\text{sol}}$ (conditions 1–5 and Assumption 3).

Given $\mathbf{d} = (d_0, \ldots, d_c) \in \mathbb{Z}_{\geq 0}^{c+1}$, set $D = d_1 + \cdots + d_c$. Let $\mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\leq \mathbf{d}}$ be the space of polynomials of multi-degree at most $\mathbf{d}$, and let $\mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)}$ be the space of polynomials of degree at most d_0 in $\mathbf{x}$ and at most D in $\mathbf{y} = (y_1, \ldots, y_r)$. The following lemma shows that the map identifying the c kernel columns with the single column $\mathbf{y}$ is surjective.

Lemma 1. *Let $\mathbf{d} = (d_0, \ldots, d_c) \in \mathbb{Z}_{\geq 0}^{c+1}$, and define the ring homomorphism φ by*

$$\begin{aligned} \varphi : \mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\leq \mathbf{d}} &\to \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)} \\ k_{i,j} &\mapsto y_i \\ x_i &\mapsto x_i, \end{aligned} \tag{26}$$

where $(k_{1,j}, \ldots, k_{r,j})$ is the j-th column of $\mathbf{K}_{[1..c]}$. Then φ is surjective.

Proof. A monomial of multi-degree at most $\mathbf{d}$ has degree at most d_0 in $\mathbf{x}$ and at most D in the kernel variables, so $\varphi(\mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\leq \mathbf{d}}) \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)}$. Conversely, let $\mathbf{x}^{\boldsymbol{\alpha}} \mathbf{y}^{\boldsymbol{\beta}} \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)}$ with $|\alpha| \leq d_0$ and $|\beta| \leq D$. Consider c distinct lists $L_1, \ldots, L_c$. Append each linear factor y_i of $\mathbf{y}^{\boldsymbol{\beta}}$ ($|\boldsymbol{\beta}| = \beta_1 + \ldots + \beta_r$ in total) to one list L_j so that $\mathbf{y}^{\boldsymbol{\beta}} = \prod_{z \in L_1 \| \cdots \| L_c} z$ and the length of L_j is at most d_j. This assignment is possible because $|\boldsymbol{\beta}| \leq d_1 + \cdots + d_c$. Replacing each factor y_i in $\mathbf{y}^{\boldsymbol{\beta}}$ assigned the list L_j by $k_{i,j}$ gives a monomial m of multi-degree at most $(0, d_1, \ldots, d_c)$ such that $\varphi(\mathbf{x}^{\alpha} \mathrm{m}) = \mathbf{x}^{\alpha} \mathbf{y}^{\beta}$. □

Let $\mathsf{eq}_{\ell,j}$ be the KS equations as in (25). The row space of $\mathbf{A}$ is

$$\mathcal{A}_{\leq \mathbf{d}} := \mathsf{Span}\{\mathrm{m} \cdot \mathsf{eq}_{\ell,j} : \ \mathrm{m} \cdot \mathsf{eq}_{\ell,j} \in \mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\leq \mathbf{d}}\},$$

so the corank of $\mathbf{A}$ is $\mathsf{dim}\mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\leq \mathbf{d}} - \mathsf{dim}(\mathcal{A}_{\leq \mathbf{d}})$.

The space $\varphi(\mathcal{A}_{\leq \mathbf{d}}) \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)}$, with $D = d_1 + \ldots + d_c$ and $d_0, d_1, \ldots, d_c \geq 1$, is given by

$$\begin{aligned} \varphi(\mathcal{A}_{\leq \mathbf{d}}) &= \mathsf{Span}\{\varphi(\mathrm{m}) \cdot \varphi(\mathsf{eq}_{\ell,j}) : \ \mathrm{m} \cdot \mathsf{eq}_{\ell,j} \in \mathcal{A}_{\leq \mathbf{d}}\} \\ &= \mathsf{Span}\{\tilde{\mathrm{m}} \cdot \varphi(\mathsf{eq}_{\ell,j}) : \ \tilde{\mathrm{m}} \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0 - 1, D-1)}\} \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0, D)}. \end{aligned}$$

The monomials $\varphi(\mathrm{m})$ belong to $\mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\leq (d_0 - 1, D-1)}$ and

$$\varphi(\mathsf{eq}_{\ell,j}) = \sigma_{\ell,j}(\mathbf{x}) + g_{\ell}(\mathbf{x}, \mathbf{y}), \tag{27}$$

where $g_{\ell}(\mathbf{x}, \mathbf{y}) := \sum_{i=1}^{r} y_i \, \sigma_{\ell, n-r+i}(\mathbf{x})$ is bilinear in $\mathbf{x}$ and $\mathbf{y}$, and does not depend on j.

Theorem 1. *Let $\mathbf{\Sigma_x}$ be as in Algorithm 12 and $1 \le c \le n - r$. Given $\mathbf{d} = (d_0, \ldots, d_c)$ with $d_0, \ldots, d_c \ge 1$ and the polynomials g_ℓ as in (27), define*

$$Q_D := \mathsf{Span}\{\tilde{\mathrm{m}}\, g_\ell : \deg_{\mathbf{x}} \tilde{\mathrm{m}} \le d_0 - 1,\ \deg_{\mathbf{y}} \tilde{\mathrm{m}} = D - 1\} \subseteq U,$$

where $D = d_1 + \ldots + d_c$ and U denotes the space of polynomials of degree at most d_0 in $\mathbf{x}$ and exactly D in $\mathbf{y}$. Then, the corank of the Macaulay matrix $\mathbf{A}$ at step 4 of Algorithm 12 is at least $\mathsf{dim}(U/Q_D)$.

Proof. By Lemma 1, φ induces a surjection

$$\mathbb{F}_q[\mathbf{x}, \mathbf{K}_{[1..c]}]_{\le \mathbf{d}} / \mathcal{A}_{\le \mathbf{d}} \to \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D)} / \varphi(\mathcal{A}_{\le \mathbf{d}}).$$

Hence the corank of $\mathbf{A}$ is at least $\mathsf{dim}(\mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D)} / \varphi(\mathcal{A}_{\le \mathbf{d}}))$.

For each $\ell = 1, \ldots, m$, $\varphi(\mathcal{A}_{\le \mathbf{d}})$ contains one bilinear equation $\varphi(\mathsf{eq}_{\ell,1})$ and the $c - 1$ linear ones

$$l_{\ell,j} := \varphi(\mathsf{eq}_{\ell,j}) - \varphi(\mathsf{eq}_{\ell,1}) = \sigma_{\ell,j}(\mathbf{x}) - \sigma_{\ell,1}(\mathbf{x}), \qquad j = 2, \ldots, c.$$

Then $\varphi(\mathcal{A}_{\le \mathbf{d}}) = L + Q$, where

$$L := \mathsf{Span}\{\tilde{\mathrm{m}} \cdot l_{\ell,j} :\ \tilde{\mathrm{m}} \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0 - 1, D-1)}\} \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)}.$$
$$Q := \mathsf{Span}\{\tilde{\mathrm{m}} \cdot \varphi(\mathsf{eq}_{\ell,1}) :\ \tilde{\mathrm{m}} \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0 - 1, D-1)}\} \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D)}.$$

Since $L \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)}$ and $\tilde{\mathrm{m}}\, \varphi(\mathsf{eq}_{\ell,1}) \in \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)} + Q_D$, we have $\varphi(\mathcal{A}_{\le \mathbf{d}}) \subseteq \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)} \oplus Q_D$. Consequently,

$$\mathsf{dim}(U/Q_D) \le \mathsf{dim}(\mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D)} / \varphi(\mathcal{A}_{\le \mathbf{d}})),$$

where equality holds when $L = \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)}$, since

$$\mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D)} = \mathbb{F}_q[\mathbf{x}, \mathbf{y}]_{\le (d_0, D-1)} \oplus U.$$

□

In other words, if

$$B(\mathbf{d}) := \#\{\text{monomials of degree at most } d_0 \text{ in } \mathbf{x} \text{ and exactly } D \text{ in } \mathbf{y}\} - \mathsf{rank}\{\mathrm{m}\, g_\ell : \deg_{\mathbf{x}}(\mathrm{m}) \le d_0 - 1,\ \deg_{\mathbf{y}}(\mathrm{m}) = D - 1\},$$

then, by Theorem 1, the corank of $\mathbf{A}$ is at least $B(\mathbf{d})$.

We estimate $B(\mathbf{d})$ using the Hilbert function in bidegree (d_0, D) of m generic bilinear forms in $k + 1$ and r variables, which is

$$\mathcal{H}^{\mathrm{top}}_{m,k,r}(t_0, t_1) := \frac{(1 - t_0 t_1)^m}{(1 - t_0)^{k+1}(1 - t_1)^r}. \tag{28}$$

We estimate $B(\mathbf{d})$ by the coefficient of $t_0^{d_0} t_1^D$ if all coefficients of $t_0^a t_1^b$ with $1 \le a \le d_0$ and $1 \le b \le D$ are positive, and by 0 otherwise.

We select the arguments c and $\mathbf{d}_{\mathrm{sol}} = (d_0, \ldots, d_c)$ of MULTI-XL$_{\mathrm{KS}}$ by the following conditions.

1. $c \geq 2$ and $k \leq m - r$, so that the g_ℓ have no common zero; this implies that the KS system is overdefined and that $(c-1)m \geq k+1$,
2. $d_0 \leq r$, to avoid the syzygies coming from the Jacobian with respect to any $\mathbf{k}_j$ [43],
3. $\mathbf{d}_{\text{sol}}$ has at most $r+1$ nonzero entries, to avoid the syzygies coming from the Jacobian with respect to $\mathbf{x}$ [43],
4. the coefficient $a_{\mathbf{d}_{\text{sol}}}$ of $\mathbf{t}^{\mathbf{d}_{\text{sol}}}$ in $\mathcal{H}(\mathbf{t})$ is nonpositive,
5. $B(\mathbf{d}_{\text{sol}}) = 0$.

Conditions 2 and 3 exclude the syzygies of Verbel et al. [43] of multi-degree $(r+1)\mathbf{e}_0 + \mathbf{e}_j$, coming from the Jacobian with respect to any $\mathbf{k}_j$, and those of multi-degree $\mathbf{e}_0 + \mathbf{e}_{j_1} + \cdots + \mathbf{e}_{j_{r+1}}$, which involve $r+1$ kernel columns, come from the left kernel of $\mathbf{I}_m \otimes \mathbf{K}_{[1..c]}^\top$ and exist whenever $r < c$. Condition 4 is the selection rule for a generic system. Condition 5 excludes the multi-degrees at which Theorem 1 forces a corank larger than one.

Assumption 3 *For a random MinRank problem $\mathbf{M}_0, \ldots, \mathbf{M}_k$ with a unique planted solution and parameters (q, m, n, k, r), and for c and $\mathbf{d}_{sol}$ satisfying conditions 1–5, the corank of the Macaulay matrix $\mathbf{A}$ at step 4 of Algorithm 12 is 1 with high probability. More generally, for $\mathbf{d} \leq \mathbf{d}_{sol}$ the corank is $\max(a_{\mathbf{d}}, 1 + B(\mathbf{d}))$ with high probability.*

Under Assumption 3, the right kernel of $\mathbf{A}$ at $\mathbf{d}_{\text{sol}}$ is spanned by the evaluation vector of the solution, and ExtractSolution reads $\mathbf{s}$ from its coordinates at the monomials $x_1, \ldots, x_k$, divided by its coordinate at the monomial 1.

Experimental verification of Assumption 3. Table 5 reports the corank of $\mathbf{A}$ for KS systems of random MinRank instances with a planted solution, together with $a_{\mathbf{d}}$, $B(\mathbf{d})$ and the prediction of Assumption 3. The rows marked † have $k > m - r$ and test the formula $\max(a_{\mathbf{d}}, 1 + B(\mathbf{d}))$ beyond the hypotheses of the assumption. Over $\mathbb{F}_{16}$ the corank equals $\max(a_{\mathbf{d}}, 1 + B(\mathbf{d}))$ in every instance. Over $\mathbb{F}_2$ it is occasionally slightly larger, as it is for random systems, which we attribute to the small field size. In every row, the exact value of $B(\mathbf{d})$ on the instance agrees with its series estimate.

7.5 Complexity Estimates

In this section we give the cost of Multi-XL_{KS} under Assumption 3 and the resulting estimates for the Mirath parameter sets. Given $(\mathbf{M}_0, \ldots, \mathbf{M}_k, r, c, \mathbf{d}_{\text{sol}})$ with $\mathbf{d}_{\text{sol}} = (d_0, \ldots, d_c)$, the number of $\mathbb{F}_q$-multiplications of Multi-XL_{KS} with the Wiedemann algorithm (1) is

$$\mathcal{C}_{\text{KS}} = 3(k+1)(r+1)(\text{N}^{\text{cols}}_{\mathbf{d}_{\text{sol}},c})^2, \qquad \text{N}^{\text{cols}}_{\mathbf{d}_{\text{sol}},c} = \binom{k+d_0}{d_0} \prod_{j=1}^{c} \binom{r+d_j}{d_j}, \tag{29}$$

q	m	n	k	r	c	$\mathbf{d}$	$a_{\mathbf{d}}$	$B(\mathbf{d})$	$\max(a_{\mathbf{d}}, 1+B(\mathbf{d}))$	corank
16	19	9	5	5	2	(1,1,1)	-12	0	1	1
16	42	8	15	4	2	(1,1,1)	-20	0	1	1
16	10	5	$10^{\dagger}$	2	3	(1,1,1,2)	-36	15	16	16
16	12	8	8	3	3	(1,1,1,1)	0	18	19	19
16	12	8	7	3	3	(1,1,1,1)	-64	8	9	9
16	8	5	4	3	2	(1,2,1)	-8	2	3	3
16	10	6	5	4	2	(1,3,1)	-50	10	11	11
16	7	7	3	4	3	(1,1,1,1)	-25	10	11	11
16	11	8	4	5	3	(1,1,1,1)	-108	10	11	11
2	42	8	15	4	2	(1,1,1)	-20	0	1	1–6
2	50	10	24	5	2	(1,3,1)	-700	0	1	1–9
2	50	10	24	5	3	(1,1,1,1)	0	125	126	126–129
16	8	5	4	3	2	(2,1,1)	-16	0	1	1
16	8	5	4	3	2	(2,2,1)	-72	0	1	1
16	7	7	3	4	3	(2,1,1,1)	-115	4	5	5
16	5	6	$3^{\dagger}$	3	3	(2,1,1,1)	-20	10	11	11
16	5	5	$4^{\dagger}$	2	2	(2,2,2)	-15	5	6	6
16	5	5	$4^{\dagger}$	2	2	(2,3,1)	-10	5	6	6
16	6	6	$6^{\dagger}$	2	2	(2,2,2)	0	17	18	18
16	6	6	$6^{\dagger}$	2	2	(2,2,1)	27	16	27	27
16	6	6	$6^{\dagger}$	2	2	(2,1,1)	36	15	36	36

Table 5: Verification of Assumption 3: corank of the Macaulay matrix of KS systems of random MinRank instances with a planted solution, at multi-degrees with $d_0 = 1$ (upper part) and $d_0 \geq 2$ (lower part), compared with $a_{\mathbf{d}}$ and $B(\mathbf{d})$ (2 to 10 instances each; a range gives the smallest and largest corank observed). Rows marked † have $k > m - r$ and lie outside the hypotheses of the assumption.

where $\mathrm{N}^{\mathrm{cols}}_{\mathbf{d}_{\mathrm{sol}},c}$ is the number of columns of $\mathbf{A}$, that is, of monomials of multi-degree at most $\mathbf{d}_{\mathrm{sol}}$, and $(k+1)(r+1)$ is the number of monomials of each KS equation, that is, the row density of $\mathbf{A}$ in (1).

Table 6 presents the resulting bit-complexity estimates for the Mirath parameter sets [1], combined with the hybrid approach of Section 3: a kernel vectors are guessed, the KS system of the reduced instance $(q, m, n-a, k-am-l_v, r)$ is solved with c and $\mathbf{d}_{\mathrm{sol}}$ chosen by conditions 1–5 so as to minimize $\mathrm{N}^{\mathrm{cols}}_{\mathbf{d}_{\mathrm{sol}},c}$, and the cost is given by (4) with $\mathcal{C}_{\mathrm{MR}} = \mathcal{C}_{\mathrm{KS}}$, scaled by $2(\log_2 q)^2 + \log_2 q$.

8 Conclusion and Further Directions

We revisited the combinatorial and algebraic attacks on MinRank. On the combinatorial side, the improved kernel search lowers the cost of the best attack on four of the six Mirath parameter sets, and we solved Courtois challenge D after 25 years. On the algebraic side, the field equations extend the Support-Minors and Minors attacks to instances and degrees where the original analyses do not apply. The Kipnis–Shamir attack turned out to be less understood than

q	m	n	k	r	a	l_v	Kipnis-Shamir			Best other attack		
							c	$\mathbf{d}_{\text{sol}}$	Cost	Attack	a	Cost
16	16	16	143	4	8	3	2	(4,1,1)	183	$\text{Kernel}_{\text{imp}}$	8	160
16	19	19	195	5	10	0	2	(1,1,1)	228	SM	10	227
16	22	22	255	6	11	0	2	(3,1,3)	314	SM	7	299
2	42	42	1443	4	34	0	2	(1,1,1)	167	$\text{Kernel}_{\text{imp}}$	31	159
2	50	50	2024	5	40	0	2	(1,1,3)	236	$\text{Kernel}_{\text{imp}}$	38	224
2	56	56	2499	6	44	8	2	(1,1,4)	313	$\text{Kernel}_{\text{imp}}$	42	289

Table 6: Bit-complexity estimates ($\log_2$ of the number of gates) of the Kipnis-Shamir attack against Mirath parameters with the hybrid approach, which guesses a kernel vectors and l_v entries of the solution vector, and the lowest estimate among the other attacks of this paper ($\text{Kernel}_{\text{imp}}$: improved kernel search; SM: Support-Minors with field equations).

the literature suggested: the generic Hilbert series does not predict the rank of its Macaulay matrices, because identifying the c kernel columns collapses the cm equations onto m quadratic parts, and Theorem 1 turns this collapse into a lower bound $B(\mathbf{d})$ on the corank that is tight in all our experiments. Two consequences follow. The multi-degrees selected by the generic prediction fail for three Mirath parameter sets, and the attack can only succeed on an instance with $k \leq m - r$, which the hybrid approach must first produce by guessing kernel vectors and linear variables. With the corrected selection, the Kipnis–Shamir attack is 1 to 24 bits above the best of the other attacks on every Mirath parameter set (Table 6). The resulting estimates are in the MinRank estimator of the CryptographicEstimators library.

Two questions are left open. The first is a proof of the rank predictions we rely on: Assumptions 1 and 2 for the field equations, and Assumption 3 for the KS equations, where the corank is only bounded, and predicted by $\max(a_{\mathbf{d}}, 1 + B(\mathbf{d}))$; a Hilbert series of the KS equations that accounts for the syzygies of Section 7.3 at every multi-degree would settle it. The second is whether the KS equations can be solved below the cost of the multi-homogeneous XL analyzed here, for instance by exploiting the degree falls that the same syzygies produce.

9 Acknowledgements

The authors wish to thank Dustin Moody for his insightful suggestions during the preparation of this manuscript. Computer time for this study was provided by the computing facilities of the MCIA (Mésocentre de Calcul Intensif Aquitain).

10 AI Disclosure

We used the Anthropic models Claude Opus 5.5 and Claude Fable 5.1 to improve the editorial quality of this document and the presentation of the accompanying

artifacts, which the authors first implemented themselves. We also used these models to derive the initial ideas of the corank analysis of MULTI-XL_{KS} in Section 7.4. The authors take full responsibility for the results presented in this document and its artifacts.

References

1. Adj, G., Aragon, N., Barbero, S., Bardet, M., Bellini, E., Bidoux, L., Chi-Domínguez, J.J., Dyseryn, V., Esser, A., Feneuil, T., Gaborit, P., Neveu, R., Rivain, M., Luis Rivera-Zamarripa, C.S., Tillich, J.P., Verbel, J., Zweydinger, F.: Mirath signature scheme, Round 2 submission (2025), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/mirath-spec-round2-web.pdf
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015)
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. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part III. LNCS, vol. 13509, pp. 376–405. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15982-4_13
4. Bardet, M., Bertin, M.: Improvement of algebraic attacks for solving superdetermined MinRank instances. In: Cheon, J.H., Johansson, T. (eds.) PQCrypto 2022. pp. 107–123. Springer, Cham (Sep 2022). https://doi.org/10.1007/978-3-031-17234-2_6
5. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Tillich, J.P.: Revisiting algebraic attacks on MinRank and on the rank decoding problem. Designs, Codes and Cryptography **91**(11), 3671–3707 (2023). https://doi.org/10.1007/s10623-023-01265-x, https://doi.org/10.1007/s10623-023-01265-x
6. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R.A., Smith-Tone, D., Tillich, J.P., Verbel, J.A.: Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 507–536. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_17
7. Bardet, M., Gilard, A.: Computation of the hilbert series for the Support-Minors modeling of the MinRank problem (2025), https://arxiv.org/abs/2502.12721
8. Bellini, E., Makarim, R.H., Sanna, C., Verbel, J.A.: An estimator for the hardness of the MQ problem. In: Batina, L., Daemen, J. (eds.) AFRICACRYPT 22. LNCS, vol. 2022, pp. 323–347. Springer, Cham (Jul 2022). https://doi.org/10.1007/978-3-031-17433-9_14
9. Beullens, W., Campos, F., Celi, S., Hess, B., Kannwischer, M.J.: MAYO – round 3 version (August 2026), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/mayo_specification.pdf, available at NIST CSRC
10. Beullens, W., Chen, M.S., Ding, J., Gong, B., Kannwischer, M.J., Patarin, J., Peng, B.Y., Schmidt, D., Shih, C.J., Tao, C., Yang, B.Y.: UOV: Specification document – round 2 (August 2026), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/UOV_Specification_R3.pdf, available at NIST CSRC
11. Bidoux, L., Feneuil, T., Gaborit, P., Neveu, R., Rivain, M.: Dual support decomposition in the head: Shorter signatures from rank SD and MinRank. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part II. LNCS, vol. 15485, pp. 38–69.

Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0888-1_2
12. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The computational complexity of some problems of linear algebra. J. Comput. Syst. Sci. **58**(3), 572–596 (1999). https://doi.org/10.1006/JCSS.1998.1608, https://doi.org/10.1006/jcss.1998.1608
13. Cabarcas, D., Gaggero, G., Gorla, E.: The complexity of the Support-Minors modeling for the MinRank problem. Designs, Codes and Cryptography **94**(4), 73 (2026). https://doi.org/10.1007/s10623-025-01800-y, https://doi.org/10.1007/s10623-025-01800-y
14. Cabarcas, D., Li, P., Verbel, J.A., Villanueva-Polanco, R.: Improved attacks for SNOVA by exploiting stability under a group action. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VI. LNCS, vol. 16005, pp. 358–389. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01887-8_12
15. Chatterjee, R., Mu, C., Vasudevan, P.N.: Public-key encryption from the MinRank problem. In: Daemen, J., Thomé, E. (eds.) EUROCRYPT 2026, Part IV. LNCS, vol. 16545, pp. 449–480. Springer, Cham (May 2026). https://doi.org/10.1007/978-3-032-25327-9_16
16. Cheng, C.M., Chou, T., Niederhagen, R., Yang, B.Y.: Solving quadratic equations with XL on parallel architectures. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 356–373. Springer, Berlin, Heidelberg (Sep 2012). https://doi.org/10.1007/978-3-642-33027-8_21
17. Courtois, N.: Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 402–421. Springer, Berlin, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_24
18. Courtois, N.: La sécurité des primitives cryptographiques basées sur les problemes algébriques multivariables MQ, IP, MinRank, et HFE. Université Paris VI France (2001)
19. Debris-Alazard, T., Gaborit, P., Neveu, R., Ruatta, O.: A minrank-based encryption scheme à la alekhnovich-regev. In: Daemen, J., Thomé, E. (eds.) EUROCRYPT 2026, Part IV. LNCS, vol. 16545, pp. 481–510. Springer, Cham (May 2026). https://doi.org/10.1007/978-3-032-25327-9_17
20. Ding, J., Chen, M.S., Kannwischer, M., Patarin, J., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow – algorithm specification and documentation. the 3rd round proposal. Submission to the NIST Post-Quantum Cryptography Standardization Process (2020), https://www.pqcrainbow.org/
21. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y., Kannwischer, M., Patarin, J.: Rainbow: Proposal for NISTPQC: Digital signature algorithms version 3.0 (October 2020), https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions, available at NIST CSRC
22. Esser, A., Bellini, E.: Syndrome decoding estimator. In: Public-Key Cryptography–PKC 2022: 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8–11, 2022, Proceedings, Part I. pp. 112–141. Springer (2022)
23. Esser, A., Verbel, J.A., Zweydinger, F., Bellini, E.: SoK: CryptographicEstimators - a software library for cryptographic hardness estimation. In: Zhou, J., Quek, T.Q.S., Gao, D., Cárdenas, A.A. (eds.) ASIACCS 24. ACM Press (Jul 2024). https://doi.org/10.1145/3634737.3645007
24. Faugère, J.C.: A new efficient algorithm for computing gröbner bases (F4). Journal of Pure and Applied Algebra **139**(1–3), 61–88 (1999)

25. Faugère, J.C.: A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation. p. 75–83. ISSAC '02, Association for Computing Machinery, New York, NY, USA (2002)
26. Faugère, J., Din, M.S.E., Spaenlehauer, P.: On the complexity of the generalized minrank problem. J. Symb. Comput. **55**, 30–58 (2013)
27. Faugère, J.C., El Din, M.S., Spaenlehauer, P.J.: Computing loci of rank defects of linear matrices using gröbner bases and applications to cryptology. In: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation. pp. 257–264. ACM (2010)
28. Faugère, J.C., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of minrank. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 280–296. Springer, Berlin, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_16
29. Furue, H., Ikematsu, Y., Hoshino, F., Takagi, T., Sugai, R., Amagasa, H., Homma, N., Ueno, R., Sakata, K., Kosuge, H., Yamakoshi, K., Kinjo, K., Akiyama, R., Nakamura, S., Origara, S., Kawahara, Y.: QR-UOV (August 2026), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/qruov_specifications.pdf, available at NIST CSRC
30. Gopalakrishnan, S., Neiger, V., Safey El Din, M.: Refined f5 algorithms for ideals of minors of square matrices. In: Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation. p. 270–279. ISSAC '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3597066.3597077
31. Goubin, L., Courtois, N.T.: Cryptanalysis of the TTM Cryptosystem. In: Advances in Cryptology — ASIACRYPT 2000. pp. 44–57. Springer Berlin Heidelberg (2000)
32. Guo, H., Ding, J.: Algebraic relation of three MinRank algebraic modelings. In: Mesnager, S., Zhou, Z. (eds.) Arithmetic of Finite Fields. pp. 239–249. Springer International Publishing, Cham (2023)
33. Jeannerod, C.P., Pernet, C., Storjohann, A.: Rank-profile revealing Gaussian elimination and the CUP matrix decomposition. Journal of Symbolic Computation **56**, 46–68 (2013). https://doi.org/10.1016/j.jsc.2013.04.004
34. Kaltofen, E.: Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. Mathematics of Computation **64**(210), 777–806 (1995). https://doi.org/10.1090/S0025-5718-1995-1270621-1
35. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 19–30. Springer, Berlin, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1_2
36. Nakamura, S., Wang, Y., Ikematsu, Y.: A new analysis of the Kipnis–Shamir method solving the MinRank problem. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E106.A**(3), 203–211 (2023). https://doi.org/10.1587/transfun.2022CIP0014
37. Ning, K.C., Ran, L., Samardjiska, S.: Multi-homogeneous XL. Cryptology ePrint Archive, Paper 2025/2060 (2025), https://eprint.iacr.org/2025/2060
38. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 33–48. Springer, Berlin, Heidelberg (May 1996). https://doi.org/10.1007/3-540-68339-9_4
39. Perlner, R., Smith-Tone, D.: Rainbow band separation is better than we thought. Cryptology ePrint Archive, Paper 2020/702 (2020), https://eprint.iacr.org/2020/702

40. Smith-Tone, D.: The generating series of support minors MinRank ideals. Cryptology ePrint Archive, Paper 2023/1456 (2023), https://eprint.iacr.org/2023/1456
41. Steel, A.: Direct solution of the (11,9,8)-MinRank problem by the Block Wiedemann algorithm in Magma with a Tesla GPU. In: Proceedings of the 2015 International Workshop on Parallel Symbolic Computation. PASCO '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2790282.2791392
42. Strassen, V.: Gaussian elimination is not optimal. Numerische Mathematik **13**(4), 354–356 (1969). https://doi.org/10.1007/BF02165411
43. Verbel, J.A., Baena, J., Cabarcas, D., Perlner, R.A., Smith-Tone, D.: On the complexity of "superdetermined" minrank instances. In: Ding, J., Steinwandt, R. (eds.) PQCrypto 2019. pp. 167–186. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_10
44. Wang, L.C., Chou, C.Y., Ding, J., Kuan, Y.L., Leegwater, J.A., Li, M.S., Tseng, B.S., Tseng, P.E., Wang, C.C.: SNOVA: Specification Document – Round 2 (August 2026), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/SNOVA_specifications.pdf, available at NIST CSRC
45. Wiedemann, D.H.: Solving sparse linear equations over finite fields. IEEE Transactions on Information Theory **32**(1), 54–62 (1986). https://doi.org/10.1109/TIT.1986.1057137