Blockwise Rank Decoding Problem and LRPC Codes: Cryptosystems with Smaller Sizes

Yongcheng Song[1], Jiang Zhang[1](✉), Xinyi Huang[2], and Wei Wu[3]

[1] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing, 100878, China
yongchengsong@outlook.com, jiangzhang09@gmail.com,
[2] Artificial Intelligence Thrust, Information Hub,
Hong Kong University of Science and Technology (Guangzhou), Guangzhou, 511455, China
xinyi@ust.hk
[3] School of Mathematics and Statistics, Fujian Normal University, Fuzhou, 350117, China
weiwu@fjnu.edu.cn

Abstract. In this paper, we initiate the study of the Rank Decoding (RD) problem and LRPC codes with blockwise structure in rank-based cryptosystems. First, we introduce the blockwise errors (ℓ-errors) where each error consists of ℓ blocks of coordinates with disjoint supports, and define the blockwise RD (ℓ-RD) problem as a natural generalization of the RD problem whose solutions are ℓ-errors (note that the standard RD problem is actually a special ℓ-RD problem with $\ell = 1$). We adapt the typical attacks on the RD problem to the ℓ-RD problem, and find that the blockwise structure does not ease the problem too much: the ℓ-RD problem is still exponentially hard for appropriate choices of $\ell > 1$. Second, we introduce blockwise LRPC (ℓ-LRPC) codes as generalizations of the standard LPRC codes whose parity-check matrices can be divided into ℓ sub-matrices with disjoint supports, i.e., the intersection of two subspaces generated by the entries of any two sub-matrices is a null space, and investigate the decoding algorithms for ℓ-errors. We find that the gain of using ℓ-errors in decoding capacity outweighs the complexity loss in solving the ℓ-RD problem, which makes it possible to design more efficient rank-based cryptosystems with flexible choices of parameters.

As an application, we show that the two rank-based cryptosystems submitted to the NIST PQC competition, namely, RQC and ROLLO, can be greatly improved by using the ideal variants of the ℓ-RD problem and ℓ-LRPC codes. Concretely, for 128-bit security, our RQC has total public key and ciphertext sizes of 2.5 KB, which is not only about 50% more compact than the original RQC, but also smaller than the NIST Round 4 code-based submissions HQC, BIKE, and Classic McEliece.

Keywords: Post-Quantum Cryptography, NIST PQC Candidates, Rank Metric Code-Based Cryptography, Rank Decoding Problem, LRPC Codes

1 Introduction

Since traditional cryptographic schemes based on number theoretic assumptions are at risk from the possible attacks using quantum computers, the design of post-quantum cryptosystems, such as code-based cryptosystems, has become the consensus of industry and academia. Last year, three code-based cryptosystems using the Hamming metric codes, namely, BIKE, Classic McEliece and HQC had been selected to the fourth round of NIST post-quantum standardization process for future standardization [46]. As a nice alternative to Hamming metric code-based cryptography, code-based cryptography using the rank metric, namely, rank-based cryptography, is typically more efficient in computational efficiency and bandwidth, and deserves further research as encouraged by NIST [45].

$\mathbb{F}_{q^m}$-Linear Codes with Rank Metric and Rank Decoding Problem. Codes used in rank-based cryptography are $\mathbb{F}_{q^m}$-linear codes with rank metric over a degree m extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{F}_{q^m}^m$ be a basis of $\mathbb{F}_{q^m}$ viewed as an m-dimensional vector space over $\mathbb{F}_q$. Then, any $\boldsymbol{e} = (e_1, e_1, \ldots, e_n) \in \mathbb{F}_{q^m}^n$ has an associated matrix $\mathrm{Mat}(\boldsymbol{e}) \in \mathbb{F}_q^{m \times n}$ such that $\boldsymbol{e} = \boldsymbol{\alpha}\mathrm{Mat}(\boldsymbol{e})$. *The rank weight* $\|\boldsymbol{e}\|_{\mathrm{R}}$ of $\boldsymbol{e}$ is defined as the rank of $\mathrm{Mat}(\boldsymbol{e})$. *The support* $\mathrm{Supp}(\boldsymbol{e})$ of $\boldsymbol{e}$ is the $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ spanned by the coordinates of $\boldsymbol{e}$. It follows from definition that $\|\boldsymbol{e}\|_{\mathrm{R}}$ equals to the dimension of $\mathrm{Supp}(\boldsymbol{e})$. The set of errors of length n and weight r is denoted by $\mathcal{S}_r^n$. An $\mathbb{F}_{q^m}$-linear code ($[n,k]_{q^m}$) with rank metric of length n and dimension k is a dimension k subspace of $\mathbb{F}_{q^m}^n$, which can be represented by a generator matrix of size $k \times n$ or a parity-check matrix of size $(n-k) \times n$ over $\mathbb{F}_{q^m}$.

Let $\boldsymbol{G}$ be a generator matrix of random $[n,k]_{q^m}$-linear codes, $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$, and $r \in \mathbb{N}$. The Rank Decoding (RD) problem is to find $\boldsymbol{x} \in \mathbb{F}_{q^m}^k$ and $\boldsymbol{e} \in \mathcal{S}_r^n$ such that $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$. Although the RD problem is not shown to be NP-hard, it is very close to the Hamming metric decoding problem which is NP-hard [31], and can be seen as a structured version of the MinRank problem which is also NP-hard [25]. Moreover, after more than three decades of study, the best known algorithms for solving the RD problem are all exponential. This makes the RD problem a promising hard problem to construct secure cryptosystems.

Rank-Based Cryptography. The first rank-based cryptosystem, known as the GPT cryptosystem [27], was based on Gabidulin codes [26] which have analogous structures to Reed-Solomon codes. The GPT cryptosystem and its early variants were broken by Overbeck attack [49], in the much same way as McEliece schemes based on Reed-Solomon codes were attacked in [21,50]. The recent variant [39] was analyzed with some insecure parameters region being found in [20,32]. As these attacks [49,21,50,20,32] mainly expose the security flaws of the GPT cryptosystem by exploiting the strong algebraic structure of Gabidulin codes, it is still possible to construct secure and efficient rank-based cryptosystems.

A very significant step was using the Low Rank Parity Check (LRPC) codes [28,5] and the Gabidulin codes to build cryptosystems [28,30,41,40,3], which can be viewed as rank metric analogues of the MDPC cryptosystem [44], NTRU [35], or Alekhnovich [1]. Four cryptosystems of this kind, namely, RQC [41], Lake,

Locker [40], and Ouroboros-R [3] were submitted to the NIST PQC standardization process in 2017, with the latter three being merged into ROLLO in the second round. The combinatorial attacks [48,29,6] were once considered to be the most efficient attacks against the parameters region of RQC and ROLLO. However, it turned out later that the improved dedicated algebraic attacks [8,10] could greatly reduce the concrete security of RQC and ROLLO. This is the main reason that RQC and ROLLO were not selected to the third round of the NIST PQC standardization process. New parameter sets [41,40,3] for RQC and ROLLO were proposed to provide adequate security against algebraic attacks. As the new key and ciphertext sizes of RQC and ROLLO remain competitive, NIST encourages further research on rank-based cryptography [45].

1.1 Our Contribution

We initiate the study of the RD problem and LRPC codes with blockwise structures to design secure and efficient rank-based cryptosystems. First, we introduce the blockwise errors (ℓ-errors) where each error consists of ℓ blocks of coordinates with disjoint supports, and define the blockwise RD (ℓ-RD) problem as a natural generalization of the RD problem whose solutions are ℓ-errors. Notably, the standard RD problem can be seen as a special ℓ-RD problem with $\ell = 1$, or equivalently the ℓ-RD problem can be treated as a structured RD problem. Since the attacks may benefit from the blockwise structure, the ℓ-RD problem is inherently not harder than the standard one. Fortunately, this structure does not ease the problem too much: we only observe a reduction about ℓ times in the exponent to solve the ℓ-RD problem by carefully examining the best known attacks for the standard RD problem, implying that the ℓ-RD problem is still exponentially hard for appropriate choices of constant $\ell > 1$.

Second, we introduce the blockwise LRPC (ℓ-LRPC) codes as generalizations of the standard LPRC codes whose parity-check matrices can be divided into ℓ sub-matrices with disjoint supports, i.e., the intersection of two subspaces generated by the entries of any two sub-matrices is a null space, and investigate the decoding algorithms for ℓ-errors. We find that the decoding algorithm can also benefit from the blockwise structures: the decoding capacity can be significantly improved by a factor of ℓ. In particular, a suitably defined $[n, k]_{q^m}$ ℓ-LRPC code can actually decode an ℓ-error with weight up to $(n - k)/2$, which achieves the decoding capacity of rank codes of optimal distance. This makes it possible to design more efficient rank-based cryptosystems with flexible choices of parameters, by making a tradeoff between the hardness of the ℓ-RD problem and the decoding capacity of the ℓ-LPRC codes.

Finally, we show that the two rank-based cryptosystems submitted to the NIST PQC competition, namely, RQC and ROLLO, can be greatly improved by using the ideal variants of the ℓ-RD problem and ℓ-LRPC codes. Concretely, for 128-bit security, our RQC has total public key and ciphertext sizes of 2.5 KB, which is not only about 50% more compact than the original RQC, but also smaller than the NIST Round 4 code-based submissions HQC, BIKE, and Classic McEliece. A detailed comparison with related works is given in Subsection 1.2.

1.2 Technical Overview

Recall that the set of errors of length n and weight r is denoted by $\mathcal{S}_r^n$. By definition, all n coordinates of an error $\boldsymbol{e} \in \mathcal{S}_r^n$ belong to the same support of dimension r. In particular, let $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_r) \in \mathbb{F}_{q^m}^r$ be a basis of the support $\mathrm{Supp}(\boldsymbol{e})$, then there is an $r \times n$ coefficient matrix $\boldsymbol{C}$ such that $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C}$.

The Blockwise Errors (ℓ-errors). Let $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ be vectors of positive integers. We say that an error $\boldsymbol{e} \in \mathcal{S}_r^n$ with $n = \sum_{i=1}^{\ell} n_i$ and $r = \sum_{i=1}^{\ell} r_i$ is an ℓ-error with parameters $\boldsymbol{n}$ and $\boldsymbol{r}$ if it can be divided into ℓ sub-vectors $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell)$ such that 1) the sub-vector $\boldsymbol{e}_i \in \mathbb{F}_{q^m}^{n_i}$ has weight r_i for all $i \in \{1..\ell\}$; and 2) the supports of these sub-vectors are mutually disjoint, namely, $\mathrm{Supp}(\boldsymbol{e}_i) \cap \mathrm{Supp}(\boldsymbol{e}_j) = \{0\}$ for all $i \neq j$. Denote $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ as the set of blockwise errors with parameters $\boldsymbol{n}$ and $\boldsymbol{r}$. By definition, the set $\mathcal{S}_r^n$ is exactly the set $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ of ℓ-errors with $\ell = 1$. For $\ell > 1$, $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ is a proper subset of $\mathcal{S}_r^n$. In particular, for any $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$, if we let $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{ir_i}) \in \mathbb{F}_{q^m}^{r_i}$ be a basis of $\mathrm{Supp}(\boldsymbol{e}_i)$, then the coefficient matrix $\boldsymbol{C}$ of $\boldsymbol{e}$ w.r.t. the basis $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_\ell)$, i.e., $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C}$, has a special block-diagonal form:

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_\ell \end{pmatrix} \in \mathbb{F}_q^{r \times n} \tag{1}$$

where $\boldsymbol{e}_i = \varepsilon_i \boldsymbol{C}_i$. As we will show later, the attacks can benefit from the block-diagonal structure.

The Blockwise RD (ℓ-RD) Problem. We define the ℓ-RD problem as a natural generalization of the RD problem whose solutions are ℓ-errors. Recall that the RD problem asks an algorithm given as inputs a generator matrix $\boldsymbol{G}$ of random $[n,k]_{q^m}$-linear code $\mathcal{C}$, a vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$, and an integer $r \in \mathbb{N}$, outputs $\boldsymbol{x} \in \mathbb{F}_{q^m}^k$ and $\boldsymbol{e} \in \mathcal{S}_r^n$ such that $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$. The RD problem can be solved by finding a codeword $\boldsymbol{e} \in \mathcal{S}_r^n$ in the $[n, k+1]_{q^m}$ extended code $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$ of $\mathcal{C}$. Let $\boldsymbol{H}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$ be the parity-check matrix of $\mathcal{C}_{\boldsymbol{y}}$. The problem can be further reduced to find an $\boldsymbol{e} \in \mathcal{S}_r^n$ such that $\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{\varepsilon}\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{0}$.

There are two main kinds of attacks for the RD problem, i.e., combinatorial attacks [19,48,29,6] and algebraic attacks [29,8,10,9]. The basic idea of the combinatorial attacks [19,48,29,6] is to guess some unknown variables about the equations $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$ or $\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{\varepsilon}\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{0}$ so that they can be directly solved by using Gaussian eliminations (note that number of equations are much less than that of the variables). The guess complexity is the main cost for the combinatorial attacks. In contrast, the algebraic attacks [29,8,10,9] resort to establish sufficiently more equations using different algebraic properties such as the annulator polynomial, so that the error $\boldsymbol{e}$ can be directly found by solving those equations. The complexity of the algebraic attacks is mainly determined by the number of the unknown variables of those equations. By carefully investigating

the best known attacks, we find that both combinatorial and algebraic attacks can benefit from the blockwise structures, the basic reason is that the coefficient matrix C for an ℓ-error has a special block-diagonal form, which allows to greatly reduce the number of the unknown variables. The take-away message is that the best cost for solving the ℓ-RD problem is roughly equal to the ℓ-th square root of the cost for solving the standard RD problem (with the same parameters). This means that for appropriate choices of constant $\ell > 1$ such as $\ell = 2$ or 3 in our applications, the ℓ-RD problem is still exponentially hard.

The Blockwise LRPC (ℓ-LRPC) Codes. Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be the parity-check matrix of an $[n,k]_{q^m}$ LRPC code. The entries of $\boldsymbol{H}$ generate an $\mathbb{F}_q$-linear subspace F of dimension d (for simplicity, we call $\boldsymbol{H}$ a matrix of weight d and support F). Let $\boldsymbol{e} \in \mathcal{S}_r^n$ be an error of support E and let $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$. Let EF be the product space of E and F, whose dimension is equal to rd with overwhelming probability when rd is sufficiently smaller than m. The decoding algorithm works by first recovering the product space EF using the support $\mathrm{Supp}(\boldsymbol{s})$ of $\boldsymbol{s}$ (which requires the weight $\|\boldsymbol{s}\|_{\mathrm{R}}$ is equal to the dimension of EF), then recovering the error support E from EF, and finally solving the linear equations $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$ using E. The Decode Failure Rate (DFR) is about $q^{\|\boldsymbol{s}\|_{\mathrm{R}}-(n-k)} = q^{rd-(n-k)}$, implying that an LPRC code of weight d can decode errors of weight up to $\frac{n-k}{d}$.

We define the blockwise LRPC (ℓ-LRPC) codes as generalizations of the standard LPRC codes whose parity-check matrices can be divided by columns into ℓ sub-matrices with disjoint supports. Let $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{d} = (d_1, \ldots, d_\ell)$ be vectors of positive integers and $k \in \mathbb{N}$. We say that an $[n,k]_{q^m}$ LRPC code is an ℓ-LRPC code with parameters $n = \sum_{i=1}^{\ell} n_i$ and $d = \sum_{i=1}^{\ell} d_i$ if its parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ can be divided into ℓ sub-matrices $\boldsymbol{H} = (\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_\ell)$ such that 1) the sub-matrix $\boldsymbol{H}_i \in \mathbb{F}_{q^m}^{n_i}$ has small weight d_i for all $i \in \{1..\ell\}$; and 2) the supports $\{F_i = \mathrm{Supp}(\boldsymbol{H}_i)\}$ of these sub-matrices are mutually disjoint, namely, $F_i \cap F_j = \{0\}$ for all $i \neq j$.

The decoding algorithm for ℓ-LRPC codes works the same way as the one for standard LRPC codes. For traditional errors, an ℓ-LRPC code has the same decoding capacity as a standard LRPC code. However, it is more powerful when decoding ℓ-errors. This is because for an ℓ-error $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ with supports $(E_1, E_2, \ldots, E_\ell)$ and $\boldsymbol{r} = (r_1, \ldots, r_\ell)$, the product space in consideration becomes $\sum_{i=1}^{\ell} E_i F_i$, whose dimension is upper bounded by $\sum_{i=1}^{\ell} r_i d_i < rd$, where $r = \sum_{i=1}^{\ell} r_i$. This means that the ℓ-LRPC code can decode an ℓ-error with a much larger weight r. Formally, we have the following Theorem 1.1 (see the proofs in Section 4).

Theorem 1.1 *When $d_1 = d_2 = \cdots = d_\ell$, the ℓ-LRPC code allows to decode ℓ-errors of weight up to $r = \sum_{j=1}^{\ell} r_j = \frac{n-k}{d_1}$. By setting $d_1 = d_2 = \cdots = d_\ell = 2$, it can decode ℓ-errors of weight up to $\frac{n-k}{2}$.*

Theorem 1.1 implies that when dealing with ℓ-errors, the decoding capacity for the ℓ-LRPC codes is ℓ times larger than that of the standard LRPC codes.

For example, fixing $d = 4$, $r = 8$, and the DFR of q^{32-n-k}, an $[n, k]_{q^m}$ LRPC code can decode errors of weight 8, but an $[n, k]_{q^m}$ 2-LRPC codes with parameter $\boldsymbol{d} = (d_1, d_2) = (2, 2)$ can decode ℓ-errors with parameter $\boldsymbol{r} = (r_1, r_2) = (8, 8)$ of weight up to $r = r_1 + r_2 = 16$.

Applications. By making a tradeoff between the hardness of the ℓ-RD problem and the decoding capacity of the ℓ-LRPC codes, it is possible to design more efficient and secure rank-based cryptosystems with flexible choices of parameters. In particular, the blockwise structures would lead to larger parameters to reserve the security, but the gain in decoding capacity still allows us to design more efficient cryptosystems. As an application, we show that both the RQC and ROLLO cryptosystems can be greatly improved by using the ideal variant of the ℓ-RD problem and ℓ-LRPC codes. A brief comparison with related coded-based cryptosystems at the same 128-bit security is summarized in Table 1, which shows that our RQC is about 50% more compact than the original RQC, and has smaller sizes than the three code-based cryptosystems using the Hamming metric, namely, HQC, BIKE, and Classic McEliece.

Table 1. Comparisons of size and DFR for 128-bit security.

Schemes		pks (bytes)	cts (bytes)	total (bytes)	DFR
RQC	**Our**	860	1704	2564	-
	NIST [41]	1834	3652	5486	-
Lake (ROLLO-I)	**Our**	511	511	1022	2^{-31}
	NIST [40]	696	696	1392	2^{-28}
Locker (ROLLO-II)	**Our**	1814	1942	3756	2^{-131}
	NIST [40]	1941	2089	4030	2^{-134}
Ouroboros-R (ROLLO-III)	**Our**	623	1166	1789	2^{-33}
	TIT 2022 [3]	736	1431	2167	2^{-28}
HQC	NIST [42]	2249	4497	6746	-
BIKE	NIST [2]	1541	1573	3114	2^{-128}
Classic McEliece	NIST [12]	261120	96	261216	-
Ouroboros	TIT 2022 [3]	1566	3100	4666	2^{-128}

The public key size (pks), the ciphertext size (cts), total = pks+cts.

1.3 Other Related Works

The idea of using blockwise errors can be seen as an adaption of the LPN/LWE problem in rank metric [13]. Our blockwise codes are also related to the sum-rank metric codes [18], where the error is also divided into ℓ blocks and the sum-rank weight is defined as the sum of rank weight of each block. One main difference is that we explicitly require the ℓ blocks to have disjoint supports, which is very crucial for our results in this paper.

1.4 Organization

After some notations given in Section 2, we define the ℓ-errors and analyze the complexity of solving the ℓ-RD problem in Section 3. Section 4 defines the ℓ-LRPC codes and analyzes decoding failure probability and error-correcting capability. In Section 5, we apply the ideal ℓ-RD problem and the ideal ℓ-LRPC codes to improve RQC and ROLLO. We conclude this paper in Section 6.

2 Notations

- We denote by $\mathbb{N}$ the set of positive integer numbers, q prime or prime power, and $\mathbb{F}_{q^m}$ an extension of degree m of the finite field $\mathbb{F}_q$.
- Let $\alpha \in \mathbb{F}_{q^m}$ be a primitive element and $\boldsymbol{\alpha} = (1, \alpha, \ldots, \alpha^{m-1})$ be a basis of $\mathbb{F}_{q^m}$ viewed as an $\mathbb{F}_q$ vector space.
- Vectors (resp. matrices) are represented by lower-case (resp. upper-case) bold letters. We say that an algorithm is a PPT algorithm if it is a probabilistic polynomial-time algorithm.
- If $\mathcal{X}$ is a finite set, $x \xleftarrow{\$} \mathcal{X}$ (resp. $x \xleftarrow{\mathsf{seed}} \mathcal{X}$) denotes that x is chosen uniformly and randomly from the set $\mathcal{X}$ (resp. by a seed seed).
- For integers $a \leq b$, let $\{a..b\}$ denote all integers from a to b.
- The number of $\mathbb{F}_q$-subspaces of dimension r of $\mathbb{F}_{q^m}$ is given by the Gaussian coefficient $\begin{bmatrix} m \\ r \end{bmatrix}_q = \prod_{i=0}^{r-1} \frac{q^m - q^i}{q^r - q^i} \approx q^{r(m-r)}$.
- The submatrix of a matrix $\boldsymbol{M}$ formed from the rows in I and columns in J is denoted by $\boldsymbol{M}_{I,J}$. When I (resp. J) consists of all the rows (resp. columns), we use the notation $\boldsymbol{M}_{*,J}$ (resp. $\boldsymbol{M}_{I,*}$).
- $|\boldsymbol{M}|$, $|\boldsymbol{M}|_{I,J}$, and $|\boldsymbol{M}|_{*,J}$ are the determinant of the matrix $\boldsymbol{M}$, the submatrix $\boldsymbol{M}_{I,J}$, and the submatrix $\boldsymbol{M}_{*,J}$, respectively.
- $\mathrm{GL}_\eta(\mathbb{F}_q)$ is a general linear group and represents the set of all invertible matrices of size η over $\mathbb{F}_q$. The matrix $\boldsymbol{I}_r$ is the identity matrix of size r.
- The maximal minor c_T of a matrix $\boldsymbol{C}$ of size $r \times n$ is the determinant of its submatrix $\boldsymbol{C}_{*,T}$ whose column indexes $T \subset \{1..n\}$ and $\#T = r$.
- Cauchy-Binet formula that computes the determinant of the product of $\boldsymbol{A} \in \mathbb{F}_{q^m}^{r \times n}$ and $\boldsymbol{B} \in \mathbb{F}_{q^m}^{n \times r}$ is expressed as $|\boldsymbol{AB}| = \sum_{T \subset \{1..n\}, \#T = r} |\boldsymbol{A}|_{*,T} |\boldsymbol{B}|_{T,*}$.
- The Gaussian elimination of a $\mu \times \nu$ matrix of rank ρ over an $\mathbb{F}_q$ has a complexity of $\mathcal{O}(\rho^{\omega-2}\mu\nu)$ operations in $\mathbb{F}_q$, where ω is the exponent of matrix multiplication with $2 \leq \omega \leq 3$ and a practical value is 2.81 when more than a few hundreds rows and columns.
- The complexities are estimated by operations in $\mathbb{F}_q$ if there is no ambiguity. All logarithms are of base 2.

3 The ℓ-RD Problem and Its Complexity

In this section, we first introduce the blockwise errors (ℓ-errors) and the blockwise RD (ℓ-RD) problem in Subsection 3.1. Then, to analyze the complexity of

the ℓ-RD problem, we refine a universal reduction from existing attacks on the RD problem and analyze the support and coefficient matrices of the ℓ-error in Subsection 3.2. Finally, we adapt the typical combinatorial and algebraic attacks to the ℓ-RD problem in Subsection (3.3 - 3.5), and find that the ℓ-errors do not ease the problem too much: the ℓ-RD problem is still exponentially hard for appropriate choices of $\ell > 1$.

3.1 The ℓ-errors and ℓ-RD Problem

Let $\ell, k \in \mathbb{N}$. Let $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ be vectors of positive integers. Let $n = \sum_{i=1}^{\ell} n_i$ and $r = \sum_{i=1}^{\ell} r_i$. We first define the disjointness of multiple subspaces. We say that ℓ $\mathbb{F}_q$-subspaces $\{V_i\}_{i\in\{1..\ell\}}$ of $\mathbb{F}_{q^m}$ are mutually *disjoint* if $\forall\ i, j \in \{1..\ell\}, i \neq j, V_i \cap V_j = \{0\}$.

Definition 3.1 (Blockwise Errors (ℓ-errors)) *Let $\boldsymbol{e}_i \in \mathbb{F}_{q^m}^{n_i}$ be a vector of weight r_i for $i \in \{1..\ell\}$. An error $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathbb{F}_{q^m}^{n}$ is called an ℓ-error if the supports of ℓ vectors $\boldsymbol{e}_i$'s are mutually disjoint.*

Recall that $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ are two vectors of positive integers. We denote the set of such ℓ-errors by $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$. Let E_i be the support of dimension r_i of $\boldsymbol{e}_i$. Because all supports are mutually disjoint, the ℓ-error $\boldsymbol{e}$ can be viewed as the error of weight r and support $E = \sum_{i=1}^{\ell} E_i$.

We now define the ℓ-RD problem. This problem is the Rank Decoding (RD) problem finding the ℓ-errors.

Definition 3.2 (Blockwise RD (ℓ-RD) Problem) *Let $\boldsymbol{G}$ be the generator matrix of a random $[n, k]_{q^m}$-linear code $\mathcal{C}$ and $\boldsymbol{y} \in \mathbb{F}_{q^m}^{n}$. The problem is to find $\boldsymbol{x} \in \mathbb{F}_{q^m}^{k}$ and $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ such that $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$.*

Like the dual version of the RD problem using the generator matrix is the Rank Syndrome Decoding (RSD) problem [31] using the parity-check matrix, the dual version of the ℓ-RD problem is defined as the ℓ-RSD problem.

Definition 3.3 (Blockwise RSD (ℓ-RSD) Problem) *Let $\boldsymbol{H}$ be the parity-check matrix of a random $[n, k]_{q^m}$-linear code $\mathcal{C}$ and $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$. The problem is to find $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ such that $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$.*

Two variants is exactly the standard RD and RSD problems when $\ell = 1$. By the duality, the hardness of two variants is equivalent. Intuitively, two variants are also hard because they still find a small-weight error.

3.2 Reduction, Support and Coefficient Matrices

In this subsection, we first recall existing attacks on the RD problem, then adapt the reduction refined from typical attacks to the ℓ-RD problem, finally analyze support and coefficient matrices of the ℓ-error.

Attacks on the RD problem. There currently exist the combinatorial and algebraic attacks [19,48,29,6,29,8,10,9] on the RD problem. Please see Appendix B for detailed overviews of these attacks. The first combinatorial attack [19] starts with the RSD problem and is significantly improved in [48] and further refined in [29,6]. The combinatorial attacks [19,29,6] consist of subtly guessing the support of error and solving a linear system. The attack [48] transforms a quadratic multivariate system obtained from the RD problem into a linear system by guessing the entries of support matrix and coefficient matrix. Another way is the algebraic attack [29], where one solves a multivariate system induced from the RD problem based on the annulator polynomial by linearization and Gröbner basis. A breakthrough paper [8] shows that the $\mathbb{F}_{q^m}$-linearity allows to devise a dedicated algebraic attack, i.e., the MaxMinors (MM) modeling. Then the MM modeling is refined and improved in [10] where the authors also introduced another algebraic modeling, the Support-Minors (SM) modeling. The SM modeling later is combined with the MM modeling (i.e., the SM-$\mathbb{F}_{q^m}^+$ modeling [9]). Both SM and MM modelings reduce the RD problem to solving a linear system. The analysis in [9] shows that the cost of the SM-$\mathbb{F}_{q^m}^+$ modeling is close to those of the combinatorial attack [6] and the MM modeling [10].

To measure the potential complexity loss and ensure the security of schemes, we adapt typical combinatorial attacks [48,6] and algebraic attacks [29,10] to the ℓ-RD problem in Subsection (3.3 - 3.5). The reduction technique in attacks [48,6,29,10] is still available to the ℓ-RD problem. We refine the reduction in Theorem 3.4.

Theorem 3.4 *Solving the ℓ-RD(q, m, n, k, r, ℓ) problem defined by $[n, k]_{q^m}$ linear code $\mathcal{C}$ (see Definition 3.2) can be reduced to finding a blockwise codeword (i.e., an ℓ-error) of weight r in the $[n, k+1]_{q^m}$ extended code of $\mathcal{C}$.*

Proof. Once obtaining word $\boldsymbol{y}$, one adds $\boldsymbol{y}$ to code $\mathcal{C}$ and obtains an $[n, k+1]_{q^m}$ extended code $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$ with a generator matrix $\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix}$ of size $(k+1) \times n$. In this way, $\boldsymbol{e} = \begin{pmatrix} 1 & -\boldsymbol{m} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix}$ is exactly a codeword of weight r of $\mathcal{C}_{\boldsymbol{y}}$. Let $\boldsymbol{G}_{\boldsymbol{y}} = (\boldsymbol{I}_{k+1} \ \boldsymbol{R}) \in \mathbb{F}_{q^m}^{(k+1)\times n}$ be a systematic generator matrix of $\mathcal{C}_{\boldsymbol{y}}$ and $\boldsymbol{H}_{\boldsymbol{y}} = \begin{pmatrix} -\boldsymbol{R}^\top & \boldsymbol{I}_{n-k-1} \end{pmatrix} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$ be a systematic parity-check matrix of $\mathcal{C}_{\boldsymbol{y}}$, where $\boldsymbol{R} \in \mathbb{F}_{q^m}^{(k+1)\times(n-k-1)}$. Then solving the ℓ-RD problem consists in finding an $\boldsymbol{u} \in \mathbb{F}_{q^m}^{k+1}$ of weight $\leq r$ such that

$$\boldsymbol{u}\boldsymbol{G}_{\boldsymbol{y}} = \boldsymbol{e}, \tag{2}$$

or finding an ℓ-error $\boldsymbol{e}$ of weight r such that

$$\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{0}. \tag{3}$$

□

The support and coefficient matrices of the ℓ-error are crucial tools to construct the specific attack modelings by exploiting the reduction in Theorem 3.4. The entries of two matrices determine the number of variables of algebraic equations in the attack modelings. We next analyze the forms of two matrices.

Support and Coefficient Matrices of the ℓ-error. Let $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ be vectors of positive integers. Let $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ be an ℓ-error. If let $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{ir_i}) \in \mathbb{F}_{q^m}^{r_i}$ be a basis of support of dimension r_i, then there exists a matrix $\boldsymbol{C}_i \in \mathbb{F}_q^{r_i \times n_i}$ of rank r_i such that $\boldsymbol{e}_i = \boldsymbol{\varepsilon}_i \boldsymbol{C}_i$, If one expresses the basis $\boldsymbol{\varepsilon}_i$ as a matrix $\boldsymbol{S}_i \in \mathbb{F}_q^{m \times r_i}$ of rank r_i under the basis $\boldsymbol{\alpha}$, then $\boldsymbol{e}_i = \boldsymbol{\alpha} \boldsymbol{S}_i \boldsymbol{C}_i$. We have $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C}$, where $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \ldots, \boldsymbol{\varepsilon}_\ell) \in \mathbb{F}_{q^m}^{r}$,

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_1 & \boldsymbol{S}_2 & \cdots & \boldsymbol{S}_\ell \end{pmatrix} \in \mathbb{F}_q^{m \times r}, \qquad \boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_\ell \end{pmatrix} \in \mathbb{F}_q^{r \times n}. \tag{4}$$

We call $\boldsymbol{S}$ and $\boldsymbol{C}$ respectively support matrix and coefficient matrix of $\boldsymbol{e}$.

Remark 1. The main difference with the standard rank metric error is that the form of the coefficient matrix $\boldsymbol{C}$ of the ℓ-error is of block-diagonal form. For a standard rank metric error $\boldsymbol{e} \in \mathcal{S}_r^n$, let $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \in \mathbb{F}_{q^m}^r$ be a basis of $\mathrm{Supp}(\boldsymbol{e})$, the there is a coefficient matrix $\boldsymbol{C} \in \mathbb{F}_q^{r \times n}$ of rank r such that $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C}$. Under the basis $\boldsymbol{\alpha}$, there is a support matrix $\boldsymbol{S} \in \mathbb{F}_q^{m \times r}$ of rank r such that $\boldsymbol{\varepsilon} = \boldsymbol{\alpha}\boldsymbol{S}$. Then $\boldsymbol{e} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C}$.

Support and Coefficient Matrices with Less Entries. Because all multiples $\lambda\boldsymbol{e}$ for $\lambda \in \mathbb{F}_{q^m}^*$ are solutions of Equation (3) due to $\|\lambda\boldsymbol{e}\|_{\mathrm{R}} = r$, one can specify λ to be the inverse of the first coordinate of $\boldsymbol{e}$. Without loss of generality, let the first coordinate of $\boldsymbol{e}$ be 1, then one can set the first column of $\boldsymbol{C}$ to $(1\ 0\ \cdots\ 0)^\top$ and the first column of $\boldsymbol{S}$ to $(1\ 0\ \cdots\ 0)^\top$. Then $\boldsymbol{S}$ and $\boldsymbol{C}$ can be further reduced to two forms with less entries.

- $\boldsymbol{S}_{\{1..r\},*} = \boldsymbol{I}_r$. By Gaussian elimination on column of $\boldsymbol{S}$, there is a matrix $\boldsymbol{P} \in \mathrm{GL}_r(q)$ such that $\boldsymbol{S}\boldsymbol{P} = \left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S}' \end{array}\right)$ and $\boldsymbol{P}^{-1}\boldsymbol{C} = \left(\begin{array}{c|c} 1 & \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}' \end{array}\right)$ where $\boldsymbol{S}' \in \mathbb{F}_q^{(m-r)\times(r-1)}$ and $\boldsymbol{C}' \in \mathbb{F}_q^{r\times(n-1)}$. Then

$$\boldsymbol{e} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{P}\boldsymbol{P}^{-1}\boldsymbol{C} = \boldsymbol{\alpha}\left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S}' \end{array}\right)\left(\begin{array}{c|c} 1 & \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}' \end{array}\right). \tag{5}$$

  Let $\boldsymbol{s} := \boldsymbol{S}\boldsymbol{P}$ and $\boldsymbol{C} := \boldsymbol{P}^{-1}\boldsymbol{C}$.
- $\boldsymbol{C}_i$ is of systematic form. By Gaussian elimination on row of $\boldsymbol{C}$, there is a matrix $\boldsymbol{Q}_i \in \mathrm{GL}_{r_i}(q)$ such that $\boldsymbol{Q}_i\boldsymbol{C}_i = (\boldsymbol{I}_{r_i}\ \boldsymbol{C}_i')$ and $\boldsymbol{S}\boldsymbol{Q}^{-1} = \left(\begin{array}{c|c} 1 & \\ \hline \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}' \end{array}\right)$

where $\boldsymbol{C}'_i \in \mathbb{F}_q^{r_i \times (n_i - r_i)}$, $\boldsymbol{S}' \in \mathbb{F}_q^{m \times (r-1)}$, and $\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_\ell \end{pmatrix} \in$ $\mathrm{GL}_r(q)$. Then

$$\boldsymbol{e} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{Q}^{-1}\boldsymbol{Q}\boldsymbol{C} = \boldsymbol{\alpha}\left(\begin{array}{c|c} 1 & \\ \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}' \end{array}\right) \begin{pmatrix} \boldsymbol{Q}_1\boldsymbol{C}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2\boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Q}_\ell\boldsymbol{C}_\ell \end{pmatrix}. \tag{6}$$

Let $\boldsymbol{S} := \boldsymbol{S}\boldsymbol{Q}^{-1}$ and $\boldsymbol{C} := \boldsymbol{Q}\boldsymbol{C}$.

For solving the ℓ-RD problem, most attacks aim to recover $\boldsymbol{S}$ and $\boldsymbol{C}$ by solving the algebraic equations obtained from Equations (2 - 6). Equation (3) is used to build the AGHT attacks (Subsection 3.3). Equations (2, 5, 6) are used to build the OJ attack (Subsection 3.3). Equations (3, 6) are used to build the algebraic attack, the MM modeling (Subsection 3.5). The details of constructing the algebraic equations can refer to the specific attacks in Subsection (3.3 - 3.5).

3.3 Combinatorial Attacks on the ℓ-RD Problem

In this subsection, we use the AGHT attack [6] and the OJ attack [48] to analyze the complexity of solving the ℓ-RD problem.

AGHT Attack [6]. The idea is that the solver tries to guess a subspace that contains the support of the ℓ-error, then checks if the choice is correct. The cost depends on how to successfully guess such a subspace.

- Guess randomly a t-dimensional subspace F that contains the support $\mathrm{Supp}(\boldsymbol{e})$ of dimension $r = \sum_{i=1}^{\ell} r_i$ of the ℓ-error $\boldsymbol{e}$.
- Let $(f_1, f_2, \ldots, f_t) \in \mathbb{F}_{q^m}^t$ be a basis of F. One expresses $\boldsymbol{e}$ under this basis

$$\boldsymbol{e} = (e_1, e_2, \ldots, e_n) = (f_1, f_2, \ldots, f_t) \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ e_{t1} & e_{t2} & \cdots & e_{tn} \end{pmatrix} = (f_1, f_2, \ldots, f_t) \begin{pmatrix} \overline{\boldsymbol{e}}_1 \\ \overline{\boldsymbol{e}}_2 \\ \vdots \\ \overline{\boldsymbol{e}}_t \end{pmatrix},$$

where $\overline{\boldsymbol{e}}_i = (e_{i1}, e_{i2}, \ldots, e_{in}) \in \mathbb{F}_q^n$ for $i \in \{1..t\}$. By Equation (3): $\boldsymbol{H_y e}^\top = \boldsymbol{0}$, let $\boldsymbol{h}_j$ is the j-th row of $\boldsymbol{H_y}$, we have

$$\begin{aligned}\boldsymbol{H_y e}^\top &= \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_{n-k-1} \end{pmatrix} \left(\overline{\boldsymbol{e}}_1^\top, \overline{\boldsymbol{e}}_2^\top, \ldots, \overline{\boldsymbol{e}}_t^\top\right) \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_t \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{h}_1 f_1 & \boldsymbol{h}_1 f_2 & \cdots & \boldsymbol{h}_1 f_t \\ \boldsymbol{h}_2 f_1 & \boldsymbol{h}_2 f_2 & \cdots & \boldsymbol{h}_2 f_t \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{h}_{n-k-1} f_1 & \boldsymbol{h}_{n-k-1} f_2 & \cdots & \boldsymbol{h}_{n-k-1} f_t \end{pmatrix} \begin{pmatrix} \overline{\boldsymbol{e}}_1^\top \\ \overline{\boldsymbol{e}}_2^\top \\ \vdots \\ \overline{\boldsymbol{e}}_t^\top \end{pmatrix} = \boldsymbol{0}_{n-k-1}. \end{aligned} \tag{7}$$

- Express Equation (7) as a linear system over $\mathbb{F}_q$ and solve $\overline{\boldsymbol{e}}_i$. By expressing $\boldsymbol{h}_j f_i$ as a matrix $\mathrm{Mat}(\boldsymbol{h}_j f_i) \in \mathbb{F}_q^{m \times n}$ under the basis $\boldsymbol{\alpha}$ for $j \in \{1..n-k-1\}$ and $i \in \{1..t\}$, a linear system over $\mathbb{F}_q$ with nt unknowns and $m(n-k-1)$ equations is obtained. The linear system has only one solution with overwhelming probability if $nt \leq m(n-k-1)$.
- The probability of $F \supset E$ is estimated as $\frac{\begin{bmatrix} t \\ r \end{bmatrix}_q}{\begin{bmatrix} m \\ r \end{bmatrix}_q} \approx q^{-r(m-t)}$. In this way, the complexity is $\mathcal{O}\left(((n-k-1)m)^\omega q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil}\right)$.
- Use $\mathbb{F}_{q^m}$-linearity to decrease the cost. Since, for any $\lambda \in \mathbb{F}_{q^m}^*$, $\|\lambda \boldsymbol{e}\|_{\mathrm{R}} = r$ and all multiples $\lambda \boldsymbol{e}$ are solutions of Equation (3): $\boldsymbol{H_y e}^\top = \boldsymbol{0}$, the complexity is divided by about q^m.

As a result, this attack has a complexity of $\mathcal{O}\left(((n-k-1)m)^\omega q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil - m}\right)$.

In [15], the authors adapted the AGHT attack to the RD problem finding so-called non-homogeneous errors. Here, inspired by [15], the strategy guessing the subspace F is that the solver randomly guesses a subspace F_i of dimension t_i that contains the support $E_i = \mathrm{Supp}(\boldsymbol{e}_i)$ of dimension r_i of $\boldsymbol{e}_i$ such that all F_i's are mutually disjoint, and sets $F = \sum_{i=1}^{\ell} F_i$. In this way, the dimension of F is of $\sum_{i=1}^{\ell} t_i$, and F must contain the support of the ℓ-error $\boldsymbol{e}$.

If one knows F_i, then each entry of $\boldsymbol{e}_i$ can be expressed as an $\mathbb{F}_q$-linear combination of t_i elements in a basis of F_i. This means that one can write $\boldsymbol{e}_i$ using $n_i t_i$ unknowns in $\mathbb{F}_q$. Doing the same for all $\boldsymbol{e}_i$'s, one obtains $\sum_{i=1}^{\ell} n_i t_i$ unknowns. Then one solves the linear system with $\sum_{i=1}^{\ell} n_i t_i$ unknowns and $m(n-k-1)$ equations for single solution $\boldsymbol{e}$ as long as $\sum_{i=1}^{\ell} n_i t_i \leq m(n-k-1)$. The most costly part of the attack consists in finding the F_i's containing E_i for $i \in \{1..\ell\}$. We estimate this probability in Lemma 3.5.

Lemma 3.5 *Let $E_1, E_2, \ldots, E_\ell$ be fixed $\mathbb{F}_q$-subspaces of dimension respectively $r_1, r_2, ..., r_\ell$ of $\mathbb{F}_{q^m}$. The probability that one successfully guesses $\mathbb{F}_q$-subspaces $F_1, F_2, ..., F_\ell$ dimension respectively $t_1, t_2, \ldots, t_\ell$ of $\mathbb{F}_{q^m}$ such that all F_i's are mutually disjoint and $E_i \subset F_i$ is estimated as $\mathcal{O}\left(q^{-mr + \sum_{i=1}^{\ell-1} r_i^2 + \sum_{j=2}^{\ell} r_j \sum_{i=1}^{j-1} r_i + t_\ell r_\ell}\right)$.*

We give the detailed proof for Lemma 3.5 in Appendix C.1. Finally, one takes advantage of the $\mathbb{F}_{q^m}$-linearity to raise this probability: for any $\lambda \in \mathbb{F}_{q^m}^*$, $\|\lambda \boldsymbol{e}\|_{\mathrm{R}} = r$ and all multiples $\lambda \boldsymbol{e}$ are solutions of Equation (3): $\boldsymbol{H_y} \boldsymbol{e}^\top = \boldsymbol{0}$, hence the complexity is divided by about q^m. The complexity of solving the ℓ-RD problem by the variant of AGHT attack is estimated as

$$\mathcal{O}\left((m(n-k-1))^\omega q^{mr-\sum_{i=1}^{\ell-1} r_i^2 - \sum_{j=2}^{\ell} r_j \sum_{i=1}^{j-1} r_i - t_\ell r_\ell - m}\right)$$

where t_i is chosen to maximize $t_\ell r_\ell$ under the constraints

$$\begin{cases} r_i \le t_i, \quad for\ i \in \{1..\ell\}; \\ \sum_{i=1}^{\ell} t_i \le m-1; \\ \sum_{i=1}^{\ell} n_i t_i \le m(n-k-1). \end{cases}$$

OJ Attack. We now analyze the complexity of solving the ℓ-RD problem by the OJ attack [48]. Let $\overline{\boldsymbol{e}}_1$ and $\overline{\boldsymbol{e}}_2$ be the first $k+1$ and the last $n-k-1$ coordinates of $\boldsymbol{e}$. Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be the first $k+1$ columns and the last $n-k-1$ columns of $\boldsymbol{C}$. Then $\boldsymbol{e} = (\overline{\boldsymbol{e}}_1, \overline{\boldsymbol{e}}_2) = \boldsymbol{\varepsilon}(\boldsymbol{A}_1, \boldsymbol{A}_2) = (\boldsymbol{\alpha S A}_1, \boldsymbol{\alpha S A}_2)$. Equation (2) means

$$\boldsymbol{u G_y} = \boldsymbol{e} \Longleftrightarrow (\boldsymbol{u}\ \boldsymbol{uR}) = (\overline{\boldsymbol{e}}_1, \overline{\boldsymbol{e}}_2) \Longleftrightarrow \overline{\boldsymbol{e}}_1 \boldsymbol{R} = \overline{\boldsymbol{e}}_2 \Longleftrightarrow \boldsymbol{\alpha S A}_1 \boldsymbol{R} = \boldsymbol{\alpha S A}_2. \quad (8)$$

We first analyze the case of the 2-RD problem, then extend conclusions into general cases. By Equation (8), for $j \in \{1..n-k-1\}$, let $\boldsymbol{r}_j$ and $\boldsymbol{a}_j$ be the j-th column of $\boldsymbol{R}$ and $\boldsymbol{A}_2$, respectively, then

$$\boldsymbol{\alpha S A}_1 \boldsymbol{r}_j = \boldsymbol{\alpha S a}_j \iff \boldsymbol{\alpha S} \begin{pmatrix} \boldsymbol{A}_1\ \boldsymbol{a}_j \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_j \\ -1 \end{pmatrix} = 0. \quad (9)$$

Let $\begin{pmatrix} \boldsymbol{r}_j \\ -1 \end{pmatrix} = \boldsymbol{T}_j \boldsymbol{\alpha}^\top$ where $\boldsymbol{T}_j \in \mathbb{F}_q^{(k+2)\times m}$ is the matrix expression of $\begin{pmatrix} \boldsymbol{r}_j \\ -1 \end{pmatrix}$ under the basis $\boldsymbol{\alpha}$. Equation (9) can be written $\boldsymbol{\alpha S} \begin{pmatrix} \boldsymbol{A}_1\ \boldsymbol{a}_j \end{pmatrix} \boldsymbol{T}_j \boldsymbol{\alpha}^\top = 0$. This means

$$\boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j = \boldsymbol{0}_{m\times m}. \quad (10)$$

The entries of $\boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j$ are quadratic polynomials. Then Equation (10) gives a quadratic multivariate system over $\mathbb{F}_q$ with m^2 quadratic polynomials in the entries of $\boldsymbol{S}$ and $\boldsymbol{C}$.

The OJ attack uses the basis enumeration and the coordinates enumeration to transform the quadratic multivariate system into a linear system. The former guesses all entries of $\boldsymbol{S}$ and solves the linear system about the entries of $(\boldsymbol{A}_1\ \boldsymbol{a}_j)$ to determine $\boldsymbol{C}$. The latter guesses the entries of $\boldsymbol{C}$ and solves the linear system about the entries of $\boldsymbol{S}$ to determine $\boldsymbol{S}$.

When $\boldsymbol{S}$ and $\boldsymbol{C}$ are in the form of Equation (5) and Equation (6), the complexities are presented in Theorem 3.6 and Theorem 3.7. We give their detailed proofs in Appendix C.2 and Appendix C.3. The ideas of proofs can be easily extended to the ℓ-RD problem.

Theorem 3.6 *If $\boldsymbol{S}$ and $\boldsymbol{C}$ are in the form of Equation (5), the 2-RD problem can be solved with complexity $\mathcal{O}\left((kr+r)^{\omega}q^{(m-r)(r-1)}\right)$ by the basis enumeration.*

Theorem 3.7 *If $k=n_1$, $\boldsymbol{S}$ and $\boldsymbol{C}$ are in the form of Equation (6), the 2-RD problem can be solved with complexity $\mathcal{O}\left((m(r-1)+(n_1-r_1))^{\omega}q^{(r_1-1)(n_1-r_1)+r_2}\right)$ by the coordinates enumeration.*

Theorem 3.8 *If $k=n_1$, the complexity of solving the ℓ-RD problem by the OJ attack is estimated as*

$$\begin{cases}\mathcal{O}\big((kr+r)^{\omega}q^{(m-r)(r-1)}\big), & \textit{Basis Enumeration};\\ \mathcal{O}\left((m(r-1)+(n_1-r_1))^{\omega}q^{(r_1-1)(n_1-r_1)+\gamma}\right), & \textit{Coordinates Enumeration},\end{cases}$$

where $\gamma=\max\left\{r_i : i\in\{2..\ell\}\right\}$ and $r=\sum_{i=1}^{\ell}r_i$.

3.4 Algebraic Attack by Annulator Polynomial

This algebraic attack [29] differs from attacks aiming to recover $\boldsymbol{S}$ and $\boldsymbol{C}$ with reductions described in Subsection 3.2. It directly solves $\boldsymbol{x}$ from a multivariate system obtained from the ℓ-RD instance and the theory of q-polynomials [47], more specifically annulator polynomials (see Appendix A). The attack details are outlined in Appendix B.2.

For the ℓ-RD problem finding the ℓ-error $\boldsymbol{e}=(\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_\ell)\in\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$, the solver splits $\boldsymbol{y}$ as $(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_\ell)$ and splits $\boldsymbol{G}$ as $(\boldsymbol{G}_1,\boldsymbol{G}_2,\ldots,\boldsymbol{G}_\ell)$ by columns $\boldsymbol{n}$. Then

$$(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_\ell)=\boldsymbol{x}(\boldsymbol{G}_1,\boldsymbol{G}_2,\ldots,\boldsymbol{G}_\ell)+(\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_\ell).$$

In this way, the ℓ-RD problem is divided into ℓ subproblems, for $\nu\in\{1..\ell\}$, $\boldsymbol{y}_\nu=\boldsymbol{x}\boldsymbol{G}_\nu+\boldsymbol{e}_\nu$, then one solves $\boldsymbol{x}$ from *one* of ℓ subproblems.

Let $\boldsymbol{x}=(x_1,x_2,\ldots,x_k)$. For $\nu\in\{1..\ell\}$, let $\boldsymbol{y}_\nu=(y_1,y_2,\ldots,y_{n_\nu})$, $\boldsymbol{G}_\nu=(g_{ij})_{\substack{i\in\{1..k\}\\ j\in\{1..n_\nu\}}}$, and $\boldsymbol{e}_\nu=(e_1,e_2,\ldots,e_{n_\nu})$. Since the entries of $\boldsymbol{e}_\nu$ lie in the support $\mathrm{Supp}(\boldsymbol{e}_\nu)$ of dimension r_ν, there exists a unique monic q-polynomials $P^{(\nu)}(u)=\sum_{\delta=0}^{r_\nu}p_\delta^{(\nu)}u^{q^\delta}$ of q-degree r_ν such that for $j\in\{1..n_\nu\}$

$$P^{(\nu)}\left(y_j-\sum_{i=1}^{k}x_ig_{ij}\right)=\sum_{\delta=0}^{r_\nu}\left(p_\delta^{(\nu)}y_j^{q^\delta}-\sum_{i=1}^{k}p_\delta^{(\nu)}x_i^{q^\delta}g_{ij}^{q^\delta}\right)=P^{(\nu)}\left(e_j\right)=0. \quad (11)$$

Equation (11) gives a multivariate system with n_ν polynomials and $(r_\nu+k)$ variables $p_\delta^{(\nu)}$ and x_i. For solving the ℓ-RD problem, one solves x_i from this multivariate system.

The linearization and Gröbner basis techniques are applied to solve x_i. The complexities are given in Theorem 3.9 and the detailed proof is presented in Appendix C.4.

Theorem 3.9 *The complexity of solving the ℓ-RD problem by annulator polynomials is estimated as*

$$\begin{cases} \mathcal{O}\left(\min\left\{(r_\nu k)^\omega q^{r_\nu\left\lceil\frac{(k+1)(r_\nu+1)-(n_\nu+1)}{r_\nu}\right\rceil} : \nu\in\{1..\ell\}\right\}\right), & \textit{Linearization;}\\ \mathcal{O}\left(\min\left\{n_\nu\binom{r_\nu+k+d_{reg}^{(\nu)}-1}{d_{reg}^{(\nu)}}^\omega : \nu\in\{1..\ell\}\right\}\right), & \textit{Gröbner Basis.}\end{cases}$$

where $d_{reg}^{(\nu)}$ is the degree of regularity of the semi-regular system.

3.5 Algebraic Attacks by the MaxMinors Modeling

The MaxMinors (MM) modeling [10] is a powerful algebraic attack for cryptographic parameters and reduces the RD problem to solving a linear system. Equation $\boldsymbol{\varepsilon C H_y^\top} = \mathbf{0}_{n-k-1}$ (obtained from Equation (3) and $\boldsymbol{e} = \boldsymbol{\varepsilon C}$) implies that $\boldsymbol{C H_y^\top} \in \mathbb{F}_{q^m}^{r\times(n-k-1)}$ is not of row full rank because a non-zero vector $\boldsymbol{s}$ belongs to its left kernel. Then all maximal minors $|\boldsymbol{C H_y^\top}|_{*,J}$ of $\boldsymbol{C H_y^\top}$ are equal to 0 for $J\subset\{1..n-k-1\}$ and $\#J = r$. By the Cauchy-Binet formula, each $|\boldsymbol{C H_y^\top}|_{*,J}$ can be viewed a non-zero linear combination about all maximal minors $c_T = |\boldsymbol{C}|_{*,T}$ for $T\subset\{1..n\}$ and $\#T = r$. One views non-zero c_T as unknowns and solves c_T from a linear system with $\binom{n}{r}$ unknowns and $\binom{n-k-1}{r}$ equations. Finally, one determines the entries of $\boldsymbol{C}$ from the c_T by using the fact that it is in systematic form. The MM modeling over $\mathbb{F}_{q^m}$ is built

$$\left\{P_J = |\boldsymbol{C H_y^\top}|_{*,J} : J\subset\{1..n-k-1\}, \#J = r\right\}, \quad (\text{MM-}\mathbb{F}_{q^m}) \tag{12}$$

Unknowns: $\binom{n}{r}$ variables $c_T\in\mathbb{F}_q$ for $T\subset\{1..n\}$ and $\#T = r$,
Equations: $\binom{n-k-1}{r}$ linear equations $P_J = 0$ over $\mathbb{F}_{q^m}$ in c_T.

However, this system has many solutions due to $\binom{n-k-1}{r} < \binom{n}{r}$ whereas one wants more equations than unknowns for a unique solution. To obtain more equations than unknowns, one unfolds the coefficients of P_J over $\mathbb{F}_q$ and obtains the MM-$\mathbb{F}_q$ modeling

$$\left\{P_{i,J} = |\boldsymbol{C H_y^\top}|_{*,J} : J\subset\{1..n-k-1\}, \#J = r, i\in\{1..m\}\right\}, \quad (\text{MM-}\mathbb{F}_q) \tag{13}$$

Unknowns: $\binom{n}{r}$ variables $c_T\in\mathbb{F}_q$ for $T\subset\{1..n\}$ and $\#T = r$,
Equations: $m\binom{n-k-1}{r}$ linear equations $P_{i,J} = 0$ over $\mathbb{F}_q$ in c_T.

We first analyze the case of the 2-RD problem, then extend conclusions to general cases. By Equation (6), the matrix $\boldsymbol{C}$ is of form

$$\boldsymbol{C} = \left(\begin{array}{c|c} \boldsymbol{I}_{r_1} & \boldsymbol{C}_1' \end{array}\middle|\begin{array}{c} \mathbf{0}_{r_1\times n_2} \\ \hline \boldsymbol{I}_{r_2} \end{array}\begin{array}{c} \\ \hline \boldsymbol{C}_2' \end{array}\right) \in \mathbb{F}_q^{r\times n}, \tag{14}$$

where $\boldsymbol{C} = (c_{ij})_{\substack{i\in\{1..r\}\\ j\in\{1..n\}}} \in \mathbb{F}_q^{r\times n}$, $\boldsymbol{C}_1'\in\mathbb{F}_q^{r_1\times(n_1-r_1)}$, and $\boldsymbol{C}_2'\in\mathbb{F}_q^{r_2\times(n_2-r_2)}$. One can easily check

- $|\boldsymbol{C}|_{*,(\{1..r_1\}\setminus\{i\}\cup\{j\})\cup\{n_1+1..n_1+r_2\}} = (-1)^{r_1-i}c_{ij}$ for $i \in \{1..r_1\}$ and $j \in \{r_1+1..n_1\}$,
- $|\boldsymbol{C}|_{*,\{1..r_1\}\cup(\{n_1+1..n_1+r_2\}\setminus\{i\}\cup\{j\})} = (-1)^{n_1+r_2-i}c_{ij}$ for $i \in \{n_1+1..n_1+r_2\}$ and $j \in \{n_1+r_2+1..n\}$,
- $|\boldsymbol{C}|_{*,\{1..r_1\}\cup\{n_1+1..n_1+r_2\}} = 1$.

Therefore, once all c_T's are solved, one can determine the entries of the matrix $\boldsymbol{C}$. Lemma 3.10 bounds the number of equations and unknowns c_T.

Lemma 3.10 *Under block form of $\boldsymbol{C}$ in Equation (14), the MM-$\mathbb{F}_q$ modeling obtained from the 2-RD problem contains $\binom{n_1}{r_1}\binom{n_2}{r_2}$ unknowns c_T and at most $m\binom{n-k-1}{r}$ equations.*

We give the detailed proof for Lemma 3.10 in Appendix C.5.

Remark 2. Our analysis follows the idea of updated RQC [41], where authors bounded the maximal number of equations. On the one hand, considering less equations could lead to a higher complexity because in this case one is more likely to solve an underdetermined system with more unknowns and would guess more entries of $\boldsymbol{C}$ to transform the system into an overdetermined case (see hybrid method in the proof of Theorem 3.11). This means that using the maximal number of equations would give a lower bound of complexity. Cryptographic parameters often lead to an underdetermined case. On the other hand, the number of zero and dependent equations is negligible to the maximal number $m\binom{n-k-1}{r}$ and their impact on complexity is very limited. A thorough analysis in [15,9] supported this point and we also experimentally verified this when $\ell = 2, 3$.

Remark 3. The number of non-zero variables c_T is easy to compute. When n and r are divisible by ℓ, by Stirling approximation, the loss of variables c_T is large due to $\binom{n/\ell}{r/\ell}^{\ell} \approx \ell^{\frac{\ell}{2}}\left(\frac{n}{2\pi r(n-r)}\right)^{\frac{\ell-1}{2}}\binom{n}{r}$ while comparing with the MM-$\mathbb{F}_q$ modeling obtained from the standard RD problem. See Lemma C.1 in Appendix C.6 for this proof.

Theorem 3.11 *The complexity of solving the 2-RD problem by the MM-$\mathbb{F}_q$ modeling is estimated as*

$$\begin{cases} \mathcal{O}\left(m\binom{n-p-k-1}{r}\left(\binom{n_1}{r_1}\binom{n_2-p}{r_2}\right)^{\omega-1}\right), & m\binom{n-k-1}{r} \geq \binom{n_1}{r_1}\binom{n_2}{r_2} - 1; \\ \mathcal{O}\left(q^{a_1r_1+a_2r_2}m\binom{n-k-1}{r}\left(\binom{n_1-a_1}{r_1}\binom{n_2-a_2}{r_2}\right)^{\omega-1}\right), & m\binom{n-k-1}{r} < \binom{n_1}{r_1}\binom{n_2}{r_2} - 1. \end{cases}$$

where $p = \max\left\{i \mid m\binom{n-i-k-1}{r} \geq \binom{n_1}{r_1}\binom{n_2-i}{r_2} - 1\right\}$ and (a_1, a_2) is an integer pair such that $m\binom{n-k-1}{r} \geq \binom{n_1-a_1}{r_1}\binom{n_2-a_2}{r_2} - 1$ exactly holds.

We give a proof with full details for Theorem 3.11 in Appendix C.7. Theorem 3.11 can be extended to the case of the ℓ-RD problem.

Theorem 3.12 *The complexity of solving the ℓ-RD problem by the MM-$\mathbb{F}_q$ modeling is estimated as*

$$\begin{cases} \mathcal{O}\left(m\binom{n-p-k-1}{r}\left(\binom{n_\ell - p}{r_\ell}\prod_{i=1}^{\ell-1}\binom{n_i}{r_i}\right)^{\omega-1}\right), & m\binom{n-k-1}{r} \geq \prod_{i=1}^{\ell}\binom{n_i}{r_i} - 1; \\ \mathcal{O}\left(q^{\sum_{i=1}^{\ell} a_i r_i} m\binom{n-k-1}{r}\left(\prod_{i=1}^{\ell}\binom{n_i - a_i}{r_i}\right)^{\omega-1}\right), & m\binom{n-k-1}{r} < \prod_{i=1}^{\ell}\binom{n_i}{r_i} - 1. \end{cases}$$

where $p = \max\left\{ i \;\middle|\; m\binom{n-i-k-1}{r} \geq \binom{n_\ell - i}{r_\ell}\prod_{i=1}^{\ell-1}\binom{n_i}{r_i} - 1 \right\}$ *and* $(a_1, a_2, \ldots, a_\ell)$ *is an integers sequence such that* $m\binom{n-k-1}{r} \geq \prod_{i=1}^{\ell}\binom{n_i - a_i}{r_i} - 1$ *exactly holds.*

3.6 Summary of Complexities for Solving the ℓ-RD Problem

At the end of this section, we summarize the complexity gain of solving the ℓ-RD problem compared with the standard RD problem in Table 2. For the first three attacks, we only compare the exponential terms.

Table 2. Complexity comparisons of solving the ℓ-RD and RD problems.

Attacks	RD(q, m, n, k, r)	ℓ-RD(q, m, n, k, r, ℓ)
AGHT	$q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil - m}$	$q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil - m}$
OJ	$q^{(m-r)(r-1)+2}$ $q^{(r-1)(k+1)}$	$q^{(m-r)(r-1)}$ $q^{(r_1-1)(k-r_1)+\gamma}$ $\gamma = \max\{r_i : i \in \{2..\ell\}\}$
Annulator Polynomial	$q^{r\left\lceil \frac{(k+1)(r+1)-(n+1)}{r} \right\rceil}$ $n\binom{r+k+d_{reg}-1}{d_{reg}}^{\omega}$	$\min\left\{ q^{r_\nu\left\lceil \frac{(k+1)(r_\nu+1)-(n_\nu+1)}{r_\nu} \right\rceil} : \nu \in \{1..\ell\} \right\}$ $\min\left\{ n_\nu\binom{r_\nu+k+d_{reg}^{(\nu)}-1}{d_{reg}^{(\nu)}}^{\omega} : \nu \in \{1..\ell\} \right\}$
MM	$m\binom{n-p-k-1}{r}\left(\binom{n-p}{r}\right)^{\omega-1}$ $q^{ar} m\binom{n-k-1}{r}\left(\binom{n-a}{r}\right)^{\omega-1}$	$m\binom{n-p-k-1}{r}\left(\binom{n_\ell - p}{r_\ell}\prod_{i=1}^{\ell-1}\binom{n_i}{r_i}\right)^{\omega-1}$ $q^{\sum_{i=1}^{\ell} a_i r_i} m\binom{n-k-1}{r}\left(\prod_{i=1}^{\ell}\binom{n_i - a_i}{r_i}\right)^{\omega-1}$

Remark 4. The complexity analysis shows that the gain of most attacks on the ℓ-RD problem benefits from the blockwise structure of ℓ-errors. (1) the OJ and MM attacks benefits from the block-diagonal form of coefficient matrix $\boldsymbol{C}$ because the sparse $\boldsymbol{C}$ enables one to solve less variables (multivariable or linear) system; (2) the AGHT attack is limited because its cost depends on how to successfully guess a subspace that contains the support of the error; (3) the annulator polynomials attack benefits from the fact that the ℓ-errors allow to divide the ℓ-RD problem into ℓ subproblems with the smaller parameters.

For the powerful MM-$\mathbb{F}_q$ modeling, in the "underdetermined" case, an interesting result is that the complexity of solving the ℓ-RD problem allows to divide by a factor ℓ that of solving the standard RD problem.

Let $\ell|n$, $\ell|r$, $n' = n/\ell$, and $r' = r/\ell$. For both RD and ℓ-RD instances, when the parameters (m, n, k, r) satisfy respectively the "underdetermined" conditions: $m\binom{n-k-1}{r} < \binom{n}{r} - 1$ and $m\binom{n-k-1}{r} < \binom{n'}{r'}^{\ell} - 1$. The attacker chooses appropriate a and $(a_1, a_2, \ldots, a_\ell)$ such that

$$m\binom{n-k-1}{r} \geq \binom{n-a}{r} - 1 \quad \text{and} \quad m\binom{n-k-1}{r} \geq \prod_{i=1}^{\ell} \binom{n'-a_i}{r'} - 1$$

exactly hold. This means $\binom{n-a}{r} \approx \prod_{i=1}^{\ell} \binom{n'-a_i}{r'}$. From Lemma C.1 in Appendix C.6, an appropriate choice is $a_1 = a_2 = \cdots = a_\ell$ and $a_i = a/\ell$. At this point,

$$\frac{\log_q(T_{\text{RD}})}{\log_q(T_{\ell\text{-RD}})} \approx \frac{ar}{\sum_i^{\ell} a_i r_i} = \ell \quad \Longrightarrow \quad T_{\ell\text{-RD}} \approx \sqrt[\ell]{T_{\text{RD}}},$$

where T_{RD} and $T_{\ell\text{-RD}}$ are the complexity of solving the RD and ℓ-RD problems, respectively. This further shows that the speedup really benefits from the block-diagonal form of $\boldsymbol{C}$ because having $\boldsymbol{C}$ sparse enables one to guess $\sum_{i=1}^{\ell} a_i r_i$ entries of $\boldsymbol{C}$ to convert the "underdetermined" system into an "overdetermined" system, instead of ar entries in the standard RD problem.

We simulate the complexity of MM-$\mathbb{F}_q$ for RD, 2-RD, and 3-RD in Figure 1.

- (a) The RD instances are estimated with $(q, m, n, k) = (2, 200, 200, 100)$ and various even values $r = 2r'$ ($r' \in \{3..30\}$). The 2-RD instances are estimated with $(q, m, n, k, n_1, n_2) = (2, 200, 200, 100, 100, 100)$ and various values $r_1 = r_2 \in \{3..30\}$.
- (b) The RD instances are estimated with $(q, m, n, k) = (2, 100, 200, 100)$ and various even values $r \in \{6..40\}$. The 2-RD instances are estimated with $(q, m, n, k, n_1, n_2) = (2, 100, 200, 100, 100, 100)$ and various values $r_1 = r_2 \in \{3..20\}$.
- (c) The RD instances are estimated with $(q, m, n, k) = (2, 100, 300, 100)$ and various values $r = 3r'$ ($r' \in \{2..20\}$). The 3-RD instances are estimated with $(q, m, n, k, n_1, n_2, n_3) = (2, 100, 300, 100, 100, 100, 100)$ and various values $r_1 = r_2 = r_3 \in \{2..20\}$.

Our simulations become interesting as r increases. (a) and (b) in Figure 1 show that, when r is divided equally into (r_1, r_2), the exponential complexity allows to divide by a factor 2 for $r \geq 10$, i.e., $T_{\text{2-RD}} \approx \sqrt{T_{\text{RD}}}$. (c) in Figure 1 shows that, when r is divided equally into (r_1, r_2, r_3), the exponential complexity allows to divide by a factor 3 for $r \geq 12$, i.e., $T_{\text{3-RD}} \approx \sqrt[3]{T_{\text{RD}}}$. The parameters sizes in (b) and (c) are exactly the case of cryptography parameters in Section 5.

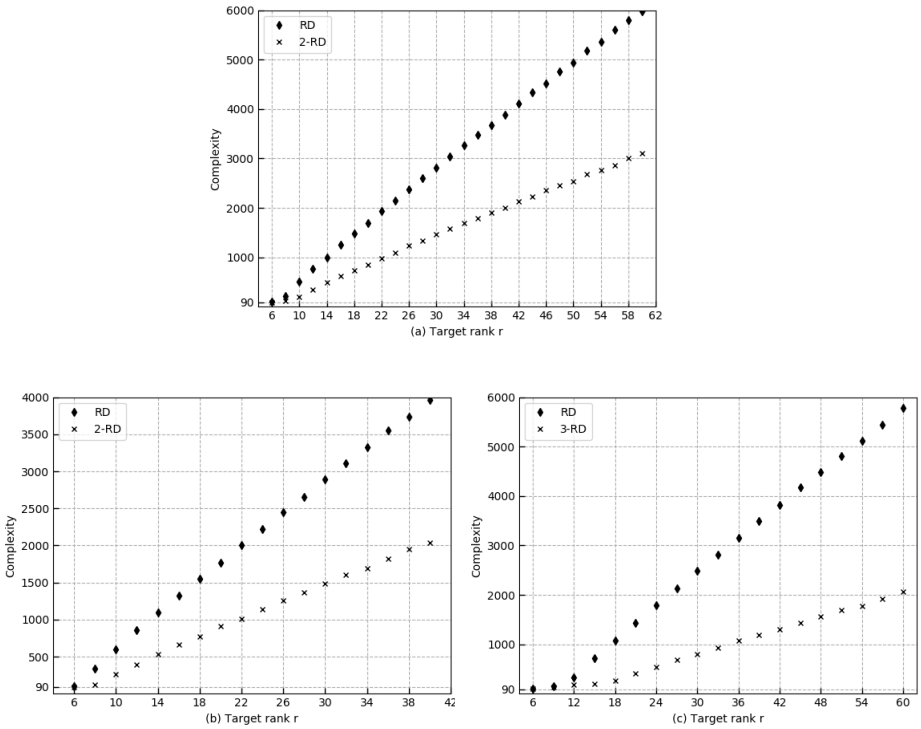

Fig. 1. Complexity trend of RD, 2-RD, and 3-RD by MM-$\mathbb{F}_q$.

4 The ℓ-LRPC Codes and Decoding Algorithm

In this section, we define the blockwise LRPC (ℓ-LRPC) codes, give its decoding algorithm, and analyze the decoding failure probability and the error-correcting capability. We find that the decoding algorithm can benefit from the blockwise structure: the decoding capacity can be significantly improved by a factor of ℓ. For cryptography applications in Section 5, we finally give the ℓ-Rank Support Recover (ℓ-RSR) algorithm which is used to recover the support of the ℓ-error.

4.1 The ℓ-LRPC Codes

An $[n,k]_{q^m}$ LRPC code [28,5] is defined by a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ with small weight. Our $[n,k]_{q^m}$ ℓ-LRPC code is defined by a parity-check matrix consisting of ℓ small-weight matrices of size $(n-k)\times n_i$.

Definition 4.1 (Blockwise LRPC (ℓ-LRPC) Codes) *Let $\ell, k \in \mathbb{N}$, $n_i, d_i \in \mathbb{N}$ for $i \in \{1..\ell\}$, and $n = \sum_{i=1}^{\ell} n_i$. Let $\boldsymbol{H}_i \in \mathbb{F}_{q^m}^{(n-k)\times n_i}$ be a matrix of weight*

d_i. Let the supports of ℓ matrices $\boldsymbol{H}_i$'s are mutually disjoint. An $[n,k]_{q^m}$ ℓ-LRPC code of length n and dimension k is defined by a parity-check matrix $\boldsymbol{H} = (\boldsymbol{H}_1\ \boldsymbol{H}_2\ \cdots\ \boldsymbol{H}_\ell) \in \mathbb{F}_{q^m}^{(n-k)\times n}$.

Let $\boldsymbol{n} = (n_1, n_2, \ldots, n_\ell)$ and $\boldsymbol{d} = (d_1, d_2, \ldots, d_\ell)$ be vectors of positive integers. We denote the set of such parity-check matrices by $\mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$. Let F_i be the support of dimension d_i of $\boldsymbol{H}_i$. Because all supports are mutually disjoint, the matrix $\boldsymbol{H}$ can be viewed as the matrix of weight $d = \sum_{i=1}^{\ell} d_i$ and support $F = \sum_{i=1}^{\ell} F_i$.

We next consider decoding algorithms for two error distributions: the ℓ-errors and the standard rank metric errors. In this subsection, we analyze the case of decoding the ℓ-errors. The decoding algorithm is also applied to ROLLO in Section 5. The latter is presented in Appendix D, where we show that for the standard errors, the ℓ-LRPC code has the same decoding capacity as the standard LRPC code.

4.2 Decoding ℓ-errors

Let $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ be a vector of positive integers. Consider an $[n,k]_{q^m}$ ℓ-LRPC code $\mathcal{C}$ with generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ and parity-check matrix $\boldsymbol{H} = (\boldsymbol{H}_1\ \boldsymbol{H}_2\ \cdots\ \boldsymbol{H}_\ell) \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$ of support $(F_1, F_2, \ldots, F_\ell)$. Let $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$ be a received word, where $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$ and $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ with the support $(E_1, E_2, \ldots, E_\ell)$. The syndrome $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \sum_{j=1}^{\ell} \boldsymbol{H}_j \boldsymbol{e}_j^\top$.

The general idea of decoding ℓ-error $\boldsymbol{e}$ uses the fact that the subspace $S = \langle s_1, s_2, \ldots, s_{n-k}\rangle_{\mathbb{F}_q}$ generated by $\boldsymbol{s}$ enables one to recover the space $\sum_{i=1}^{\ell} E_i F_i$. Once obtaining $\sum_{j=1}^{\ell} E_j F_j$, one recovers $E_1, E_2, \ldots, E_\ell$ and computes the support $E = \sum_{j=1}^{\ell} E_j$ of the error $\boldsymbol{e}$. Finally, the coordinates of $\boldsymbol{e}$ are computed by solving a linear system. The decoding algorithm is described in Algorithm 1.

4.3 Correctness of the Decoding Algorithm

The correctness of Algorithm 1 depends on the recovery of correct E_j, which requires $\dim S = \dim\left(\sum_{j=1}^{\ell} E_j F_j\right)$ and $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) = r_j$ for $j \in \{1..\ell\}$. We assume that these two conditions hold.

Step 1: the first step of the algorithm is obvious.

Step 2: we prove that $E_j = \bigcap_{i=1}^{d_j} S_{ji}$ for $j \in \{1..\ell\}$. Let $(\varepsilon_{j1}, \varepsilon_{j2}, \ldots, \varepsilon_{jr_j}) \in \mathbb{F}_{q^m}^{r_j}$ be the basis of E_j. Since $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top = \sum_{j=1}^{\ell} \boldsymbol{H}_j \boldsymbol{e}_j^\top$, $\boldsymbol{H} \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$ is a matrix of support $(F_1, F_2, \ldots, F_\ell)$, and $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ is an ℓ-error of support $(E_1, E_2, \ldots, E_\ell)$, we have that the entries of $\boldsymbol{H}_j \boldsymbol{e}_j^\top$ respectively lie in $E_j F_j$. Thus, $S \subset \sum_{j=1}^{\ell} E_j F_j$. By assumption $\dim S = \dim\left(\sum_{j=1}^{\ell} E_j F_j\right)$, we have $S = \sum_{j=1}^{\ell} E_j F_j$. Further, for any $i \in \{1..d_j\}$, since $f_{ji}\varepsilon_{j\kappa} \in \sum_{j=1}^{\ell} E_j F_j$ for all $\kappa \in \{1..r_j\}$, we have $\varepsilon_{j\kappa} \subset S_{ji} = \{f_{ji}^{-1} x : x \in S\} \Rightarrow E_j \subset S_{ji}$. Then, $E_j \subset \bigcap_{i=1}^{d_j} S_{ji}$. By assumption $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) = r_j$, we have $E_j = \bigcap_{i=1}^{d_j} S_{ji}$.

Algorithm 1 Decoding ℓ-errors for ℓ-LRPC codes

Input: the vector $\boldsymbol{y}$ and the parity-check matrix $\boldsymbol{H}$.
Output: the message $\boldsymbol{m}$

1: Computing syndrome space :
   – Compute the syndrome $\boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \sum_{i=1}^{\ell} \boldsymbol{H}_i \boldsymbol{e}_i^\top = \boldsymbol{s} = (s_1, s_2, \ldots, s_{n-k})^\top$ and the syndrome space $S = \langle s_1, s_2, \ldots, s_{n-k} \rangle_{\mathbb{F}_q}$.
2: Recovering the support E of the error $\boldsymbol{e}$:
   – Compute F_j from $\boldsymbol{H}$ for $j \in \{1..\ell\}$
   – Compute the basis $(f_{j1}, f_{j2}, \ldots, f_{jd_j}) \in \mathbb{F}_{q^m}^{d_j}$ of F_j for $j \in \{1..\ell\}$
   – Compute $S_{ji} = f_{ji}^{-1} S$, where all generators of S are multiplied by f_{ji}^{-1} for $j \in \{1..\ell\}$ and $i \in \{1..d_j\}$
   – Compute $E_j = \bigcap_{i=1}^{d_j} S_{ji}$ for $j \in \{1..\ell\}$
   – Compute $E = \sum_{j=1}^{\ell} E_j$
3: Recovering the error $\boldsymbol{e}$:
   – Compute the basis $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \in \mathbb{F}_{q^m}^r$ of E
   – Write each entry e_j of $\boldsymbol{e}$ as $e_j = \sum_{i=1}^{r} e_{ij}\varepsilon_j$ for $j \in \{1..n\}$ in the basis $\boldsymbol{\varepsilon}$
   – Solve e_{ij} from the linear system $\boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{s}$.
4: Recovering $\boldsymbol{m}$ from $\boldsymbol{m}\boldsymbol{G} = \boldsymbol{y} - \boldsymbol{e}$.

Step 3: one expresses $\boldsymbol{e}$ under the basis $\boldsymbol{\varepsilon}$ of E:

$$\boldsymbol{e} = (e_1, e_2, \ldots, e_n) = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1n} \\ e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ e_{r1} & e_{r2} & \cdots & e_{rn} \end{pmatrix} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \begin{pmatrix} \overline{\boldsymbol{e}}_1 \\ \overline{\boldsymbol{e}}_2 \\ \vdots \\ \overline{\boldsymbol{e}}_r \end{pmatrix},$$

where $\overline{\boldsymbol{e}}_i = (e_{i1}, e_{i2}, \ldots, e_{in})$ for $i \in \{1..r\}$, and computes $\overline{\boldsymbol{e}}_i$ from Equation (15):

$$\begin{aligned} \boldsymbol{H}\boldsymbol{e}^\top &= \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_{n-k} \end{pmatrix} (\overline{\boldsymbol{e}}_1^\top, \overline{\boldsymbol{e}}_2^\top, \ldots, \overline{\boldsymbol{e}}_r^\top) \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_r \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{h}_1\varepsilon_1 & \boldsymbol{h}_1\varepsilon_2 & \cdots & \boldsymbol{h}_1\varepsilon_r \\ \boldsymbol{h}_2\varepsilon_1 & \boldsymbol{h}_2\varepsilon_2 & \cdots & \boldsymbol{h}_2\varepsilon_r \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{h}_{n-k}\varepsilon_1 & \boldsymbol{h}_{n-k}\varepsilon_2 & \cdots & \boldsymbol{h}_{n-k}\varepsilon_r \end{pmatrix} \begin{pmatrix} \overline{\boldsymbol{e}}_1^\top \\ \overline{\boldsymbol{e}}_2^\top \\ \vdots \\ \overline{\boldsymbol{e}}_r^\top \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{n-k} \end{pmatrix}, \end{aligned} \tag{15}$$

where $\boldsymbol{h}_j$ is the j-th row of $\boldsymbol{H}$.
There are two methods to solve Equation (15):

1. **Solve-$\mathbb{F}_{q^m}$:** Obtaining a linear system with nr unknowns and $m(n-k)$ equations over $\mathbb{F}_q$ by expressing $\boldsymbol{h}_j\varepsilon_i$ and s_j as a matrix $\mathrm{Mat}(\boldsymbol{h}_j\varepsilon_i) \in \mathbb{F}_q^{m \times n}$ and column vector of length m, respectively, under the basis $\boldsymbol{\alpha}$. The system has one solution with overwhelming probability if $nr \leq m(n-k)$;

2. **Solve-EF:** As $\sum_{j=1}^{\ell} E_j F_j \subset EF$, where $F = \sum_{j=1}^{\ell} F_j$, the entries of $\boldsymbol{h}_j\varepsilon_i$ and s_j lie in EF. We then can express Equation (15) under the basis of EF by expressing $\boldsymbol{h}_j\varepsilon_i$ and s_j as a matrix of $rd \times n$ and column vector of length rd, respectively. Finally, we will obtain a linear system with nr unknowns and $rd(n-k)$ equations over $\mathbb{F}_q$. The system has one solution with overwhelming probability if $nr \leq rd(n-k)$, where $d = \sum_{j=1}^{\ell} d_j$ and $r = \sum_{j=1}^{\ell} r_j$.

Once all $\overline{\boldsymbol{e}}_i$'s are obtained, one can recover $\boldsymbol{e}$. We experimentally find that **Solve-$\mathbb{F}_{q^m}$** is more efficient than **Solve-EF** on SageMath 9.0.

Step 4: the fourth step of the algorithm is obvious.

4.4 The Decoding Complexity

The most costly part is the intersection in Step 2 and solving linear systems in Step 3. The intersection $\bigcap_{i=1}^{d_j} S_{ji}$ of spaces S_{ji} of dimension $\mu = \sum_{j=1}^{\ell} r_j d_j$ costs $\mathcal{O}\left(4\mu^2 m \sum_{j=1}^{\ell} d_j\right)$ operations in $\mathbb{F}_q$ for $j \in \{1..\ell\}$. By **Solve-EF**, expressing $\boldsymbol{h}_j\varepsilon_i$ as a matrix of $rd \times n$ in the basis of EF consists in solving n linear systems with rd unknowns and m equations. This costs $(n-k)nr^{\omega+1}d^{\omega}$ operations in $\mathbb{F}_q$. Expressing s_j as a column vector of length rd in the basis of EF consists in solving a linear system with rd unknowns and m equations. This costs $(n-k)(rd)^{\omega}$ operations in $\mathbb{F}_q$. Solving the linear system $\boldsymbol{H}\boldsymbol{e}^{\top} = \boldsymbol{s}$ with nr unknowns and $rd(n-k)$ equations costs about $\mathcal{O}((nr)^{\omega})$ operations in $\mathbb{F}_q$. Thus, the complexity of the decoding algorithm is bounded by $\mathcal{O}((nr)^{\omega})$.

4.5 Decoding Failure Probability

By the correctness assumption of Algorithm 1, two cases can make the algorithm fail: (i) $\dim S < \dim\left(\sum_{j=1}^{\ell} E_j F_j\right)$; (ii) $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ for $j \in \{1..\ell\}$. Propositions (4.2 and 4.3) estimate the probability of two cases.

Proposition 4.2 *The probability of* $\dim S < \dim\left(\sum_{j=1}^{\ell} E_j F_j\right)$ *is bounded by* $q^{-(n-k-\mu)}$ *where* $\mu = \sum_{j=1}^{\ell} r_j d_j$.

Proposition 4.3 *The probability that there exists* $j \in \{1..\ell\}$ *such that* $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ *is bounded by* $\sum_{j=1}^{\ell} q^{\mu - r_j}\left(\frac{q^{\mu-r_j}-1}{q^{m-r_j}}\right)^{d_j-1}$ *where* $\mu = \sum_{j=1}^{\ell} r_j d_j$.

We give the detailed proofs for Propositions (4.2 and 4.3) in Appendices (C.8 and C.9). Combining these two propositions, we deduce the decoding failure probability of Algorithm 1 in Theorem 4.4.

Theorem 4.4 *Under assumptions that* S_{ji} *behaves as independent and random subspaces containing* E_j*, the decoding failure probability of Algorithm 1 is bounded by* $q^{-(n-k-\mu)} + \sum_{j=1}^{\ell} q^{\mu - r_j}\left(\frac{q^{\mu-r_j}-1}{q^{m-r_j}}\right)^{d_j-1}$ *where* $\mu = \sum_{j=1}^{\ell} r_j d_j$.

The analysis shows that the failure probability can be made arbitrarily small.

4.6 Error Correction Capability

From the correctness of Algorithm 1, we have $nr \le rd(n-k) \Rightarrow d \ge \frac{n}{n-k}$. Under this condition, the decoding capacity is constrained by $\sum_{j=1}^{\ell} r_j d_j \le n-k$. The following Theorem 4.5 is obvious.

Theorem 4.5 *When $d_1 = d_2 = \cdots = d_\ell$, the ℓ-LRPC code allows to decode ℓ-errors of weight up to $r = \sum_{j=1}^{\ell} r_j = \frac{n-k}{d_1}$. By setting $d_1 = d_2 = \cdots = d_\ell = 2$, it can decode ℓ-errors of weight up to $\frac{n-k}{2}$.*

Theorem 4.5 implies that the decoding algorithm can benefit from the blockwise structure: the decoding capacity can be significantly improved by a factor of ℓ. An $[n,k]_{q^m}$ LRPC code defined by a parity-check matrix of weight d can decode the standard errors of weight up to $r = \frac{n-k}{d}$ with a DFR of about q^{rd-n-k}. Let $\ell | d$, $d_i = d/\ell$, $\boldsymbol{H} \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$ be a parity-check matrix of an $[n,k]_{q^m}$ ℓ-LRPC code. This ℓ-LRPC code can decode ℓ-errors in $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ of weight up to ℓr with the same DFR, which comes from

$$\sum_{j=1}^{\ell} r_j d_j = \frac{d}{\ell} \sum_{j=1}^{\ell} r_j \le n-k \implies \sum_{j=1}^{\ell} r_j \le \frac{\ell(n-k)}{d} = \ell r.$$

For example, fixing $d = 4$, $r = 8$, and the DFR of q^{32-n-k}, an $[n,k]_{q^m}$ LRPC code can decode errors of weight 8, but an $[n,k]_{q^m}$ 2-LRPC codes with parameter $\boldsymbol{d} = (d_1, d_2) = (2,2)$ can decode ℓ-errors with parameter $\boldsymbol{r} = (r_1, r_2) = (8,8)$ of weight up to $r = r_1 + r_2 = 16$.

For the accurate failure probability of decoding *maximal errors*, the theoretical value is hard to be estimated and the value in Theorem 4.4 seems not practical for $q > 2$. We give a simulation of the decoding algorithm for 2-LRPC codes on SageMath 9.0. When $\ell = 2$ and $d_1 = d_2 = 2$, the 2-LRPC codes can decode 2-errors of weight up to $\frac{n-k}{2}$. The simulated result shows that the failure probability is about 0.73 for $q = 2$. Figure 2 shows the decreasing trend of the failure probability as q increases. For $q = 2$, the failure probability is bounded by $q^{-\left(n-k-\sum_{j=1}^{2} r_j d_j\right)} = 1$. For $q > 2$, the upper bound of failure probability seems to be $q^{-\left(n-k+1-\sum_{j=1}^{2} r_j d_j\right)}$. The code parameters are $(m, n, k, n_1, n_2, r_1, r_2, d_1, d_2) = (43, 44, 22, 22, 22, 6, 5, 2, 2)$ for $q = 2, 3, 5, 7, 11, 13, 17, 19$.

4.7 The ℓ-RSR Algorithm

For cryptography applications in Section 5, one just recovers the support of the error. In this subsection, we give the ℓ-Rank Support Recover (ℓ-RSR) algorithm (Algorithm 2), which is a shortened version of the decoding Algorithm 1 without the computation of the error. The correctness follows Algorithm 1. The failure probability follows Theorem 4.4. The cost is only the recovery of support and is given in Subsection 4.4.

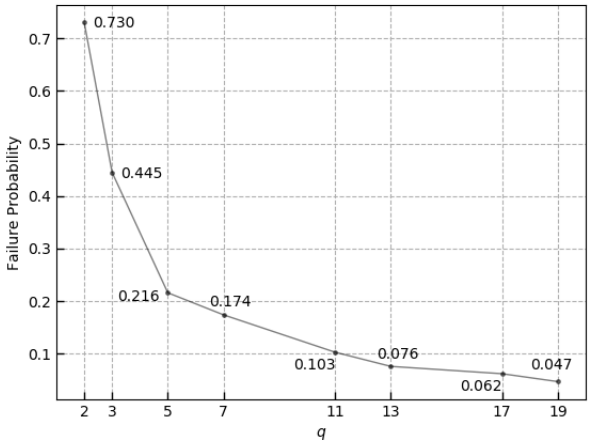

Fig. 2. Simulated failure probability of decoding 2-errors of the weight $\frac{n-k}{2}$ for 2-LRPC codes.

Algorithm 2 ℓ-RSR Algorithm

Input: a parity-check matrix $\boldsymbol{H} = (\boldsymbol{H}_1\ \boldsymbol{H}_2\ \cdots\ \boldsymbol{H}_\ell) \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$, a syndrome $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$, $\boldsymbol{r} = (r_1, r_2, \ldots, r_\ell)$.
Output: ℓ spaces E_j of dimensions r_j.

1: Compute the syndrome space $S = \langle s_1, s_2, \ldots, s_{n-k} \rangle_{\mathbb{F}_q}$.
2: Recovering the support E_j for $j \in \{1..\ell\}$:
   – Compute F_j from $\boldsymbol{H}_j$
   – Compute the basis $(f_{j1}, f_{j2}, \ldots, f_{jd_j})$ of F_j
   – Compute $S_{ji} = f_{ji}^{-1} S$, where all generators of S are multiplied by f_{ji}^{-1} for $i \in \{1..d_j\}$
   – Compute $E_j = \bigcap_{i=1}^{d_j} S_{ji}$

5 Applications to Cryptography

In this section, we apply the ideal variants of the ℓ-RD problem and the ℓ-LRPC codes to improve RQC [41] and ROLLO [40] kept in NIST PQC Round 2. Due to space limitations, we present the ideal variants in Appendix E and only list improved schemes and comparisons in this section.

RQC [41] and ROLLO [40] include Public Key Encryptions (PKE) and Key Encapsulation Mechanisms (KEM). RQC is an IND-CCA2 KEM built from its IND-CPA PKE construction based on the HHK transformation [36] and uses the Gabidulin codes. We only consider the PKE version of RQC for simplicity. ROLLO is the merge of the three cryptosystems Laker, Locker, and Ouroboros-R which all share the same decryption algorithm for the LRPC codes. Laker (ROLLO-I) and Ouroboros-R (ROLLO-III) are two IND-CPA KEM. Locker

(ROLLO-II) is an IND-CCA2 PKE scheme built from its IND-CPA PKE construction based on the HHK transformation [36]. We only consider the IND-CPA PKE version of Locker for simplicity.

5.1 Improved RQC

In this subsection, we improve RQC [41] based on the 2-IRSD and 3-IRSD problems. Our RQC uses three types of codes: a Gabidulin code $\mathcal{C}$ [26] with generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ which can correct up to $\lfloor \frac{n-k}{2} \rfloor$ errors by a deterministic decoding algorithm $\mathcal{C}.\mathsf{Decode}$ [38,7], a random $[2n, n]_{q^m}$-ideal code with parity-check matrix $(\boldsymbol{1} \;\; \boldsymbol{h})$, and a random $[3n, n]_{q^m}$-ideal code with parity-check matrix $\begin{pmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{h} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{s} \end{pmatrix}$.

- $\mathsf{RQC.KGen}(\lambda)$: Taking 1^λ as input, it randomly samples $\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_{q^m}^n$ and $(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(w_{\boldsymbol{x}}, w_{\boldsymbol{y}})}$, computes $\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{h}\boldsymbol{y}$, and sets the public key $pk = (\boldsymbol{h}, \boldsymbol{s})$ and the private key $sk = (\boldsymbol{x}, \boldsymbol{y})$.
- $\mathsf{RQC.Enc}(pk, \boldsymbol{m})$: Taking the public key $pk = (\boldsymbol{s}, \boldsymbol{h})$ and a message $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$ as input, it randomly samples $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e}) \xleftarrow{\$} \mathcal{S}^{(n,n,n)}_{(w_{\boldsymbol{r}_1}, w_{\boldsymbol{r}_2}, w_{\boldsymbol{e}})}$, computes $\boldsymbol{u} = \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2$ and $\boldsymbol{v} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{e}$, and returns the ciphertext $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$.
- $\mathsf{RQC.Dec}(sk, \boldsymbol{c})$: Taking a private key $sk = (\boldsymbol{x}, \boldsymbol{y})$ and the ciphertext $\boldsymbol{c}$ as input, it computes $\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y}$ and returns $\boldsymbol{m} \leftarrow \mathcal{C}.\mathsf{Decode}(\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y})$.

Fig. 3. Description of our RQC PKE scheme.

Correctness. We have $\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{x}\boldsymbol{r}_2 + \boldsymbol{e} - \boldsymbol{r}_1\boldsymbol{y}$. The correctness of our encryption scheme is based on the decoding capability of the Gabidulin code $\mathcal{C}$, i.e., the error term $\boldsymbol{x}\boldsymbol{r}_2 + \boldsymbol{e} - \boldsymbol{r}_1\boldsymbol{y}$ must fulfill: $\|\boldsymbol{x}\boldsymbol{r}_2 + \boldsymbol{e} - \boldsymbol{r}_1\boldsymbol{y}\|_{\mathrm{R}} = w_{\boldsymbol{x}}w_{\boldsymbol{r}_2} + w_{\boldsymbol{y}}w_{\boldsymbol{r}_1} + w_{\boldsymbol{e}} \leq \lfloor \frac{n-k}{2} \rfloor$.

In the decryption step, one needs to decode an error of weight $w_{\boldsymbol{x}}w_{\boldsymbol{r}_2} + w_{\boldsymbol{y}}w_{\boldsymbol{r}_1} + w_{\boldsymbol{e}}$. This weight increase is slow, which brings the gain of decoding capacity and saves code parameters. Although the ℓ-errors can also be used to speed up the attacks for decoding problems, the performance analysis in Table 3 shows that the gain in the decoding method greatly outweighs the gain in the attacks, and eventually allows scheme with small parameters.

Theorem 5.1 *Under the decisional 2-IRSD and 3-IRSD problems, our RQC PKE in Figure 3 is IND-CPA secure.*

Proof. The proof is similar to [41] with 2-IRSD and 3-IRSD instances. The two instances are defined by

$$\boldsymbol{s} = \begin{pmatrix} \boldsymbol{1} & \boldsymbol{h} \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} - \boldsymbol{m}\boldsymbol{G} \end{pmatrix} = \begin{pmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{h} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{s} \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ \boldsymbol{e} \end{pmatrix}.$$

□

5.2 Improved Lake (ROLLO-I)

In this subsection, we improve Lake based on the 2-IRSD problem and the 2-ILRPC codes indistinguishability problem. Our Laker has three building blocks: a random $[2n, n]_{q^m}$ 2-ILRPC code with parity-check matrix $(\boldsymbol{x} \ \ \boldsymbol{y})$, the algorithm 2-RSR (see Algorithm 2), and a random $[2n, n]_{q^m}$-ideal code with parity-check matrix $(\mathbf{1} \ \ \boldsymbol{h})$.

- Lake.KGen(λ): Taking 1^λ as input, it samples $(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(d_1,d_2)}$ and computes $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y}$, then it sets the public key $pk = \boldsymbol{h}$ and the private key $sk = (\boldsymbol{x}, \boldsymbol{y})$.
- Lake.Encap(pk): Taking the public key $\boldsymbol{h}$ as input, it randomly chooses $(\boldsymbol{e}_1, \boldsymbol{e}_2) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(r_1,r_2)}$ and computes $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$, $E_1 = \mathrm{Supp}(\boldsymbol{e}_1)$, $E_2 = \mathrm{Supp}(\boldsymbol{e}_2)$, $E = E_1 + E_2$, and $K = \mathrm{Hash}(E)$, and returns $(\boldsymbol{c}, K)$.
- Lake.Decap($sk, \boldsymbol{c}$): Taking $(\boldsymbol{x}, \boldsymbol{y})$ and $\boldsymbol{c}$ as input, it computes $\boldsymbol{x}\boldsymbol{c} = \boldsymbol{x}\boldsymbol{e}_1 + \boldsymbol{y}\boldsymbol{e}_2$, executes $(E_1, E_2) \leftarrow$ 2-RSR$((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}\boldsymbol{c}, r_1, r_2)$, computes $E = E_1 + E_2$, and returns $K = \mathrm{Hash}(E)$.

Fig. 4. Description of our Lake KEM scheme.

5.3 Improved Locker (ROLLO-II)

Locker (ROLLO-II [40]) is a PKE scheme and is obtained from ROLLO-I. In this subsection, we improve ROLLO-II by the 2-IRSD problem. As our Lake, our Locker has three building blocks: a random $[2n, n]_{q^m}$ 2-ILRPC code with parity-check matrix $(\boldsymbol{x} \ \ \boldsymbol{y})$, the algorithm 2-RSR (see Algorithm 2), and a random $[2n, n]_{q^m}$-ideal code with parity-check matrix $(\mathbf{1} \ \ \boldsymbol{h})$.

- Locker.KGen(λ): Taking 1^λ as input, it samples $(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(d_1,d_2)}$ and computes $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y}$, then it sets the public key $pk = \boldsymbol{h}$ and the private key $sk = (\boldsymbol{x}, \boldsymbol{y})$.
- Locker.Enc(pk, M): Taking the public key $\boldsymbol{h}$ and a message M as input, it randomly chooses $(\boldsymbol{e}_1, \boldsymbol{e}_2) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(r_1,r_2)}$, computes $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$, $E_1 = \mathrm{Supp}(\boldsymbol{e}_1)$, $E_2 = \mathrm{Supp}(\boldsymbol{e}_2)$, $E = E_1 + E_2$, and the ciphertext $C = (\boldsymbol{c}, M \oplus \mathrm{Hash}(E)) = (\boldsymbol{c}, \boldsymbol{c}')$, and returns C.
- Locker.Dec(sk, C): Taking the private key $(\boldsymbol{x}, \boldsymbol{y})$ and the ciphertext C as input, it computes $\boldsymbol{x}\boldsymbol{c} = \boldsymbol{x}\boldsymbol{e}_1 + \boldsymbol{y}\boldsymbol{e}_2$, executes $(E_1, E_2) \leftarrow$ 2-RSR$((\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}\boldsymbol{c}, r_1, r_2)$, computes $E = E_1 + E_2$, and returns $M = \boldsymbol{c}' \oplus \mathrm{Hash}(E)$.

Fig. 5. Description of our Locker PKE scheme.

In Laker and Locker, the decapsulation and decryption steps obtain the support of $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ from $\boldsymbol{x}\boldsymbol{e}_1 - \boldsymbol{y}\boldsymbol{e}_2$ of weight $r_1d_1 + r_2d_2$. This weight increase implies that the parameters (r_1, r_2) and (d_1, d_2) can be increased a lot. Although the 2-errors and the 2-LRPC codes can also be used to speed up the attacks for decoding problems, the performance analysis in Table (4, 5, 7) shows that the gain in the decoding method outweighs the gain in the attacks, and eventually allows schemes with small parameters.

Theorem 5.2 *Under the* 2*-ILRPC codes indistinguishability, and* 2*-IRSR problems our Lake KEM in Figure 4 and Locker PKE in Figure 5 are IND-CPA secure in the random oracle model.*

Proof. The proofs are similar to [40] with the 2-ILRPC codes indistinguishability and 2-IRSR instances. The two instances are defined by

$$\mathbf{0} = \begin{pmatrix} \mathbf{1} & \boldsymbol{h} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ -\boldsymbol{x} \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} \mathbf{1} & \boldsymbol{h} \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix}.$$

□

5.4 Improved Ouroboros-R (ROLLO-III)

In this subsection, we improve ROLLO-III based on the 2-IRSD and 3-IRSD problems. Our Ouroboros-R has three building blocks: a 3-ILRPC code with parity-check matrix $(\boldsymbol{h}_0\ \boldsymbol{h}_1\ \mathbf{1})$, the algorithm 3-RSR (see Algorithm 2), a $[2n, n]_{q^m}$-ideal code with parity-check matrix $(\mathbf{1}\ \boldsymbol{f}_1)$, and a $[3n, n]_{q^m}$-ideal code with parity-check matrix $\begin{pmatrix} \mathbf{1} & \mathbf{0} & \boldsymbol{f}_0 \\ \mathbf{0} & \mathbf{1} & \boldsymbol{f}_1 \end{pmatrix}$.

- Ouroboros-R.KGen(λ): Taking 1^λ as input, it samples $\boldsymbol{f}_1 \xleftarrow{\text{seed}} \mathbb{F}_{q^m}^n$, and $(\boldsymbol{h}_0, \boldsymbol{h}_1) \xleftarrow{\$} \mathcal{S}_{(d_1,d_2)}^{(n,n)}$, then it computes $\boldsymbol{f}_0 = \boldsymbol{h}_1 + \boldsymbol{f}_1\boldsymbol{h}_0$ and sets the public key $pk = (\boldsymbol{f}_0, \text{seed})$ and the private key $sk = (\boldsymbol{h}_0, \boldsymbol{h}_1)$.
- Ouroboros-R.Encap(pk): Taking the public key $(\boldsymbol{f}_0, \text{seed})$ as input, it randomly chooses $(\boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}) \xleftarrow{\$} \mathcal{S}_{(r_1,r_2,r_3)}^{(n,n,n)}$, computes $\boldsymbol{c}_0 = \boldsymbol{f}_0\boldsymbol{e}_1 + \boldsymbol{e}$, $\boldsymbol{c}_1 = \boldsymbol{f}_1\boldsymbol{e}_1 + \boldsymbol{e}_0$, $E_1 = \text{Supp}(\boldsymbol{e}_1)$, $E_2 = \text{Supp}(\boldsymbol{e}_2)$, $E = E_1 + E_2$, and $K = \text{Hash}(E)$, sets $\boldsymbol{c} = (\boldsymbol{c}_0, \boldsymbol{c}_1)$, and returns $(\boldsymbol{c}, K)$.
- Ouroboros-R.Decap($sk, \boldsymbol{c}$): Taking $(\boldsymbol{h}_0, \boldsymbol{h}_1)$ and $\boldsymbol{c}$ as input, it computes $\boldsymbol{s} = \boldsymbol{c}_0 - \boldsymbol{h}_0\boldsymbol{c}_1 = -\boldsymbol{h}_0\boldsymbol{e}_0 + \boldsymbol{h}_1\boldsymbol{e}_1 + \boldsymbol{e}$, executes $(E_1, E_2) \leftarrow \text{3-RSR}((\boldsymbol{h}_0, \boldsymbol{h}_1, \mathbf{1}), \boldsymbol{s}, r_1, r_2, r_3)$, computes $E = E_1 + E_2$, and returns $K = \text{Hash}(E)$.

Fig. 6. Description of our Ouroboros-R KEM scheme.

In the decapsulation step, one obtains the support of $(\boldsymbol{e}_0, \boldsymbol{e}_1)$ from $\boldsymbol{h}_1\boldsymbol{e}_1 - \boldsymbol{h}_0\boldsymbol{e}_0 + \boldsymbol{e}$ of weight $r_1d_1 + r_2d_2 + r_3$. This weight increasing implies that the parameters (r_1, r_2, r_3) and (d_1, d_2) can be increased a lot. Although the blockwise

errors and LRPC codes can also be used to speed up the attacks for decoding problems, the performance analysis in Table (6, 7) shows that the gain in the decoding method outweighs the gain in the attacks, and eventually allows scheme with small parameters.

Theorem 5.3 *Under the decisional* 2*-IRSD and* 3*-IRSD problems, our Ouroboros-R KEM in Figure 6 is IND-CPA secure in the random oracle model.*

Proof. The proof is similar to [3] with the (decisional) 2-IRSD and 3-IRSD instances. The two instances are defined by

$$\boldsymbol{f}_0 = \begin{pmatrix} \mathbf{1} & \boldsymbol{f}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_0 \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \boldsymbol{f}_0 \\ \mathbf{0} & \mathbf{1} & \boldsymbol{f}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{e} \\ \boldsymbol{e}_0 \\ \boldsymbol{e}_1 \end{pmatrix}.$$

□

5.5 Performance and Comparison

In this subsection, we compare performance of our RQC and ROLLO with original versions.

In Table 3 - 5, parameters are chosen in two principles. First, the hardness of decoding problems (the 2-IRSD and 3-IRSD problems) is ensured to reach the target security level. The hardness is estimated by our complexity formulas. Secondly, the error-correcting capacity of rank metric codes is ensured to satisfy the decryption correctness condition. $[n,k]_{q^m}$ Gabidulin codes used in RQC require $k < n \leq m$ and correct errors of weight up to $\lfloor (n-k)/2 \rfloor$; in the decryption step, the weight of the decoded errors must $\leq \lfloor (n-k)/2 \rfloor$. The ℓ-LRPC codes used in ROLLO must satisfy a reasonable DFR in Theorem 4.4. In Table 3 and Table 6, "$2n$" ("$3n$") represents the complexity of solving the 2-IRSD (3-IRSD) instances in RQC and Ouroboros-R. In Table 4 and Table 5, the structural attack is estimated with parameters $(m,n,k,r_1,r_2) = \left(m, 2n - \lfloor \frac{n}{d} \rfloor, n - \lfloor \frac{n}{d} \rfloor, d_1.d_2\right)$; the message attack is estimated with parameters $(m,n,k,r_1,r_2) = (m,2n,n,r_1,r_2)$.

From Table 3 - 6, our parameters sizes are smaller than those of the original ones due to the blockwise stricture, which brings a low complexity redundancy, improved the public key/ciphertext sizes, and more efficient implementations. The improved performance benefits from that the gain of using ℓ-errors and ℓ-LRPC codes in decoding capacity outweighs the complexity loss in solving the ℓ-RD problem. As an example, we provide concrete timings of implementations for our ROLLO and original versions (Table 7). The benchmark is performed on Intel(R) Core(TM) i5-7440HQ CPU@ 3.40 GHz with SageMath 9.0. The tests are available online at https://github.com/YCSong232431/NH-ROLLO. Note that, we do not compare with most recent works [15,43], where the authors constructed a series of efficient PKE and KEM schemes without ideal structure by proposing augmented Gabidulin codes and LRPC codes with multiple syndromes. Our techniques are different from [15,43] and we only consider cryptosystems with the ideal structure and only one syndrome.

Table 3. Comparison of parameters and sizes for RQC.

Schemes	m	n	k	$w_{\boldsymbol{x}}$	$w_{\boldsymbol{y}}$	$w_{\boldsymbol{r}_1}$	$w_{\boldsymbol{r}_2}$	$w_{\boldsymbol{e}}$	pks (bytes)	cts (bytes)	total (KB)	Attack $(2n, 3n)$	Security
Our RQC	83	79	7	4	4	4	4	4	860	1704	2.5	$(2^{130}, 2^{163})$	128
Our RQC	127	113	3	5	5	5	5	5	1834	3652	5.3	$(2^{258}, 2^{214})$	192
Our RQC	139	137	5	5	5	6	6	6	2421	4826	7.1	$(2^{271}, 2^{274})$	256

Schemes	m	n	k	$w_{\boldsymbol{x}}$	$w_{\boldsymbol{y}}$	$w_{\boldsymbol{r}_1}$	$w_{\boldsymbol{r}_2}$	$w_{\boldsymbol{e}}$	pks (bytes)	cts (bytes)	total (KB)	Security
RQC (NIST [41])	127	113	3	7	7	7	7	13	1834	3652	5.3	128
RQC (NIST [41])	151	149	5	8	8	8	8	16	2853	5690	8.3	192
RQC (NIST [41])	181	179	3	9	9	9	9	16	4090	8164	12.0	256

pks: $\left(\lceil \frac{mn}{8} \rceil + 40\right)$ bytes; cts: $\left(2\lceil \frac{mn}{8} \rceil + 64\right)$ bytes; total = pks + cts.

Table 4. Comparison of parameters and sizes for Lake (ROLLO-I).

Schemes	m	n	r_1	r_2	d_1	d_2	DFR	pks/cts (bytes)	Structural attack $\boldsymbol{y} - \boldsymbol{x}\boldsymbol{h} = \boldsymbol{0}$	Message attack $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$	Security
Our Lake	61	67	4	4	5	4	2^{-31}	511	2^{160}	2^{144}	128
Our Lake	71	79	5	5	5	5	2^{-29}	702	2^{225}	2^{255}	192
Our Lake	79	89	5	5	6	5	2^{-34}	879	2^{281}	2^{266}	256

Schemes	m	n	r	d	DFR	pks/cts (bytes)	Security
Lake (NIST [40])	67	83	7	8	2^{-28}	696	128
Lake (NIST [40])	79	97	8	8	2^{-34}	958	192
Lake (NIST [40])	97	113	9	9	2^{-33}	1371	256

pks: $\lceil \frac{mn}{8} \rceil$ bytes. cts: $\lceil \frac{mn}{8} \rceil$ bytes.

Table 5. Comparison of parameters and sizes for Locker (ROLLO-II).

Schemes	m	n	r_1	r_2	d_1	d_2	DFR	pks (bytes)	cts (bytes)	Structural attack $\boldsymbol{y} - \boldsymbol{x}\boldsymbol{h} = \boldsymbol{0}$	Message attack $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$	Security
Our Locker	89	163	4	4	4	4	2^{-131}	1814	1942	2^{134}	2^{139}	128
Our Locker	97	179	4	5	5	5	2^{-134}	2171	2299	2^{254}	2^{231}	192
Our Locker	101	181	5	5	5	5	2^{-131}	2286	2414	2^{267}	2^{357}	256

Schemes	m	n	r	d	DFR	pks (bytes)	cts (bytes)	Security
Locker (NIST [40])	83	189	7	8	2^{-134}	1941	2089	128
Locker (NIST [40])	97	193	8	8	2^{-130}	2341	2469	192
Locker (NIST [40])	97	211	8	9	2^{-136}	2559	2687	256

pks: $\lceil \frac{mn}{8} \rceil$ bytes; cts: $\lceil \frac{mn}{8} \rceil + 64$ bytes. To obtain the IND-CCA2 security, another hash is added to the ciphertext such that cts $= \lceil \frac{mn}{8} \rceil + 2 * 64$ bytes.

6 Conclusion and Future Work

In this paper, we studied blockwise structures in rank-based cryptosystems and introduced ℓ-errors, ℓ-RD problem, and ℓ-LRPC codes. They are natural generalizations of the standard errors, RD problem, and LRPC codes. We found that

Table 6. Comparison of parameters and sizes for Ouroboros-R (ROLLO-III).

Schemes	m	n	r_1	r_2	r_3	d_1	d_2	DFR	pks (bytes)	cts (bytes)	Attacks $(2n, 3n)$	Security
Our Ouroboros-R	53	79	4	4	5	4	4	2^{-33}	623	1166	$(2^{147}, 2^{175})$	128
Our Ouroboros-R	89	101	6	6	6	4	5	2^{-33}	1164	2248	$(2^{196}, 2^{266})$	192
Our Ouroboros-R	97	103	6	6	7	5	5	2^{-42}	1362	2644	$(2^{275}, 2^{308})$	256

Schemes	m	n	w	w_r	δ	DFR	pks (bytes)	cts (bytes)	Security
Ouroboros-R (TIT [3])	67	83	7	7	7	2^{-28}	736	1431	128
Ouroboros-R (TIT [3])	107	113	9	9	9	2^{-24}	1552	3023	192
Ouroboros-R (TIT [3])	149	151	11	11	11	2^{-20}	2853	5625	256

pks: $\left(\lceil \frac{mn}{8} \rceil + 40\right)$ bytes and cts: $\lceil \frac{2mn}{8} \rceil$ bytes. We update DFR of Ouroboros-R.

Table 7. Timings comparisons of our ROLLO and original ROLLO.

Schemes	KGen (ms)	Encap (ms)	Decap (ms)	Security
Our Lake	715	73	257	128
Our Lake	737	100	499	192
Our Lake	1020	118	553	256
Lake (NIST [40])	995	109	391	128
Lake (NIST [40])	1220	134	525	192
Lake (NIST [40])	1390	181	838	256

Schemes	KGen (ms)	Enc (ms)	Dec (ms)	Security
Our Locker	2300	232	388	128
Our Locker	2940	280	614	192
Our Locker	3210	301	644	256
Locker (NIST [40])	2760	258	446	128
Locker (NIST [40])	3410	314	583	192
Locker (NIST [40])	2780	333	715	256

Schemes	KGen (ms)	Encap (ms)	Decap (ms)	Security
Our Ouroboros-R	101	120	246	128
Our Ouroboros-R	206	247	633	192
Our Ouroboros-R	224	262	798	256
Ouroboros-R (TIT [3])	130	153	368	128
Ouroboros-R (TIT [3])	275	308	1040	192
Ouroboros-R (TIT [3])	504	614	2560	256

(1) the blockwise structures do not ease the problem too much: the ℓ-RD problem is still exponentially hard for appropriate choices of $\ell > 1$; (2) the decoding algorithm can benefit from the blockwise structures: the decoding capacity can be significantly improved by a factor of ℓ. Interestingly, the gain of the decoding capacity outweighs the complexity loss in solving the ℓ-RD problem, which allows to improve RQC and ROLLO. For 128-bit security, our RQC has total public key and ciphertext sizes of 2.5 KB, which is not only about 50% more compact than the original RQC, but also smaller than the NIST Round 4 code-based submissions HQC, BIKE, and Classic McEliece.

Recent works [15,43,4] proposed unstructured PKE and KEM without ideal structure for more reliable security. We would in next work analyze the complexity of blockwise rank support learning problem and apply the ℓ-LRPC codes with multiple syndromes to improve unstructured schemes.

Acknowledgement

We would like to thank the anonymous reviewers of ASIACRYPT 2023 for their helpful comments and suggestions on earlier versions of our paper. Jiang Zhang, the corresponding author, is supported by the National Key Research and Development Program of China (Grant No. 2022YFB2702000), and by the National Natural Science Foundation of China (Grant Nos. 62022018, 61932019). Xinyi Huang is supported by the National Natural Science Foundation of China (Grant No. 62032005). Wei Wu is supported by the National Natural Science Foundation of China (Grant No. 62372108). This research is also funded in part by the National Natural Science Foundation of China (Grant No. 62172096).

References

1. Alekhnovich, M.: More on average case vs approximation complexity. In: Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS). pp. 298–307. IEEE Computer Society (2003)
2. Aragon, N., Barreto, P., Bettaieb, S., et al.: BIKE. Fourth round submission to the NIST post-quantum cryptography call (2022), `https://bikesuite.org/`
3. Aragon, N., Blazy, O., Deneuville, J., Gaborit, P., Zémor, G.: Ouroboros: An efficient and provably secure KEM family. IEEE Transactions on Information Theory **68**(9), 6233–6244 (2022)
4. Aragon, N., Dyseryn, V., Gaborit, P., Loidreau, P., Renner, J., Wachter-Zeh, A.: LowMS: A new rank metric code-based KEM without ideal structure. IACR Cryptology ePrint Archive p. 1596 (2022), `https://eprint.iacr.org/2022/1596`
5. Aragon, N., Gaborit, P., Hauteville, A., Ruatta, O., Zémor, G.: Low rank parity check codes: new decoding algorithms and applications to cryptography. IEEE Transactions on Information Theory **65**(12), 7697–7717 (2019)
6. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.: A new algorithm for solving the rank syndrome decoding problem. In: International Symposium on Information Theory (ISIT). pp. 2421–2425. IEEE (2018)
7. Augot, D., Loidreau, P., Robert, G.: Generalized Gabidulin codes over fields of any characteristic. Designs, Codes and Cryptography **86**(8), 1807–1848 (2018)
8. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Neiger, V., Ruatta, O., Tillich, J.: An algebraic attack on rank metric code-based cryptosystems. In: Advances in Cryptology - EUROCRYPT. vol. 12107, pp. 64–93. Springer (2020)
9. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Tillich, J.: Revisiting algebraic attacks on MinRank and on the rank decoding problem. IACR Cryptology ePrint Archive p. 1031 (2022), `https://eprint.iacr.org/2022/1031`
10. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R.A., Smith-Tone, D., Tillich, J., Verbel, J.A.: Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Advances in Cryptology - ASIACRYPT. vol. 12491, pp. 507–536. Springer (2020)

11. Bardet, M., Faugère, J.C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: International Conference on Polynomial System Solving. pp. 71–75. Springer (2004)
12. Bernstein, D.J., Chou, T., Cid, C., et al.: Classic McEliece. Fourth round submission to the NIST post-quantum cryptography call (2022), https://classic.mceliece.org/
13. Bettaieb, S., Bidoux, L., Connan, Y., Gaborit, P., Hauteville, A.: The Learning with Rank Errors problem and an application to symmetric authentication. In: International Symposium on Information Theory, ISIT. pp. 2629–2633. IEEE (2018)
14. Bettale, L., Faugère, J., Perret, L.: Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. Designs, Codes and Cryptography **69**(1), 1–52 (2013)
15. Bidoux, L., Briaud, P., Bros, M., Gaborit, P.: RQC revisited and more cryptanalysis for rank-based cryptography. CoRR (2022), https://doi.org/10.48550/arXiv.2207.01410
16. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal. Ph. D. Thesis, Universitat Innsbruck, Austria (1965)
17. Buchberger, B., Collins, G.E., Loos, R., Albrecht, R.: Computer algebra symbolic and algebraic computation. ACM Communications in Computer Algebra (ACM SIGSAM Bull) **16**(4), 5 (1982)
18. Byrne, E., Gluesing-Luerssen, H., Ravagnani, A.: Fundamental properties of sum-rank-metric codes. IEEE Transactions on Information Theory **67**(10), 6456–6475 (2021)
19. Chabaud, F., Stern, J.: The cryptographic security of the syndrome decoding problem for rank distance codes. In: Advances in Cryptology - ASIACRYPT. vol. 1163, pp. 368–381. Springer (1996)
20. Coggia, D., Couvreur, A.: On the security of a Loidreau rank metric code based encryption scheme. Designs, Codes and Cryptography **88**(9), 1941–1957 (2020)
21. Couvreur, A., Gaborit, P., Gauthier-Umaña, V., Otmani, A., Tillich, J.: Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes. Designs, Codes and Cryptography **73**(2), 641–666 (2014)
22. Faugère, J.C.: A new efficient algorithm for computing Gröbner basis (F_4). Journal of Pure and Applied Algebra **139**(1-3), 61–88 (1999)
23. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In: International Symposium on Symbolic and Algebraic Computation. pp. 75–83. ACM (2002)
24. Faugère, J., Din, M.S.E., Spaenlehauer, P.: Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. In: International Symposium on Symbolic and Algebraic Computation. pp. 257–264. ACM (2010)
25. Faugère, J., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of MinRank. In: Advances in Cryptology - CRYPTO. vol. 5157, pp. 280–296. Springer (2008)
26. Gabidulin, E.M.: Theory of codes with maximum rank distance. Problemy peredachi informatsii **21**(1), 3–16 (1985)
27. Gabidulin, E.M., Paramonov, A.V., Tretjakov, O.V.: Ideals over a non-commutative ring and thier applications in cryptology. In: Advances in Cryptology - EUROCRYPT. vol. 547, pp. 482–489. Springer (1991)
28. Gaborit, P., Murat, G., Ruatta, O., Zémor, G.: Low rank parity check codes and their application to cryptography. In: The Workshop on Coding and Cryptography (WCC). http://www.selmer.uib.no/WCC2013/pdfs/Gaborit.pdf

29. Gaborit, P., Ruatta, O., Schrek, J.: On the complexity of the rank syndrome decoding problem. IEEE Transactions on Information Theory **62**(2), 1006–1019 (2016)
30. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: New results for rank-based cryptography. In: Progress in Cryptology - AFRICACRYPT. vol. 8469, pp. 1–12. Springer (2014)
31. Gaborit, P., Zémor, G.: On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Transactions on Information Theory **62**(12), 7245–7252 (2016)
32. Ghatak, A.: Extending coggia-couvreur attack on Loidreau's rank-metric cryptosystem. Designs, Codes and Cryptography **90**(1), 215–238 (2022)
33. Goubin, L., Courtois, N.T.: Cryptanalysis of the TTM cryptosystem. In: Advances in Cryptology - ASIACRYPT. vol. 1976, pp. 44–57. Springer (2000)
34. Hauteville, A., Tillich, J.: New algorithms for decoding in the rank metric and an attack on the LRPC cryptosystem. In: IEEE International Symposium on Information Theory (ISIT). pp. 2747–2751. IEEE (2015)
35. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Third International Symposium on Algorithmic Number Theory (ANTS-III). vol. 1423, pp. 267–288. Springer (1998)
36. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Theory of Cryptography Conference (TCC). vol. 10677, pp. 341–371. Springer (2017)
37. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Advances in Cryptology - CRYPTO. vol. 1666, pp. 19–30. Springer (1999)
38. Loidreau, P.: A Welch-Berlekamp like algorithm for decoding Gabidulin codes. In: International Workshop on Coding and Cryptography (WCC). vol. 3969, pp. 36–45. Springer (2005)
39. Loidreau, P.: A new rank metric codes based encryption scheme. In: Post-Quantum Cryptography (PQCrypto). vol. 10346, pp. 3–17. Springer (2017)
40. Melchor, C.A., Aragon, N., Bardet, M., et al.: ROLLO. Second round submission to the NIST post-quantum cryptography call (2020), `https://pqc-rollo.org/`
41. Melchor, C.A., Aragon, N., Bettaieb, S., et al.: RQC. Second round submission to the NIST post-quantum cryptography call (2020), `http://pqc-rqc.org/`
42. Melchor, C.A., Aragon, N., Bettaieb, S., et al.: HQC. Fourth Round Submission to the NIST Post-quantum Cryptography Call (2023), `http://pqc-hqc.org`
43. Melchor, C.A., Aragon, N., Dyseryn, V., Gaborit, P., Zémor, G.: LRPC codes with multiple syndromes: Near ideal-size KEMs without ideals. In: Post-Quantum Cryptography (PQCrypto). vol. 13512, pp. 45–68. Springer (2022)
44. Misoczki, R., Tillich, J., Sendrier, N., Barreto, P.S.L.M.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In: IEEE International Symposium on Information Theory (ISIT). pp. 2069–2073. IEEE (2013)
45. NIST: Status report on the second round of the NIST post-quantum cryptography standardization process (2020), `https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf`
46. NIST: Status report on the third round of the NIST post-quantum cryptography standardization process (2022), `https://doi.org/10.6028/NIST.IR.8413-upd1`
47. Ore, O.: On a special class of polynomials. Transactions of the American Mathematical Society **35**(3), 559–584 (1933)
48. Ourivski, A.V., Johansson, T.: New technique for decoding codes in the rank metric and its cryptography applications. Problems of Information Transmission **38**(3), 237–246 (2002)

49. Overbeck, R.: A new structural attack for GPT and variants. In: The First International Conference on Cryptology in Malaysia (Mycrypt). vol. 3715, pp. 50–63. Springer (2005)
50. Sidelnikov, V.M., Shestakov, S.O.: On insecurity of cryptosystems based on generalized Reed-Solomon codes. Discrete Applied Mathematics **2**(4), 439–444 (1992)

A. Preliminaries

A.1 $\mathbb{F}_{q^m}$-Linear Codes with Rank Metric

Definition A.1 (Rank Metric). *Let $\boldsymbol{\alpha}$ be a basis of $\mathbb{F}_{q^m}$ viewed as an m-dimensional vector space over $\mathbb{F}_q$. For $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_{q^m}^n$, each coordinate x_i is associated to a vector of $\mathbb{F}_q^m$ w.r.t. the basis $\boldsymbol{\alpha}$. Then $\boldsymbol{x}$ is associated to an $m \times n$ matrix given by* $\mathrm{Mat}(\boldsymbol{x}) = (x_{ij})_{\substack{i\in\{1..m\}\\ j\in\{1..n\}}}$*: $\boldsymbol{x} = \boldsymbol{\alpha}\mathrm{Mat}(\boldsymbol{x})$* The rank weight $\|\boldsymbol{x}\|_{\mathrm{R}}$ *of $\boldsymbol{x}$ is defined as the rank of* $\mathrm{Mat}(\boldsymbol{x})$*. The rank distance* $\mathrm{d}_{\mathrm{R}}(\boldsymbol{x}, \boldsymbol{y})$ *between elements $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{F}_{q^m}^n$ is defined by* $\mathrm{d}_{\mathrm{R}}(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\|_{\mathrm{R}}$.

The support $\mathrm{Supp}(\boldsymbol{x})$ of $\boldsymbol{x}$ is defined as the $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ generated by linear combinations over $\mathbb{F}_q$ of coordinates of $\boldsymbol{x}$, i.e., $\mathrm{Supp}(\boldsymbol{x}) = \langle x_1, x_2, \ldots, x_n\rangle_{\mathbb{F}_q}$. It follows from definition that $\|\boldsymbol{x}\|_{\mathrm{R}} = \dim(\mathrm{Supp}(\boldsymbol{x}))$. The set of such errors of weight r and length n is denoted by $\mathcal{S}_r^n$. The weight and support definitions can also be extended to matrices.

Definition A.2 ($\mathbb{F}_{q^m}$-Linear Codes with Rank Metric). *An $\mathbb{F}_{q^m}$-linear code embedded with rank metric of length n and dimension k is a subspace of dimension k of $\mathbb{F}_{q^m}^n$. Such $\mathbb{F}_{q^m}$-linear codes are denoted by $[n, k]_{q^m}$.*

Given an $[n, k]_{q^m}$-linear code $\mathcal{C}$, a matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ is called *generator matrix* iff $\mathcal{C} = \{\boldsymbol{mG} \mid \boldsymbol{m} \in \mathbb{F}_{q^m}^k\}$ and a matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is called *parity-check matrix* iff $\mathcal{C} = \{\boldsymbol{c} \in \mathbb{F}_{q^m}^n \mid \boldsymbol{Hc}^\top = \boldsymbol{0}\}$. The systematic forms of $\boldsymbol{G}$ and $\boldsymbol{H}$ are respectively defined as $(\boldsymbol{I}_k \ \boldsymbol{U})$ and $(-\boldsymbol{U}^\top \ \boldsymbol{I}_{n-k})$ where $\boldsymbol{U} \in \mathbb{F}_{q^m}^{k\times(n-k)}$.

An $\mathbb{F}_{q^m}$-linear code in the rank metric is closely related to a matrix code. The reason for considering $\mathbb{F}_{q^m}$-linear codes instead of matrix codes in rank-based cryptography is that $[n, k]_{q^m}$-linear codes can compress the description of $[m \times n, km]$-matrix codes and bring a small key size. An $[m \times n, K]$-matrix code of size $m \times n$ and dimension K is a subspace of dimension K of $\mathbb{F}_q^{m\times n}$, and the weight of the codeword (a matrix of size $m \times n$) is defined as the rank of a matrix. An $[n, k]_{q^m}$-linear code can be viewed as an $[m \times n, km]$-matrix code by expressing its codeword as a matrix of size $m \times n$. The latter can be defined by a systematic generator matrix $(\boldsymbol{I}_{km} \ \boldsymbol{P}_1)$ of size $km \times mn$ over $\mathbb{F}_q$ that is stored in $km(nm - km)\log q = k(n - k)m^2 \log q$ bits, whereas the $[n, k]_{q^m}$-linear code with systematic $\boldsymbol{G} = (\boldsymbol{I}_k \ \boldsymbol{P}_2)$ size $k \times n$ over $\mathbb{F}_{q^m}$ is stored in only $k(n - k)\log q^m = k(n - k)m \log q$ bits.

A.2 q-Polynomials, Gröbner Basis, and Degree of Regularity

Definition A.3 (q-Polynomials [47]). *A q-polynomial of q-degree r in $\mathbb{F}_{q^m}$ is a polynomial of the form $P(x) = \sum_{i=0}^{r} p_i x^{q^i}$ for $p_i \in \mathbb{F}_{q^m}$, $p_r \neq 0$.*

A q-polynomial P satisfies $\forall x_1, x_2 \in \mathbb{F}_q$, $\alpha_1, \alpha_2 \in \mathbb{F}_{q^m}$, $P(\alpha_1 x_1 + \alpha_2 x_2) = \alpha_1 P(x_1) + \alpha_2 P(x_2)$. If x_1 and x_2 are roots of a q-polynomial P, then $P(x_1) = P(x_2) = P(x_1 + x_2) = 0$, which implies that the roots of a q-polynomial of q-degree r form a vector space over $\mathbb{F}_q$ of dimension at most r.

Proposition A.1 (Ore [47]). *For any subspace E of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ of dimension r, there exists a unique monic q-polynomial $P(x) = \sum_{i=0}^{r} p_i x^{q^i}$ of q-degree r such that $P(z) = 0$ for any $z \in E$. This q-polynomial is called an annulator polynomial.*

Definition A.4 (Gröbner Basis). *Let $\mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial ring over a finite field $\mathbb{F}$, $\prec$ be a monomial ordering, $\mathrm{LM}_\prec(f)$ be leading monomial w.r.t. $\prec$ of a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, I be an ideal generated by a polynomial sequence, $\mathrm{LM}_\prec(I) = \{\mathrm{LM}_\prec(f) : f \in I\}$, and $\langle \mathrm{LM}_\prec(I) \rangle$ be the leading monomial ideal of I. A polynomial set $G = \{g_1, g_2, \ldots, g_m\} \subset I$ is a Gröbner basis w.r.t. $\prec$ of I if $\langle \mathrm{LM}_\prec(I) \rangle = \langle \mathrm{LM}_\prec(g_1), \mathrm{LM}_\prec(g_2), \ldots, \mathrm{LM}_\prec(g_m) \rangle$, i.e., the leading monomial ideal of I is spanned by the leading monomials of $g_1, g_2, \ldots, g_m$.*

This means that a set $G = \{g_1, g_2, \ldots, g_m\}$ is a Gröbner basis w.r.t. $\prec$ of I iff $\forall f \in I$, $\exists g \in G$ s.t., $\mathrm{LM}_\prec(g) | \mathrm{LM}_\prec(f)$.

The definition depends on the monomial ordering. This ordering has also a direct impact on the structure of a Gröbner basis. For instance, a Gröbner basis for a lexicographical order of a zero-dimensional system (i.e., with a finite number of solutions) has the following shape:

$$\{g_1(x_1), \ldots, g_2(x_1, x_2), \ldots, g_{k_1}(x_1, x_2), g_{k_1+1}(x_1, x_2, x_3), \ldots, g_{k_n}(x_1, \ldots, x_n)\}.$$

With such structure, the solutions to the zero-dimensional system can be easily computed by successively eliminating variables, namely computing solutions of univariate polynomials $g_1(x_1)$ and back-substituting the results.

The historical method for computing Gröbner bases was introduced by Buchberger in [16,17]. Many improvements have been done and have led to more efficient algorithms such as F_4 and F_5 [22,23]. The algorithm F_4 for example is the default algorithm in the computer algebra softwares MAGMA, MAPLE, and SageMath. The F_5 algorithm is even more efficient. An important quantity for computing Gröbner basis of an ideal is the degree of regularity (denoted by d_{reg}) of the system. The d_{reg} is the biggest degree reached in the Gröbner basis computation by the F_5 algorithm. The authors in [24] give a way to bound the complexity of the algorithm w.r.t. the d_{reg} of the system.

Proposition A.2. *The complexity of computing a Gröbner basis of a zero-dimensional system of m equations in n variables with F_5 is bounded by*

$$\mathcal{O}\left(m\binom{n + d_{reg} - 1}{d_{reg}}^{\omega}\right),$$

where d_{reg} is the degree of regularity of the system and $2 \leq \omega \leq 3$ is the linear algebra constant.

For a semi-regular system, the more precise complexity of computing a Gröbner basis is bounded by $\mathcal{O}\left(\binom{n+d_{reg}}{d_{reg}}^{\omega}\right)$ [11]. It has also been proven in [11] that the degree of regularity of semi-regular system can be computed explicitly.

Definition A.5 (Degree of regularity [11]). *The degree of regularity d_{reg} of a semi-regular system $f_1, \ldots, f_m$ of respective degrees $d_1, \ldots, d_m$ is given by the smallest integer such that the coefficient of the term $z^{d_{reg}}$ in the power series expansion of*

$$\frac{\prod_{i=1}^{m}(1-z^{d_i})}{(1-z)^n}.$$

is non-positive.

For semi-regular systems, it has been proven that the degree decreases as m goes larger. Thus, the more a system is overdetermined, the faster its Gröbner basis can be computed.

A.3 Product of Two Subspaces

Definition A.6 (Product of Two Subspaces [5]). *Let A and B be two $\mathbb{F}_q$-subspaces of $\mathbb{F}_{q^m}$. The product space of A and B is denoted by AB, the $\mathbb{F}_q$-linear span of the set of products $\{ab : a \in A, b \in B\}$.*

If A and B have dimensions α and β, and are generated respectively by the basis $(a_1, a_2, \ldots, a_\alpha) \in \mathbb{F}_{q^m}^{\alpha}$ and the basis $(b_1, b_2, \ldots, b_\beta) \in \mathbb{F}_{q^m}^{\beta}$, then the product space AB is obviously generated by the set $\{a_i b_j : i \in \{1..\alpha\}, j \in \{1..\beta\}\}$ and its dimension is bounded by $\alpha\beta$.

Proposition A.3 (Dimension of Product of Two Subspaces [5]). *Let A and B be two $\mathbb{F}_q$-subspaces of $\mathbb{F}_{q^m}$ of dimension α and β, respectively.* $\dim(AB) = \alpha\beta$ *with probability* $\geq 1 - \alpha\frac{q^{\alpha\beta}}{q^m}$.

A.4 Key Encapsulation Mechanism

Definition A.7 (Key Encapsulation Mechanism). *A Key Encapsulation Mechanism (KEM) scheme with a key space $\mathcal{K}$ consists of three polynomial-time algorithms* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$*:*

- KGen*: The key generation algorithm that takes the security parameter λ as input and outputs a public key pk and a private key sk. It is denoted as* $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$.
- Encap*: The encapsulation algorithm that takes pk as input and outputs a encapsulation c and a key $K \in \mathcal{K}$. It is denoted as* $(c, K) \leftarrow \mathsf{Encap}(pk)$.
- Decap*: The decapsulation algorithm that takes sk and c as inputs and outputs a key K. It is denoted as* $K \leftarrow \mathsf{Decap}(sk, c)$.

The correctness of KEM requires that for all $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$ and any $(c, K) \leftarrow \mathsf{Encap}(pk)$, the equation $K = \mathsf{Decap}(sk, c)$ hold with overwhelming probability.

A IND-CPA secure KEM scheme is defined by the experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

1. $\mathcal{A}$ takes λ as inputs.
2. $\mathcal{C}$ computes $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$, gives pk to $\mathcal{A}$, and keeps sk to itself.
3. $\mathcal{C}$ returns $(c, K) = \mathsf{Encap}(pk)$ to $\mathcal{A}$.
4. $\mathcal{C}$ randomly chooses a bit $b^* \in \{0,1\}$: if $b^* = 0$, sets $K^* = K$; if $b^* = 1$, chooses random $K^* \in \mathcal{K}$, and returns (c, K^*) to $\mathcal{A}$.
5. $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, $\mathcal{C}$ outputs 1, else outputs 0.

Definition A.8 (IND-CPA Security). *A KEM scheme is Indistinguishability under Chosen Plaintext Attacks (IND-CPA) if for any* PPT *adversary $\mathcal{A}$, its advantage*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{KEM},\mathcal{A}}(\lambda) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

A.5 Public Key Encryption

Definition A.9 (Public Key Encryption). *A Public Key Encryption (PKE) scheme with message space $\mathcal{M}$ consists of three polynomial-time algorithms* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*:*

- KGen*: The key generation algorithm that takes the security parameter λ as input and outputs a public key pk and a private key sk. It is denoted as $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$.*
- Enc*: The encryption algorithm that takes pk and a plaintext $M \in \mathcal{M}$ as inputs and outputs a ciphertext C. It is denoted as $C \leftarrow \mathsf{Enc}(pk, M)$.*
- Dec*: The decryption algorithm that takes sk and C as inputs and outputs a plaintext M. It is denoted as $M \leftarrow \mathsf{Dec}(sk, C)$.*

The correctness of PKE requires that for all $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$, any plaintext M, and any $C \leftarrow \mathsf{Enc}(pk, M)$, the equation $M = \mathsf{Dec}(sk, C)$ hold with overwhelming probability.

An IND-CPA secure PKE scheme is defined by the experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

1. $\mathcal{A}$ takes λ as inputs.
2. $\mathcal{C}$ computes $(pk, sk) \leftarrow \mathsf{KGen}(1^\lambda)$, gives pk to $\mathcal{A}$, and keeps sk to itself.
3. $\mathcal{A}$ outputs two plaintexts $M_0, M_1 \in \mathcal{M}$. $\mathcal{C}$ randomly chooses a bit $b^* \in \{0,1\}$ and returns the challenge ciphertext $C^* = \mathsf{Enc}(pk, M_{b^*})$ to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, $\mathcal{C}$ outputs 1, else outputs 0.

Definition A.10 (IND-CPA Security). *A PKE scheme is Indistinguishability under Chosen Plaintext Attacks (IND-CPA) if for any* PPT *adversary $\mathcal{A}$, its advantage*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

B. Overview of Attacks on the RD Problem

Due to the reduction [25], all the methods to solve the MinRank problem can be applied to the RD problem. These methods include: Kipnis-Shamir modeling [37], Kernel Attack [33], Minors Modeling [14], and Support-Minors (SM) modeling [10]. However, the first three MinRank solvers would not be the most suitable as they forget the $\mathbb{F}_{q^m}$-linearity. The first attack [19] was of combinatorial nature and started with the RSD problem. Then it was significantly improved in [48] and further refined in [29,6]. These combinatorial attacks consist of subtly guessing the support and coordinates of error and solving a linear system. These works can be viewed as the continuation of the kernel attack on generic MinRank, where one first guesses sufficiently many vectors in the kernel of the matrix of the rank r and then solves a linear system. The major difference in the case of RD is that the success probability of guess can be greatly increased due to the $\mathbb{F}_{q^m}$-linearity. Another way is the algebraic attack [29], where one solves an equations system induced from the RD problem based on the annulator polynomial by linearization and Gröbner basis. This algebraic attack [29] is considered to be less efficient than the combinatorial ones for a long time, especially for small values of q. A breakthrough paper [8] showed that the $\mathbb{F}_{q^m}$-linearity allows to devise a dedicated algebraic attack, i.e., the MaxMinors (MM) modeling. This was further improved in [10], where authors introduced another algebraic modeling, the Support-Minors (SM) modeling. The SM modeling is a generic MinRank attack and it later was combined with the MM modeling (the SM-$\mathbb{F}_{q^m}^+$ modeling [9]) to solve the RD problem. The attacks [8,10,9] are suitable for the parameters size of ROLLO and RQC. The analysis in [9] shows that the cost of the SM-$\mathbb{F}_{q^m}^+$ modeling is close to those of the combinatorial attack [6] and the MM modeling [10] for the parameters of ROLLO-I.

Recall that the RD problem asks an algorithm given as inputs a generator matrix $\boldsymbol{G}$ of random $[n,k]_{q^m}$-linear code $\mathcal{C}$, a vector $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$, and an integer $r \in \mathbb{N}$, outputs $\boldsymbol{x} \in \mathbb{F}_{q^m}^k$ and $\boldsymbol{e} \in \mathcal{S}_r^n$ such that $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$. The support and coefficient matrices of the standard rank metric error $\boldsymbol{e} \in \mathcal{S}_r^n$ are described in Remark 1. The two matrices can be reduced to less entries by Theorem 3.4.

Support and Coefficient Matrices with Less Entries. If let the first coordinate of $\boldsymbol{e}$ be 1, one can set the first column of $\boldsymbol{C}$ to $(1\ 0\ \cdots\ 0)^\top$ and the first column of $\boldsymbol{S}$ to $(1\ 0\ \cdots\ 0)^\top$. $\boldsymbol{S}$ and $\boldsymbol{C}$ can be further reduced to the following two forms with less entries.

- $\boldsymbol{S}_{\{1..r\},*} = \boldsymbol{I}_r$. By Gaussian elimination on columns of $\boldsymbol{S}$, there exists a matrix $\boldsymbol{P} = \left(\begin{array}{c|c} 1 & \boldsymbol{p} \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{P'} \end{array}\right) \in \mathrm{GL}_r(\mathbb{F}_q)$ such that $\boldsymbol{S}\boldsymbol{P} = \left(\begin{array}{c} \boldsymbol{I}_r \\ \hline \boldsymbol{0}_{(m-r)\times 1} \end{array}\middle|\, \boldsymbol{S'}\right)$ and $\boldsymbol{P}^{-1}\boldsymbol{C} = \left(\begin{array}{c} 1 \\ \hline \boldsymbol{0}_{(r-1)\times 1} \end{array}\middle|\, \boldsymbol{C'}\right)$ where $\boldsymbol{S'} \in \mathbb{F}_q^{(m-r)\times(r-1)}$ and $\boldsymbol{C'} \in \mathbb{F}_q^{r\times(n-1)}$. Then

$$\boldsymbol{e} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{P}\boldsymbol{P}^{-1}\boldsymbol{C} = \boldsymbol{\alpha}\left(\begin{array}{c} \boldsymbol{I}_r \\ \hline \boldsymbol{0}_{(m-r)\times 1} \end{array}\middle|\, \boldsymbol{S'}\right)\left(\begin{array}{c} 1 \\ \hline \boldsymbol{0}_{(r-1)\times 1} \end{array}\middle|\, \boldsymbol{C'}\right). \tag{16}$$

Let $\boldsymbol{S} := \boldsymbol{SP}$ and $\boldsymbol{C} := \boldsymbol{P}^{-1}\boldsymbol{C}$.

- $\boldsymbol{C}_{*,\{1..r\}} = \boldsymbol{I}_r$. By Gaussian elimination on rows of $\boldsymbol{C}$, there exists a matrix $\boldsymbol{Q} = \left(\begin{array}{c|c} 1 & \boldsymbol{q} \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{Q'} \end{array}\right) \in \mathrm{GL}_r(\mathbb{F}_q)$ such that $\boldsymbol{QC} = (\boldsymbol{I}_r\ \boldsymbol{C'})$ where $\boldsymbol{S'} \in \mathbb{F}_q^{m\times(r-1)}$ and $\boldsymbol{C'} \in \mathbb{F}_q^{r\times(n-r)}$. Then

$$\boldsymbol{e} = \boldsymbol{\alpha SC} = \boldsymbol{\alpha SQ}^{-1}\boldsymbol{QC} = \boldsymbol{\alpha}\left(\begin{array}{c|c} 1 & \\ \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S'} \end{array}\right)(\boldsymbol{I}_r\ \boldsymbol{C'}). \tag{17}$$

  Let $\boldsymbol{S} := \boldsymbol{SQ}^{-1}$ and $\boldsymbol{C} := \boldsymbol{QC}$.

For solving the RD problem, the most attacks aim to recover $\boldsymbol{S}$ and $\boldsymbol{C}$.

B.1 Combinatorial Attacks

CS Attack [19]: The combinatorial attack stems from the CS attack due to Chabaud and Stern. It solves a multivariate quadratic system obtained from the parity-check equation $\boldsymbol{He}^\top = \boldsymbol{s}$ and the error $\boldsymbol{e} = \boldsymbol{\alpha SC}$. This attack consists of searching all possible $\boldsymbol{S} = \left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S'} \end{array}\right)$ in Equation (17) and solving a generic $\boldsymbol{C}$ from a linear system. The cost is bounded by $\mathcal{O}\left((nr)^\omega q^{(m-r)(r-1)}\right)$.
OJ Attack [48]: Ourivski and Johannson proposed two combinatorial attacks to solve a quadratic multivariate system about the entries of $\boldsymbol{S}$ and $\boldsymbol{C}$. The solver guesses *some* coordinates variables of $\left(\begin{array}{c|c} 1 & \\ \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C'} \end{array}\right)$ or all variables of $\left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S'} \end{array}\right)$ in Equation (17), then solves a linear system. The cost is bounded by $\mathcal{O}\left(\min\left\{(k+r)^\omega q^{(m-r)(r-1)+2}, ((k+r)r)^\omega q^{(r-1)(k+1)}\right\}\right)$.
AGHT Attack [6]: This is an improvement of the previous method [29] and is considered to be the best combinatorial attack. The solver tries to guess a subspace that contains the support of $\boldsymbol{e}$, then solves a linear system and checks if the choice is correct. Please see Section 3.3 for details.

B.2 Algebraic Attack by Annulator Polynomial

Gaborit *et al.* [29] proposed an algebraic attack based on the theory of q-polynomials [47], more specifically annulator polynomials, whose roots form an $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ (see Appendix A). For the RD problem, since all coordinates of $\boldsymbol{e} = (e_1, e_2, ..., e_n)$ lie in a support E of dimension r, then there exists a unique monic q-polynomials $P(u) = \sum_{\delta=0}^{r} p_\delta u^{q^\delta}$ of q-degree r such that for $j \in \{1...n\}$

$$P\left(y_j - \sum_{i=1}^{k} x_i g_{ij}\right) = \sum_{\delta=0}^{r}\left(p_\delta y_j^{q^\delta} - \sum_{i=1}^{k} p_\delta x_i^{q^\delta} g_{ij}^{q^\delta}\right) = P\left(e_j\right) = 0, \tag{18}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)$, and $\boldsymbol{G} = (g_{ij})_{\substack{i\in\{1..k\}\\ j\in\{1..n\}}}$.

Equation (18) gives a multivariate system with n equations and $(r+k)$ variables $p_\delta^{(\nu)}$ and x_i. For solving the RD problem, one solves x_i from Equation (18). One views $(r+1)(k+1)-1$ monomials in p_δ and x_i as unknowns: kr terms of the form $p_\delta x_i^{q^\delta}$; k terms of the form $x_i^{q^r}$ due to $p_r = 1$; r terms of the form p_δ. Once $x_i^{q^r}$ is solved, one will obtain x_i. In the cryptography setting, usually $n < (r+1)(k+1)-1$, two methods are proposed to solve this case:

Linearization: Guessing e_j such that the number of equations is more than that of unknowns. If $e_j = 0$ is guessed, then the number of equations is reduced by one and the number of x_i is reduced by one, hence the number of unknowns is reduced by $r+1$ terms. The entries of $\boldsymbol{e}$ lie in the support E of dimension r, then the probability that $e_j = 0$ is correctly guessed is q^{-r}. If t $(t \leq k)$ coordinates e_j are correctly guessed and $n - t \geq (r+1)(k+1-t)-1$, then the cost of this attack is bounded by $\mathcal{O}\left((rk)^\omega q^{r\left\lceil \frac{(k+1)(r+1)-(n+1)}{r} \right\rceil}\right)$.

Gröbner Basis: Viewing Equation (18) as semi-regular polynomials system of degree of q^r+1 with n polynomials and $r+k$ variables p_l and x_i, and one solves this system by computing its Gröbner basis. The cost is bounded by $\mathcal{O}\left(n\binom{r+k+d_{reg}-1}{d_{reg}}^\omega\right)$, where d_{reg} is the degree of regularity of semi-regular system (see Appendix A for Gröbner basis and d_{reg}).

B.3 Algebraic Attack by MaxMinors Modeling

Bardet *et al.* [10] refined the MM modeling from [8] and achieved a more significant improvement in complexity. Equation (3) and $\boldsymbol{e} = \boldsymbol{\varepsilon C}$ imply that the matrix $\boldsymbol{C H_y^\top} \in \mathbb{F}_{q^m}^{r \times (n-k-1)}$ is not of row full rank because a non-zero vector $\boldsymbol{s}$ belongs to its left kernel. Then the maximal minors of $\boldsymbol{C H_y^\top}$ are equal to 0. By Cauchy-Binet formula, each P_J can be expressed as a linear combination over $\mathbb{F}_{q^m}$ of the maximal minors $c_T = |\boldsymbol{C}|_{*,T}$ of $\boldsymbol{C}$ for $T \in \{1..n\}$ and $\#T = r$. The MM modeling over $\mathbb{F}_{q^m}$ is built

$$\left\{P_J = \left|\boldsymbol{C H_y^\top}\right|_{*,J} : J \subset \{1..n-k-1\}, \#J = r\right\} \qquad \text{(MM-}\mathbb{F}_{q^m}\text{)} \qquad (19)$$

This leads to a linear system over $\mathbb{F}_{q^m}$ with $\binom{n}{r}$ unknowns c_T and $\binom{n-k-1}{r}$ equations. Once all c_T's are solved, one can recover $\boldsymbol{C}$ because, when $\boldsymbol{C}$ is of systematic form in Equation (17), $c_T = (-1)^{r-i} c_{ij}$ for $T \in (\{1..r\} \setminus \{i\}) \cup \{j\}$, $i \in \{1..r\}$, and $j \in \{r+1..n\}$. However, this system has many solutions due to $\binom{n-k-1}{r} < \binom{n}{r}$ whereas one wants more equations than unknowns for a unique solution. Because all c_T's are in $\mathbb{F}_q$, one further expresses P_J over $\mathbb{F}_q$ to obtain $m\binom{n-k-1}{r}$ equations. The MM modeling over $\mathbb{F}_q$ is built

$$\left\{P_{i,J} = |\boldsymbol{C H_y^\top}|_{*,J} : J \subset \{1..n-k-1\}, \#J = r, i \subset \{1..m\}\right\}, \quad \text{(MM-}\mathbb{F}_q\text{)} \quad (20)$$

Unknowns: $\binom{n}{r}$ variables $c_T \in \mathbb{F}_q$ for $T \subset \{1..n\}$ and $\#T = r$,
Equations: $m\binom{n-k-1}{r}$ linear equations $P_{i,J} = 0$ over $\mathbb{F}_q$ in c_T.

Once obtaining $\boldsymbol{C}$, one solves $\boldsymbol{\varepsilon} \in \mathbb{F}_{q^m}^r$ from linear system $\boldsymbol{\varepsilon C H_y}^\top = \boldsymbol{0}$. The cost of this attack is dominated by computing an echelon form of a matrix over $\mathbb{F}_q$ of size $m\binom{n-k-1}{r} \times \binom{n}{r}$ by Gaussian elimination.

Theorem B.1. *The complexity of solving the RD problem by the MM-$\mathbb{F}_q$ modeling is estimated as*

$$\begin{cases} \mathcal{O}\left(m\binom{n-p-k-1}{r}\binom{n-p}{r}^{\omega-1}\right), & m\binom{n-k-1}{r} \geq \binom{n}{r} - 1; \\ \mathcal{O}\left(q^{ar} m\binom{n-k-1}{r}\binom{n-a}{r}^{\omega-1}\right), & m\binom{n-k-1}{r} < \binom{n}{r} - 1. \end{cases}$$

where $p = \max\left\{i \mid m\binom{n-i-k-1}{r} \geq \binom{n-i}{r} - 1\right\}$ *and* a *is an integer such that* $m\binom{n-k-1}{r} \geq \binom{n-a}{r} - 1$ *exactly holds.*

B.4 Combining MaxMinors and Support-Minors Modelings

The SM modeling [8] for the RD problem is first proposed over $\mathbb{F}_q$ (i.e., SM-$\mathbb{F}_q$), but it is hard to achieve an overdetermined system due to a large number of linear variables. Later, the SM modeling is compacted over $\mathbb{F}_{q^m}$ (i.e., SM-$\mathbb{F}_{q^m}^+$) [9]. The analysis in [9] shows that the cost of the SM-$\mathbb{F}_{q^m}^+$ modeling is close to those of the combinatorial attack [6] and the MM modeling [10].

Equations $\boldsymbol{y} - \boldsymbol{xG} = \boldsymbol{e}$ and $\boldsymbol{e} = \boldsymbol{\alpha SC}$ imply that $\boldsymbol{y} - \boldsymbol{xG}$ is in the row space of $\boldsymbol{C}$, which in turn means that all maximal minors of the matrix $\binom{\boldsymbol{y}-\boldsymbol{xG}}{\boldsymbol{C}}$ are equal to 0. By Laplace expansion of such maximal minors w.r.t. the first row, each Q_J can be written as a bilinear polynomial in the entries x_i of $\boldsymbol{x}$ and in the maximal minors c_T of $\boldsymbol{C}$. Then the SM modeling over $\mathbb{F}_{q^m}$ (SM-$\mathbb{F}_{q^m}$) is obtained

$$\left\{ Q_J = \left| \binom{\boldsymbol{y} - \boldsymbol{xG}}{\boldsymbol{C}} \right|_{*,J} : J \subset \{1..n\}, \#J = r+1 \right\}, \qquad \text{(SM-}\mathbb{F}_{q^m}\text{)} \qquad (21)$$

Unknowns: $\binom{n}{r}$ minor variables $c_T \in \mathbb{F}_q$ for $T \subset \{1..n\}$, $\#T = r$, k linear variables x_i over $\mathbb{F}_{q^m}$,
Equations: $\binom{n}{r+1}$ equations $Q_J = 0$ which are affine bilinear equations over $\mathbb{F}_{q^m}$ in x_i and c_T.

For solving SM-$\mathbb{F}_{q^m}$, if there are more linearly independent equations than bilinear monomials, the system would be solved by linearization (i.e., by replacing the bilinear monomials in the form of $x_i c_T$, i.e., bi-degree $(1,1)$, by single variables). Otherwise, one solves at higher degree by multiplying the SM-$\mathbb{F}_{q^m}$ equations by monomials of degree $b-1$ in the linear variables to obtain equations in the linear variables of degree b and the c_T of degree 1. However, this system always has more monomials than equations for cryptographic parameters and cannot be solved at any degree b. The authors in [9] proved that the MM-$\mathbb{F}_q$ equations and some multiples are included in the vector space generated by the SM-$\mathbb{F}_{q^m}$ equations. By combining SM-$\mathbb{F}_{q^m}$ and MM-$\mathbb{F}_q$, the number of monomials can be further decreased and the SM-$\mathbb{F}_{q^m}^+$ modeling is proposed

$$\text{SM-}\mathbb{F}_{q^m}^+ = \text{SM-}\mathbb{F}_{q^m} \mod \text{(MM-}\mathbb{F}_q\text{)}, \qquad \text{(SM-}\mathbb{F}_{q^m}^+\text{)} \qquad (22)$$

Unknowns: $\binom{n}{r} - m\binom{n-k-1}{r}$ variables $c_T \in \mathbb{F}_q$ for $T \subset \{1..n\}$, $\#T = r$, and k variables $x_1, x_2, \ldots, x_k$ over $\mathbb{F}_{q^m}$,
Equations: $\binom{n}{r+1} - \binom{n-k-1}{r+1} - (k+1)\binom{n-k-1}{r}$ equations of the form $\widetilde{Q}_J = 0$ over $\mathbb{F}_{q^m}$ with $J \subset \{1..n\}, \#J = r+1, \#(J \cap \{1...k+1\}) \geq 2$, where $\widetilde{Q}_J = Q_J \mod (\text{MM-}\mathbb{F}_q)$.

Let the number of monomials be $\mathcal{M}_b^{\mathbb{F}_q} = k\left(\binom{n}{r} - m\binom{n-k-1}{r}\right)$ and the number of equations be $\mathcal{N}_b^{\mathbb{F}_q} = \binom{n}{r+1} - \binom{n-k-1}{r+1} - (k+1)\binom{n-k-1}{r}$. If $\mathcal{N}_b^{\mathbb{F}_q} \geq \mathcal{M}_b^{\mathbb{F}_q} - 1$, then the complexity of solving the RD problem is bounded by $T_{\text{plain}}(m,n,k,r) = \mathcal{O}\left(m^2 \mathcal{N}_b^{\mathbb{F}_q} {\mathcal{M}_b^{\mathbb{F}_q}}^{\omega-1}\right)$. If $\mathcal{N}_b^{\mathbb{F}_q} < \mathcal{M}_b^{\mathbb{F}_q} - 1$, to decrease the number of monomials, the authors proposed more sophisticated hybrid technique by permutating a entries of $\boldsymbol{e}$ into 0 and applying the SM-$\mathbb{F}_{q^m}^+$ modeling to the shorted code with parameters $(m, n-a, k-a, r)$. The complexity of the hybrid technique is given by $T_{\text{hybrid}}(m,n,k,r) = \min_{a \geq 0}\left(q^{ar} \cdot T_{\text{plain}}(m, n-a, k-a, r)\right)$.

C. Our Proofs of Lemmas, Propositions, and Theorems

C.1 Proof of Lemma 3.5

Proof. Let $\Pi(\ell) = \Pr_{F_1,F_2,\ldots,F_\ell}[E_1 \subset F_1, E_2 \subset F_2, \ldots, E_\ell \subset F_\ell, \text{all } F_i\text{'s are disjoint}]$ where the randomness comes from the choice of all F_i's. Our method to find such F_i containing E_i for $i \in \{1..\ell\}$ is that one first randomly guesses F_1 containing E_1 in $\mathbb{F}_{q^m}$, then successively guesses randomly F_i containing E_i in quotient space $\mathbb{F}_{q^m}/\sum_{j=1}^{i-1} E_j$.

For $\ell = 2$, the probability $\Pi(2) \approx q^{-mr+r_1^2+r_2r_1+t_2r_2}$ is estimated by three steps:

- guess randomly F_1 of dimension t_1 that contains E_1 of dimension r_1 in $\mathbb{F}_{q^m}$, and the success probability is $\frac{\begin{bmatrix} t_1 \\ r_1 \end{bmatrix}_q}{\begin{bmatrix} m \\ r_1 \end{bmatrix}_q} \approx q^{-r_1(m-t_1)}$,
- guess randomly E_1 from F_1, and the success probability is $\frac{1}{\begin{bmatrix} t_1 \\ r_1 \end{bmatrix}_q} \approx q^{-r_1(t_1-r_1)}$,
- guess randomly F_2 of dimension t_2 that contains E_2 of dimension r_2 in $\mathbb{F}_{q^m}/E_1$, and the success probability is $\frac{\begin{bmatrix} t_2 \\ r_2 \end{bmatrix}_q}{\begin{bmatrix} m-r_1 \\ r_2 \end{bmatrix}_q} \approx q^{-r_2(m-r_1-t_2)}$.

For $\ell = 3$, the probability $\Pi(3) \approx q^{-mr+r_1^2+r_2^2+r_2r_1+r_3(r_1+r_2)+t_3r_3}$ is estimated by three steps:

- proceeds the case of $\ell = 2$,
- guess randomly E_2 from F_2, and the success probability is $\frac{1}{\begin{bmatrix} t_2 \\ r_2 \end{bmatrix}_q} \approx q^{-r_2(t_2-r_2)}$,
- guess randomly F_3 of dimension t_3 that contains E_3 of dimension r_3 in $\mathbb{F}_{q^m}/(E_2+E_1)$, and the success probability is $\frac{\begin{bmatrix} t_3 \\ r_3 \end{bmatrix}_q}{\begin{bmatrix} m-r_1-r_2 \\ r_3 \end{bmatrix}_q} \approx q^{-r_3(m-r_1-r_2-t_3)}$.

For general ℓ, following the steps above in sequence, the expected probability will be obtained. □

C.2 Proof of Theorem 3.6

Proof. When $\boldsymbol{S}$ and $\boldsymbol{C}$ are in the form of Equation (5),

$$\boldsymbol{S} = \boldsymbol{\alpha}\left(\begin{array}{c|c} \boldsymbol{I}_r & \\ \hline \boldsymbol{0}_{(m-r)\times 1} & \boldsymbol{S}' \end{array}\right), \qquad \boldsymbol{C} = \left(\begin{array}{c|c} 1 & \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}' \end{array}\right),$$

where $\boldsymbol{S}' \in \mathbb{F}_q^{(m-r)\times(r-1)}$ and $\boldsymbol{C}' \in \mathbb{F}_q^{r\times(n-1)}$.

Splitting $\boldsymbol{C}$ into $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be the first $k+1$ columns and the last $n-k-1$ columns of $\boldsymbol{C}$. Equation $\boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j = \boldsymbol{0}_{m\times m}$ gives a quadratic multivariate system with m^2 quadratic polynomials and $(m-r)(r-1)+kr+r$ variables.

By first guessing the entries of $\boldsymbol{S}'$, one can obtain a linear system $\boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j = \boldsymbol{0}_{m\times m}$ about $(\boldsymbol{A}_1\ \boldsymbol{a}_j)$ with m^2 equations and at most $kr+r$ unknowns for $j \in \{1..n-k-1\}$. Thus, the complexity is bounded by $\mathcal{O}\left((kr+r)^\omega q^{(m-r)(r-1)}\right)$. □

C.3 Proof of Theorem 3.7

Proof. When $\boldsymbol{S}$ and $\boldsymbol{C}$ are in the form of Equation (6),

$$\boldsymbol{S} = \left(\begin{array}{c|c} 1 & \\ \hline \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}' \end{array}\right), \qquad \boldsymbol{C} = \left(\begin{array}{cc|c} \boldsymbol{I}_{r_1} & \boldsymbol{C}_1' & \boldsymbol{0}_{r_1\times n_2} \\ \hline \boldsymbol{0}_{r_2\times n_1} & \boldsymbol{I}_{r_2} & \boldsymbol{C}_2' \end{array}\right),$$

where $\boldsymbol{S}' \in \mathbb{F}_q^{m\times(r-1)}$, $\boldsymbol{C}_1' \in \mathbb{F}_q^{r_1\times(n_1-r_1)}$, and $\boldsymbol{C}_2' \in \mathbb{F}_q^{r_2\times(n_2-r_2)}$.

Let $k = n_1$ which is also the case of cryptography and other cases easily are extended. Let $\boldsymbol{S} = \left(\begin{array}{c|c} 1 & \boldsymbol{s}' \\ \hline \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}' \end{array}\right) \in \mathbb{F}_q^{m\times r}$ where $\boldsymbol{s}' = \boldsymbol{S}_{\{1\},\{2..r\}} \in \mathbb{F}_q^{1\times(r-1)}$ and $\boldsymbol{S}' = \boldsymbol{S}_{\{2..m\},\{2..r\}} \in \mathbb{F}_q^{(m-1)\times(r-1)}$. Splitting $\boldsymbol{C}$ into $\boldsymbol{A}_1 = \left(\begin{array}{c|c} 1 & \boldsymbol{c}^* \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}_1^* \end{array}\right) \in \mathbb{F}_q^{r\times(n_1+1)}$ and $\boldsymbol{A}_2 = \left(\begin{array}{c} \boldsymbol{0}_{1\times(n_2-1)} \\ \hline \boldsymbol{C}_2^* \end{array}\right) \in \mathbb{F}_q^{r\times(n_2-1)}$, where $\boldsymbol{c}^* = \boldsymbol{C}_{\{1\},\{2..n_1+1\}} \in \mathbb{F}_q^{1\times n_1}$, $\boldsymbol{C}_1^* = \boldsymbol{C}_{\{2..r\},\{2..n_1+1\}} \in \mathbb{F}_q^{(r-1)\times n_1}$, and $\boldsymbol{C}_2^* = \boldsymbol{C}_{\{2..r\},\{n_1+2..n\}} \in \mathbb{F}_q^{(r-1)\times(n_2-1)}$. Let $\boldsymbol{a}_j$ be the j-th column of $\boldsymbol{A}_2$ and $\boldsymbol{c}_2^*(j)$ be the j-th column of $\boldsymbol{C}_2^*$ for $j \in \{1..n_2-1\}$. Then $(\boldsymbol{A}_1\ \boldsymbol{a}_j) = \left(\begin{array}{c|c|c} 1 & \boldsymbol{c}^* & 0 \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}_1^* & \boldsymbol{c}_2^*(j) \end{array}\right)$ and

$$\begin{aligned} \boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j &= \left(\begin{array}{c|c} 1 & \boldsymbol{s}' \\ \hline \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}' \end{array}\right)\left(\begin{array}{c|c|c} 1 & \boldsymbol{c}^* & 0 \\ \hline \boldsymbol{0}_{(r-1)\times 1} & \boldsymbol{C}_1^* & \boldsymbol{c}_2^*(j) \end{array}\right)\boldsymbol{T}_j \\ &= \left(\begin{array}{c|c|c} 1 & \boldsymbol{c}^* + \boldsymbol{s}'\boldsymbol{C}_1^* & \boldsymbol{s}'\boldsymbol{c}_2^*(j) \\ \hline \boldsymbol{0}_{(m-1)\times 1} & \boldsymbol{S}'\boldsymbol{C}_1^* & \boldsymbol{S}'\boldsymbol{c}_2^*(j) \end{array}\right)\boldsymbol{T}_j = \boldsymbol{0}_{m\times m}. \end{aligned} \tag{23}$$

Equation (23) gives a quadratic multivariate system with m^2 quadratic polynomials and at most $m(r-1)+(n_1-r_1)r_1+r_2$ variables.

By first guessing the entries of $\boldsymbol{C}_1^*$ and $\boldsymbol{c}_2^*(j)$, we can obtain a linear system $\boldsymbol{S}(\boldsymbol{A}_1\ \boldsymbol{a}_j)\boldsymbol{T}_j = \boldsymbol{0}$ about the entries of $\boldsymbol{S}$ with m^2 equations and $m(r-1)+(n_1-r_1)$

unknowns. Since there are at most $q^{(r_1-1)(n_1-r_1)+r_2}$ numbers of $\boldsymbol{C}_1^*$ and $\boldsymbol{c}_2^*(j)$, the total complexity is bounded by $\mathcal{O}\left((m(r-1)+(n_1-r_1))^\omega q^{(r_1-1)(n_1-r_1)+r_2}\right)$. □

C.4 Proof of Theorem 3.9

Proof. To solve x_i from the multivariate system with n_ν polynomials and $(r_\nu+k)$ variables $p_\delta^{(\nu)}$ and x_i obtained from Equation (11), the linearization and Gröbner basis techniques are used.

Linearization: For the multivariate system obtained from Equation (11), one views $(r_\nu+1)(k+1)-1$ monomials in $p_\delta^{(\nu)}$ and x_i as unknowns: kr_ν terms of the form $p_\delta^{(\nu)}x_i^{q^\delta}$; k terms of the form $x_i^{q^{r_\nu}}$ due to $p_{r_\nu}^{(\nu)}=1$; r_ν terms of the form $p_\delta^{(\nu)}$. Then the multivariate system is transformed into a linear system over $\mathbb{F}_{q^m}$ with $(r_\nu+1)(k+1)-1$ unknowns and n_ν equations. Once $x_i^{q^{r_\nu}}$ is solved, one will obtain x_i.

When $n_\nu \geq (r_\nu+1)(k+1)-1$, this linear system is easily obtained in polynomial time with $\mathcal{O}\left(((r_\nu+1)(k+1)-1)^\omega\right)$ operations in $\mathbb{F}_{q^m}$. However, in the cryptography setting, usually $n_\nu < (r_\nu+1)(k+1)-1$. At this point, one guesses $e_j=0$ such that the number of equations is more than that of unknowns. If $e_j=0$ is correctly guessed, then the number of equations is reduced by one and the number of x_i is reduced by one. Hence, the number of unknowns is reduced by $r_\nu+1$ terms for Equation (11). The entries of $\boldsymbol{e}_\nu$ lie in the support $\mathrm{Supp}(\boldsymbol{e}_\nu)$ of dimension r_ν, for random $e_j \in \mathrm{Supp}(\boldsymbol{e}_\nu)$, the probability that $e_j=0$ is correctly guessed is q^{-r_ν}. If t_ν ($t_\nu \leq k$) coordinates e_j are correctly guessed such that $n_\nu - t_\nu \geq (r_\nu+1)(k+1-t_\nu)-1$, then the complexity is bounded by $\mathcal{O}\left((r_\nu k)^\omega q^{r_\nu t_\nu}\right)$ operations in $\mathbb{F}_{q^m}$. More specifically, for $\nu \in \{1..\ell\}$, if $\frac{(k+1)(r_\nu+1)-(n_\nu+1)}{r_\nu} \leq t_\nu \leq k$, then the complexity of solving the ℓ-RD problem is bounded by

$$\mathcal{O}\left(\min\left\{(r_\nu k)^\omega q^{r_\nu\left\lceil\frac{(k+1)(r_\nu+1)-(n_\nu+1)}{r_\nu}\right\rceil} : \nu \in \{1..\ell\}\right\}\right).$$

Gröbner Basis: For each $\nu \in \{1..\ell\}$, viewing the multivariate system obtained from Equation (11) as a semi-regular polynomials system of degree $q^{r_\nu}+1$ with n_ν polynomials and $r_\nu+k$ variables $p_\ell^{(\nu)}$ and x_i. Then one solves this semi-regular system by computing its Gröbner basis to obtain x_i. This complexity is dominated by the computation of Gröbner basis of semi-regular system. Then the complexity of solving the ℓ-RD problem is bounded by

$$\mathcal{O}\left(\min\left\{n_\nu\binom{r_\nu+k+d_{reg}^{(\nu)}-1}{d_{reg}^{(\nu)}}^\omega : \nu \in \{1..\ell\}\right\}\right),$$

where $d_{reg}^{(\nu)}$ is the degree of regularity of the semi-regular system (see Appendix A for Gröbner basis and d_{reg}). □

C.5 Proof of Lemma 3.10

Proof. Given the instance of the 2-RD problem defined by $[n,k]_{q^m}$-linear codes $\mathcal{C}$, let $\boldsymbol{H_y} = \left(-\boldsymbol{R}^\top \ \boldsymbol{I}_{n-k-1}\right) \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$ be a systematic parity-check matrix of the extend code $\mathcal{C}_{\boldsymbol{y}}$ of $\mathcal{C}$ (see Subsection 3.2). Let $k = n_1$ which is also the case of cryptography and other cases easily are extended.

When the matrix $\boldsymbol{C}$ is in the form of Equation (14), let $\boldsymbol{C} = (\boldsymbol{A}_1 \ \boldsymbol{A}_2)$, where $\boldsymbol{A}_1 = \left(\begin{array}{c|c} \boldsymbol{I}_{r_1} \ \boldsymbol{C}'_1 & \boldsymbol{0}_{r_1\times 1} \\ \hline \boldsymbol{0}_{r_2\times n_1} & \boldsymbol{c}^* \end{array}\right) = \left(\begin{array}{c|c} \overline{\boldsymbol{C}}_1 & \boldsymbol{0}_{r_1\times 1} \\ \hline \boldsymbol{0}_{r_2\times n_1} & \boldsymbol{c}^* \end{array}\right) \in \mathbb{F}_{q^m}^{r\times(n_1+1)}$, $\boldsymbol{A}_2 = \begin{pmatrix} \boldsymbol{0}_{r_1\times(n_2-1)} \\ \boldsymbol{C}_2^* \end{pmatrix} \in \mathbb{F}_{q^m}^{r\times(n_2-1)}$, $\overline{\boldsymbol{C}}_1 = (\boldsymbol{I}_{r_1} \ \boldsymbol{C}'_1) \in \mathbb{F}_q^{r_1\times n_1}$, $\boldsymbol{c}^* = \boldsymbol{C}_{\{r_1+1..r\},\{n_1+1\}} = (1,0,\ldots,0)^\top \in \mathbb{F}_q^{r_2\times 1}$, and $\boldsymbol{C}_2^* = \boldsymbol{C}_{\{r_1+1..r\},\{n_1+2..n\}} \in \mathbb{F}_q^{r_2\times(n_2-1)}$. Let $\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}^* \\ \boldsymbol{r}^* \end{pmatrix}$, where $\boldsymbol{R}^* = \boldsymbol{R}_{\{1..n_1\},*} \in \mathbb{F}_q^{n_1\times(n_2-1)}$ and $\boldsymbol{r}^* = \boldsymbol{R}_{\{n_1+1\},*} \in \mathbb{F}_q^{1\times(n_2-1)}$. Then we have

$$\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{A}_2 - \boldsymbol{A}_1\boldsymbol{R} = \begin{pmatrix} -\overline{\boldsymbol{C}}_1\boldsymbol{R}^* \\ \boldsymbol{C}_2^* - \boldsymbol{c}^*\boldsymbol{r}^* \end{pmatrix}.$$

This means that the most maximal minors $|\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top|_{*,J}$ of $\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top$ is non-zero for $J \subset \{1..n-k-1\}$ and $\#J = r$. Thus, the number of non-zero polynomials $P_J = |\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top|_{*,J}$ over $\mathbb{F}_{q^m}$ is dominated by $\binom{n-k-1}{r}$. When unfolding P_J over $\mathbb{F}_q$, at most $m\binom{n-k-1}{r}$ non-zero polynomials are obtained.
By Cauchy-Binet formula, $P_J = |\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top|_{*,J}$ can be viewed as a non-zero linear combination about all maximal minors c_T of $\boldsymbol{C}$ for $T \subset \{1..n\}$ and $\#T = r$. For the number of c_T, $c_T \neq 0$ iff $c_T = \left|\overline{\boldsymbol{C}}_1\right|_{*,T_1} \times \left|\overline{\boldsymbol{C}}_2\right|_{*,T_2}$ for $T = T_1 \cup T_2$, $T_1 \subset \{1..n_1\}$, $T_2 \subset \{n_1+1..n\}$, $\#T_1 = r_1$, and $\#T_2 = r_2$. Then the number of non-zero c_T is $\binom{n_1}{r_1}\binom{n_2}{r_2}$. □

C.6 Conclusion of Remark 3

Lemma C.1. *For $n, r, b \in \mathbb{N}$, n and r are divided by b, we have*

$$\left(\frac{1}{b}\right)^{\frac{b}{2}} \left(\frac{2\pi r(n-r)}{n}\right)^{\frac{b-1}{2}} \binom{n/b}{r/b}^b \approx \binom{n}{r}.$$

Proof. By Stirling's approximation, $n! \approx (2\pi n)^{\frac{1}{2}}(\frac{n}{e})^n$, we have

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \approx \frac{(2\pi n)^{\frac{1}{2}}(\frac{n}{e})^n}{(2\pi r)^{\frac{1}{2}}(\frac{r}{e})^r(2\pi(n-r))^{\frac{1}{2}}(\frac{n-r}{e})^{n-r}} = \frac{n^{\frac{1}{2}}n^n}{r^{\frac{1}{2}}(2\pi(n-r))^{\frac{1}{2}}r^r(n-r)^{n-r}},$$

$$\binom{n/b}{r/b} = \frac{\frac{n}{b}!}{\frac{r}{b}!\frac{n-r}{b}!} \approx \frac{(2\pi\frac{n}{b})^{\frac{1}{2}}(\frac{n}{be})^{\frac{n}{b}}}{(2\pi\frac{r}{b})^{\frac{1}{2}}(\frac{r}{be})^{\frac{r}{b}}(2\pi\frac{n-r}{b})^{\frac{1}{2}}(\frac{n-r}{be})^{\frac{n-r}{b}}} = \frac{n^{\frac{1}{2}}n^{\frac{n}{b}}}{(\frac{r}{b})^{\frac{1}{2}}(2\pi(n-r))^{\frac{1}{2}}r^{\frac{r}{b}}(n-r)^{\frac{n-r}{b}}},$$

$$\frac{\binom{n}{r}}{\binom{n/b}{r/b}^b} = \frac{n^{\frac{1}{2}}}{r^{\frac{1}{2}}(2\pi(n-r))^{\frac{1}{2}}} \times \frac{(\frac{r}{b})^{\frac{b}{2}}(2\pi(n-r))^{\frac{b}{2}}}{n^{\frac{b}{2}}} = \left(\frac{1}{b}\right)^{\frac{b}{2}} \left(\frac{2\pi r(n-r)}{n}\right)^{\frac{b-1}{2}}.$$

C.7 Proof of Theorem 3.11

Proof. From Lemma 3.10, for solving the 2-RD problem by the MM-$\mathbb{F}_q$ modeling, one needs to solve a linear system with $\binom{n_1}{r_1}\binom{n_2}{r_2}$ unknowns c_T and $m\binom{n-k-1}{r}$ equations. One wants a solution and hopes that the kernel of matrix of size $m\binom{n-k-1}{r} \times \binom{n_1}{r_1}\binom{n_2}{r_2}$ is one-dimensional.
"Overdetermined" Case: The parameters $(q, m, n, k, r, 2)$ fulfill

$$m\binom{n-k-1}{r} \geq \binom{n_1}{r_1}\binom{n_2}{r_2} - 1. \tag{24}$$

The cost is estimated as $\mathcal{O}\left(m\binom{n-k-1}{r}\left(\binom{n_1}{r_1}\binom{n_2}{r_2}\right)^{\omega-1}\right)$.

When the case is "super-overdetermined", i.e., parameters $(q, m, n, k, r, 2)$ *wildly fulfill* Inequality (24): $m\binom{n-k-1}{r} \gg \binom{n_1}{r_1}\binom{n_2}{r_2} - 1$. To obtain system whose equations are exactly more than unknowns, one constructs the system by puncturing on the last p coordinates of code $\mathcal{C}_{\boldsymbol{y}}$. The obtained puncturing code $\mathcal{C}'_{\boldsymbol{y}}$ is an $[n-p, k+1]_{q^m}$-code and one solves the 2-RD problem about code $\mathcal{C}'_{\boldsymbol{y}}$ with the same support of error. In this case, the MM-$\mathbb{F}_q$ modeling contains $m\binom{n-k-p-1}{r}$ equations and $\binom{n_1}{r_1}\binom{n_2-p}{r_2}$ variables c_T with $T \subset \{1..n-p\}$ and $\#T = r$. If there exists a maximal p such that $m\binom{n-i-k-1}{r} \geq \binom{n_1}{r_1}\binom{n_2-p}{r_2} - 1$ *exactly* holds, then the rank of system is $\binom{n_1}{r_1}\binom{n_2-p}{r_2} - 1$. The complexity is estimated as

$$\mathcal{O}\left(m\binom{n-p-k-1}{r}\left(\binom{n_1}{r_1}\binom{n_2-p}{r_2}\right)^{\omega-1}\right).$$

"Underdetermined" Case: The parameters $(q, m, n, k, r, 2)$ do not satisfy Inequality (24). One can reduce this "underdetermined" case to "overdetermined" case with hybrid method by exhaustively searching on some variables of $\boldsymbol{C}'_1$ and $\boldsymbol{C}'_2$ to obtain a linear system satisfying overdetermined case. In case that the last a_1 columns of $\boldsymbol{C}'_1$ and the last a_2 columns of $\boldsymbol{C}'_2$ are exhaustively searched, the number of unknowns is bounded by $\binom{n_1-a_1}{r_1}\binom{n_2-a_2}{r_2}$. The cost is about

$$\mathcal{O}\left(q^{a_1 r_1 + a_2 r_2} m\binom{n-k-1}{r}\left(\binom{n_1-a_1}{r_1}\binom{n_2-a_2}{r_2}\right)^{\omega-1}\right), \tag{25}$$

where (a_1, a_2) is an integer pair such that the overdetermined condition $m\binom{n-k-1}{r} \geq \binom{n_1-a_1}{r_1}\binom{n_2-a_2}{r_2} - 1$ exactly holds and the minimal complexity is obtained. □

C.8 Proof of Proposition 4.2

Lemma C.2. *For $a, \delta \in \mathbb{N}$, let $\Pr(a, \delta)$ denote the probability that a random matrix in $\mathbb{F}_q^{a \times (a+\delta)}$ has rank $< a$, then $\Pr(a, \delta) \leq q^{-\delta}$.*

Proof. Assume that a random matrix in $\mathbb{F}_q^{a\times(a+\delta)}$ is constructed by sampling the a rows one by one. For $i \in \{1..a\}$, let E_i and $\neg E_i$ denote respectively the events that the first i rows are linearly independent and dependent, then $\Pr[\neg E_i|E_{i-1}] = \frac{q^{i-1}}{q^{a+\delta}} = q^{i-1-a-\delta}$ because $\neg E_i$ occurs iff the i-th row (sampled uniformly from the space of size $q^{a+\delta}$) falls into the space (of size q^{i-1}) spanned by the first $i-1$ rows. Here, $\Pr[\neg E_1|E_0] = \Pr[\neg E_1] = \frac{q^{1-1}}{q^{a+\delta}} = q^{-a-\delta}$. We obtain further $\Pr(a,\delta) = \Pr[\neg E_a] = \sum_{i=1}^{a} \Pr[\neg E_i|E_{i-1}] = \sum_{i=1}^{a} q^{i-1-a-\delta} \leq q^{-\delta}$. □

Proof (Proposition 4.2). Since the error $\boldsymbol{e}$ of weight $(r_1, r_2, \ldots, r_\ell)$ and support $(E_1, E_2, \ldots, E_\ell)$ is chosen randomly, every s_i can be seen as a random element of $\sum_{j=1}^{\ell} E_j F_j$. The probability that $n-k$ elements s_i do not generate the whole space $\sum_{j=1}^{\ell} E_j F_j$ of dimension $\mu = \sum_{j=1}^{\ell} r_j d_j$ is given by the probability that a random $\mu \times (n-k)$ matrix over $\mathbb{F}_q$ is not full rank. This probability has been well analyzed (see Lemma C.2) and is bounded by $q^{-(n-k-\mu)}$. □

C.9 Proof of Proposition 4.3

Lemma C.3. *Let E be a fixed subspace of dimension r of $\mathbb{F}_{q^m}$. Let S_i, for $i \in \{1..d\}$, be d independently and randomly chosen subspaces of dimension μ containing the subspace E. The probability of* $\dim\left(\bigcap_{i=1}^{d} S_i\right) > r$ *is bounded by* $q^{\mu-r}\left(\frac{q^{\mu-r}-1}{q^{m-r}}\right)^{d-1}$.

Proof. As in [40], it suffices to prove the proposition for $r = 0$ by considering the quotient space S_i/E and $\dim\left(\bigcap_{i=1}^{d}(S_i/E)\right) > 0$. Fix the first subspace S_1, let $y \in S_1/E$ and $y \neq 0$. The probability of $y \in S_i/E$ equals $\frac{q^{\mu-r}-1}{q^{m-r}}$. By independence, the probability that this occurs for all $i \in \{2..d\}$ is $\left(\frac{q^{\mu-r}-1}{q^{m-r}}\right)^{d-1}$. The expected number of non-zero y in the intersection of d spaces S_i/E is $q^{\mu-r} - 1$, hence the result holds. □

Proof (Proposition 4.3). From Step 2 in the correctness of Algorithm 1, S_{ji} can be viewed as independent and random subspaces of dimension $\mu = \sum_{j=1}^{\ell} r_j d_j$ containing the subspace E_j of dimension r_j. From Lemma C.3, for each $j \in \{1..\ell\}$, the probability of $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ is $p_j = q^{\mu-r_j}\left(\frac{q^{\mu-r_j}-1}{q^{m-r_j}}\right)^{d_j-1}$. Then the probability of $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ for all $j \in \{1..\ell\}$ is $1 - \prod_{i=1}^{l}(1 - p_j) \approx \sum_{j=1}^{\ell} p_j$. □

D. Decoding Standard Errors for ℓ-LRPC Codes

In this section, we give decoding algorithm for the standard rank metric errors and analyze the decoding capability. We show that for the standard errors, the ℓ-LRPC code has the same decoding capacity as the standard LRPC code.

Consider an $[n,k]_{q^m}$ ℓ-LRPC code $\mathcal{C}$ with generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ and parity-check matrix $\boldsymbol{H} = (\boldsymbol{H}_1\ \boldsymbol{H}_2\ \cdots\ \boldsymbol{H}_\ell) \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$ of support $(F_1, F_2, \ldots, F_\ell)$. Let $\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G}+\boldsymbol{e}$ be a received word, where $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$ and $\boldsymbol{e} \in \mathcal{S}_r^n$ with the support E. The syndrome $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{H}\boldsymbol{e}^\top$.

The general idea of decoding $\boldsymbol{e}$ uses the fact that the subspace $S = \langle s_1, s_2, \ldots, s_{n-k}\rangle_{\mathbb{F}_q}$ generated by $\boldsymbol{s}$ enables one to recover the space $\sum_{i=1}^{\ell} EF_i$. Once obtaining $\sum_{j=1}^{\ell} EF_j$, one recovers the support E of the error $\boldsymbol{e}$. Finally, one recovers the coordinates of $\boldsymbol{e}$ by solving a linear system. The decoding algorithm is described in Algorithm 3.

Algorithm 3 Decoding standard errors for ℓ-LRPC codes

Input: the vector $\boldsymbol{y}$ and the parity-check matrix $\boldsymbol{H}$.
Output: the message $\boldsymbol{m}$

1: Computing syndrome space:
   – Compute the syndrome $\boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{s} = (s_1, s_2, \ldots, s_{n-k})^\top$ and the syndrome space $S = \langle s_1, s_2, \ldots, s_{n-k}\rangle_{\mathbb{F}_q}$
2: Recovering the support E:
   – Compute F_j from $\boldsymbol{H}$ for *one* $j \in \{1..\ell\}$
   – Compute the basis $(f_{j1}, f_{j2}, \ldots, f_{jd_j}) \in \mathbb{F}_{q^m}^{d_j}$ of F_j for *one* $j \in \{1..\ell\}$
   – Compute $S_{ji} = f_{ji}^{-1}S$, where all generators of S are multiplied by f_{ji}^{-1} for $i \in \{1..d_j\}$ for *one* $j \in \{1..\ell\}$
   – Compute $E = \bigcap_{i=1}^{d_j} S_{ji}$
3: Recovering the error $\boldsymbol{e}$:
   – Compute the basis $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r) \in \mathbb{F}_{q^m}^r$ of E
   – Write each entry e_j of $\boldsymbol{e}$ as $e_j = \sum_{i=1}^{r} e_{ij}\varepsilon_j$ for $j \in \{1..n\}$ in the basis $\boldsymbol{\varepsilon}$
   – Solve e_{ij} from the linear system $\boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{s}$
4: Recovering $\boldsymbol{m}$ from $\boldsymbol{m}\boldsymbol{G} = \boldsymbol{y} - \boldsymbol{e}$.

D.1 Correctness of the Decoding Algorithm

The correctness of Algorithm 3 depends on the recovery of correct E, which requires $\dim S = \dim\left(\sum_{i=1}^{\ell} EF_i\right)$ and $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) = r$ for *one* $j \in \{1..\ell\}$. We assume that these two conditions hold. Let $F = \sum_{j=1}^{\ell} F_j$.

Step 1: the first step of the algorithm is obvious.

Step 2: we prove that $E = \bigcap_{i=1}^{d_j} S_{ji}$ for *one* $j \in \{1..\ell\}$. Since $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$, $\boldsymbol{H} \in \mathcal{M}_{(d_1,d_2,\ldots,d_\ell)}^{(n_1,n_2,\ldots,n_\ell)}(n-k)$ is the matrix of support $(F_1, F_2, \ldots, F_\ell)$, and $\boldsymbol{e} \in \mathcal{S}_r^n$ is the error of support E, we have that the entries of $\boldsymbol{H}\boldsymbol{e}^\top$ lie in $E\sum_{j=1}^{\ell} F_j = EF$. Thus, $S \subset EF$. By assumption $\dim S = \dim\left(\sum_{i=1}^{\ell} EF_i\right)$, we have $S = EF$. Further, for any $i \in \{1..d_j\}$, since $f_{ji}\varepsilon_\kappa \in EF$ for one $\kappa \in \{1..r\}$, we have $\varepsilon_\kappa \subset S_{ji} = \{f_{ji}^{-1}x : x \in S\} \Rightarrow E \subset S_{ji}$. Then, $E \subset \bigcap_{i=1}^{d_j} S_{ji}$. By assumption $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) = r$, we have $E = \bigcap_{i=1}^{d_j} S_{ji}$.

Step 3: once the correct E is recovered, one expresses $\boldsymbol{e}$ under the basis of E, then computes the coordinates of $\boldsymbol{e}$ from Equation (15).

Step 4: the fourth step of the algorithm is obvious.

D.2 The Decoding Complexity

The most costly part is the intersection in Step 2 and solving linear systems in Step 3. The intersection $\bigcap_{i=1}^{d_j} S_{ji}$ of subspaces S_{ji} of dimension $\eta = r\sum_{j=1}^{\ell} d_j$ costs $\mathcal{O}\left(4d_j\eta^2 m\right)$ operations in $\mathbb{F}_q$. By **Solve-**EF, expressing $\boldsymbol{h}_i\varepsilon_j$ as a matrix of $rd \times n$ in the basis of EF consists in solving n linear systems with rd unknowns and m equations. This costs $(n-k)nr^{\omega+1}d^{\omega}$ operations in $\mathbb{F}_q$. Expressing s_i as a column vector of length rd in the basis of EF consists in solving a linear systems with rd unknowns and m equations. This costs $(n-k)(rd)^{\omega}$ operations in $\mathbb{F}_q$. Solving the linear system $\boldsymbol{H}\boldsymbol{e}^{\top} = \boldsymbol{s}$ with nr unknowns and $rd(n-k)$ equations costs $\mathcal{O}((nr)^{\omega})$ operations in $\mathbb{F}_q$. Thus, the complexity of the decoding algorithm is bounded by $\mathcal{O}((nr)^{\omega})$.

D.3 Decoding Failure Probability

By the correctness assumption of Algorithm 3, two cases can make the algorithm fail: (i) $\dim S < \dim\left(\sum_{i=1}^{\ell} EF_i\right)$; (ii) $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r$ for *one* $j \in \{1..\ell\}$. Propositions (D.1 and D.2) estimate the probability of two cases.

Proposition D.1. *The probability that* $\dim S < \dim\left(\sum_{i=1}^{\ell} EF_i\right)$ *is bounded by* $q^{-(n-k-\eta)}$ *where* $\eta = \sum_{j=1}^{\ell} rd_j$.

Proof. Since $\boldsymbol{e}$ of weight r and support E is chosen randomly, every s_i can be seen as a random element of $\sum_{i=1}^{\ell} EF_i$, the probability that $n-k$ elements s_i do not generate the space $\sum_{i=1}^{\ell} EF_i$ of dimension $\eta = \sum_{j=1}^{\ell} rd_j$ is given by the probability that a random $\eta \times (n-k)$ matrix over $\mathbb{F}_q$ is not full rank. This probability is bounded by $q^{-(n-k-\eta)}$ (see Lemma C.2). □

Proposition D.2. *The probability of* $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ *for one* $j \in \{1..\ell\}$ *is bounded by*

$$\max\left\{q^{\eta-r_j}\left(\frac{q^{\eta-r_j}-1}{q^{m-r_j}}\right)^{d_j-1}\right\},$$

where $\eta = \sum_{j=1}^{\ell} rd_j$.

Proof. From Step 2 in the correctness analysis of Algorithm 3, S_{ji} can be viewed as independent and random subspaces of dimension $\eta = \sum_{j=1}^{\ell} rd_j$ containing the subspace E of dimension r. From Lemma C.3, for each $j \in \{1..\ell\}$, the probability of $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ is $q^{\eta-r_j}\left(\frac{q^{\eta-r_j}-1}{q^{m-r_j}}\right)^{d_j-1}$. We obtain the probability of $\dim\left(\bigcap_{i=1}^{d_j} S_{ji}\right) > r_j$ for *one* $j \in \{1..\ell\}$. □

Combining Proposition D.1 and Proposition D.2, we deduce the decoding failure probability of Algorithm 3 in Theorem D.1.

Theorem D.1. *Under assumptions that S_{ji} behaves as random subspaces containing E, the decoding failure probability of Algorithm 1 is bounded by*

$$q^{-(n-k-\eta)} + \max\left\{ q^{\eta - r_j}\left(\frac{q^{\eta-r_j}-1}{q^{m-r_j}}\right)^{d_j-1} : j \in \{1..\ell\}\right\},$$

where $\eta = \sum_{j=1}^{\ell} rd_j$.

The analysis shows that the failure probability can be made arbitrarily small.

D.4 Error Correction Capability

From the correctness of Algorithm 3, we have $nr \leq rd(n-k) \Rightarrow d \geq \frac{n}{n-k}$. Under this condition, the decoding capacity is constrained by $r\sum_{j=1}^{\ell} d_j \leq n-k$. The following Theorem D.2 is obvious.

Theorem D.2. *When $d_1 = d_2 = \cdots = d_\ell$, the ℓ-LRPC code allows to decode the errors of weight up to $r = \frac{n-k}{\ell d_1}$.*

Theorem D.2 implies that for the standard rank metric errors, the ℓ-LRPC code has the same decoding capacity as the standard LRPC code.

For the accurate failure probability of decoding maximal errors, the theoretical probability is hard to be estimated and the value in Theorem D.1 is not practical for $q > 2$. We give a simulation of the decoding algorithm of 2-LRPC codes on SageMath 9.0. When $\ell = 2$ and $d_1 = d_2 = 2$, the 2-LRPC codes can decode the errors of weight up to $\frac{n-k}{4}$. The simulated result shows that the failure probability is about 0.721 for $q = 2$. Figure 7 shows the decreasing trend of the failure probability as q increases. For $q = 2$, the failure probability is bounded by $q^{-\left(n-k-\sum_{j=1}^{2} rd_j\right)} = 1$. For $q > 2$, the upper bound of failure probability seems to be $q^{-\left(n-k+1-\sum_{j=1}^{2} rd_j\right)}$. The code parameters are $(m, n, k, n_1, n_2, r, d_1, d_2) = (43, 40, 20, 20, 20, 5, 2, 2)$ for $q = 2, 3, 5, 7, 11, 13, 17, 19$.

E Ideal ℓ-RD Problem and Ideal ℓ-LRPC Codes

To improve RQC and ROLLO in Section 5, in this section, we give the ideal variants of the ℓ-RD problem and the ℓ-LRPC codes. Let $P(X)$ be a polynomial of degree n in $\mathbb{F}_q[X]$ and $\mathcal{R} = \mathbb{F}_{q^m}[X]/\langle P(X)\rangle$. By the map $\psi : \mathbb{F}_{q^m}^n \to \mathcal{R}$, the element of $\mathbb{F}_{q^m}^n$ is viewed as one of $\mathcal{R}$ and vice versa. The polynomial associated the vector $\boldsymbol{u} = (u_0, \ldots, u_{n-1}) \in \mathbb{F}_{q^m}^n$ is defined as $\boldsymbol{u}(X) = \sum_{i=0}^{n-1} u_i X^i \in \mathcal{R}$.

For $\boldsymbol{u} \in \mathbb{F}_{q^m}^n$ and $\boldsymbol{v} \in \mathbb{F}_{q^m}^n$, the product $\boldsymbol{uv}$ is defined as the vector of coefficients of polynomials $\boldsymbol{u}(X)\boldsymbol{v}(X) \mod P(X)$.

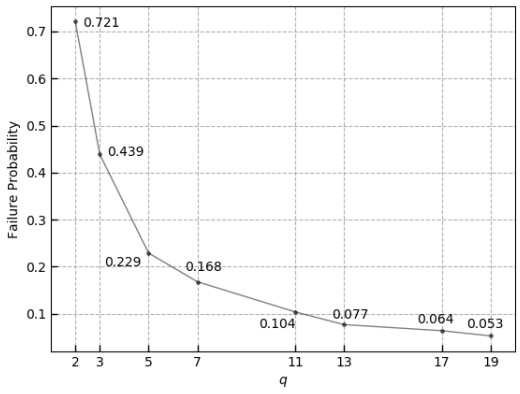

Fig. 7. Simulated failure probability of decoding the errors of the weight $\frac{n-k}{4}$ for 2-LRPC codes.

Definition E.1 (Ideal Matrix). *The ideal matrix generated by* $\boldsymbol{v} \in \mathbb{F}_{q^m}^n$ *is defined as an* $n \times n$ *matrix*

$$\mathcal{IM}(\boldsymbol{v}) = \begin{pmatrix} \boldsymbol{v}(X) \\ X\boldsymbol{v}(X) \mod P(X) \\ \vdots \\ X^{n-1}\boldsymbol{v}(X) \mod P(X) \end{pmatrix}.$$

The product $\boldsymbol{uv}$ is equivalent to the vector-matrix product $\boldsymbol{uv} = \boldsymbol{u}\mathcal{IM}(\boldsymbol{v}) = \mathcal{IM}(\boldsymbol{u})^\top \boldsymbol{v}^\top = \boldsymbol{vu}$. It is clear that $\boldsymbol{h}_1\boldsymbol{e}_1 + \boldsymbol{h}_2\boldsymbol{e}_2 = \boldsymbol{s} \iff (\boldsymbol{H}_1 \ \ \boldsymbol{H}_2)\,(\boldsymbol{e}_1 \ \ \boldsymbol{e}_2)^\top = \boldsymbol{s}$ where $\boldsymbol{H}_1 = \mathcal{IM}(\boldsymbol{h}_1)^\top$ and $\boldsymbol{H}_2 = \mathcal{IM}(\boldsymbol{h}_2)^\top$. We say that $(\boldsymbol{h}_1 \ \boldsymbol{h}_2)$ defines a parity-check matrix of a code $\mathcal{C}$ if $(\boldsymbol{H}_1 \ \boldsymbol{H}_2)$ is a parity-check matrix of $\mathcal{C}$.

To reduce the size of rank-based cryptosystems, the family of ideal codes is introduced in rank-based cryptography. Another advantage is that cryptosystems using ideal codes can resist the folding attack [34]. The ideal codes are codes with a systematic generator matrix consisting of blocks of ideal matrices. Please refer to [3,5] for more detailed definitions.

Definition E.2 (Ideal Codes [3,5]). *Let* $P(X)$ *be a polynomial of degree* n *in* $\mathbb{F}_q[X]$. *An* $[n\ell, nt]_{q^m}$ *code* $\mathcal{C}$ *is an* (ℓ, t)*-ideal code if its generator matrix under systematic form is of the form*

$$\boldsymbol{G} = \begin{pmatrix} & \mathcal{IM}(\boldsymbol{g}_{1,1}) & \dots & \mathcal{IM}(\boldsymbol{g}_{1,\ell-t}) \\ \boldsymbol{I}_{tn} & \vdots & \ddots & \vdots \\ & \mathcal{IM}(\boldsymbol{g}_{t,1}) & \dots & \mathcal{IM}(\boldsymbol{g}_{t,\ell-t}) \end{pmatrix}$$

where $(\boldsymbol{g}_{i,j})_{\substack{i\in\{1\dots\ell-t\}\\ j\in\{1\dots t\}}}$ *are vectors of* $\mathbb{F}_{q^m}^n$.

It has been proven [41] that if m and n are two different prime numbers and $P(X)$ is irreducible, then a non-zero ideal matrix is always non-singular. In this case, the generator matrix of ideal codes can be always reduced to the systematic form. We only use $[\ell n, n]_{q^m}$-ideal codes with the systematic parity-check matrix

$$\boldsymbol{H} = \begin{pmatrix} & \mathcal{IM}\left(\boldsymbol{h}_1\right)^\top \\ \boldsymbol{I}_{(\ell-1)n} & \vdots \\ & \mathcal{IM}\left(\boldsymbol{h}_{\ell-1}\right)^\top \end{pmatrix} \in \mathbb{F}_{q^m}^{(\ell-1)n \times \ell n}.$$

Let $\ell, k, n \in \mathbb{N}$. Let $\boldsymbol{n} = (n, \ldots, n)$, $\boldsymbol{r} = (r_1, \ldots, r_\ell)$, and $\boldsymbol{d} = (d_1, \ldots, d_\ell)$ be vectors of positive integers.

Definition E.3 (Ideal ℓ-RSD (ℓ-IRSD) Problem). *Let $\boldsymbol{H}$ be the systematic parity-check matrix of a random $[\ell n, n]_{q^m}$-ideal code and $\boldsymbol{s} \in \mathbb{F}_{q^m}^{(\ell-1)n}$. The problem is to find an ℓ-error $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$ such that $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$.*

This is also called computational ℓ-IRSD problem. In security proof of cryptosystems, the decisional version is often used.

Definition E.4 (Decisional ℓ-IRSD Problem). *Let $\boldsymbol{H}$ be the systematic parity-check matrix of a random $[\ell n, n]_{q^m}$-ideal code and $\boldsymbol{s} \in \mathbb{F}_{q^m}^{(\ell-1)n}$. The problem is to distinguish $\mathcal{D}_1$ and $\mathcal{D}_2$*

$$\mathcal{D}_1 = \left\{ (\boldsymbol{H}, \boldsymbol{s}) : \boldsymbol{s} \xleftarrow{\$} \mathbb{F}_{q^m}^{(\ell-1)n} \right\}, \quad \mathcal{D}_2 \quad = \left\{ (\boldsymbol{H}, \boldsymbol{s}) : \boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top, \boldsymbol{e} \xleftarrow{\$} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}} \right\}.$$

In ROLLO, to reduce the computational cost, one just recovers the support of the error without the computation of the error. This involves the Rank Support Recovery (RSR) problem which is as hard as solving the RSD problem. The definition of the ℓ-IRSR problem is naturally extended from the ℓ-IRSD problem.

Definition E.5 (Ideal ℓ-RSR (ℓ-IRSR) Problem). *Let $\boldsymbol{H}$ be the systematic parity-check matrix of a random $[\ell n, n]_{q^m}$-ideal code and $\boldsymbol{s} \in \mathbb{F}_{q^m}^{(\ell-1)n}$. The problem is to recover the space E_i of dimension r_i such that $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^\top$, $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$, and $\mathrm{Supp}(\boldsymbol{e}_i) = E_i$.*

As hardness assumptions [3,40] of the ideal decoding problems and the RSR problem, we argue that: the ℓ-IRSD problem is considered to be as hard as the ℓ-RSD problem and there is no known strong improvement on the complexity of solving the ideal version, typically choosing a $P(X)$ with many small factors and in our case $P(X)$ is an irreducible polynomial; the attacks on the decisional ℓ-IRSD problem remain the direct attacks on the computational ℓ-IRSD problem, thus decisional and computational versions have similar hardness; solving the ℓ-IRSR problem is as hard as solving the ℓ-IRSD problem.

Definition E.6 (Ideal ℓ-LRPC (ℓ-ILRPC) Codes). *Let F_i be an $\mathbb{F}_q$-subspace of dimension d_i of $\mathbb{F}_{q^m}$. Let $\boldsymbol{h}_i \in \mathbb{F}_{q^m}^n$ be a vector of support F_i. Let $\boldsymbol{H}_i = \mathcal{IM}(\boldsymbol{h}_i)^\top$ and $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 & \cdots & \boldsymbol{H}_\ell \end{pmatrix}$. The code $\mathcal{C}$ with the parity-check matrix $\boldsymbol{H}$ is called an ℓ-ILRPC code of type $[\ell n, (\ell-1)n]_{q^m}$.*

Since $P(X) \in \mathbb{F}_q[X]$ and the support of $P(X)\boldsymbol{h}_i$ is still F_i, it is necessary to choose $P(X)$ with coefficients in the base field $\mathbb{F}_q$ to keep the ideal ℓ-ILRPC structure. When m and n are two different prime numbers and $P(X)$ is irreducible, the parity-check matrix $\boldsymbol{H}$ of the $[\ell n, (\ell-1)n]_{q^m}$ ℓ-ILRPC code always can be reduced to the systematic form $\left(\boldsymbol{I}_n \ \boldsymbol{H}_1^{-1}\boldsymbol{H}_2 \ \boldsymbol{H}_1^{-1}\boldsymbol{H}_3 \ \cdots \ \boldsymbol{H}_1^{-1}\boldsymbol{H}_\ell\right)$.

Definition E.7 (2-ILRPC Codes Indistinguishability). *Given a vector* $\boldsymbol{h} \in \mathbb{F}_{q^m}^n$*. The problem is to distinguish* $\mathcal{D}_1'$ *and* $\mathcal{D}_2'$

$$\mathcal{D}_1' = \left\{\boldsymbol{h} : \boldsymbol{h} \xleftarrow{\$} \mathbb{F}_{q^m}^n\right\}, \qquad \mathcal{D}_2' = \left\{\boldsymbol{h} : \boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y}, (\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}_{(d_1,d_2)}^{(n,n)}\right\}.$$

By $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y} \iff \boldsymbol{y} - \boldsymbol{x}\boldsymbol{h} = \boldsymbol{0}$, solving this problem consists in finding $(\boldsymbol{y}, -\boldsymbol{x}) \in \mathcal{S}_{(d_1,d_2)}^{(n,n)}$ for the 2-IRSD problem with $\boldsymbol{H} = (\boldsymbol{1} \ \boldsymbol{h})$ and $\boldsymbol{s} = \boldsymbol{0}$ or consists in finding the codeword $(\boldsymbol{y}, -\boldsymbol{x}) \in \mathcal{S}_{(d_1,d_2)}^{(n,n)}$ in a $[2n, n]_{q^m}$-ideal code with parity-check matrix $\boldsymbol{H} = (\boldsymbol{1} \ \boldsymbol{h})$. Because any $\boldsymbol{r} \in \mathbb{F}_q^n$ fulfills $\boldsymbol{r}\boldsymbol{y} - \boldsymbol{r}\boldsymbol{x}\boldsymbol{h} = \boldsymbol{0}$ and $(\boldsymbol{r}\boldsymbol{y}, -\boldsymbol{r}\boldsymbol{x}) \in \mathcal{S}_{(d_1,d_2)}^{(n,n)}$, the complexity of the combinatorial attack is divided by q^n and is given by $\mathcal{O}\left((nm)^\omega q^{(d_1+d_2)\lceil \frac{m}{2} \rceil - m - n}\right)$. When the algebraic attacks in Section 3 is applied to this case, we do not find significant improvements.