Subspace Guessing and Rank-Metric Solvers with Hints *

Anmoal Porwal[1], Harrison Banda[2], Jan Brinkmann[2],
Anna Baumeister[1], Juliane Krämer[2], Antonia Wachter-Zeh[1]

[1]Department of Computer Engineering, Technical University of Munich, Munich, Germany
[2]Faculty of Informatics and Data Science, University of Regensburg, Regensburg, Germany

anmoal.porwal(at)tum.de, harrison.banda(at)ur.de, jan.brinkmann(at)ur.de,
anna.baumeister(at)tum.de, juliane.kraemer(at)ur.de,
antonia.wachter-zeh(at)tum.de

Abstract

We show how to improve rank-metric solvers when certain side information (hints) about the secret is available. Concretely, we adapt the kernel search algorithm for MinRank and the GRS algorithm for the Rank Syndrome Decoding problem when some entries in the rank decomposition of the error matrix are known. This setting is motivated by side-channel leakage and cryptographic applications: Mirath and RYDE, two signature candidates in the NIST post-quantum competition, rely on these problems and employ secret keys in this decomposed form. As a main technical ingredient, we give an optimal procedure for guessing a subspace containing the row space of a systematic matrix given only partial knowledge of its entries. Further, we describe a profiling side-channel attack on the reference implementation of Mirath to demonstrate the plausibility of obtaining such hints.

Keywords: rank-metric, coding theory, cryptography, side-channel.

1 Introduction

The MinRank and the Rank Syndrome Decoding (RSD) problems form the basis of the security of many recently proposed cryptographic schemes and, in particular, of Mirath [AAB+25] and RYDE [ABB+25] which are two digital signature candidates in the current NIST post-quantum standardization process [ABC+24]. Understanding the complexity of solving these problems is integral to determining the security level of these schemes. More generally, there is significant interest in understanding how this complexity changes when partial information (*hints*) about the secret becomes available, for instance through side-channel leakage.

These leakage settings have been extensively studied in cryptography, both for classical schemes (e.g., RSA and Diffie–Hellman), see [MH24], and for post-quantum schemes, see, e.g., [EMVW22]. These works typically focus on bit-level hints (also called *partial key exposure* attacks), i.e., knowledge of some random bits of the secret. Other forms of hints that are not purely bit-level have also been considered, for example by Dachman-Soled et al. [DSDGR20] for lattice problems and by Horlemann et al. [HPR+22] for the Syndrome Decoding problem in the Hamming metric. As an example, the latter considers hints such as knowing the $\mathbb{F}_q$-entries of the error vector or weights of its subvectors.

In both MinRank and RSD, the secret can be represented as an $m \times n$ matrix $\mathbf{E}$ over $\mathbb{F}_q$ of small rank r. In cryptographic implementations, $\mathbf{E}$ is often represented via a rank decomposition with a systematic part

*This work has been supported by the ANR-DFG project "CROWD: Cryptography with Skew Codes" (project numbers WA3907/9-1 and KR5152/2-1), Deutsche Forschungsgemeinschaft (DFG) – KR 5152/2-1 and SFB 1119 – 236615297, and the German Federal Ministry of Research, Technology and Space (BMFTR) under the project QUORYPTAN (16KIS2034).

Table 1: Rank-Metric Hints Considered in the Literature

Work[a]	Problem	Hint[b]	Method	Motivation
[DEGS25]	RSD	Bits of $\boldsymbol{e} \in \mathbb{F}_{2^\nu}^n$	Adapt GRS	Older RYDE versions used $\boldsymbol{e}$ as secret key
[DEGS25]	RSD	Bits of the coefficients of the q-polynomial that annihilates $\boldsymbol{e} \in \mathbb{F}_{2^\nu}^n$	Reduce to MinRank	Older RYDE versions used this q-polynomial as the signing witness
[DEGS25]	MinRank	Bits of $\boldsymbol{x} \in \mathbb{F}_2^n$	Reduce to MinRank	MIRA and MiRiTH (older NIST candidates) used $\boldsymbol{x}$ as secret key
[DEGS25]	MinRank	Bits of the coefficients of the q-polynomial that annihilates $\mathbf{E} \in \mathbb{F}_2^{m\times n}$ when $\mathbf{E}$ is represented as an element of $\mathbb{F}_{2^\nu}^n$	Reduce to MinRank	MIRA used this q-polynomial as the signing witness
This work	MinRank	$\mathbb{F}_q$-entries or bits of $\mathbf{S}$ and $\mathbf{C}'$ in $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{m\times n}$	Adapt kernel search	Mirath uses $(\mathbf{S}, \mathbf{C}')$ as secret key
This work	RSD	$\mathbb{F}_q$-entries or bits of $\mathbf{S}$ and $\mathbf{C}'$ in $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{m\times n}$ where $\mathbf{E}$ is the matrix representation of $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$	Adapt GRS	RYDE uses $(\boldsymbol{s}, \mathbf{C}')$ as secret key where[c] $\boldsymbol{s} \in \mathbb{F}_{q^m}^r$ is the vector representation of $\mathbf{S} \in \mathbb{F}_q^{m\times r}$

[a]The work [DEGS25] also describe the *error* model, where some hints may be corrupted, unlike the *erasure* model, where hints are always correct (as in our work). Their proposed algorithm is to guess which hints are correct and then run the erasure-model solver, thus effectively reducing the problem to the erasure setting.
[b]See Sections 3 and 4 for definitions of $\boldsymbol{e}$, $\boldsymbol{x}$, etc.
[c]In addition to choosing $\mathbf{C}$ systematic, RYDE also specializes the first entry of $\boldsymbol{s}$ to 1.
Note this table is an expanded version of the corresponding table in the published ISIT version.

which is more efficient in terms of key size, for example:

$$\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}') \text{ where } \mathbf{S} \in \mathbb{F}_q^{m\times r},\ \mathbf{C}' \in \mathbb{F}_q^{r\times(n-r)}.$$

In this work, the hints consist of partial knowledge of the $\mathbb{F}_q$-entries of $\mathbf{S}$ and $\mathbf{C}'$. We also consider the *bit-level* variants, i.e., knowledge of some bits of some $\mathbb{F}_q$-entries (when $q = 2^\nu$). These choices are motivated by the fact that the secret key in both Mirath and RYDE is of the form $(\mathbf{S}, \mathbf{C}') \in \mathbb{F}_q^{m\times r} \times \mathbb{F}_q^{r\times(n-r)}$, making it plausible for a side-channel attack to reveal such information.

Concretely, we adapt the kernel search [GC00] solver for MinRank and the GRS solver [GRS16] for the RSD problem under the hints model mentioned above. These two combinatorial algorithms form one of the two main classes of rank-metric solvers for cryptographically relevant parameters (the other being the algebraic solvers of [BBC$^+$20, BBB$^+$23]).

The only prior work to our knowledge that considers rank-metric hints is the recent work of D'Alconzo et al. [DEGS25]. All the hints analyzed by them are bit-level. We list them alongside our hints in Table 1 for easier comparison.

A Contribution

1) Subspace Guessing with Hints (Section 2)

As the main building block (Theorem 1), we solve the following problem, which may be of independent interest: Given a random systematic matrix $\mathbf{C} := (\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{r \times n}$ and $\ell > 0$ with some entries of $\mathbf{C}'$ unknown, find a subspace of dimension $r + \ell$ that contains the row space of $\mathbf{C}$ with highest probability. For example, $\mathbf{C}$ might be

$$\left[\begin{array}{ccc|ccccc} 1 & 0 & 0 & 1 & * & 2 & * & * \\ 0 & 1 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 1 & * & 1 & * & * & * \end{array}\right] \in \mathbb{F}_3^{3\times 8},$$

where the $*$ represent unknown entries.

2) Kernel Search and GRS with Hints (Sections 3, 4; Appendices A, B)

The MinRank problem asks to find a matrix $\mathbf{E}$ of small rank r in a space spanned by K matrices in $\mathbb{F}_q^{m\times n}$. Then, for typical parameters, the complexities of our modified kernel search algorithm with hints and (for comparison) the original kernel search are

$$\tilde{K}^\omega q^{r\lceil \tilde{K}/m\rceil - h_c + t_c} \quad \text{and} \quad K^\omega q^{r\lceil K/m\rceil}$$

respectively. Here h_s denotes the number of known entries in $\mathbf{S}$, $\tilde{K} := \max\{0, K - h_s\}$, h_c is the number of known entries in $\mathbf{C}'$ and t_c is the number of known entries within any $\ell := n - r - \lceil \tilde{K}/m\rceil$ columns of $\mathbf{C}'$ with the fewest hints. Section 3 describes the complexity for bit-level hints.

The RSD problem asks to find an error $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ of small rank r over $\mathbb{F}_q$ that satisfies a certain parity-check equation. This error can also be represented as a matrix $\mathbf{E} \in \mathbb{F}_q^{m\times n}$ of rank r. Upper bounds on the complexities of our modified GRS algorithm (for the "row space" version) and (for comparison) the original approach are

$$\begin{aligned} &(n-k)^\omega m^\omega q^{r(k-\lfloor h_s/m\rfloor)-h_c+t_c} \\ \text{and } &(n-k)^\omega m^\omega q^{rk} \end{aligned}$$

respectively.[1] Here h_s, h_c and t_c have similar meanings as in the previous part but with $\ell := n-k-r+\lfloor h_s/m\rfloor$. See Section 4 for the complexities of the "column-space" version and bit-level hints.

Appendix A states all our modified algorithms in full with their precise complexities. Appendix B lists the concrete costs for various hint fractions of all Mirath and RYDE parameter sets.

Limitations Both kernel search and GRS can guess on either $\mathbf{C}$ or $\mathbf{S}$. When $m = n$ this choice does not matter, but for other parameter sets it can affect the complexity. However, our method of incorporating hints only allows using the matrix that is in systematic form which may not be optimal depending on m and n. (Although we note that cryptographic schemes typically use $m = n$.) Moreover, we adapt only the original GRS algorithm [GRS16], and not the improvement in [AGHT06], which performs better for certain parameter regions. For details, see Section 6.

3) Side-Channel Implementation (Section 5)

To show that this hints setting is plausible, we performed a profiling attack on Mirath [AAB+25] that recovers the entries of $\mathbf{C}$ with high probability. The attack only targets the reference implementation (no optimized

[1] For the original row-GRS algorithm, Proposition 4 gives the cost $P_0 = \max\{m(n-k)rk\min\{r,k\}^{\omega-3}\ \mathbb{F}_q\text{-operations}, (n-k)^\omega\ \mathbb{F}_{q^m}\text{-operations}\}$; the displayed term is a common upper bound.

versions were available) and standard countermeasures such as masking would prevent it. Still, together with our theoretical results, this highlights the need for side-channel countermeasures in rank-metric schemes.

B Organization

Section 2 shows (Theorem 1) how to solve the "subspace guessing with hints" problem. Section 3 defines the MinRank problem and then, using this result, shows how to incorporate hints into the kernel search solver. Section 4 defines the RSD problem and then does the same for the GRS solver. Section 5 describes our side-channel profiling attack. Section 6 concludes with future work. Appendix A gives the full algorithms and cost analyses, and Appendix B gives concrete costs.

C Notation

We assume row vectors. Let $\mathbf{A} \in \mathbb{F}_q^{m \times n}$ be an $m \times n$ matrix over the finite field $\mathbb{F}_q$. Then $A_{i,j} \in \mathbb{F}_q$ is the entry in row i and column j. Denote by $[a : b]$ the set of integers from a to b, inclusive, and let $[b]$ be shorthand for $[1 : b]$. If $\mathcal{I} \subseteq [m]$ and $\mathcal{J} \subseteq [n]$, then $\mathbf{A}_{\mathcal{I},\mathcal{J}} \in \mathbb{F}_q^{|\mathcal{I}| \times |\mathcal{J}|}$ is the submatrix of $\mathbf{A}$ with rows and columns indexed by $\mathcal{I}$ and $\mathcal{J}$ respectively. The notation $\mathbf{A}_{*,\mathcal{J}} \in \mathbb{F}_q^{m \times |\mathcal{J}|}$ denotes the submatrix consisting of columns indexed by $\mathcal{J}$ and all rows, and analogously for $\mathbf{A}_{\mathcal{I},*}$. If $\mathcal{I} \subseteq [m]$, then $\bar{\mathcal{I}}$ denotes $[m] \setminus \mathcal{I}$. We sometimes write $\mathbf{I}_m$ or $\mathbf{0}_{m \times n}$ to make the size of the identity or zero matrix clear. The matrix multiplication exponent is ω.

2 The Subspace Guessing with Hints Problem

Proposition 1 (Optimal Subspace)**.** *Let $\mathbf{C} := (\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{r \times n}$ be a matrix in systematic form, where only some entries of $\mathbf{C}' \in \mathbb{F}_q^{r \times (n-r)}$ are known, and let $0 \leq \ell \leq n - r$. Let $\mathcal{C}$ be the set of all possible matrices $(\mathbf{I}_r \mid \mathbf{C}'') \in \mathbb{F}_q^{r \times n}$ such that $\mathbf{C}'' \in \mathbb{F}_q^{r \times (n-r)}$ agrees with $\mathbf{C}'$ at the known entries. Then there exists an $(r + \ell)$-dimensional subspace $\mathcal{R} \subseteq \mathbb{F}_q^n$ such that*

$$\left|\{\ (\mathbf{I}_r \mid \mathbf{C}'') \in \mathcal{C} : \text{rowspace}(\mathbf{I}_r \mid \mathbf{C}'') \subseteq \mathcal{R}\ \}\right| = q^{r\ell - t_c}$$

and for no other $(r + \ell)$-dimensional $\mathcal{R}$ is this number larger. Here

$$t_c := \min_{\substack{S \subseteq \{1,\ldots,n-r\} \\ |S| = \ell}} \sum_{i \in S} h_i, \tag{1}$$

where h_i is the number of known entries in column i of $\mathbf{C}'$. In other words, t_c is the sum of the ℓ smallest numbers in the list $h_1, \ldots, h_{n-r}$.

Proof. Given $\mathbf{R} \in \mathbb{F}_q^{(r+\ell) \times n}$, we say a matrix $\mathbf{C}'' \in \mathbb{F}_q^{r \times (n-r)}$ is $\mathbf{R}$-*compatible* if $\mathbf{C}''$ agrees with the known entries of $\mathbf{C}' \in \mathbb{F}_q^{r \times (n-r)}$ and $\text{rowspace}(\mathbf{I}_r \mid \mathbf{C}'') \subseteq \text{rowspace}(\mathbf{R})$. Our goal is to construct an $\mathbf{R}$ such that the number of $\mathbf{R}$-compatible matrices is maximized. By way of illustration, suppose

$$(\mathbf{I}_r \mid \mathbf{C}') = \left[\begin{array}{c|ccccc} & 1 & * & 2 & * & * \\ \mathbf{I}_r & 0 & * & * & 0 & * \\ & * & 1 & * & * & * \end{array}\right] \in \mathbb{F}_q^{r \times n}$$

where the symbols $*$ denote unknown entries. If $\mathbf{R}$ has at least one $\mathbf{R}$-compatible matrix $\mathbf{C}''$ then, up to row

operations, $\mathbf{R}$ can always be written in the form

$$\mathbf{R} = \left[\begin{array}{c|ccccc} & 1 & * & 2 & * & * \\ \mathbf{I}_r & 0 & * & * & 0 & * \\ & * & 1 & * & * & * \\ \hline \mathbf{0}_{\ell\times r} & \multicolumn{5}{c}{\mathbf{A}} \\ & & & & & \end{array}\right] \in \mathbb{F}_q^{(r+\ell)\times n}$$

for some $\mathbf{A} \in \mathbb{F}_q^{\ell\times(n-r)}$. This is because rowspace($\mathbf{R}$) contains the rows of $(\mathbf{I}_r \mid \mathbf{C}'')$ and we may as well assume these are the first r rows of $\mathbf{R}$. Further, the submatrix below $\mathbf{I}_r$ can be made $\mathbf{0}_{\ell\times r}$ after appropriate row operations. Now call the top-right submatrix $\mathbf{B} \in \mathbb{F}_q^{r\times(n-r)}$ so that $\mathbf{R} = \begin{pmatrix} \mathbf{I}_r & \mathbf{B} \\ \mathbf{0} & \mathbf{A} \end{pmatrix}$. To understand how to pick $\mathbf{A}$ and the $*$ entries in $\mathbf{B}$, let $\mathcal{H}_i$ be the set of column indices of known entries in the ith row of $\mathbf{B}$. For example, if the ith row of $\mathbf{B}$ is $\begin{pmatrix} 1 & * & 2 & * & * \end{pmatrix}$ then $\mathcal{H}_i = \{1, 3\}$. One can observe that the number of $\mathbf{R}$-compatible $\mathbf{C}''$ is equal to the number of ways one can add rows of $\mathbf{A}$ to $\mathbf{B}$ without disturbing the known entries. This in turn equals

$$\prod_{i=1}^{r} \left|\{\boldsymbol{x} \in \text{rowspace}(\mathbf{A}) \subseteq \mathbb{F}_q^{n-r} \mid \boldsymbol{x}_{\mathcal{H}_i} = \mathbf{0}\}\right| \tag{2}$$

where $\boldsymbol{x}_{\mathcal{H}_i}$ means the subvector of $\boldsymbol{x}$ restricted to the indices in $\mathcal{H}_i$. Note $\mathbf{A}$ should be chosen full-rank since only then rowspace($\mathbf{R}$) has the maximum dimension $r + \ell$. It then follows that the ith factor equals the cardinality of the left null space of $\mathbf{A}_{*,\mathcal{H}_i} \in \mathbb{F}_q^{\ell\times|\mathcal{H}_i|}$ and hence equals $q^{\ell-\text{rank}(\mathbf{A}_{*,\mathcal{H}_i})}$ by rank-nullity. Thus (2) equals $q^{r\ell-\sum_{i=1}^{r}\text{rank}(\mathbf{A}_{*,\mathcal{H}_i})}$.

Hence, the values of $*$ in $\mathbf{B}$ do not matter and can be chosen arbitrarily, while $\mathbf{A} \in \mathbb{F}_q^{\ell\times(n-r)}$ should be chosen full-rank such that the sum $\sum_{i=1}^{r}\text{rank}(\mathbf{A}_{*,\mathcal{H}_i})$ is minimized. We can always pick $\mathbf{A}$ to be all-zero except at some ℓ columns where it equals the identity. This is because, given any (size-ℓ) information set $\mathcal{I} \subseteq [n-r]$ of $\mathbf{A}$, the matrix obtained by zeroing all columns of $\mathbf{A}$ outside $\mathcal{I}$ cannot increase this sum. Thus we can pick $\mathbf{A}_{*,\bar{\mathcal{I}}} = \mathbf{0}_{\ell\times(n-r-\ell)}$, and since row operations on $\mathbf{A}$ do not matter either, also let $\mathbf{A}_{*,\mathcal{I}} = \mathbf{I}_\ell$. Thus $\text{rank}(\mathbf{A}_{*,\mathcal{H}_i}) = |\mathcal{H}_i \cap \mathcal{I}|$ and so $\sum_{i=1}^{r}\text{rank}(\mathbf{A}_{*,\mathcal{H}_i}) = \sum_{i=1}^{r}|\mathcal{H}_i \cap \mathcal{I}|$. The latter is minimum if $\mathcal{I}$ indexes any ℓ columns with the fewest known entries. If we denote this number by t_c, then the number of $\mathbf{R}$-compatible matrices is $q^{r\ell-t_c}$. □

The above proposition suggests an optimal strategy to guess an $(r+\ell)$-dimensional space such that it contains the row space of a systematic matrix in $\mathbb{F}_q^{r\times n}$ with partially known entries.

Theorem 1 (Subspace Guessing Algorithm)**.** *Let $\mathbf{C} := (\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{r\times n}$ where some entries of the secret matrix $\mathbf{C}' \in \mathbb{F}_q^{r\times(n-r)}$ are known, and let $0 \le \ell \le n-r$. Suppose an oracle $\mathcal{O}$ takes as input an $(r+\ell)$-dimensional space $\mathcal{R} \subseteq \mathbb{F}_q^n$ and outputs success if and only if $\mathcal{R}$ contains the row space of $(\mathbf{I}_r \mid \mathbf{C}')$.*

Define the following algorithm `GuessSubspace` *with oracle access to $\mathcal{O}$ and which takes the known entries of $\mathbf{C}'$ as input (i.e., both entry position and entry value).*

1. *In each iteration, sample $\mathcal{R}$ independently and identically distributed as follows. Set $\mathcal{R} :=$ rowspace($\mathbf{R}$) with $\mathbf{R} \in \mathbb{F}_q^{(r+\ell)\times n}$ of the form*

$$\mathbf{R} = \begin{pmatrix} \mathbf{I}_r & \mathbf{B} \\ \mathbf{0}_{\ell\times r} & \mathbf{A} \end{pmatrix} \in \mathbb{F}_q^{(r+\ell)\times n},$$

where:

- *$\mathbf{B} \in \mathbb{F}_q^{r\times(n-r)}$ agrees with $\mathbf{C}'$ at all known entries and the other entries are sampled uniformly at random, and*
- *$\mathbf{A} \in \mathbb{F}_q^{\ell\times(n-r)}$ is a fixed matrix such that $\mathbf{A}_{*,\mathcal{I}} = \mathbf{I}_\ell$ and $\mathbf{A}_{*,\bar{\mathcal{I}}} = \mathbf{0}_{\ell\times(n-r-\ell)}$ for some (fixed) set $\mathcal{I} \subseteq [n-r]$ of ℓ columns of $\mathbf{C}'$ containing the fewest known entries.*

2. If $\mathcal{O}(\mathcal{R})$ returns success, output $\mathcal{R}$, else go to Step 1.

If $\mathbf{C}'$ is drawn uniformly at random, then the above algorithm finds such an $\mathcal{R}$ in expected number of oracle queries

$$q^{r(n-r-\ell)-h_c+t_c}$$

where h_c is the total number of known entries in $\mathbf{C}'$ and t_c is the number of known entries within any ℓ columns of $\mathbf{C}'$ that have the fewest known entries (i.e., as defined in (1)). Further, this procedure is optimal up to a constant factor: no other algorithm with oracle access to $\mathcal{O}$ can find such an $\mathcal{R}$ in fewer expected oracle queries up to a constant factor.

Note that this theorem states that even an adaptive guessing algorithm cannot do better up to a constant factor.

Proof. Call a matrix $\mathbf{C}'' \in \mathbb{F}_q^{r\times(n-r)}$ *compatible* if it agrees with the known entries of $\mathbf{C}'$. There are $q^{r(n-r)-h_c}$ compatible matrices, exactly one of which is the correct answer (the matrix $\mathbf{C}'$). Proposition 1 says that any oracle query with an $(r+\ell)$-dimensional space $\mathcal{R} \subseteq \mathbb{F}_q^n$ as input can only check for at most $q^{r\ell-t_c}$ compatible matrices. Thus, in the worst case, any algorithm has to make at least $q^{r(n-r)-h_c}/q^{r\ell-t_c} = q^{r(n-r-\ell)-h_c+t_c}$ oracle queries. In the average case, even an adaptive algorithm cannot do better than checking disjoint size-$q^{r\ell-t_c}$ sets of compatible matrices without replacement in each iteration. But this improves upon sampling with replacement by at most a factor of 2, and hence the expected number of oracle queries is at least $q^{r(n-r-\ell)-h_c+t_c}$, up to a constant factor.

`GuessSubspace` samples $\mathcal{R}$ as $\text{rowspace}((\mathbf{I}_r \mid \mathbf{C}'')) + \text{rowspace}((\mathbf{0}_{\ell\times r} \mid \mathbf{A}))$ where $\mathbf{C}''$ is drawn uniformly from the set of compatible matrices. This mapping from $\mathbf{C}''$ to $\mathcal{R}$ is many-to-one, and two compatible matrices give the same $\mathcal{R}$ if and only if they differ, row by row, by vectors in $\text{rowspace}(\mathbf{A})$. Hence each fiber (i.e., the set of preimages of a given $\mathcal{R}$) has size $q^{r\ell-t_c}$, by the proof of Proposition 1. These fibers are pairwise disjoint and partition the set of all compatible matrices. Further, since $\mathbf{C}''$ is sampled uniformly and all fibers have the same size, the induced distribution on $\mathcal{R}$ is also uniform. Thus, `GuessSubspace` checks one of these size-$q^{r\ell-t_c}$ fibers uniformly at random in each iteration. The expected number of oracle queries is then $q^{r(n-r)-h_c}/q^{r\ell-t_c}$. □

Finally, the above results easily extend to *bit-level hints*.

Corollary 1 (Subspace Guessing with Bit-Level Hints)**.** *Suppose only some bits of some entries of $\mathbf{C}' \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$ are known. Then Proposition 1 and Theorem 1 remain valid with "known entries" replaced by "known bits". In other words:*

- *There exists an $(r+\ell)$-dimensional subspace $\mathcal{R} \subseteq \mathbb{F}_{2^\nu}^n$ which contains the row space of the largest number of systematic matrices that agree with $(\mathbf{I}_r \mid \mathbf{C}')$ at the* known bits*, and this number is $2^{\nu r\ell-t_c}$, where t_c is the number of* known bits *within any ℓ columns of $\mathbf{C}'$ with the fewest* known bits *(so, for example, the maximum value of t_c is $\nu r\ell$ instead of $r\ell$).*
- *Modify `GuessSubspace` in Theorem 1 so that $\mathbf{A}$ equals the identity on any set of ℓ columns with the fewest* known bits*. Then the expected number of oracle queries made by this algorithm is $2^{\nu r(n-r-\ell)-h_c+t_c}$ where h_c is the total number of* known bits*. Further, this is optimal up to a constant factor.*

See Appendix C for the proof and a fully explicit restatement of this corollary.

3 Kernel Search with Hints (MinRank)

Problem 1 (MinRank Problem)**.** *Given $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_K \in \mathbb{F}_q^{m\times n}$ and a small integer $r > 0$, find a vector $\boldsymbol{x} \in \mathbb{F}_q^K$ such that $\text{rank}(\mathbf{E}(\boldsymbol{x})) = r$ where $\mathbf{E}(\boldsymbol{x}) := \mathbf{M}_0 + \sum_{i=1}^K x_i\mathbf{M}_i$.*

Let $\mathbf{E} := \mathbf{E}(\boldsymbol{x}_0) \in \mathbb{F}_q^{m \times n}$ be the (unknown) matrix of rank r obtained from a correct solution $\boldsymbol{x}_0$ to the problem. Further, assume we can write $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}')$ for $\mathbf{S} \in \mathbb{F}_q^{m \times r}$ and $\mathbf{C}' \in \mathbb{F}_q^{r \times (n-r)}$. Suppose we know some hints from both $\mathbf{S}$ and $\mathbf{C}'$. For example, we might know

$$\mathbf{E} = \overbrace{\begin{bmatrix} * & * & * \\ 2 & 1 & * \\ * & 0 & * \\ * & * & * \\ * & * & * \end{bmatrix}}^{\mathbf{S}} \overbrace{\left[\begin{array}{ccc|ccccc} 1 & 0 & 0 & 1 & * & 2 & * & * \\ 0 & 1 & 0 & 0 & * & * & 0 & * \\ 0 & 0 & 1 & * & 1 & * & * & * \end{array}\right]}^{(\mathbf{I} \mid \mathbf{C}')}.$$

We will adapt the kernel search solver [GC00] to exploit these hints. First let us describe the original algorithm.

Kernel Search without Hints

Since $\mathbf{E} \in \mathbb{F}_q^{m \times n}$ has low rank, it has a large kernel. Thus, we can guess vectors $\boldsymbol{v}_i \in \mathbb{F}_q^n$ hoping they lie in the kernel and solve the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0} \in \mathbb{F}_q^m$ for $\boldsymbol{x} \in \mathbb{F}_q^K$. The latter yields m scalar equations, so we need to guess $b := \lceil K/m \rceil$ many vectors.

1. Guess b linearly independent vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b \in \mathbb{F}_q^n$ uniformly at random.
2. Solve the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T} \in \mathbb{F}_q^m$ for all $i = 1, \ldots, b$, for the unknowns $\boldsymbol{x} = (x_1, \ldots, x_K)$.
3. If $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) \neq r$, go to Step 1, else return $\boldsymbol{x} \in \mathbb{F}_q^K$.

Kernel Search with Hints

Hints from $\mathbf{S}$. If we know $S_{i,j}$ at position (i, j), then since $\mathbf{E}(\boldsymbol{x}) = (\mathbf{S} \mid \mathbf{SC}')$, we get the linear equation $E(\boldsymbol{x})_{i,j} = S_{i,j}$ over the unknowns $x_1, \ldots, x_K$. Thus we can reduce the given problem to a new MinRank instance with $K - 1$ unknowns for each known entry $S_{i,j}$, assuming each entry yields linearly independent equations. But the latter holds with high probability for random MinRank instances. Note that this reduction can always be done and is not specific to kernel search.

Hints from $\mathbf{C}'$. We now adapt the guessing step (Step 1) in kernel search to use the hints in $\mathbf{C}'$. Kernel search hopes to draw a b-dimensional space $\mathcal{K} \subseteq \mathbb{F}_q^n$ such that it is contained in the $(n-r)$-dimensional kernel $\mathcal{E} \subseteq \mathbb{F}_q^n$ of $\mathbf{E} \in \mathbb{F}_q^{m \times n}$. But note $\mathcal{K} \subseteq \mathcal{E}$ if and only if $\mathcal{K}^\perp \supseteq \mathcal{E}^\perp$. Thus we can guess an $(r + \ell)$-dimensional space $\mathcal{K}^\perp$ containing the r-dimensional space $\mathcal{E}^\perp$ instead, where we set $\ell = n - r - b$. Now $\mathcal{E}^\perp$ is the row space of $\mathbf{E}$ and, since $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}')$, also the row space of $(\mathbf{I}_r \mid \mathbf{C}')$. Thus, we can use Theorem 1 to guess $\mathcal{K}^\perp$ optimally, as described next.

Let $\mathcal{I} \subseteq [n - r]$ be the set of indices of any ℓ columns in $\mathbf{C}'$ containing the fewest hints. Construct $\mathbf{A} \in \mathbb{F}_q^{\ell \times (n-r)}$ such that $\mathbf{A}_{*,\mathcal{I}} = \mathbf{I}_\ell$ and $\mathbf{A}_{*,\bar{\mathcal{I}}} = \mathbf{0}_{\ell \times (n-r-\ell)}$.

1. Draw $\mathbf{B} \in \mathbb{F}_q^{r \times (n-r)}$ uniformly at random such that $\mathbf{B}$ agrees with $\mathbf{C}'$ at the hint positions. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b \in \mathbb{F}_q^n$ be a basis of the kernel (i.e., right null space) of $\mathbf{R} \in \mathbb{F}_q^{(r+\ell) \times n}$ where $\mathbf{R} := \begin{pmatrix} \mathbf{I}_r & \mathbf{B} \\ \mathbf{0}_{\ell \times r} & \mathbf{A} \end{pmatrix}$.
2. Solve the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T} \in \mathbb{F}_q^m$ for all $i = 1, \ldots, b$, for the unknowns $\boldsymbol{x} = (x_1, \ldots, x_K)$.
3. If $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) \neq r$, go to Step 1, else return $\boldsymbol{x} \in \mathbb{F}_q^K$.

Complexity

The complexity without hints is $P\ q^{r\lceil K/m\rceil}$ where the polynomial cost P (up to a constant factor) is K^ω $\mathbb{F}_q$-operations for typical parameters. *Derivation:* We succeed exactly when the b-dimensional space $\langle \boldsymbol{v}_1, \dots, \boldsymbol{v}_b\rangle$ is a subspace of the $(n-r)$-dimensional kernel of $\mathbf{E}$ (here $b := \lceil K/m\rceil$). This happens with probability $\begin{bmatrix} n-r \\ b \end{bmatrix}_q / \begin{bmatrix} n \\ b \end{bmatrix}_q \approx q^{-br} = q^{-r\lceil K/m\rceil}$ where $\begin{bmatrix} a \\ b \end{bmatrix}_q$ denotes the Gaussian binomial coefficient and where we use $q^{(a-b)b} \le \begin{bmatrix} a \\ b \end{bmatrix}_q \le 4q^{(a-b)b}$ for the approximation. The most expensive steps are 2 and 3. Step 2 sets up the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T}$ for $i = 1, \dots b$, which costs $Kmnb$, and solves for $\boldsymbol{x}$, which costs K^ω $\mathbb{F}_q$-operations. Step 3 computes the rank of $\mathbf{E}(\boldsymbol{x}) \in \mathbb{F}_q^{m\times n}$, costing $\max\{m,n\}\min\{m,n\}^{\omega-1}$ $\mathbb{F}_q$-operations. For typical parameters (including Mirath's), K^ω dominates.

The complexity with hints is

$$P\ q^{r\lceil \tilde{K}/m\rceil - h_c + t_c}$$

where h_s is the number of hints in $\mathbf{S}$, $\tilde{K} := \max\{0, K - h_s\}$, h_c is the total number of hints in $\mathbf{C}'$, and t_c is the number of hints within any $\ell := n - r - \lceil \tilde{K}/m\rceil$ columns of $\mathbf{C}'$ with the fewest hints. The polynomial cost P (up to a constant factor) is typically $\tilde{K}^\omega$ $\mathbb{F}_q$-operations. *Derivation:* The hints from $\mathbf{S}$ allow us to reduce the K unknowns down to $\tilde{K} := K - h_s$ (assuming all the h_s additional equations are independent). The exponential cost follows from Theorem 1. The polynomial cost is as follows. We first reduce the MinRank instance with parameter K to $\tilde{K}$, which is a one-time cost and thus can be neglected. The only step in each iteration that is different from the without-hints algorithm is Step 1 where we compute a basis of the right null space of $\mathbf{R}$. But note that $\mathbf{R}$ can easily be brought into row reduced echelon form (since $\mathbf{A}$ equals the identity at ℓ columns and is zero otherwise) and thus the basis can be directly read off from $\mathbf{R}$. Thus, the polynomial cost is the same (after replacing K with $\tilde{K}$ throughout). The full algorithm and cost are given in Proposition 2.

Bit-level Hints

Suppose $q = 2^\nu$. Then, using Corollary 1, one can extend this algorithm to the case where we know some bits of some entries of $\mathbf{C}'$ and $\mathbf{S}$ and not full $\mathbb{F}_q$-entries. The complexity becomes

$$P\ 2^{\nu r\lceil \tilde{K}/m\rceil - h_c + t_c}$$

where h_s, h_c and t_c are as before but they count bit-level hints instead, and $\tilde{K} := \max\{K - h_s/\nu, 0\}$. The polynomial cost P (up to a constant factor) is $(\nu\tilde{K})^\omega$ $\mathbb{F}_2$-operations for typical parameters. The detailed cost is given in Proposition 3.

4 GRS with Hints (RSD)

Problem 2 (Rank Syndrome Decoding (RSD) Problem)**.** *Given* $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\boldsymbol{t} \in \mathbb{F}_{q^m}^{n-k}$ *and a small integer* $r > 0$*, find a vector* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *such that* $\mathbf{H}\boldsymbol{e}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ *and* $\mathrm{rank}(\boldsymbol{e}) = r$ *(where* $\mathrm{rank}(\boldsymbol{e})$ *is the dimension of the* $\mathbb{F}_q$*-linear space spanned by the entries of* $\boldsymbol{e}$*).*

Suppose $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ can be written as $\boldsymbol{s}(\mathbf{I}_r \mid \mathbf{C}')$ where $\boldsymbol{s} \in \mathbb{F}_{q^m}^r$ and $\mathbf{C}' \in \mathbb{F}_q^{r\times(n-r)}$. Further, let $\mathbf{E} := \mathrm{ext}(\boldsymbol{e}) \in \mathbb{F}_q^{m\times n}$ be the matrix obtained by expanding $\boldsymbol{e}$ over a fixed basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then we can write $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}')$ where $\mathbf{S} := \mathrm{ext}(\boldsymbol{s}) \in \mathbb{F}_q^{m\times r}$. Our hints will again be entries of $\mathbf{S}$ and $\mathbf{C}'$ and we will adapt the GRS solver [GRS16] to use them. First let us describe the original algorithm.

GRS without Hints

Let $\boldsymbol{e}$ be decomposed as $\boldsymbol{e} = \boldsymbol{s}\mathbf{C}$, where $\boldsymbol{s} \in \mathbb{F}_{q^m}^r$ and $\mathbf{C} \in \mathbb{F}_q^{r\times n}$. Let $\mathcal{E} \subseteq \mathbb{F}_{q^m}$ be the r-dimensional space over $\mathbb{F}_q$ spanned by the entries of $\boldsymbol{s}$. If we guess an $(r+\ell)$-dimensional space $\mathcal{C} \subseteq \mathbb{F}_{q^m}$ such that $\mathcal{C} \supseteq \mathcal{E}$, then

we can find $\boldsymbol{e}$ by solving $\mathbf{H}(\boldsymbol{a}\mathbf{B})^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ where $\boldsymbol{a} \in \mathbb{F}_{q^m}^{r+\ell}$ is a basis of $\mathcal{C}$, the matrix $\mathbf{B} \in \mathbb{F}_q^{(r+\ell)\times n}$ is unknown, and $\boldsymbol{e} = \boldsymbol{a}\mathbf{B}$. This holds with high probability provided we have as many equations as unknowns. This approach can be seen as guessing the column space of $\mathbf{S} := \mathrm{ext}(\boldsymbol{s}) \in \mathbb{F}_q^{m\times r}$. Alternatively, we can also guess an $(r+\ell)$-dimensional space $\mathcal{R} \subseteq \mathbb{F}_q^n$ containing the row space of $\mathbf{C}$.

We now state the steps for the "row space" version below. The algorithm for the "column space" version is similar.

1. Guess $r+\ell$ linearly independent vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{r+\ell} \in \mathbb{F}_q^n$ uniformly at random.
2. Write $\boldsymbol{e} = \boldsymbol{a}\begin{pmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_{r+\ell} \end{pmatrix}$ for unknown $\boldsymbol{a} \in \mathbb{F}_{q^m}^{r+\ell}$ and then substitute it into $\mathbf{H}\boldsymbol{e}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ to solve for $\boldsymbol{a}$ and hence $\boldsymbol{e}$.
3. If $\mathrm{rank}(\boldsymbol{e}) \neq r$, go to Step 1, else return $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$.

GRS with Hints

We will describe the modifications for the "row space" version of GRS, assuming $\mathbf{E} = \mathbf{S}(\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_q^{m\times n}$. The "column space" version with hints is analogous, but note it requires the first factor to be systematic instead of the second, i.e., $\mathbf{E} = \begin{pmatrix} \mathbf{I}_r \\ \mathbf{S}' \end{pmatrix}\mathbf{C}$ where $\mathbf{S}' \in \mathbb{F}_q^{(m-r)\times r}$ and $\mathbf{C} \in \mathbb{F}_q^{r\times n}$ (see Section 6).

Hints from $\mathbf{S}$. Each known entry $S_{i,j}$ in $\mathbf{S}$ gives us an equation of the form $E_{i,j} = S_{i,j}$ since $\mathbf{E} = (\mathbf{S} \mid \mathbf{S}\mathbf{C}')$. Thus, if we have h_s hints, then we obtain h_s additional equations which can lead to a larger value of ℓ in the guessing step.

Hints from $\mathbf{C}'$. Theorem 1 directly improves the guessing of the $(r+\ell)$-dimensional space in Step 1.

Complexity

The "row space" GRS variant with hints yields $(r+\ell)m$ unknowns and $(n-k)m + h_s$ equations. Assuming the heuristic that the equations are independent, having at least as many equations as unknowns gives $\ell \leq n-k-r+\left\lfloor \frac{h_s}{m} \right\rfloor$. Setting ℓ to be its maximum value and using Theorem 1, the complexity is

$$(n-k)^\omega m^\omega q^{r(k-\lfloor h_s/m \rfloor)-h_c+t_c}\ \mathbb{F}_q\text{-operations}$$

where h_s is the number of hints in $\mathbf{S}$, t_c is the number of hints within any $\ell = n-k-r+\lfloor h_s/m \rfloor$ columns of $\mathbf{C}'$ with the fewest hints, and h_c is the total number of hints in $\mathbf{C}'$. The detailed cost is given in Proposition 4. For the "column space" variant of GRS, the complexity turns out to be

$$(n-k)^\omega m^\omega q^{r\lceil (km-h_c)/n \rceil - h_s + t_s}\ \mathbb{F}_q\text{-operations}$$

where, assuming one $\mathbb{F}_{q^m}$-operation costs m $\mathbb{F}_q$-operations, t_s is the number of hints within any $\ell = m-r-\lceil (km-h_c)/n \rceil$ rows of $\mathbf{S}'$ with the fewest hints, and h_s is the total number of hints in $\mathbf{S}'$ where $\mathbf{S}' \in \mathbb{F}_q^{(m-r)\times r}$ is the non-systematic part of $\mathbf{S}$. The detailed polynomial cost is given in Proposition 6.

Bit-level Hints

Similar to kernel search, we can extend these results to bit-level hints using Corollary 1. If $q = 2^\nu$, the complexities become

$$\nu^\omega (n-k)^\omega m^\omega 2^{\nu r(k-\lfloor h_s/(\nu m) \rfloor)-h_c+t_c}\ \mathbb{F}_2\text{-operations}$$
$$\text{and } \nu^\omega (n-k)^\omega m^\omega 2^{\nu r\lceil (km-(h_c/\nu))/n \rceil - h_s + t_s}\ \mathbb{F}_2\text{-operations}$$

for the row and column space GRS variants respectively, assuming one $\mathbb{F}_{2^\nu}$-operation costs ν $\mathbb{F}_2$-operations and one $\mathbb{F}_{2^{\nu m}}$-operation costs νm $\mathbb{F}_2$-operations. If $h_s = 0$ for row GRS, respectively $h_c = 0$ for column GRS, the system need not be expanded over $\mathbb{F}_2$. A common upper bound is then $(m(n-k))^\omega$ $\mathbb{F}_{2^\nu}$-operations.[2] The detailed costs are given in Propositions 5 and 7.

5 Side-Channel Implementation

In this section, we demonstrate how side-channel information can be used to generate hints that assist in solving the MinRank or RSD problems, as discussed in this work. We show how such hints can arise in practice by presenting a side-channel attack that, with a certain success probability, leaks information suitable for constructing these hints. As a target, we focus on the Mirath signature scheme [AAB+25], which is based on the MinRank problem.

A Setup

For our side-channel analysis, we use a ChipWhisperer CW308 setup with a UFO board and an STM32L5HWC as the target device. All measurements are initialized and processed on a host PC, which communicates with the target using the SimpleSerial protocol executed from within a Jupyter Notebook environment.

For the Mirath implementation on the embedded device, we rely on a minimal, functionally relevant subset of the official reference implementation. To the best of our knowledge, no embedded implementation of Mirath currently exists, and a full reference implementation is infeasible on our target device due to runtime memory limitations. This is primarily because of the memory-intensive tree structure used in seed generation.

Our partial implementation focuses on the function `compute_polynomial_proof`, in which the secret key is used.

B Attack Target

Within the function `emulateMPC_mu`, one of the operations is the computation of $\mathbf{C}' + \mathbf{C}_{\text{rnd}}$, where $\mathbf{C}_{\text{rnd}}$ is a random masking matrix. This computation is performed by calling the function `mirath_matrix_ff_mu_add_mu1ff` (see Listing 1), where the corresponding matrix entries are XORed element-wise. The objective of our attack is to extract the matrix $\mathbf{C}'$ from this operation. Together with $\mathbf{S}$, the matrix $\mathbf{C}'$ forms the secret key of Mirath.

```
1  static inline void mirath_matrix_ff_mu_add_mu1ff(ff_mu_t *matrix1, const ff_mu_t *matrix2,
       const ff_t *matrix3, const uint32_t n_rows, const uint32_t n_cols) {
2      for (uint32_t i = 0; i < n_rows; i++) {
3          for (uint32_t j = 0; j < n_cols; j++) {
4              ff_mu_t entry2;
5
6              const ff_mu_t entry1 = mirath_matrix_ff_mu_get_entry(matrix2, n_rows, i, j);
7
8              entry2 = mirath_matrix_ff_get_entry(matrix3, n_rows, i, j);
9
10             entry2 = mirath_map_ff_to_ff_mu[entry2];
11             entry2 = entry1 ^ entry2;
12             mirath_matrix_ff_mu_set_entry(matrix1, n_rows, i, j, entry2);
13         }
14     }
15 }
```

Listing 1: Matrix addition in Mirath used in the computation of $\mathbf{C}' + \mathbf{C}_{\text{rnd}}$.

[2] For row GRS, Proposition 5 gives the cost $P_0 = \max\{m(n-k)rk \min\{r,k\}^{\omega-3}$ $\mathbb{F}_{2^\nu}$-operations, $(n-k)^\omega$ $\mathbb{F}_{2^{\nu m}}$-operations$\}$; for column GRS, Proposition 7 gives the displayed upper bound as the highlighted term.

The function `mirath_matrix_ff_mu_add_mu1ff` takes as input `matrix2` (corresponding to $\mathbf{C}_{\text{rnd}}$), `matrix3` (corresponding to $\mathbf{C}'$), the number of rows and columns, and `matrix1`, which stores the result $\mathbf{C}_{\text{aux}}$ of the operation.

For this partial implementation, a random 32-byte seed is generated by a Jupyter Notebook instance and transferred to the target. From this seed, the Mirath public and secret keys are derived as specified by the signature scheme. All remaining values ($\mathbf{S}_{\text{base}}$, $\mathbf{C}_{\text{base}}$, $\boldsymbol{v}$, $\boldsymbol{v}_{\text{base}}$, and $\boldsymbol{\Gamma}$) are generated at runtime on the target using the `rand()` function. The random seed for `rand()` is provided by `HAL_GetTick()`, which returns a millisecond-resolution system tick counter.

The program executed on the target performs a full run of `emulateMPC_mu`, and at the end of the execution, the derived matrix $\mathbf{C}'$ is returned via SimpleSerial. The numeric values of the entries of $\mathbf{C}'$ serve as labels for the side-channel analysis.

Our attacks focus on the parameter set `Mirath-1a-fast` with $\tau = 17$, $q = 16$, $m = 16$, $n = 16$, $k = 143$, and $r = 4$, where τ is the number of times `emulateMPC_mu` is called during signing.

C Side-Channel Analysis

The attack applied in this work is a profiling side-channel attack. In this setting, we assume that the attacker has access to a device identical to the actual target. On this identical device, a large number of power measurements are recorded while the computation is performed with known secret values. These measurements form the basis for the profiling phase.

Afterwards, a single trace is taken on the target device during a signing operation (or, in our case, during the execution of the partial Mirath implementation). By comparing this trace with the previously recorded measurements from the profiling device, the attacker can get information about which secret key was used during the computation.

1) Measurement

The function `emulateMPC_mu` is executed τ times during the signing procedure. In this work, we set $\tau = 17$, following the `Mirath-1a-fast` parameter set. Since one execution of the target program corresponds to one function call, the target program is called seventeen times using the same seed but with different values for $\mathbf{C}_{\text{rnd}}$. During each execution, a power trace is recorded for the function `mirath_matrix_ff_mu_add_mu1ff`, which computes the value of $\mathbf{C}' + \mathbf{C}_{\text{rnd}}$. To isolate this section, a trigger is placed immediately before and after the function call.

The seventeen traces obtained from one full execution are combined by computing the mean over all sample points, resulting in a single aggregated trace. Each aggregated trace is associated with a label containing the corresponding entries of the matrix $\mathbf{C}'$. In total, 1000 such measurements were recorded to build the dataset used in this work.

2) Attack

For the attack phase, a dataset is collected in the same manner as during profiling. The entries of the matrix $\mathbf{C}'$ are structured as follows: since each element lies in $\mathbb{F}_{16}$, four bits are required to represent a single entry, meaning one byte contains two entries. In the `Mirath-1a-fast` parameter set, the matrix $\mathbf{C}'$ consists of four rows and twelve columns.

In this phase, the measurement traces still contain labels, allowing us to verify whether the classification is correct. The attack proceeds entry by entry, which is feasible because each entry involved in the addition is loaded and processed independently. For the classification of one matrix entry, all profiling traces corresponding to the same value of that entry are grouped together, again by computing the mean of the respective sample points. Thus, for determining the first entry of $\mathbf{C}'$, all profiling traces sharing the same value in this position are aggregated, resulting in sixteen different averaged traces—one for each possible field element.

Classification is then performed by computing the correlation between the trace measured on the target device and the grouped profiling traces. Instead of using the full trace, only the relevant portion where the corresponding entry is processed is considered. To achieve this, we define a correlation window consisting of consecutive sample points that contribute most to the classification. The start point of this window corresponds to the sample index with the most significant difference, determined through differences among the profiling groups; for the first entry, this occurs at `start = 474`. From this start point, the window size and shift parameters are computed, with the best results obtained using `window = 57` and `shift = 53`. Further entries are located using a static column offset `col_offset` $= l \cdot 348$, where l denotes the column index. If a different row is targeted, the outer loop (see Listing 1) is iterated, and an additional row offset `row_offset` $= 4208 \cdot$ `k` $\cdot 2$ is applied, with k representing the row index.

Moreover, a distinction must be made between the first four bits and the last four bits of each byte, since they correspond to two independent matrix entries. Consequently, two separate classifications are performed, and the start point for the latter case is `start = 4682`. Using these observations, the correlation window boundaries `low` and `high` for each entry can be computed as:

$$\texttt{low} = \texttt{start} + \texttt{row_offset} + \texttt{offset} - \texttt{window} + \texttt{shift},$$

$$\texttt{high} = \texttt{start} + \texttt{row_offset} + \texttt{offset} + \texttt{window} + \texttt{shift}.$$

The correlation itself is calculated using the Pearson correlation coefficient, implemented using the `numpy.corrcoef` function. The class whose correlation window yields the highest coefficient is selected as the predicted value of the corresponding matrix entry.

D Results of the Attack

We performed a total of 1000 attack measurements and classified them using the method described in Section C. The results show that the individual entries of the secret matrix $\mathbf{C}'$ can be correctly classified with a success probability of 96.55%. These hints can then be input to the modified kernel search and GRS solvers of Sections 3 and 4 to recover the secret key at a lower cost. Since some hints may be incorrect, the solvers may need to be rerun with the erroneous hint positions being guessed. However, the expected number of retries is small given the high success probability of obtaining the hints.

6 Conclusion and Future Work

Our modified kernel search and GRS algorithms (Sections 3 and 4) have the following limitations which are left for future work.

1. We only incorporate hints in kernel search and GRS if one of the matrices in the rank-decomposition of $\mathbf{E} = \mathbf{SC}$ is systematic (although we note most cryptographic schemes do this as this reduces key size). Further, kernel search can be made to guess either the right null space of $\mathbf{C}$ or the left null space of $\mathbf{S}$. Similarly, GRS can guess a space containing the column space of $\mathbf{S}$ or a space containing the row space of $\mathbf{C}$. In both algorithms, both versions have the same complexity when $m = n$, the former is better when $m > n$ and the latter is better when $n > m$. However, we can only incorporate hints when the guessing is done for the systematic matrix (although again most cryptographic schemes use $m = n$ so this would not matter).

   Incorporating hints in this case amounts to finding the optimal subspace in Proposition 1 when $\mathbf{C} \in \mathbb{F}_q^{r \times n}$ is no longer known to be systematic.

2. We adapt only the original GRS algorithm [GRS16], and not the improvement in [AGHT06] which introduces a $-m$ term in the exponent and thus performs better for certain parameter regions, including $m = n$ and $m < n$.

Incorporating hints in this case amounts to the following variant of Proposition 1: find an $(r+\ell)$-dimensional $\mathcal{R}$ which maximizes the number of $\widetilde{\mathbf{C}} \in \mathbb{F}_q^{r\times n}$ such that $\widetilde{\mathbf{C}}$ agrees with $\mathbf{C}$ on the known entries and there exists some polynomial $p(x) \in \mathbb{F}_q[x]$ such that $\mathrm{rowspace}(\mathbf{C}\, p(\mathbf{M})) \subseteq \mathcal{R}$. Here $\mathbf{M} \in \mathbb{F}_q^{n\times n}$ is a fixed matrix with an irreducible degree-n minimal polynomial over $\mathbb{F}_q$.

References

[AAB+25] Gora Adj, Nicolas Aragon, Stefano Barbero, Magali Bardet, Emanuele Bellini, Loïc Bidoux, Jesús-Javier Chi-Domınguez, Victor Dyseryn, Andre Esser, Thibauld Feneuil, et al. Mirath signature scheme, 2025. URL: `https://pqc-mirath.org/`.

[ABB+25] Nicolas Aragon, Magali Bardet, Loıc Bidoux, Jesús-Javier Chi-Domınguez, Victor Dyseryn, Thibauld Feneuil, Philippe Gaborit, Antoine Joux, Romaric Neveu, Matthieu Rivain, et al. Ryde signature scheme, 2025. URL: `https://pqc-ryde.org/`.

[ABC+24] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the first round of the additional digital signature schemes for the nist post-quantum cryptography standardization process. Technical Report NISTIR 8528, National Institute of Standards and Technology, October 2024. `doi:10.6028/NIST.IR.8528`.

[AGHT06] Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. A New Algorithm for Solving the Rank Syndrome Decoding Problem. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2421–2425, 2018-06. `doi:10.1109/ISIT.2018.8437464`.

[BBB+23] Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, and Jean-Pierre Tillich. Revisiting algebraic attacks on MinRank and on the rank decoding problem. *Designs, Codes and Cryptography*, 91(11):3671–3707, November 2023. `doi:10.1007/s10623-023-01265-x`.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, Lecture Notes in Computer Science, pages 507–536. Springer International Publishing, 2020. `doi:10.1007/978-3-030-64837-4_17`.

[DEGS25] Giuseppe D'Alconzo, Andre Esser, Andrea Gangemi, and Carlo Sanna. Sneaking up the ranks: Partial key exposure attacks on rank-based schemes. *Designs, Codes and Cryptography*, 94(1):15, December 2025. `doi:10.1007/s10623-025-01738-1`.

[DSDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with Side Information: Attacks and Concrete Security Estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 329–358. Springer International Publishing, 2020. `doi:10.1007/978-3-030-56880-1_12`.

[EMVW22] Andre Esser, Alexander May, Javier Verbel, and Weiqiang Wen. Partial Key Exposure Attacks on BIKE, Rainbow and NTRU. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 346–375, Cham, 2022. Springer Nature Switzerland. `doi:10.1007/978-3-031-15982-4_12`.

[GC00] Louis Goubin and Nicolas T. Courtois. Cryptanalysis of the TTM Cryptosystem. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 44–57. Springer, 2000. `doi:10.1007/3-540-44448-3_4`.

[GRS16] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the Complexity of the Rank Syndrome Decoding Problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, February 2016. doi:10.1109/TIT.2015.2511786.

[How08] Rodney R. Howell. On asymptotic notation with multiple variables. Technical Report Technical Report 2007-4, Department of Computing and Information Sciences, Kansas State University, 2008. URL: https://people.cs.ksu.edu/~rhowell/asymptotic.pdf.

[HPR+22] Anna-Lena Horlemann, Sven Puchinger, Julian Renner, Thomas Schamberger, and Antonia Wachter-Zeh. Information-Set Decoding with Hints. In Antonia Wachter-Zeh, Hannes Bartz, and Gianluigi Liva, editors, *Code-Based Cryptography*, Lecture Notes in Computer Science, pages 60–83. Springer International Publishing, 2022. doi:10.1007/978-3-030-98365-9_4.

[MH24] Gabrielle De Micheli and Nadia Heninger. Survey: Recovering cryptographic keys from partial information, by example. *IACR Communications in Cryptology*, 1(1), April 2024. doi:10.62056/ahjbksdja.

[Rut16] Kalle Rutanen. *O-Notation in Algorithm Analysis*. PhD thesis, Tampere University of Technology, 2016. URL: https://urn.fi/URN:ISBN:978-952-15-3841-4.

A Full Algorithms and Cost Analysis

Remark 2 (Cost functions up to a constant factor). Let $f, g\colon D \to \mathbb{R}_{\geq 0}$ be functions with common domain D. Then we define "$f \leq g$ (up to a constant factor)" to mean

$$\text{there exists a constant } c > 0 \text{ such that } f(x) \leq c\, g(x) \qquad \text{for all } x \in D.$$

The expression "$f = g$ (up to a constant factor)" means $f \leq g$ and $g \leq f$ (each up to a constant factor) or, equivalently,

$$\text{there exist constants } c_1, c_2 > 0 \text{ such that } c_1\, g(x) \leq f(x) \leq c_2\, g(x) \qquad \text{for all } x \in D.$$

We use the above "up to a constant factor" language to avoid ambiguity, since the big-$\mathcal{O}$ notation for multivariate functions seems to have no universally agreed-upon definition in the literature (see, e.g., [How08] and [Rut16, Sec. 1.6, "Problem 3"]). This "up to a constant factor" relation is called "linear dominance" by Rutanen [Rut16], who studies this notion carefully.

Let f and g be two non-negative-valued functions. Then the "up to a constant factor" relation, denoted $\sim$ below, has the following properties.

- $\sim$ is an equivalence relation.
- $f + g \sim \max\{f, g\}$.
- $c_1 f + c_2 \sim f$ for any constants $c_1, c_2 > 0$, provided f is bounded below by a positive constant.

Remark 3 (Relation of the below algorithms to the original kernel search and GRS algorithms). The complexity formulas derived below also give, up to a constant factor, the complexities (both the exponential and polynomial factors) of the original kernel search and row GRS algorithms if we set $h_s = h_c = 0$.

For kernel search, Algorithm 1 differs from the original algorithm only in how the $(n - r - \ell)$-dimensional subspace of the kernel is guessed. Theorem 1 exploits the fact that we know the matrix $\mathbf{C}$ is systematic and succeeds with probability exactly $q^{-r(n-r-\ell)} = q^{-r\lceil K/m \rceil}$. The original algorithm samples a uniformly random subspace and succeeds with probability $p := \frac{\begin{bmatrix} n-r \\ n-r-\ell \end{bmatrix}_q}{\begin{bmatrix} n \\ n-r-\ell \end{bmatrix}_q}$. One can show that $c_1 q^{-r(n-r-\ell)} \leq p \leq c_2 q^{-r(n-r-\ell)}$

for some constants $c_1, c_2 > 0$. Hence, up to a constant factor, Proposition 2 below also gives the complexity of original kernel search (after setting $h_s = h_c = 0$). Similarly, for row/column GRS, Propositions 4 and 6 below give the complexities of the original algorithms (after setting $h_s = h_c = 0$).

A Kernel Search with and without Hints (Algorithms and Cost Analysis)

Algorithm 1. Kernel Search with Hints over $\mathbb{F}_q$

Input: $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_K \in \mathbb{F}_q^{m\times n}$ and a small integer $r > 0$. Assume there exists a solution $\widehat{\boldsymbol{x}} \in \mathbb{F}_q^K$ such that

$$\mathbf{E}(\widehat{\boldsymbol{x}}) := \mathbf{M}_0 + \sum_{i=1}^{K} \hat{x}_i \mathbf{M}_i = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$$

for some full-rank $\widehat{\mathbf{S}} \in \mathbb{F}_q^{m\times r}$ and some $\widehat{\mathbf{C}}' \in \mathbb{F}_q^{r\times(n-r)}$. (Note this implies $\mathbf{E}(\widehat{\boldsymbol{x}})$ has rank r.) Additionally, suppose h_s entries of $\widehat{\mathbf{S}}$ and h_c entries of $\widehat{\mathbf{C}}'$ are known.
Output: A vector $\boldsymbol{x} \in \mathbb{F}_q^K$ such that $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) = r$.

0. Set $\tilde{K} := K - h_s$, $b := \lceil \tilde{K}/m \rceil$ and $\ell := n - r - b$. Use the h_s known entries of $\widehat{\mathbf{S}}$ to derive h_s linear equations in the unknowns $x_1, \ldots, x_K$, and reduce the given MinRank instance to one with only $\tilde{K}$ unknowns.
1. Draw $\mathbf{R} \in \mathbb{F}_q^{(r+\ell)\times n}$ according to Theorem 1, and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b \in \mathbb{F}_q^n$ be a basis of the right null space of $\mathbf{R}$.
2. Solve the system

   $$\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}} \in \mathbb{F}_q^m, \qquad i = 1, \ldots, b,$$

   for the $\tilde{K}$ unknowns of the reduced MinRank instance.
3. If the obtained solution $\boldsymbol{x}$ satisfies $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) = r$, return $\boldsymbol{x}$, else go to Step 1.

Algorithm 2. Kernel Search with Bit-Level Hints over $\mathbb{F}_{2^\nu}$

Input: $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_K \in \mathbb{F}_{2^\nu}^{m\times n}$ and a small integer $r > 0$. Assume there exists a solution $\widehat{\boldsymbol{x}} \in \mathbb{F}_{2^\nu}^K$ such that

$$\mathbf{E}(\widehat{\boldsymbol{x}}) := \mathbf{M}_0 + \sum_{i=1}^{K} \hat{x}_i \mathbf{M}_i = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$$

for some full-rank $\widehat{\mathbf{S}} \in \mathbb{F}_{2^\nu}^{m\times r}$ and some $\widehat{\mathbf{C}}' \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$. (Note this implies $\mathbf{E}(\widehat{\boldsymbol{x}})$ has rank r.) Additionally, suppose h_s bits of $\widehat{\mathbf{S}}$ and h_c bits of $\widehat{\mathbf{C}}'$ are known.
Output: A vector $\boldsymbol{x} \in \mathbb{F}_{2^\nu}^K$ such that $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) = r$.

0. Set $\tilde{K} := K - h_s/\nu$, $b := \lceil \tilde{K}/m \rceil$ and $\ell := n - r - b$. Use the h_s known bits of $\widehat{\mathbf{S}}$ to derive h_s independent equations over $\mathbb{F}_2$ in the binary variables $\{x_{i,j}\}_{i\in[K], j\in[\nu]}$.
1. Draw $\mathbf{R} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n}$ according to Corollary 1, and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b \in \mathbb{F}_{2^\nu}^n$ be a basis of the right null space of $\mathbf{R}$.
2. Set up the system

   $$\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}} \in \mathbb{F}_{2^\nu}^m, \qquad i = 1, \ldots, b.$$

   Expand the above system over $\mathbb{F}_2$ by expanding each unknown $x_i \in \mathbb{F}_{2^\nu}$ using a fixed basis $\gamma_1, \ldots, \gamma_\nu$ of $\mathbb{F}_{2^\nu}$ over $\mathbb{F}_2$ as $x_i = x_{i,1}\gamma_1 + \cdots + x_{i,\nu}\gamma_\nu$ with $x_{i,j} \in \mathbb{F}_2$. Combine these equations with the h_s $\mathbb{F}_2$-equations obtained in Step 0 and then solve the system for $\boldsymbol{x}$.
3. If the obtained solution $\boldsymbol{x}$ satisfies $\operatorname{rank}(\mathbf{E}(\boldsymbol{x})) = r$, return $\boldsymbol{x}$, else go to Step 1.

Proposition 2 (Complexity of Kernel Search with $\mathbb{F}_q$-Hints (Algorithm 1))**.** *Set $\tilde{K} := K - h_s$ and $\ell := n - r - \lceil \tilde{K}/m \rceil$. So that the problem is not trivial, assume $\tilde{K} > 0$ and $\ell \geq 0$. The condition $\ell \geq 0$ is equivalent to $\lceil \frac{\tilde{K}}{m} \rceil \leq n - r$, which is equivalent[3] to $\tilde{K} \leq m(n-r)$.*

Assume the (reasonable) heuristic that the h_s equations obtained from $\widehat{\mathbf{S}}$ have rank h_s and the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 1 is

$$P\, q^{r\lceil \tilde{K}/m \rceil - h_c + t_c} \text{ (up to a constant factor)}$$

[3]Note if $\tilde{K} \geq m(n-r)$, then the MinRank problem with $\tilde{K}$ unknowns is easy to solve. Indeed, one can impose that the first $n-r$ columns of $\mathbf{M}_0 + \sum_i x_i \mathbf{M}_i$ are zero, giving $m(n-r)$ linear equations. If the number of unknowns is larger than this, then such a linear combination exists with high probability; the resulting matrix has rank at most r, and has rank exactly r with high probability which happens when the remaining $m \times r$ block has full rank.

where h_c is the total number of known entries in $\widehat{\mathbf{C}}'$ and t_c is the number of known entries contained in any ℓ columns of $\widehat{\mathbf{C}}'$ with the fewest known entries. The polynomial cost is

$$P = \max\left\{\tilde{K}^2 r,\ \tilde{K}^\omega,\ \tilde{K}mn,\ mn\min\{m,n\}^{\omega-2}\right\}\ \mathbb{F}_q\text{-operations} \quad \textit{(up to a constant factor).}$$

where the highlighted term dominates for typical parameters. Setting $h_s = h_c = 0$ recovers the original kernel-search complexity; see Remark 3 above.

Proof. Exponential cost. Each guessed kernel vector gives m linear equations in the $\tilde{K}$ remaining MinRank unknowns, which we assume to be linearly independent by the heuristic, so we take $b := \lceil \tilde{K}/m \rceil$ guessed vectors. Equivalently, kernel search guesses an $(r+\ell)$-dimensional superspace of the row space of $\mathbf{E}(\widehat{\boldsymbol{x}})$ where $\ell := n - r - b$. By Theorem 1, Step 1 succeeds after $q^{r(n-r-\ell)-h_c+t_c} = q^{rb-h_c+t_c} = q^{r\lceil \tilde{K}/m\rceil - h_c + t_c}$ iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor. Step 0 is one-time and its cost can be neglected.

1. Step 1 only constructs $\mathbf{R} \in \mathbb{F}_q^{(r+\ell)\times n}$ and reads off a basis of its right null space. Since $\mathbf{R}$ is already in systematic form, this is negligible compared to the following linear-algebra steps.
2. Step 2 sets up the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T}$ for all $i = 1, \ldots, b$. Write

$$\mathbf{R} = \begin{pmatrix} \mathbf{I}_r & \mathbf{B} \\ \mathbf{0} & \mathbf{A} \end{pmatrix}$$

   as in Theorem 1, where $\mathbf{A}$ is zero except for an $\ell \times \ell$ identity block in the column set $\mathcal{I}$. Let $\mathcal{J}$ be the complementary set of $b = n - r - \ell$ columns. Then a systematic basis of the right null space of $\mathbf{R}$ is given by the vectors whose $\mathcal{J}$-part is an identity matrix, whose $\mathcal{I}$-part is zero, and whose first r entries are the corresponding negatives of the columns of $\mathbf{B}_{*,\mathcal{J}}$. Hence each $\boldsymbol{v}_i$ has only $r+1$ possibly nonzero entries. A single scalar equation, i.e., one row of $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T}$, can therefore be formed by combining only $r+1$ columns of each of the $\tilde{K}$ coefficient matrices of the reduced MinRank instance, at cost $\tilde{K}r$ $\mathbb{F}_q$-operations. We keep exactly $\tilde{K}$ scalar equations, so that our system is square. Thus, setting up the system costs $\tilde{K}^2 r$ $\mathbb{F}_q$-operations. Then we solve the resulting $\tilde{K} \times \tilde{K}$ linear system which costs $\tilde{K}^\omega$ $\mathbb{F}_q$-operations.
3. Step 3 computes $\mathbf{E}(\boldsymbol{x}) \in \mathbb{F}_q^{m\times n}$ which costs $\tilde{K}mn$ $\mathbb{F}_q$-operations and then checks its rank, which costs $mn\min\{m,n\}^{\omega-2}$ $\mathbb{F}_q$-operations.

The polynomial cost equals the sum of the above costs. Up to a constant factor, this is just the maximum of the above costs. □

Proposition 3 (Complexity of Kernel Search with Bit-Level Hints (Algorithm 2))**.** *Suppose $q = 2^\nu$. Set $\tilde{K} := K - h_s/\nu$ and $\ell := n - r - \lceil \tilde{K}/m \rceil$. As mentioned in Proposition 2, we can assume $\tilde{K} > 0$ and $\ell \geq 0$ so that the problem is not trivial.*

Assume the (reasonable) heuristic that the equations obtained from $\widehat{\mathbf{S}}$ have rank h_s and the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 2 is[4]

$$P\ 2^{\nu r\lceil \tilde{K}/m\rceil - h_c + t_c} \quad \textit{(up to a constant factor)}$$

where h_c is the total number of known bits in $\widehat{\mathbf{C}}'$ and t_c is the number of known bits contained in any ℓ

[4]This assumes a unique rank-r solution. If there are N rank-r solutions, the expected number of iterations is divided by N. For a random affine MinRank instance, and heuristically also for planted instances, the number of rank-r solutions is, up to a constant factor, $q^{K-(m-r)(n-r)}$ when $K > (m-r)(n-r)$.

columns of $\widehat{\mathbf{C}}'$ with the fewest known bits. The polynomial cost is

$$P = \max \left\{ \begin{array}{l} Kr\lceil \tilde{K} \rceil \ \mathbb{F}_{2^\nu}\text{-operations}, \\ \nu^2 K \lceil \tilde{K} \rceil \ \mathbb{F}_2\text{-operations}, \\ (\nu\tilde{K})^\omega \ \mathbb{F}_2\text{-operations}, \\ \min\{(\nu\tilde{K})^2 h_s, (\nu K)^\omega\} \ \mathbb{F}_2\text{-operations}, \\ Kmn \ \mathbb{F}_{2^\nu}\text{-operations}, \\ mn \min\{m,n\}^{\omega-2} \ \mathbb{F}_{2^\nu}\text{-operations} \end{array} \right\} \quad \textit{(up to a constant factor)}$$

where the highlighted term dominates for typical parameters.

If $h_s = 0$, then Step 2 can instead be performed directly over $\mathbb{F}_{2^\nu}$, and the polynomial cost becomes

$$P_0 = \max \left\{ \begin{array}{l} K^2 r \ \mathbb{F}_{2^\nu}\text{-operations}, \\ K^\omega \ \mathbb{F}_{2^\nu}\text{-operations}, \\ Kmn \ \mathbb{F}_{2^\nu}\text{-operations}, \\ mn \min\{m,n\}^{\omega-2} \ \mathbb{F}_{2^\nu}\text{-operations} \end{array} \right\} \quad \textit{(up to a constant factor)}$$

where the highlighted term dominates for typical parameters.

Proof. Exponential cost. After expanding $x_1, \ldots, x_K \in \mathbb{F}_{2^\nu}$ over a fixed basis of $\mathbb{F}_{2^\nu}$ over $\mathbb{F}_2$, the unknowns are represented by νK binary variables. The h_s known bits of $\widehat{\mathbf{S}}$ give h_s equations over $\mathbb{F}_2$. Each guessed kernel vector gives m equations over $\mathbb{F}_{2^\nu}$, or equivalently νm equations over $\mathbb{F}_2$. Thus we need $\nu m b + h_s \geq \nu K$, and thus guess $b := \lceil \tilde{K}/m \rceil$ kernel vectors. Equivalently, kernel search guesses an $(r+\ell)$-dimensional superspace of the row space of $\mathbf{E}(\widehat{\boldsymbol{x}})$ with $\ell := n - r - b$. By Corollary 1, Step 1 succeeds after $2^{\nu r(n-r-\ell)-h_c+t_c} = 2^{\nu r b - h_c + t_c} = 2^{\nu r \lceil \tilde{K}/m \rceil - h_c + t_c}$ iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor. Step 0 is one-time and its cost can be neglected.

1. Step 1 only constructs $\mathbf{R} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n}$ according to Corollary 1 and reads off a basis of its right null space. As before, this is negligible compared to the following linear-algebra steps.
2. Step 2 does the following.
   (a) Set up the system $\mathbf{E}(\boldsymbol{x})\boldsymbol{v}_i^\mathsf{T} = \mathbf{0}^\mathsf{T}$ over $\mathbb{F}_{2^\nu}$ for $i = 1, \ldots, b$. As in the proof of Proposition 2, the systematic form of $\mathbf{R}$ lets us choose $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b$ so that each $\boldsymbol{v}_i$ has only $r+1$ possibly nonzero entries. Thus a single scalar equation over $\mathbb{F}_{2^\nu}$ can be formed by combining only $r+1$ columns of each of the K coefficient matrices, at cost Kr $\mathbb{F}_{2^\nu}$-operations. Now we have νK $\mathbb{F}_2$-unknowns and one $\mathbb{F}_{2^\nu}$-equation yields ν $\mathbb{F}_2$-equations. Since we have h_s $\mathbb{F}_2$-equations already from Step 0, we need $\lceil (\nu K - h_s)/\nu \rceil = \lceil \tilde{K} \rceil$ $\mathbb{F}_{2^\nu}$-equations. Hence this setup costs $Kr\lceil \tilde{K} \rceil$ $\mathbb{F}_{2^\nu}$-operations.
   (b) We then expand these $\lceil \tilde{K} \rceil$ $\mathbb{F}_{2^\nu}$-equations over $\mathbb{F}_2$. To expand a term of the form $\beta_i x_i$ with $\beta_i \in \mathbb{F}_{2^\nu}$, write $x_i = x_{i,1}\gamma_1 + \cdots + x_{i,\nu}\gamma_\nu$ for a fixed $\mathbb{F}_2$-basis $\gamma_1, \ldots, \gamma_\nu$ of $\mathbb{F}_{2^\nu}$, and multiply out $\beta_i(x_{i,1}\gamma_1 + \cdots + x_{i,\nu}\gamma_\nu)$, which costs ν $\mathbb{F}_{2^\nu}$-operations. If we have a power basis, i.e., $(\gamma_1, \ldots, \gamma_\nu) = (1, \alpha, \ldots, \alpha^{\nu-1})$, then the multiplication can instead be carried out with ν^2 $\mathbb{F}_2$-operations, which is what we assume. Now we have K unknowns per $\mathbb{F}_{2^\nu}$-equation and $\lceil \tilde{K} \rceil$ such equations, hence the expansion cost is $\nu^2 K \lceil \tilde{K} \rceil$ $\mathbb{F}_2$-operations.
   (c) We have $\nu\lceil \tilde{K} \rceil$ $\mathbb{F}_2$-equations from the previous substep, h_s additional $\mathbb{F}_2$-equations from Step 0 and $\nu K = \nu\tilde{K} + h_s$ $\mathbb{F}_2$-unknowns. We discard any additional equations from the previous substep so that we have exactly $\nu\tilde{K} + h_s$ binary equations and unknowns. Solving this square system directly costs $(\nu\tilde{K} + h_s)^\omega = (\nu K)^\omega$ $\mathbb{F}_2$-operations.

   Alternatively, Step 0 can already express h_s of the $\mathbb{F}_2$-unknowns in terms of the remaining $\nu\tilde{K}$

$\mathbb{F}_2$-unknowns, so that our system of equations has the form,

$$\begin{pmatrix} * & * \\ * & \mathbf{I}_{h_s} \end{pmatrix} \begin{pmatrix} \vdots \\ x_{i,j} \\ \vdots \end{pmatrix} = \mathbf{0}$$

The matrix on the left has size $(\nu\tilde{K} + h_s) \times (\nu\tilde{K} + h_s)$ and we assume the h_s $\mathbb{F}_2$-unknowns from Step 0 are at the end of the vector $\begin{pmatrix} \cdots x_{i,j} \cdots \end{pmatrix}^{\mathsf{T}}$, so that the matrix has an identity at the bottom right. We can then do Gaussian elimination to bring the matrix into the form $\begin{pmatrix} * & \mathbf{0} \\ * & \mathbf{I}_{h_s} \end{pmatrix}$ and then solve the smaller $\nu\tilde{K} \times \nu\tilde{K}$ system for the first $\nu\tilde{K}$ $\mathbb{F}_2$-unknowns. One can then recover the remaining h_s unknowns whose cost can be neglected as it does not dominate. The Gaussian elimination costs $(\nu\tilde{K})^2 h_s$ $\mathbb{F}_2$-operations, the solving costs $(\nu\tilde{K})^\omega$ $\mathbb{F}_2$-operations, and hence in total $\max\{(\nu\tilde{K})^2 h_s, (\nu\tilde{K})^\omega\}$ $\mathbb{F}_2$-operations (up to a constant).
We pick the method that has the smaller complexity. Therefore the cost is

$$\begin{aligned} &\min\{(\nu\tilde{K} + h_s)^\omega, \max\{(\nu\tilde{K})^2 h_s, (\nu\tilde{K})^\omega\}\} \\ =&\max\{(\nu\tilde{K})^\omega, \min\{(\nu\tilde{K})^2 h_s, (\nu\tilde{K} + h_s)^\omega\}\} \ \mathbb{F}_2\text{-operations.} \end{aligned}$$

where we use the identity $\min\{A, \max\{B, C\}\} = \max\{C, \min\{B, A\}\}$ for $A \geq C$.

If $h_s = 0$, then Steps 2(b) and 2(c) are unnecessary and $\tilde{K} = K$. In this case, Step 2(a) sets up K equations over $\mathbb{F}_{2^\nu}$ with cost $K^2 r$ $\mathbb{F}_{2^\nu}$-operations, and the resulting $K \times K$ system can be solved directly in K^ω $\mathbb{F}_{2^\nu}$-operations.

3. Step 3 reconstructs $\boldsymbol{x} \in \mathbb{F}_{2^\nu}^K$ and then computes $\mathbf{E}(\boldsymbol{x})$, which costs Kmn $\mathbb{F}_{2^\nu}$-operations. Then it checks its rank, which costs $mn \min\{m, n\}^{\omega-2}$ $\mathbb{F}_{2^\nu}$-operations.

□

B Row GRS with and without Hints (Algorithms and Cost Analysis)

Algorithm 3. Row GRS with Hints over $\mathbb{F}_q$

Input: $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\boldsymbol{t} \in \mathbb{F}_{q^m}^{n-k}$, and a small integer $r > 0$. Assume there exists a solution $\widehat{\boldsymbol{e}} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\widehat{\boldsymbol{e}}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ and

$$\mathrm{ext}(\widehat{\boldsymbol{e}}) = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$$

for some full-rank $\widehat{\mathbf{S}} \in \mathbb{F}_q^{m\times r}$ and some $\widehat{\mathbf{C}}' \in \mathbb{F}_q^{r\times(n-r)}$. (Note this implies $\widehat{\boldsymbol{e}}$ has rank r.) Additionally, suppose h_s entries of $\widehat{\mathbf{S}}$ and h_c entries of $\widehat{\mathbf{C}}'$ are known.
Output: A vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\boldsymbol{e}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ and $\mathrm{rank}(\boldsymbol{e}) = r$.

0. Since $\mathrm{ext}(\widehat{\boldsymbol{e}}) = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$, the first r columns of $\mathrm{ext}(\widehat{\boldsymbol{e}})$ are exactly $\widehat{\mathbf{S}}$. Hence each known hint position (i,j) in $\widehat{\mathbf{S}}$ gives the equation $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{S}_{i,j}$. Set $\ell := n - k - r + \lfloor h_s/m \rfloor$.
1. Draw $\mathbf{B} := \mathbf{R} \in \mathbb{F}_q^{(r+\ell)\times n}$ according to Theorem 1.
2. Set up the system
$$\mathbf{H}(\boldsymbol{a}\mathbf{B})^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$$
for the unknown $\boldsymbol{a} \in \mathbb{F}_{q^m}^{r+\ell}$. Since $\mathbf{B}$ is systematic and $\boldsymbol{e} = \boldsymbol{a}\mathbf{B}$, the first r entries of $\boldsymbol{e}$ are exactly the first r entries of $\boldsymbol{a}$, i.e., $e_j = a_j$ for $1 \le j \le r$. Further, for $1 \le j \le r$, we have $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{S}_{i,j}$. Hence the h_s equations from Step 0 become equations of the form $\mathrm{ext}(\boldsymbol{a})_{i,j} = \widehat{S}_{i,j}$. Substitute these known values into the system above and solve the resulting system for $\boldsymbol{a}$.
3. If $\mathrm{rank}(\boldsymbol{e}) \neq r$, go to Step 1. Otherwise, return $\boldsymbol{e}$.

Algorithm 4. Row GRS with Bit-Level Hints over $\mathbb{F}_{2^\nu}$

Input: $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\boldsymbol{t} \in \mathbb{F}_{q^m}^{n-k}$, and a small integer $r > 0$, where $q = 2^\nu$. Assume there exists a solution $\widehat{\boldsymbol{e}} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\widehat{\boldsymbol{e}}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ and

$$\mathrm{ext}(\widehat{\boldsymbol{e}}) = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$$

for some full-rank $\widehat{\mathbf{S}} \in \mathbb{F}_{2^\nu}^{m\times r}$ and some $\widehat{\mathbf{C}}' \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$. (Note this implies $\widehat{\boldsymbol{e}}$ has rank r.) Additionally, suppose h_s bits of $\widehat{\mathbf{S}}$ and h_c bits of $\widehat{\mathbf{C}}'$ are known.
Output: A vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\boldsymbol{e}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ and $\mathrm{rank}(\boldsymbol{e}) = r$.

0. Since $\mathrm{ext}(\widehat{\boldsymbol{e}}) = \widehat{\mathbf{S}}(\mathbf{I}_r \mid \widehat{\mathbf{C}}')$, the first r columns of $\mathrm{ext}(\widehat{\boldsymbol{e}})$ are exactly $\widehat{\mathbf{S}}$. Hence each known bit position k' in an $\mathbb{F}_{2^\nu}$-entry (i,j) of $\widehat{\mathbf{S}}$ gives one $\mathbb{F}_2$-equation obtained by expanding the $\mathbb{F}_{2^\nu}$-equation $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{S}_{i,j}$ into ν $\mathbb{F}_2$-equations and then picking the k'th equation. Set $\ell := n - k - r + \lfloor h_s/(\nu m) \rfloor$.
1. Draw $\mathbf{B} := \mathbf{R} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n}$ according to Corollary 1.
2. Set up the system
$$\mathbf{H}(\boldsymbol{a}\mathbf{B})^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$$
and expand over $\mathbb{F}_2$. Since $\mathbf{B}$ is systematic and $\boldsymbol{e} = \boldsymbol{a}\mathbf{B}$, the first r entries of $\boldsymbol{e}$ are exactly the first r entries of $\boldsymbol{a}$, i.e., $e_j = a_j$ for $1 \le j \le r$. Further, for $1 \le j \le r$, we have $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{S}_{i,j}$. Hence the h_s $\mathbb{F}_2$-equations from Step 0 become equations of the form "bit k' of $\mathrm{ext}(\boldsymbol{a})_{i,j}$ equals bit k' of $\widehat{S}_{i,j}$". Substitute these known bits into the above system and solve the resulting system for $\boldsymbol{a}$.
3. If $\mathrm{rank}(\boldsymbol{e}) \neq r$, go to Step 1. Otherwise, return $\boldsymbol{e}$.

Proposition 4 (Complexity of Row GRS with $\mathbb{F}_q$-Hints (Algorithm 3))**.** *Set $\ell := n - k - r + \lfloor h_s/m \rfloor$. So that the RSD problem is not trivial, assume*[5] *$r \le \min\{1, m/n\}(n-k)$ (this implies $0 \le \ell$) and*[6] *$\ell \le n - r$.*

Assume the (reasonable) heuristic that the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 3 is upper-bounded[7] *by*

$$P\, q^{r(k-\lfloor h_s/m \rfloor)-h_c+t_c} \quad \textit{(up to a constant factor)},$$

[5] The RSD problem is easy to solve if $r \ge \min\{1, \frac{m}{n}\}(n-k)$. To see this, write $\boldsymbol{e} = \boldsymbol{a}\mathbf{B}$ with $\boldsymbol{a} \in \mathbb{F}_{q^m}^r$ and $\mathbf{B} \in \mathbb{F}_q^{r\times n}$. The syndrome system becomes $\mathbf{H}(\boldsymbol{a}\mathbf{B})^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$. If we fix $\boldsymbol{a}$ randomly and solve for $\mathbf{B}$, then the condition #unknowns $\ge$ #equations is equivalent to $r \ge \frac{m}{n}(n-k)$, and if we fix $\mathbf{B}$ instead, then the same condition becomes $r \ge n - k$.

[6] The condition $\ell \le n - r$ is equivalent to $k - \lfloor h_s/m \rfloor \ge 0$. If this is not true, then the problem is trivial since we can set $\ell = n - r$, i.e., search for a superspace of dimension $r + \ell = n$ in $\mathbb{F}_q^n$.

[7] This assumes a unique rank-r solution. If there are N rank-r solutions satisfying the syndrome, the expected number of iterations is divided by N. For a random RSD instance, and heuristically also for planted instances, the number of rank-r solutions satisfying the syndrome is, up to a constant factor, $q^{r(m+n-r)-m(n-k)}$ when $r(m+n-r) > m(n-k)$.

where h_c is the total number of known entries in $\widehat{\mathbf{C}}'$ and t_c is the number of known entries contained in any ℓ columns of $\widehat{\mathbf{C}}'$ with the fewest known entries. An upper bound on the polynomial cost is

$$P = (m(n-k))^{\omega}\ \mathbb{F}_q\text{-operations} \quad \textit{(up to a constant factor).}$$

If $h_s = 0$, then Step 2 can instead be performed directly over $\mathbb{F}_{q^m}$, and the polynomial cost equals P_0 up to a constant factor, where

$$P_0 = \max\left\{ \begin{array}{l} m(n-k)rk\min\{r,k\}^{\omega-3}\ \mathbb{F}_q\text{-operations}, \\ (n-k)^{\omega}\ \mathbb{F}_{q^m}\text{-operations} \end{array} \right\}.$$

Depending on the parameters and the cost model for $\mathbb{F}_{q^m}$-operations, either term may dominate. Setting $h_s = h_c = 0$ recovers the original row-GRS complexity, i.e., the polynomial cost is P_0 and the exponential cost is obtained by setting $h_s = h_c = 0$ and thereby also $t_c = 0$; see Remark 3 above.

Proof. Exponential cost. The guessed matrix $\mathbf{B}$ has $r+\ell$ rows. The $(n-k)$ syndrome equations over $\mathbb{F}_{q^m}$ give $m(n-k)$ equations over $\mathbb{F}_q$, while the hints from $\widehat{\mathbf{S}}$ give h_s additional equations over $\mathbb{F}_q$. We have $m(r+\ell)$ $\mathbb{F}_q$-unknowns in total. Thus, we need $m(r+\ell) \le m(n-k)+h_s$ and so we set $\ell := n-k-r+\lfloor h_s/m \rfloor$ which is the largest possible value for ℓ. By Theorem 1, Step 1 succeeds after $q^{r(n-r-\ell)-h_c+t_c} = q^{r(k-\lfloor h_s/m \rfloor)-h_c+t_c}$ iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor.

1. Step 1 only constructs the systematic matrix $\mathbf{B}$ according to Theorem 1, and as before this cost is negligible.
2. Step 2 consists of the following substeps. Since

$$r+\ell = n-k+\lfloor h_s/m \rfloor$$

and $0 \le h_s \le mr \le m(n-k)$, we have $n-k \le r+\ell \le 2(n-k)$. Hence $r+\ell$ equals $n-k$ up to a constant factor.

(a) Compute $\mathbf{H}\mathbf{B}^{\mathsf{T}} \in \mathbb{F}_{q^m}^{(n-k)\times(r+\ell)}$. After permuting columns, $\mathbf{B}$ has the form

$$\begin{pmatrix} \mathbf{I}_r & * & * \\ \mathbf{0} & \mathbf{I}_\ell & \mathbf{0} \end{pmatrix}.$$

We can subtract suitable multiples of the bottom ℓ rows from the top r rows, which only changes the coordinates of the unknown vector $\boldsymbol{a}$. Thus, we may use the row-equivalent form

$$\begin{pmatrix} \mathbf{I}_r & \mathbf{0} & \mathbf{B}' \\ \mathbf{0} & \mathbf{I}_\ell & \mathbf{0} \end{pmatrix},$$

where $\mathbf{B}'$ has $k-\lfloor h_s/m \rfloor$ columns. Hence $\mathbf{H}\mathbf{B}^{\mathsf{T}}$ is obtained by copying the corresponding $r+\ell$ columns of $\mathbf{H}$ and adding a product $\mathbf{H}_{*,Q}(\mathbf{B}')^{\mathsf{T}}$ to the first r columns, where Q indexes the remaining $k-\lfloor h_s/m \rfloor$ columns. The latter can be computed as m matrix multiplications over $\mathbb{F}_q$, one for each $\mathbb{F}_q$-slice of $\mathbf{H}$. Otherwise, since the cost of multiplying an $a \times b$ matrix with an $b \times c$ matrix is $abc\min\{a,b,c\}^{\omega-3}$ field operations (up to a constant), this costs

$$m(n-k)r\,(k-\lfloor h_s/m \rfloor)\min\left\{n-k, r, k-\lfloor h_s/m \rfloor\right\}^{\omega-3}\ \mathbb{F}_q\text{-operations}.$$

This is at most the dense-multiplication cost $mn(r+\ell)(n-k)^{\omega-2}$ $\mathbb{F}_q$-operations (i.e., the cost when we do not exploit the systematic structure of $\mathbf{B}$), which equals $mn(n-k)^{\omega-1}$ $\mathbb{F}_q$-operations up to a constant factor. Since $r > 0$ and $r \le \min\{1, m/n\}(n-k)$ imply $m(n-k) \ge n$, this cost is upper-bounded by $(m(n-k))^{\omega}$ $\mathbb{F}_q$-operations.

(b) We now expand the system $\mathbf{H}\mathbf{B}^{\mathsf{T}}\boldsymbol{a}^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$ over $\mathbb{F}_q$. These are $(n-k)$ $\mathbb{F}_{q^m}$-equations and $(r+\ell)$ $\mathbb{F}_{q^m}$-unknowns. By the same expansion argument as in Step 2(b) of the proof of Proposition 3, this costs $m^2(n-k)(r+\ell)$ $\mathbb{F}_q$-operations. By the relation above, this equals $m^2(n-k)^2$ $\mathbb{F}_q$-operations up to a constant factor. But this will be dominated by the cost $(m(n-k))^\omega$ which appears next.

(c) We now have $m(n-k)$ $\mathbb{F}_q$-equations from the previous substep and h_s additional $\mathbb{F}_q$-equations from Step 0. The number of $\mathbb{F}_q$-unknowns is $m(r+\ell) = m(n-k) + m\lfloor h_s/m \rfloor$, and each equation from Step 0 fixes one of these unknowns to a known value. We substitute these h_s known values into the $m(n-k)$ syndrome equations, which costs $h_s m(n-k)$ $\mathbb{F}_q$-operations and leaves

$$m(r+\ell) - h_s = m(n-k) + m\lfloor h_s/m \rfloor - h_s$$

unknowns. Since this number is at most $m(n-k)$, solving the remaining system costs at most $(m(n-k))^\omega$ $\mathbb{F}_q$-operations. Moreover, $h_s \le mr \le m(n-k)$, so the substitution cost is dominated by $(m(n-k))^\omega$. Thus, the total cost of this substep is upper-bounded by $(m(n-k))^\omega$ $\mathbb{F}_q$-operations. This is a possibly loose upper bound on the cost of this substep, which we use to keep the expression for the polynomial cost simple.

If $h_s = 0$, then Steps 2(b) and 2(c) are unnecessary. In this case $\ell = n-k-r$, so the multiplication in Step 2(a) costs $m(n-k)rk\min\{r,k\}^{\omega-3}$ $\mathbb{F}_q$-operations up to a constant factor, where we used $r \le n-k$. The resulting system has $(n-k)$ equations and unknowns over $\mathbb{F}_{q^m}$ and can be solved directly with cost $(n-k)^\omega$ $\mathbb{F}_{q^m}$-operations.

3. In Step 3, since $\mathbf{B}$ has full row rank, $\operatorname{rank}(\boldsymbol{e}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{a})\mathbf{B}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{a}))$. Thus, it suffices to compute the rank of the $m \times (r+\ell)$ matrix $\operatorname{ext}(\boldsymbol{a})$, which costs $m(r+\ell)\min\{m, r+\ell\}^{\omega-2}$ $\mathbb{F}_q$-operations. By the relation above, this equals $m(n-k)\min\{m, n-k\}^{\omega-2}$ $\mathbb{F}_q$-operations up to a constant factor. This is upper-bounded by both $(m(n-k))^\omega$ and $mn(n-k)^{\omega-1}$ $\mathbb{F}_q$-operations.

The polynomial cost is upper-bounded by the maximum of the displayed costs up to a constant factor. □

Proposition 5 (Complexity of Row GRS with Bit-Level Hints (Algorithm 4))**.** *Suppose $q = 2^\nu$. Set $\ell := n-k-r+\lfloor h_s/(\nu m) \rfloor$. As mentioned in Proposition 4, we can assume $r \le \min\{1, m/n\}(n-k)$ (which implies $0 \le \ell$) and $\ell \le n-r$.*

Assume the (reasonable) heuristic that the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 4 is upper-bounded[8] *by*

$$P\ 2^{\nu r(k-\lfloor h_s/(\nu m)\rfloor)-h_c+t_c} \quad \textit{(up to a constant factor)},$$

where h_c is the total number of known bits in $\widehat{\mathbf{C}}'$ and t_c is the number of known bits contained in any ℓ columns of $\widehat{\mathbf{C}}'$ with the fewest known bits. An upper bound on the polynomial cost is

$$P = \max\left\{ \begin{array}{l} mn(n-k)^{\omega-1}\ \mathbb{F}_{2^\nu}\textit{-operations}, \\ (\nu m(n-k))^\omega\ \mathbb{F}_2\textit{-operations} \end{array} \right\}$$

up to a constant factor. The highlighted term generally dominates, e.g., if we assume one $\mathbb{F}_{2^\nu}$-operation costs ν $\mathbb{F}_2$-operations.

If $h_s = 0$, then Step 2 can instead be performed directly over $\mathbb{F}_{2^{\nu m}}$, and the polynomial cost equals P_0 up to a constant factor, where

$$P_0 = \max\left\{ \begin{array}{l} m(n-k)rk\min\{r,k\}^{\omega-3}\ \mathbb{F}_{2^\nu}\textit{-operations}, \\ (n-k)^\omega\ \mathbb{F}_{2^{\nu m}}\textit{-operations} \end{array} \right\}.$$

[8] With $q = 2^\nu$, the same caveat regarding multiple solutions as explained in footnote 7 applies.

Depending on the parameters and the cost model for $\mathbb{F}_{2^{\nu m}}$-operations, either term may dominate.

Proof. Exponential cost. The guessed matrix $\mathbf{B}$ has $r+\ell$ rows. After expansion over $\mathbb{F}_2$, the $(n-k)$ syndrome equations over $\mathbb{F}_{2^{\nu m}}$ give $\nu m(n-k)$ equations over $\mathbb{F}_2$, while the hints from $\widehat{\mathbf{S}}$ give h_s additional equations over $\mathbb{F}_2$. We have $\nu m(r+\ell)$ $\mathbb{F}_2$-unknowns. Thus, $\ell := n-k-r+\lfloor h_s/(\nu m)\rfloor$ is the largest integer for which $\nu m(r+\ell) \le \nu m(n-k)+h_s$. By Corollary 1, Step 1 succeeds after $2^{\nu r(n-r-\ell)-h_c+t_c} = 2^{\nu r(k-\lfloor h_s/(\nu m)\rfloor)-h_c+t_c}$ iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor.

1. Step 1 only constructs the systematic matrix $\mathbf{B}$ according to Corollary 1, and as before this cost is negligible.

2. Step 2 consists of the following substeps. Since

$$r+\ell = n-k+\lfloor h_s/(\nu m)\rfloor$$

and $0 \le h_s \le \nu m r \le \nu m(n-k)$, we have $n-k \le r+\ell \le 2(n-k)$. Hence $r+\ell$ equals $n-k$ up to a constant factor.

   (a) Compute $\mathbf{H}\mathbf{B}^\mathsf{T} \in \mathbb{F}_{2^{\nu m}}^{(n-k)\times(r+\ell)}$. As in Step 2(a) of the proof of Proposition 4, after permuting columns and applying row operations, we may use the row-equivalent form

$$\begin{pmatrix} \mathbf{I}_r & \mathbf{0} & \mathbf{B}' \\ \mathbf{0} & \mathbf{I}_\ell & \mathbf{0} \end{pmatrix},$$

   where $\mathbf{B}'$ has $k-\lfloor h_s/(\nu m)\rfloor$ columns. Thus computing $\mathbf{H}\mathbf{B}^\mathsf{T}$ only requires copying the pivot columns of $\mathbf{H}$ and multiplying the remaining $k-\lfloor h_s/(\nu m)\rfloor$ columns by $(\mathbf{B}')^\mathsf{T}$. This costs at most $mn(n-k)^{\omega-1}$ $\mathbb{F}_{2^\nu}$-operations up to a constant factor.

   (b) To exploit the bit-level equations from Step 0, expand $\mathbf{H}\mathbf{B}^\mathsf{T}\boldsymbol{a}^\mathsf{T} = \boldsymbol{t}^\mathsf{T}$ from a system over $\mathbb{F}_{2^{\nu m}}$ to one over $\mathbb{F}_2$. There are $(n-k)$ $\mathbb{F}_{2^{\nu m}}$-equations and $(r+\ell)$ $\mathbb{F}_{2^{\nu m}}$-unknowns. By the same expansion argument as in Step 2(b) of the proof of Proposition 3, this costs $(\nu m)^2(n-k)(r+\ell)$ $\mathbb{F}_2$-operations. By the relation above, this equals $(\nu m)^2(n-k)^2$ $\mathbb{F}_2$-operations up to a constant factor. But this will be dominated by the cost $(\nu m(n-k))^\omega$ that appears next.

   (c) The same argument as in Step 2(c) of the proof of Proposition 4 applies, with m replaced by νm and $\mathbb{F}_q$ replaced by $\mathbb{F}_2$. Thus, the cost of substituting the h_s known bits and solving the resulting system is upper-bounded by $(\nu m(n-k))^\omega$ $\mathbb{F}_2$-operations. This is a possibly loose upper bound on the cost of this substep, which we use to keep the expression for the polynomial cost simple.

   If $h_s = 0$, Steps 2(b) and 2(c) are unnecessary. In this case $\ell = n-k-r$, so the multiplication in Step 2(a) costs $m(n-k)rk\min\{r,k\}^{\omega-3}$ $\mathbb{F}_{2^\nu}$-operations up to a constant factor, where we used $r \le n-k$. The resulting system has $(n-k)$ equations and unknowns over $\mathbb{F}_{2^{\nu m}}$ and can be solved directly in $(n-k)^\omega$ $\mathbb{F}_{2^{\nu m}}$-operations.

3. In Step 3, since $\mathbf{B}$ has full row rank, $\operatorname{rank}(\boldsymbol{e}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{a})\mathbf{B}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{a}))$. Thus, as in Step 3 of the proof of Proposition 4, it suffices to compute the rank of the $m \times (r+\ell)$ matrix $\operatorname{ext}(\boldsymbol{a})$. This costs $m(r+\ell)\min\{m, r+\ell\}^{\omega-2}$ $\mathbb{F}_{2^\nu}$-operations. By the relation above, this equals $m(n-k)\min\{m, n-k\}^{\omega-2}$ $\mathbb{F}_{2^\nu}$-operations up to a constant factor, which is upper-bounded by $mn(n-k)^{\omega-1}$ $\mathbb{F}_{2^\nu}$-operations.

The polynomial cost is upper-bounded by the maximum of the displayed costs up to a constant factor. □

C Column GRS with and without Hints (Algorithms and Cost Analysis)

Algorithm 5. Column GRS with Hints over $\mathbb{F}_q$

Input: $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\boldsymbol{t} \in \mathbb{F}_{q^m}^{n-k}$, and a small integer $r > 0$. Assume there exists a solution $\widehat{\boldsymbol{e}} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\widehat{\boldsymbol{e}}^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$ and

$$\mathrm{ext}(\widehat{\boldsymbol{e}}) = \begin{pmatrix} \mathbf{I}_r \\ \widehat{\mathbf{S}}' \end{pmatrix} \widehat{\mathbf{C}}$$

for some $\widehat{\mathbf{S}}' \in \mathbb{F}_q^{(m-r)\times r}$ and some full-rank $\widehat{\mathbf{C}} \in \mathbb{F}_q^{r\times n}$. (Note this implies $\widehat{\boldsymbol{e}}$ has rank r.) Additionally, suppose h_s entries of $\widehat{\mathbf{S}}'$ and h_c entries of $\widehat{\mathbf{C}}$ are known.
Output: A vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\boldsymbol{e}^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$ and $\mathrm{rank}(\boldsymbol{e}) = r$.

0. Since $\mathrm{ext}(\widehat{\boldsymbol{e}}) = \begin{pmatrix} \mathbf{I}_r \\ \widehat{\mathbf{S}}' \end{pmatrix} \widehat{\mathbf{C}}$, the first r rows of $\mathrm{ext}(\widehat{\boldsymbol{e}})$ are exactly $\widehat{\mathbf{C}}$. Hence each known hint position (i,j) in $\widehat{\mathbf{C}}$ gives the equation $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{C}_{i,j}$. Set $\ell := m - r - \lceil (km - h_c)/n \rceil$.
1. Draw $\mathbf{A}^{\mathsf{T}} := \mathbf{R} \in \mathbb{F}_q^{(r+\ell)\times m}$ according to Theorem 1 applied to $(\mathbf{I}_r \mid \widehat{\mathbf{S}}'^{\mathsf{T}})$.
2. Set up the system
$$\mathbf{H}\big(\mathrm{ext}^{-1}(\mathbf{A})\mathbf{B}\big)^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$$
for the unknown $\mathbf{B} \in \mathbb{F}_q^{(r+\ell)\times n}$. Since $\mathbf{A}^{\mathsf{T}}$ is systematic and $\mathrm{ext}(\boldsymbol{e}) = \mathbf{AB}$, the first r rows of $\mathrm{ext}(\boldsymbol{e})$ are exactly the first r rows of $\mathbf{B}$. Further, for $1 \le i \le r$, we have $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{C}_{i,j}$. Hence the h_c equations from Step 0 become equations of the form $B_{i,j} = \widehat{C}_{i,j}$. Substitute these known values into the above system and solve the resulting system for $\mathbf{B}$, and then set $\boldsymbol{e} := \mathrm{ext}^{-1}(\mathbf{A})\mathbf{B}$.
3. If $\mathrm{rank}(\boldsymbol{e}) \neq r$, go to Step 1. Otherwise, return $\boldsymbol{e}$.

Algorithm 6. Column GRS with Bit-Level Hints over $\mathbb{F}_{2^\nu}$

Input: $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\boldsymbol{t} \in \mathbb{F}_{q^m}^{n-k}$, and a small integer $r > 0$, where $q = 2^\nu$. Assume there exists a solution $\widehat{\boldsymbol{e}} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\widehat{\boldsymbol{e}}^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$ and

$$\mathrm{ext}(\widehat{\boldsymbol{e}}) = \begin{pmatrix} \mathbf{I}_r \\ \widehat{\mathbf{S}}' \end{pmatrix} \widehat{\mathbf{C}}$$

for some $\widehat{\mathbf{S}}' \in \mathbb{F}_{2^\nu}^{(m-r)\times r}$ and some full-rank $\widehat{\mathbf{C}} \in \mathbb{F}_{2^\nu}^{r\times n}$. (Note this implies $\widehat{\boldsymbol{e}}$ has rank r.) Additionally, suppose h_s bits of $\widehat{\mathbf{S}}'$ and h_c bits of $\widehat{\mathbf{C}}$ are known.
Output: A vector $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{H}\boldsymbol{e}^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$ and $\mathrm{rank}(\boldsymbol{e}) = r$.

0. Since $\mathrm{ext}(\widehat{\boldsymbol{e}}) = \begin{pmatrix} \mathbf{I}_r \\ \widehat{\mathbf{S}}' \end{pmatrix} \widehat{\mathbf{C}}$, the first r rows of $\mathrm{ext}(\widehat{\boldsymbol{e}})$ are exactly $\widehat{\mathbf{C}}$. Hence each known bit position k' in an $\mathbb{F}_{2^\nu}$-entry (i,j) of $\widehat{\mathbf{C}}$ gives one $\mathbb{F}_2$-equation obtained by expanding the $\mathbb{F}_{2^\nu}$-equation $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{C}_{i,j}$ into ν $\mathbb{F}_2$-equations and then picking the k'th equation. Set $\ell := m - r - \lceil (km - (h_c/\nu))/n \rceil$.
1. Draw $\mathbf{A}^{\mathsf{T}} := \mathbf{R} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times m}$ according to Corollary 1 applied to $(\mathbf{I}_r \mid \widehat{\mathbf{S}}'^{\mathsf{T}})$.
2. Set up the system
$$\mathbf{H}\big(\mathrm{ext}^{-1}(\mathbf{A})\mathbf{B}\big)^{\mathsf{T}} = \boldsymbol{t}^{\mathsf{T}}$$
and expand over $\mathbb{F}_2$. Since $\mathbf{A}^{\mathsf{T}}$ is systematic and $\mathrm{ext}(\boldsymbol{e}) = \mathbf{AB}$, the first r rows of $\mathrm{ext}(\boldsymbol{e})$ are exactly the first r rows of $\mathbf{B}$. Further, for $1 \le i \le r$, we have $\mathrm{ext}(\boldsymbol{e})_{i,j} = \widehat{C}_{i,j}$. Hence the h_c $\mathbb{F}_2$-equations from Step 0 become equations of the form "bit k' of $B_{i,j}$ equals bit k' of $\widehat{C}_{i,j}$". Substitute these known bits into the above system and solve the resulting system for $\mathbf{B}$, and then set $\boldsymbol{e} := \mathrm{ext}^{-1}(\mathbf{A})\mathbf{B}$.
3. If $\mathrm{rank}(\boldsymbol{e}) \neq r$, go to Step 1. Otherwise, return $\boldsymbol{e}$.

Proposition 6 (Complexity of Column GRS with $\mathbb{F}_q$-Hints (Algorithm 5))**.** *Set $\ell := m - r - \lceil (km - h_c)/n \rceil$. As noted in footnotes 5 and 6 to Proposition 4, we restrict to $r \le \min\{1, m/n\}(n-k)$, which implies $0 \le \ell$, and assume $\ell \le m - r$ (otherwise the RSD problem is trivial).*

Assume the (reasonable) heuristic that the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 5 is upper-bounded[9] *by*

$$P\, q^{r\lceil (km-h_c)/n \rceil - h_s + t_s} \quad \textit{(up to a constant factor)},$$

[9]The same caveat regarding multiple solutions as explained in footnote 7 applies.

where h_s is the total number of known entries in $\widehat{\mathbf{S}}'$ and t_s is the number of known entries contained in any ℓ rows of $\widehat{\mathbf{S}}'$ with the fewest known entries. An upper bound on the polynomial cost is

$$P = \max\left\{ \begin{array}{l} m(n-k)^2 \ \mathbb{F}_{q^m}\text{-operations}, \\ (m(n-k))^\omega \ \mathbb{F}_q\text{-operations} \end{array} \right\} \quad \textit{(up to a constant factor).}$$

The highlighted term generally dominates, e.g., if we assume one $\mathbb{F}_{q^m}$-operation costs $m\mathbb{F}_q$-operations. Setting $h_s = h_c = 0$ recovers the original column-GRS complexity; see Remark 3 above.

Proof. Exponential cost. The unknown matrix $\mathbf{B} \in \mathbb{F}_q^{(r+\ell)\times n}$ has $n(r+\ell)$ entries. The $(n-k)$ syndrome equations over $\mathbb{F}_{q^m}$ give $m(n-k)$ equations over $\mathbb{F}_q$, while the hints from $\widehat{\mathbf{C}}$ give h_c additional equations over $\mathbb{F}_q$. Thus, $\ell = m - r - \lceil (km - h_c)/n \rceil$ is the largest integer for which $n(r+\ell) \le m(n-k) + h_c$. By Theorem 1, applied to $(\mathbf{I}_r \mid \widehat{\mathbf{S}}'^{\mathsf{T}})$, Step 1 succeeds after

$$q^{r(m-r-\ell)-h_s+t_s} = q^{r\lceil (km-h_c)/n \rceil - h_s + t_s}$$

iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor.

1. Step 1 only constructs the systematic matrix $\mathbf{A}^{\mathsf{T}} \in \mathbb{F}_q^{(r+\ell)\times m}$ according to Theorem 1, and as before this cost is negligible.

2. Step 2 consists of the following substeps. The syndrome system can be written as

$$\mathbf{H}\mathbf{B}^{\mathsf{T}} \operatorname{ext}^{-1}(\mathbf{A})^{\mathsf{T}} = \sum_{i=1}^{r+\ell} \sum_{j=1}^{n} B_{i,j} \big(\operatorname{ext}^{-1}(\mathbf{A})\big)_i \mathbf{H}_{*,j} = \boldsymbol{t}^{\mathsf{T}}$$

   where the unknowns are the $B_{i,j}$'s.

   Before proceeding, we show that $n(r+\ell)$ equals $m(n-k)$ up to a constant factor. Using $r + \ell = m - \lceil (km - h_c)/n \rceil$, we have

$$n(r+\ell) = n\left\lfloor \frac{m(n-k) + h_c}{n} \right\rfloor.$$

   Note $n(r+\ell)$ is less than $m(n-k)+h_c$ by at most $n-1$, so that $n(r+\ell) \le m(n-k)+h_c < n(r+\ell)+n \le 2n(r+\ell)$. Hence $n(r+\ell)$ equals $m(n-k)+h_c$ up to a constant factor. Further, $0 \le h_c \le rn \le m(n-k)$, so $m(n-k)+h_c$ equals $m(n-k)$ up to a constant factor and so $n(r+\ell)$ equals $m(n-k)$ up to a constant factor.

   (a) For every $i \in [r+\ell]$ and $j \in [n]$, compute the coefficient column

$$\big(\operatorname{ext}^{-1}(\mathbf{A})\big)_i \mathbf{H}_{*,j} \in \mathbb{F}_{q^m}^{n-k}. \qquad \text{(Note } (\operatorname{ext}^{-1}(\mathbf{A}))_i \text{ is a scalar.)}$$

   This requires $n(n-k)(r+\ell)$ $\mathbb{F}_{q^m}$-operations. By the relation above, this equals $m(n-k)^2$ $\mathbb{F}_{q^m}$-operations up to a constant factor. Once the coefficient columns are represented in the fixed $\mathbb{F}_q$-basis of $\mathbb{F}_{q^m}$, placing their coordinates in the system matrix requires no additional field operations.

   (b) We now have $m(n-k)$ $\mathbb{F}_q$-equations, h_c additional $\mathbb{F}_q$-equations from Step 0, and $n(r+\ell)$ unknowns. Each equation from Step 0 fixes one of these unknowns to a known value. We substitute these h_c known values into the $m(n-k)$ syndrome equations, which costs $h_c m(n-k)$ $\mathbb{F}_q$-operations and leaves $n(r+\ell) - h_c$ unknowns, whose system can be solved in $(n(r+\ell)-h_c)^\omega$ $\mathbb{F}_q$-operations. Since $h_c \le rn \le m(n-k)$ and $n(r+\ell) - h_c \le m(n-k)$, both costs are upper-bounded by $(m(n-k))^\omega$ $\mathbb{F}_q$-operations. This is a possibly loose upper bound on the cost of this substep, which we use to keep the expression for the polynomial cost simple.

3. In Step 3, since $\mathbf{A}$ has full column rank, $\operatorname{rank}(\boldsymbol{e}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{e})) = \operatorname{rank}(\mathbf{AB}) = \operatorname{rank}(\mathbf{B})$. Thus, it suffices to compute the rank of the $(r+\ell) \times n$ matrix $\mathbf{B}$, which costs $n(r+\ell)\min\{n, r+\ell\}^{\omega-2}$ $\mathbb{F}_q$-operations. By the relation above, this cost is upper-bounded by $(m(n-k))^{\omega}$ $\mathbb{F}_q$-operations up to a constant factor.

The polynomial cost is upper-bounded by the maximum of the displayed costs up to a constant factor. □

Proposition 7 (Complexity of Column GRS with Bit-Level Hints (Algorithm 6))**.** *Suppose $q = 2^{\nu}$. Set $\ell := m - r - \lceil (km - (h_c/\nu))/n \rceil$. As in Proposition 6, we restrict to $r \leq \min\{1, m/n\}(n-k)$, which implies $0 \leq \ell$, and assume $\ell \leq m - r$ (otherwise the RSD problem is trivial).*

Assume the (reasonable) heuristic that the system obtained in Step 2 is full rank with probability bounded below by a positive constant. The expected cost of Algorithm 6 is upper-bounded by

$$P\ 2^{\nu r\lceil (km-(h_c/\nu))/n\rceil - h_s + t_s} \quad \textit{(up to a constant factor)},$$

where h_s is the total number of known bits in $\widehat{\mathbf{S}}'$ and t_s is the number of known bits contained in any ℓ rows of $\widehat{\mathbf{S}}'$ with the fewest known bits. An upper bound on the polynomial cost is

$$P = \max\left\{ \begin{array}{l} m(n-k)^2\ \mathbb{F}_{2^{\nu m}}\textit{-operations}, \\ (\nu m(n-k))^{\omega}\ \mathbb{F}_2\textit{-operations}, \\ m(n-k)\min\left\{n, \frac{m(n-k)}{n}\right\}^{\omega-2}\ \mathbb{F}_{2^{\nu}}\textit{-operations} \end{array} \right\} \quad \textit{(up to a constant factor)}.$$

The highlighted term generally dominates, e.g., if we assume one $\mathbb{F}_{2^{\nu m}}$-operation costs νm $\mathbb{F}_2$-operations and one $\mathbb{F}_{2^{\nu}}$-operation costs ν $\mathbb{F}_2$-operations.

If $h_c = 0$, then Step 2 need not be expanded from $\mathbb{F}_{2^{\nu}}$ to $\mathbb{F}_2$. In this case the polynomial cost equals P_0 up to a constant factor, where

$$P_0 = \max\left\{ \begin{array}{l} m(n-k)^2\ \mathbb{F}_{2^{\nu m}}\textit{-operations}, \\ (m(n-k))^{\omega}\ \mathbb{F}_{2^{\nu}}\textit{-operations} \end{array} \right\},$$

where now $\ell = m - r - \lceil km/n \rceil$. The highlighted term generally dominates, e.g., if we assume one $\mathbb{F}_{2^{\nu m}}$-operation costs m $\mathbb{F}_{2^{\nu}}$-operations.

Proof. Exponential cost. The unknown matrix $\mathbf{B} \in \mathbb{F}_{2^{\nu}}^{(r+\ell)\times n}$ contains $\nu n(r+\ell)$ binary unknowns. After expansion over $\mathbb{F}_2$, the $(n-k)$ syndrome equations over $\mathbb{F}_{2^{\nu m}}$ give $\nu m(n-k)$ binary equations, while the hints from $\widehat{\mathbf{C}}$ give h_c additional binary equations. Thus, $\ell = m - r - \lceil (km - (h_c/\nu))/n \rceil$ is the largest integer for which $\nu n(r+\ell) \leq \nu m(n-k) + h_c$. By Corollary 1, applied to $(\mathbf{I}_r \mid \widehat{\mathbf{S}}'^{\mathsf{T}})$, Step 1 succeeds after

$$2^{\nu r(m-r-\ell) - h_s + t_s} = 2^{\nu r\lceil (km-(h_c/\nu))/n\rceil - h_s + t_s}$$

iterations in expectation.

Polynomial cost. All costs below are stated up to a constant factor.

1. Step 1 only constructs the systematic matrix $\mathbf{A}^{\mathsf{T}} \in \mathbb{F}_{2^{\nu}}^{(r+\ell)\times m}$ according to Corollary 1, and as before this cost is negligible.
2. Step 2 consists of the following substeps. As in the proof of Proposition 6,

$$n(r+\ell) = n\left\lfloor \frac{m(n-k) + (h_c/\nu)}{n} \right\rfloor$$

The left-hand side equals $m(n-k)$ up to a constant factor, where we used $0 \leq h_c \leq \nu r n \leq \nu m(n-k)$.

(a) As in Step a of the proof of Proposition 6, compute every coefficient column

$$\left(\operatorname{ext}^{-1}(\mathbf{A})\right)_i \mathbf{H}_{*,j} \quad \text{for } i \in [r+\ell],\ j \in [n].$$

This costs $n(n-k)(r+\ell)$ $\mathbb{F}_{2^{\nu m}}$-operations, which equals $m(n-k)^2$ $\mathbb{F}_{2^{\nu m}}$-operations up to a constant factor.

(b) The resulting system over $\mathbb{F}_{2^\nu}$ has $m(n-k)$ equations and $n(r+\ell)$ unknowns. By the same expansion argument as in Step 2(b) of the proof of Proposition 3, expanding it over $\mathbb{F}_2$ costs $\nu^2 mn(n-k)(r+\ell)$ $\mathbb{F}_2$-operations. Since $n(r+\ell)$ equals $m(n-k)$ up to a constant factor, this equals $(\nu m(n-k))^2$ $\mathbb{F}_2$-operations up to a constant factor and is upper-bounded by $(\nu m(n-k))^\omega$ $\mathbb{F}_2$-operations.

(c) We now have $\nu m(n-k)$ binary syndrome equations and $\nu n(r+\ell)$ binary unknowns. Each of the h_c equations from Step 0 fixes one of these unknowns to a known value. Substituting these values costs $h_c \nu m(n-k)$ $\mathbb{F}_2$-operations and leaves $\nu n(r+\ell) - h_c \leq \nu m(n-k)$ unknowns. Since $h_c \leq \nu rn \leq \nu m(n-k)$, the cost of substituting and solving the resulting system is upper-bounded by $(\nu m(n-k))^\omega$ $\mathbb{F}_2$-operations. This is a possibly loose upper bound on the cost of this substep, used to keep the polynomial-cost expression simple.

If $h_c = 0$, Steps b and c over $\mathbb{F}_2$ are unnecessary. The system can instead be expanded and solved over $\mathbb{F}_{2^\nu}$. Since $n(r+\ell)$ equals $m(n-k)$ up to a constant factor, solving costs $(m(n-k))^\omega$ $\mathbb{F}_{2^\nu}$-operations up to a constant factor.

3. In Step 3, $\mathbf{A}$ has full column rank, and hence $\operatorname{rank}(\boldsymbol{e}) = \operatorname{rank}(\operatorname{ext}(\boldsymbol{e})) = \operatorname{rank}(\mathbf{AB}) = \operatorname{rank}(\mathbf{B})$. Computing the rank of $\mathbf{B} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n}$ costs $n(r+\ell)\min\{n, r+\ell\}^{\omega-2}$ $\mathbb{F}_{2^\nu}$-operations. By the relation above, this equals

$$m(n-k)\min\left\{n, \frac{m(n-k)}{n}\right\}^{\omega-2}$$

$\mathbb{F}_{2^\nu}$-operations up to a constant factor. If $h_c = 0$, this is upper-bounded by $(m(n-k))^\omega$ $\mathbb{F}_{2^\nu}$-operations.

The polynomial cost is upper-bounded by the maximum of the displayed costs up to a constant factor. □

B Concrete Costs

Table 2 in Appendix B shows the complexity in bits of modified kernel search with hints (for Mirath) and row-space GRS with hints (for RYDE) for various fractions of known hints in $\mathbf{C}'$. For completeness, we also report the minimum fraction of hints needed for the attack to become polynomial-time. This threshold is less interesting however, since it is only reached once almost all of $\mathbf{C}'$ is revealed ($\geq 94\%$).[10] The Table also provides a link to our source code.

[10] In an earlier version of this preprint, the table had incorrect values. These are now fixed.

Table 2: Concrete Complexity (Exponential Part) of Kernel Search and GRS with Hints

Parameter Set	Fraction of hints known in $\mathbf{C}'$											Minimum Fraction
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	
`Mirath-1a`	144	124	104	89	71	54	39	24	14	3	0	0.94
`Mirath-3a`	220	192	164	137	112	88	65	44	24	7	0	0.96
`Mirath-5a`	288	248	212	176	145	113	81	55	23	10	0	0.96
`Mirath-1b`	140	125	110	94	79	65	51	37	23	10	0	0.97
`Mirath-3b`	205	183	160	137	116	95	74	53	34	15	0	0.98
`Mirath-5b`	270	240	210	180	152	124	97	70	44	19	0	0.98
`Mirath-1a (bit)`	144	126	110	93	78	64	50	36	22	9	0	0.98
`Mirath-3a (bit)`	220	193	169	145	122	100	78	57	36	16	0	0.99
`Mirath-5a (bit)`	288	253	220	189	159	130	102	73	46	20	0	0.99
`RYDE-1`	180	160	141	121	102	83	64	47	29	13	0	0.97
`RYDE-3`	255	227	199	171	144	118	92	67	42	18	0	0.98
`RYDE-5`	308	273	238	204	172	141	109	79	50	21	0	0.98

1. The table reports the exponential part of the complexity, averaged over 2000 trials with hints uniformly placed in the matrix $\mathbf{C}$ for the given hint fraction. The complexities are given in bits, i.e., as base-2 logarithms.

   The Mirath parameter sets use the complexity expression for kernel search with hints (under "Complexity" in Section 3). The RYDE parameter sets use the complexity expression for row-space GRS with hints (under "Complexity" in Section 4). We only use the exponential part of the complexity expression, i.e., the polynomial factor is ignored. We assume the guessing is performed on the right, i.e., we guess the row space of $\mathbf{C}$.

   The columns labeled by the fraction of hints refer to the fraction of known $\mathbb{F}_q$-entries in $\mathbf{C}'$, and we assume no hints in $\mathbf{S}$ are known. The "Minimum Fraction" value is the minimum fraction of hints in $\mathbf{C}'$ required for the complexity to become polynomial-time. If "(bit)" is written after a parameter set, then the (exponential part of the) bit-level complexity formula is used, and the hint fraction instead refers to the fraction of known *bits* in $\mathbf{C}'$.

2. The code can be found at: `https://github.com/mnap/subspace-guessing-with-hints`

3. *Explanation by example:* In row `Mirath-1a` and column `0.2`, the value 104 means the following. We assume that 20% of the entries of $\mathbf{C}'$ are known. For this parameter set, $\mathbf{C}'$ has size 4×12, hence 48 entries, so this corresponds to about $0.2 \cdot 48 \approx 10$ known entries. Their positions are chosen uniformly at random in $\mathbf{C}'$, and we assume $h_s = 0$.

   For each such random placement, we compute the exponential part of the kernel-search-with-hints complexity expression. This is repeated for 2000 trials, and the entry shown in the table is the base-2 logarithm of the resulting averaged cost.

   By default, this averaged cost is equal to the reciprocal of the mean success probability. (More details are in the code repository.) Using direct averaging of the trial costs instead increases the reported values only slightly, typically by about 0–3 bits.

   The code also allows the hints to be distributed as evenly as possible across the columns of $\mathbf{C}'$, rather than uniformly at random. This is the worst-case placement for our with-hints algorithms and therefore yields larger costs. In this setting, the averaging strategy does not matter because all trial costs are the same. We note that, even in this worst-case setting, the resulting "Minimum Fraction" values turn out to be same as in the above table.

4. The parameter sets are as follows:

Parameter Set	q	m	n	Dimension	r
`Mirath-1a`	16	16	16	$K = 143$	4
`Mirath-3a`	16	19	19	$K = 195$	5
`Mirath-5a`	16	22	22	$K = 255$	6
`Mirath-1b`	2	42	42	$K = 1443$	4
`Mirath-3b`	2	50	50	$K = 2024$	5
`Mirath-5b`	2	56	56	$K = 2499$	6
`RYDE-1`	2	53	53	$k = 45$	4
`RYDE-3`	2	61	61	$k = 51$	5
`RYDE-5`	2	61	57	$k = 44$	7

5. RYDE specializes the first entry of $\boldsymbol{s}$ to 1, but we do not exploit this here when computing the costs. Incorporating this would amount to increasing h_s by m, since it reveals the first column of $\mathbf{S} \in \mathbb{F}_q^{m \times r}$.

C Bit-Level Hints for Subspace Guessing

Corollary 2 (Subspace Guessing with Bit-Level Hints, Fully Explicit Restatement of Corollary 1)**.** *Let* $\mathbf{C} := (\mathbf{I}_r \mid \mathbf{C}') \in \mathbb{F}_{2^\nu}^{r\times n}$ *where only some bits of some entries of the secret matrix* $\mathbf{C}' \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$ *are known, and let* $0 \le \ell \le n-r$*.*

(1) Let $\mathcal{C}$ *be the set of all possible matrices* $(\mathbf{I}_r \mid \mathbf{C}'') \in \mathbb{F}_{2^\nu}^{r\times n}$ *such that* $\mathbf{C}'' \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$ *agrees with* $\mathbf{C}'$ *at the known bits. Then there exists an* $(r+\ell)$*-dimensional subspace* $\mathcal{R} \subseteq \mathbb{F}_{2^\nu}^n$ *such that*

$$\left|\{\ (\mathbf{I}_r \mid \mathbf{C}'') \in \mathcal{C} : \mathrm{rowspace}(\mathbf{I}_r \mid \mathbf{C}'') \subseteq \mathcal{R}\ \}\right| = 2^{\nu r\ell - t_c}$$

and for no other $(r+\ell)$*-dimensional* $\mathcal{R}$ *is this number larger. Here*

$$t_c := \min_{\substack{S\subseteq\{1,\dots,n-r\}\\ |S|=\ell}} \sum_{i\in S} h_i, \tag{3}$$

where h_i *is the number of known bits in column* i *of* $\mathbf{C}'$*. In other words,* t_c *is the sum of the* ℓ *smallest numbers in the list* $h_1, \dots, h_{n-r}$*.*

(2) Suppose an oracle $\mathcal{O}$ *takes as input an* $(r+\ell)$*-dimensional space* $\mathcal{R} \subseteq \mathbb{F}_{2^\nu}^n$ *and outputs success if and only if* $\mathcal{R}$ *contains the row space of* $(\mathbf{I}_r \mid \mathbf{C}')$*.*

Define the following algorithm `GuessSubspace` *with oracle access to* $\mathcal{O}$ *and which takes the known bits of* $\mathbf{C}'$ *as input (i.e., both bit positions and bit values).*

(a) In each iteration, sample $\mathcal{R}$ *independently and identically distributed as follows. Set* $\mathcal{R} :=$ $\mathrm{rowspace}(\mathbf{R})$ *with* $\mathbf{R} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n}$ *of the form*

$$\mathbf{R} = \begin{pmatrix} \mathbf{I}_r & \mathbf{B} \\ \mathbf{0}_{\ell\times r} & \mathbf{A} \end{pmatrix} \in \mathbb{F}_{2^\nu}^{(r+\ell)\times n},$$

where:

- $\mathbf{B} \in \mathbb{F}_{2^\nu}^{r\times(n-r)}$ *agrees with* $\mathbf{C}'$ *at all known bits and the other bits are sampled uniformly at random, and*
- $\mathbf{A} \in \mathbb{F}_{2^\nu}^{\ell\times(n-r)}$ *is a fixed matrix such that* $\mathbf{A}_{*,\mathcal{I}} = \mathbf{I}_\ell$ *and* $\mathbf{A}_{*,\bar{\mathcal{I}}} = \mathbf{0}_{\ell\times(n-r-\ell)}$ *for some (fixed) set* $\mathcal{I} \subseteq [n-r]$ *of* ℓ *columns of* $\mathbf{C}'$ *containing the fewest known bits.*

(b) If $\mathcal{O}(\mathcal{R})$ *returns success, output* $\mathcal{R}$*, else go to Step 1.*

If $\mathbf{C}'$ *is drawn uniformly at random, then the above algorithm finds such an* $\mathcal{R}$ *in expected number of oracle queries*

$$2^{\nu r(n-r-\ell)-h_c+t_c}$$

where h_c *is the total number of known bits in* $\mathbf{C}'$ *and* t_c *is the number of known bits within any* ℓ *columns of* $\mathbf{C}'$ *that have the fewest known bits (i.e., as defined in* (3)*). Further, this procedure is optimal up to a constant factor: no other algorithm with oracle access to* $\mathcal{O}$ *can find such an* $\mathcal{R}$ *in fewer expected oracle queries up to a constant factor.*

Proof. If $\boldsymbol{x} \in \mathbb{F}_{2^\nu}^{n-r}$, then define $\mathrm{bitstring}(\boldsymbol{x}) \in \mathbb{F}_2^{\nu(n-r)}$ to be the binary vector obtained by expanding each $\mathbb{F}_{2^\nu}$-entry of $\boldsymbol{x}$ into a ν-length binary vector by using a fixed basis of $\mathbb{F}_{2^\nu}$ over $\mathbb{F}_2$.

The proof of Proposition 1 still applies after replacing known entries by known bits. The number of $\mathbf{R}$-compatible matrices equals

$$\prod_{i=1}^{r} \left|\{\boldsymbol{x} \in \mathrm{rowspace}(\mathbf{A}) \subseteq \mathbb{F}_{2^\nu}^{n-r} \mid \mathrm{bitstring}(\boldsymbol{x})_{\mathcal{H}_i} = \mathbf{0}\}\right|$$

where $\mathcal{H}_i \subseteq [\nu(n-r)]$ denotes the known bit positions in row i and $\text{bitstring}(\boldsymbol{x})_{\mathcal{H}_i} \in \mathbb{F}_2^{|\mathcal{H}_i|}$ denotes the restriction of $\text{bitstring}(\boldsymbol{x}) \in \mathbb{F}_2^{\nu(n-r)}$ to the positions in $\mathcal{H}_i$.

We can again assume $\mathbf{A} \in \mathbb{F}_{2^\nu}^{\ell \times (n-r)}$ is zero outside a set $\mathcal{I} \subseteq [n-r]$ of ℓ column indices and equals $\mathbf{I}_\ell$ on $\mathcal{I}$. Then

$$\begin{aligned}
&|\{\boldsymbol{x} \in \text{rowspace}(\mathbf{A}) \subseteq \mathbb{F}_{2^\nu}^{n-r} \mid \text{bitstring}(\boldsymbol{x})_{\mathcal{H}_i} = \mathbf{0}\}| \\
&= |\{\boldsymbol{x} \in \text{rowspace}(\mathbf{A}_{*,\mathcal{I}}) \subseteq \mathbb{F}_{2^\nu}^{\ell} \mid \text{bitstring}(\boldsymbol{x})_{\mathcal{H}_i \triangleright \mathcal{I}} = \mathbf{0}\}| \\
&= |\{\boldsymbol{x} \in \mathbb{F}_{2^\nu}^{\ell} \mid \text{bitstring}(\boldsymbol{x})_{\mathcal{H}_i \triangleright \mathcal{I}} = \mathbf{0}\}| \\
&= 2^{\nu\ell - |\mathcal{H}_i \triangleright \mathcal{I}|},
\end{aligned}$$

where $\mathcal{H}_i \triangleright \mathcal{I} \subseteq [\nu\ell]$ denotes the induced set of bit positions obtained by restricting $\mathcal{H}_i$ to the columns indexed by $\mathcal{I}$ and then reindexing those columns as $1, \dots, \ell$. More formally, if $\mathcal{I} = \{i_1 < \cdots < i_\ell\}$, then

$$\mathcal{H}_i \triangleright \mathcal{I} := \{\nu(a-1) + b \mid \ a \in [\ell],\ b \in [\nu],\ \nu(i_a - 1) + b \in \mathcal{H}_i\} \subseteq [\nu\ell].$$

Thus the product equals $2^{\nu r\ell - \sum_{i=1}^{r} |\mathcal{H}_i \triangleright \mathcal{I}|}$ and is maximum when $\mathcal{I}$ consists of any ℓ columns with the fewest known bits. This proves Item (1).

For Item (2), the proof of Theorem 1 now carries over verbatim with known entries replaced by known bits, with q replaced by 2^ν, and with the above counting result showing that each oracle query can check at most $2^{\nu r\ell - t_c}$ compatible matrices. Therefore the expected number of oracle queries is $2^{\nu r(n-r-\ell) - h_c + t_c}$, and this is optimal up to a constant factor. □