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

Yongcheng Song, Jiang Zhang, Xinyi Huang, Wei Wu

State Key Laboratory of Cryptology, Beijing, China

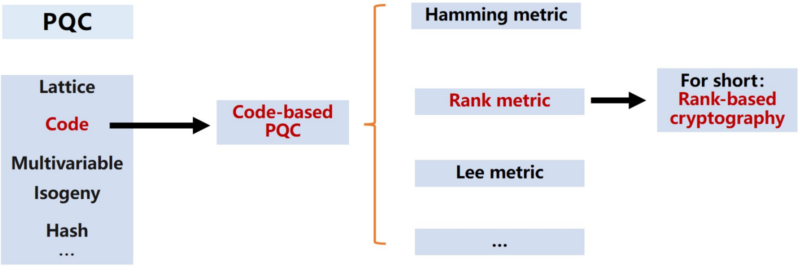
The Hong Kong University of Science and Technology (Guangzhou), China

ASIACRYPT —— December 7, 2023

Summary

1. Code-Based Cryptography using Rank Metric
   - Rank Metric, LRPC codes, and Hardness Assumptions
2. Contents
   - Basic Idea: Blockwise structure
   - Blockwise rank errors （ℓ-errors）
   - Blockwise rank decoding problems (ℓ-RD)
   - Attacks on ℓ-RD
   - Blockwise LRPC codes (ℓ-LRPC)
   - Improved RQC and ROLLO (NIST PQC Round 2)
3. Perspectives
4. References
5. Backup slides: Attack Details

Post-Quantum Cryptography (PQC)
Code-Based Cryptography: Rank-Based Cryptography



Rank Metric

In the rank metric, coordinates are in $\mathbb{F}_{q^m}$ (a field extension of $\mathbb{F}_q$ of degree m).

Definition 1 (Rank weight)

Let $\boldsymbol{\alpha} \in \mathbb{F}_{q^m}^m$ be an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^m}$. A word $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ can be expressed w.r.t. $\boldsymbol{\alpha}$ as a matrix $\mathrm{Mat}(\boldsymbol{x}) \in \mathbb{F}_{q^m}^{m \times n}$.
The rank weight of $\boldsymbol{x}$ is defined as the rank of the matrix $\mathrm{Mat}(\boldsymbol{x})$:

$$\|\boldsymbol{x}\|_{\mathrm{R}} = \mathrm{Rank}(\mathrm{Mat}(\boldsymbol{x})) \in [0, \min(m, n)].$$

Definition 2 (Rank support)

The rank support of a word $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ is the $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ generated by its coordinates:

$$\mathrm{Supp}(\boldsymbol{x}) = \langle x_1, x_2, \ldots, x_n \rangle_{\mathbb{F}_q}.$$

The rank weight is equal to the dimension of the rank support.
The weight and support definitions can also be extended to matrices.

LRPC codes, Hardness Assumptions

Definition 3 (Low Rank Parity-Check (LRPC) codes [6])

An $[n, k]_{q^m}$-LRPC code is defined by a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ of small rank weight.

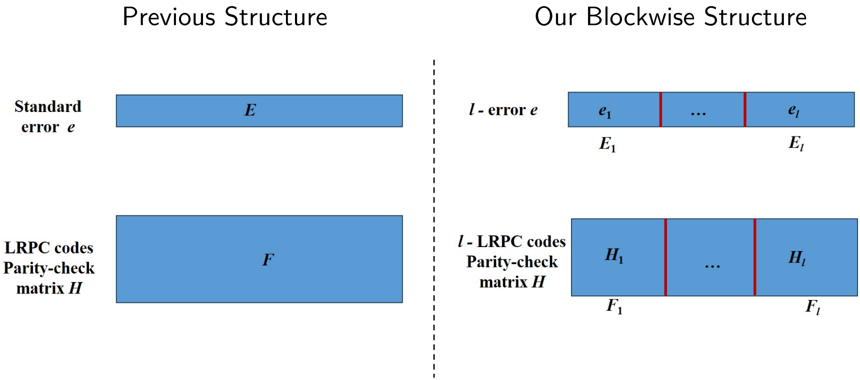
Definition 4 (Rank Decoding (RD))

Input: $\boldsymbol{G} \xleftarrow{\$} \mathbb{F}_{q^m}^{k\times n}$ (generator matrix), $\boldsymbol{y} \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$, $r \in \mathbb{N}$
Output: $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$, $\boldsymbol{m} \in \mathbb{F}_{q^m}^{k}$ s.t., $\boldsymbol{mG} + \boldsymbol{e} = \boldsymbol{y}$, $\|\boldsymbol{e}\|_{\mathrm{R}} \leq r$

Definition 5 (Rank Syndrome Decoding (RSD))

Input: $\boldsymbol{H} \xleftarrow{\$} \mathbb{F}_{q^m}^{(n-k)\times n}$ (parity-check matrix), $\boldsymbol{s} \xleftarrow{\$} \mathbb{F}_{q^m}^{n-k}$, $r \in \mathbb{N}$
Output: $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$ s.t., $\boldsymbol{He}^{\top} = \boldsymbol{s}$, $\|\boldsymbol{e}\|_{\mathrm{R}} \leq r$

Basic Ideas: Blockwise structure



Contents

In this talk:

1. Blockwise errors (ℓ-errors)
   The forms of support and coefficient matrices
2. Blockwise Rank Decoding problem (ℓ-RD) finding ℓ-errors
   Complexity loss
3. Blockwise LRPC codes (ℓ-LRPC)
   Decoding complexity, Decoding failure probability, Decoding capacity
4. Improve RQC and ROLLO (NIST PQC Round 2) with a smaller bandwidth than HQC, BIKE, and Classic McEliece (NIST PQC Round 4)
   Feasible

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.
$n = \sum_{i=1}^{\ell} n_i$ and $r = \sum_{i=1}^{\ell} r_i$

Definition 6 (ℓ-errors)

An error $\boldsymbol{e} \in \mathcal{S}_r^n$ is an ℓ-error if it can be divided into ℓ sub-vectors
$\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell)$ such that:
1) $\boldsymbol{e}_i \in \mathbb{F}_{q^m}^{n_i}$, $\|\boldsymbol{e}_i\|_{\mathrm{R}} = r_i$ for all $i \in \{1..\ell\}$
2) $\mathrm{Supp}(\boldsymbol{e}_i) \cap \mathrm{Supp}(\boldsymbol{e}_j) = \{0\}$ for all $i \neq j$.
The set of ℓ-errors: $\mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$.

Standard errors

- $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C}$
- Support matrix: $\boldsymbol{S} \in \mathbb{F}_q^{m \times r}$
- Coefficient matrix: $\boldsymbol{C} \in \mathbb{F}_q^{r \times n}$

ℓ-errors

- $\boldsymbol{e}_i = \boldsymbol{\varepsilon}_i \boldsymbol{C}_i$
  $\boldsymbol{e} = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_\ell)\mathrm{diag}(\boldsymbol{C}_1, ..., \boldsymbol{C}_\ell)$
  $= \boldsymbol{\alpha}\boldsymbol{S}\mathrm{diag}(\boldsymbol{C}_1, ..., \boldsymbol{C}_\ell)$
- Support matrix: $\boldsymbol{S} \in \mathbb{F}_q^{m \times r}$
- Coefficient matrix:
  $\mathrm{diag}(\boldsymbol{C}_1, ..., \boldsymbol{C}_\ell)$, $\boldsymbol{C}_i \in \mathbb{F}_q^{r_i \times n_i}$

Blocwise Rank Decoding Problems

Definition 7 (ℓ-RD)

Input: $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k\times n}$ (generator matrix), $\boldsymbol{y} \in \mathbb{F}_{q^m}^{n}$.
Output: $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$, $\boldsymbol{x} \in \mathbb{F}_{q^m}^{k}$, s.t., $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$ and $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$.

Definition 8 (ℓ-RSD)

Input: $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ (parity-check matrix), $\boldsymbol{s} \in \mathbb{F}_{q^m}^{n-k}$.
Output: $\boldsymbol{e} \in \mathbb{F}_{q^m}^{n}$, s.t., $\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top$ and $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$.

Recall Reduction

Theorem 9

Solving ℓ-$RD(q,m,n,k,r,\ell)$ problem defined by the $[n,k]_{q^m}$ linear code $\mathcal{C}$ $\Longrightarrow$ Finding an ℓ-codeword (i.e., ℓ-error) of weight r in $[n,k+1]_{q^m}$ linear code $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$.

The generator matrix of $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$ is $\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix} \in \mathbb{F}_{q^m}^{(k+1)\times n}$.

$\boldsymbol{e} = \begin{pmatrix} 1 & -\boldsymbol{m} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix}$ is an ℓ-codeword (i.e., ℓ-error) of weight r of $\mathcal{C}_{\boldsymbol{y}}$.

$\boldsymbol{G}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(k+1)\times n}$: generator matrix of $\mathcal{C}_{\boldsymbol{y}}$.

$\boldsymbol{H}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$: parity-check matrix of $\mathcal{C}_{\boldsymbol{y}}$.

Solving ℓ-$RD(q,m,n,k,r,\ell)$ problem $\Longrightarrow$

– Finding $\boldsymbol{u} \in \mathbb{F}_{q^m}^{k+1}$, s.t.,

$$\boldsymbol{u}\boldsymbol{G}_{\boldsymbol{y}} = \boldsymbol{e}, \tag{1}$$

– Finding ℓ-error $\boldsymbol{e}$ of weight r, s.t.,

$$\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^{\top} = \boldsymbol{0}. \tag{2}$$

Attacks Methods

- Combinatorial attacks: guess the entries of $\boldsymbol{S}$ or $\boldsymbol{C}$, then solve a linear system. The cost depends on the guessing way.
- Algebraic attacks: solve a multivariate or linear system in the entries of $\boldsymbol{S}$ and $\boldsymbol{C}$ by some sophisticated transformation. The cost depends on the number of the entries of $\boldsymbol{S}$ and $\boldsymbol{C}$.

For example, $\boldsymbol{e} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C}$ and $\boldsymbol{e}\boldsymbol{H}^\top = \mathbf{0} \Rightarrow \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{C}\boldsymbol{H}^\top = \mathbf{0}$, (known $\boldsymbol{\alpha}$ and $\boldsymbol{H}$).

Complexity comparison of solving RD and ℓ-RD

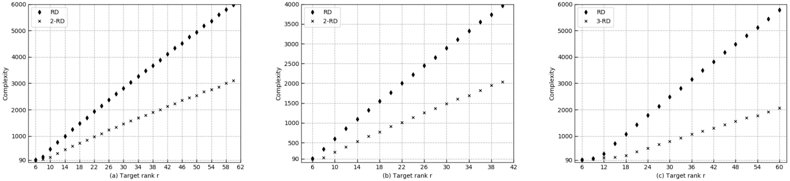
Attacks	RD(q, m, n, k, r)	ℓ-RD(q, m, n, k, r, ℓ)
AGHT [1, 2] (TIT 2016 & ISIT 2018)	$q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil - m}$	$q^{r\left\lceil \frac{(k+1)m}{n} \right\rceil - m}$
OJ [3] (PIT 2002)	$q^{(m-r)(r-1)+2}$ $q^{(r-1)(k+1)}$	$q^{(m-r)(r-1)}$ $q^{(r_1-1)(k-r_1)+\gamma}$ $\gamma = \max\{r_i : i \in \{2..\ell\}\}$
Annulator Polynomial [1] (TIT 2016)	$q^{r\left\lceil \frac{(k+1)(r+1)-(n+1)}{r} \right\rceil}$ $n\binom{r+k+d_{reg}-1}{d_{reg}}^{\omega}$	$\min\left\{ q^{r_\nu \left\lceil \frac{(k+1)(r_\nu+1)-(n_\nu+1)}{r_\nu} \right\rceil} : \nu \in \{1..\ell\} \right\}$ $\min\left\{ n_\nu \binom{r_\nu+k+d_{reg}^{(\nu)}-1}{d_{reg}^{(\nu)}}^{\omega} : \nu \in \{1..\ell\} \right\}$
Maximal Minors (MM) (Asiacrypt & Eurocrypt 2020) [4, 5]	$m\binom{n-p-k-1}{r}\left(\binom{n-p}{r}\right)^{\omega-1}$ $q^{ar}m\binom{n-k-1}{r}\left(\binom{n-a}{r}\right)^{\omega-1}$	$m\binom{n-p-k-1}{r}\left(\binom{n_\ell-p}{r_\ell}\prod_{i=1}^{\ell-1}\binom{n_i}{r_i}\right)^{\omega-1}$ $q^{\sum_{i=1}^{\ell} a_i r_i} m\binom{n-k-1}{r}\left(\prod_{i=1}^{\ell}\binom{n_i-a_i}{r_i}\right)^{\omega-1}$

The gain of most attacks benefits from the blockwise structure of ℓ-errors.

- OJ and MM: the block-diagonal form of coefficient matrix $\boldsymbol{C}$ allows to solve (multivariate or linear) systems with less variables;
- AGHT is limited because its cost depends on how to successfully guess a subspace that contains the support of the error;
- Annulator polynomials attack: the ℓ-errors allow to divide the ℓ-RD problem into ℓ subproblems with the smaller parameters.

Complexity loss of factor ℓ in the exponent

$$\ell | n, \quad \ell | r, \quad T_{\ell\text{-RD}} \approx \sqrt[\ell]{T_{\text{RD}}}$$



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

$$T_{2\text{-RD}} \approx \sqrt[2]{T_{\text{RD}}}, \quad T_{3\text{-RD}} \approx \sqrt[3]{T_{\text{RD}}}.$$

Blockwise LRPC Codes (ℓ-LRPC)

Let $\boldsymbol{n} = (n_1, \ldots, n_\ell)$ and $\boldsymbol{d} = (d_1, \ldots, d_\ell)$ be vectors of positive integers.
$n = \sum_{i=1}^{\ell} n_i$ and $d = \sum_{i=1}^{\ell} d_i$.

Definition 10 (ℓ-LRPC codes)

An LRPC code is an ℓ-LRPC 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) $\boldsymbol{H}_i \in \mathbb{F}_{q^m}^{(n-k)\times n_i}$, $\|\boldsymbol{H}_i\|_{\mathrm{R}} = d_i$ for all $i \in \{1..\ell\}$
2) $\mathrm{Supp}(\boldsymbol{H}_i) \cap \mathrm{Supp}(\boldsymbol{H}_j) = \{0\}$ for all $i \neq j$.

Decoding algorithm for ℓ-LRPC codes

The generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$,
The parity-check matrix $\boldsymbol{H} = (\boldsymbol{H}_1 \ \boldsymbol{H}_2 \ \cdots \ \boldsymbol{H}_\ell) \in \mathbb{F}_{q^m}^{(n-k)\times n}$, $\mathrm{Supp}(\boldsymbol{H}_i) = F_i$,
The error $\boldsymbol{e} = (\boldsymbol{e}_1 \ \boldsymbol{e}_2 \ \cdots \ \boldsymbol{e}_\ell)$, $\mathrm{Supp}(\boldsymbol{e}_i) = E_i$
The column of $\boldsymbol{H}_i$ matches the length of $\boldsymbol{e}_i$.

Decoding Steps: syndrome space S, recover support E, recover the error e

$$\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$$
$$\Downarrow$$
$$\boldsymbol{s} = (s_1, s_2, \ldots, s_{n-k}) = \boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{H}_1\boldsymbol{e}_1^\top + \boldsymbol{H}_2\boldsymbol{e}_2^\top + \ldots + \boldsymbol{H}_\ell\boldsymbol{e}_\ell^\top$$
$$\Downarrow$$
$$S = \langle s_1, s_2, \ldots, s_{n-k}\rangle_{\mathbb{F}_q} = F_1E_1 + F_2E_2 + \ldots + F_\ell E_\ell$$
$$\Downarrow$$
$$E_j = \bigcap_{i=1}^{d_j} f_{ji}^{-1}S,\ F_j = \langle f_{j1}, f_{j2}, \ldots, f_{jd_j}\rangle_{\mathbb{F}_q}$$
$$\Downarrow$$
$$E = \sum_{j=1}^{\ell} E_j$$
$$\Downarrow$$
$$E = \langle \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r\rangle_{\mathbb{F}_q},\ e_j = \sum_{i=1}^{r} e_{ij}\varepsilon_j,\ \ j \in \{1..n\}$$
$$\Downarrow$$

Solve e_{ij} from the linear system $\boldsymbol{H}\boldsymbol{e}^\top = \boldsymbol{s}$

Decoding complexity, failure probability, capacity

- Decoding complexity: $\mathcal{O}((nr)^{\omega})$
- Failure probability: $\approx q^{-(n-k-\mu)}$, ($\mu = \sum_{j=1}^{\ell} r_j d_j$ is the weight of syndrome $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{e}^{\top} = \sum_{i=1}^{\ell} \boldsymbol{H}_i \boldsymbol{e}_i$)
- Decoding capacity: $d_1 = d_2 = \cdots = d_\ell$, decoding ℓ-error of weight up to $\frac{n-k}{d_\ell}$. $d_1 = d_2 = \cdots = d_\ell = 2$, decoding ℓ-error of weight up to $\frac{n-k}{2}$.

A gain of factor ℓ in decoding capacity

- $[n, k]_{q^m}$ LRPC codes, parity-check matrix $\boldsymbol{H}$ of weight d, decoding error of weight $r = \frac{n-k}{d}$, DFR: q^{rd-n-k}.
- $[n, k]_{q^m}$ ℓ-LRPC codes, parity-check matrix $\boldsymbol{H} \in \mathcal{M}_{\boldsymbol{d}}^{\boldsymbol{n}}(k)$ ($\ell | d$, $d_i = d/\ell$), decoding ℓ-error of weight ℓr, the same DFR:

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

Codes	(q, m, n, k)	d	r	$\boldsymbol{d} = (d_1, d_2)$	$\boldsymbol{r} = (r_1, r_2)$	DFR	d_{RGV}
LRPC	$(2, 60, 66, 33)$	4	8	-	-	2^{-1}	18
2-LRPC		4	**16**	(2, 2)	(8, 8)	2^{-1}	

Rank setting: Ideal structure compresses size (the generalization of circulant structure)

$$P(X) \in \mathbb{F}_q[X]; \quad \mathcal{R} = \mathbb{F}_{q^m}[X]/\langle P(X) \rangle$$

$$\begin{aligned} \Psi : \quad \mathbb{F}_{q^m}^n &\simeq \mathcal{R} \\ (v_0, \ldots, v_{n-1}) &\mapsto \sum_{i=0}^{n-1} v_i X^i \end{aligned}$$

Vectors are equally viewed as polynomials.
P(X) is set as a reducible polynomial for the security.

Ideal matrix

$$\boldsymbol{u}\boldsymbol{v} = \boldsymbol{u}(X)\boldsymbol{v}(X) \mod P(X) = \sum_{i=0}^{n-1} u_i X^i \boldsymbol{v}(X) \mod P(X)$$

$$= \sum_{i=0}^{n-1} u_i \left(X^i \boldsymbol{v}(X) \mod P(X)\right) = (u_0, \ldots, u_{n-1}) \begin{pmatrix} \boldsymbol{v}(X) & \mod P(X) \\ X\boldsymbol{v}(X) & \mod P(X) \\ \vdots & \\ X^{n-1}\boldsymbol{v}(X) & \mod P(X) \end{pmatrix}$$

Definition 11 (Ideal matrix)

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

$$\boldsymbol{u}\boldsymbol{v} = \boldsymbol{u}\mathcal{IM}(\boldsymbol{v}) = \mathcal{IM}(\boldsymbol{u})^\top \boldsymbol{v}^\top = (\boldsymbol{v}\mathcal{IM}(\boldsymbol{u}))^\top = (\boldsymbol{v}\boldsymbol{u})^\top = \boldsymbol{v}\boldsymbol{u}.$$

Ideal ℓ-RD and ℓ-LRPC codes

Definition 12 (Ideal ℓ-RD)

Input: $\boldsymbol{G} = \left(\mathcal{IM}(\boldsymbol{g}_1)^\top, \mathcal{IM}(\boldsymbol{g}_2)^\top, ..., \mathcal{IM}(\boldsymbol{g}_\ell)^\top\right) \in \mathbb{F}_{q^m}^{n\times \ell n}$, $\boldsymbol{g}_i \in \mathbb{F}_{q^m}^n$, $\boldsymbol{y} \in \mathbb{F}_{q^m}^{\ell n}$.
Output: $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathbb{F}_{q^m}^{\ell n}$, $\boldsymbol{x} \in \mathbb{F}_{q^m}^k$, s.t., $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{G} + \boldsymbol{e}$ and $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$.

Definition 13 (Ideal ℓ-RSD)

Input: $\boldsymbol{H} = \left(\mathcal{IM}(\boldsymbol{h}_1)^\top, \mathcal{IM}(\boldsymbol{h}_2)^\top, ..., \mathcal{IM}(\boldsymbol{h}_\ell)^\top\right) \in \mathbb{F}_{q^m}^{n\times \ell n}$, $\boldsymbol{h}_i \in \mathbb{F}_{q^m}^n$, $\boldsymbol{s} \in \mathbb{F}_{q^m}^n$.
Output: $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_\ell) \in \mathbb{F}_{q^m}^{\ell n}$, s.t., $\boldsymbol{s}^\top = \boldsymbol{H}\boldsymbol{e}^\top = \sum_{i=1}^{\ell} \boldsymbol{h}_i \boldsymbol{e}_i$ and $\boldsymbol{e} \in \mathcal{S}_{\boldsymbol{r}}^{\boldsymbol{n}}$.

Definition 14 (Ideal ℓ-LRPC Codes)

F_i: $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ of dimension d_i. $\boldsymbol{h}_i \in \mathbb{F}_{q^m}^n$ and $\mathrm{Supp}(\boldsymbol{h}_i) = F_i$.
$\boldsymbol{H}_i = \mathcal{IM}(\boldsymbol{h}_i)^\top$, $\boldsymbol{H} = \left(\boldsymbol{H}_1 \ \boldsymbol{H}_2 \ \cdots \ \boldsymbol{H}_\ell\right)$.
An $[\ell n, (\ell-1)n]_{q^m}$ ℓ-LRPC code is called ℓ-ILRPC if its parity-check matrix is $\boldsymbol{H}$.

Improved RQC (PKE)

- RQC.KGen(λ): $\boldsymbol{h} \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$, $(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}_{(w_{\boldsymbol{x}}, w_{\boldsymbol{y}})}^{(n,n)}$, $\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{h}\boldsymbol{y}$.
  $pk = (\boldsymbol{h}, \boldsymbol{s})$, $sk = (\boldsymbol{x}, \boldsymbol{y})$.
- RQC.Enc($pk, \boldsymbol{m}$): $pk = (\boldsymbol{s}, \boldsymbol{h})$, $\boldsymbol{m} \in \mathbb{F}_{q^m}^{k}$,
  $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{e}) \xleftarrow{\$} \mathcal{S}_{(w_{\boldsymbol{r}_1}, w_{\boldsymbol{r}_2}, w_{\boldsymbol{e}})}^{(n,n,n)}$, $\boldsymbol{u} = \boldsymbol{r}_1 + \boldsymbol{h}\boldsymbol{r}_2$, $\boldsymbol{v} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{s}\boldsymbol{r}_2 + \boldsymbol{e}$,
  $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{v})$.
- RQC.Dec($sk, \boldsymbol{c}$): $sk = (\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{c}$, $\boldsymbol{m} \leftarrow \mathcal{C}.\text{Decode}\,(\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y})$.

$$\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{x}\boldsymbol{r}_2 - \boldsymbol{r}_1\boldsymbol{y} + \boldsymbol{e}$$

$$\|\boldsymbol{x}\boldsymbol{r}_2 + \boldsymbol{e} - \boldsymbol{r}_1\boldsymbol{y}\|_{\mathrm{R}} = w_{\boldsymbol{x}} w_{\boldsymbol{r}_2} + w_{\boldsymbol{y}} w_{\boldsymbol{r}_1} + w_{\boldsymbol{e}} \leq \left\lfloor \frac{n-k}{2} \right\rfloor$$

The security of improved RQC

Theorem 15

Under decisional 2-IRSD and 3-IRSD are hard, then our RQC PKE is IND-CPA secure.

Proof.

The 2-IRSD and 3-IRSD instances are

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

□

Improved ROLLO-I (Lake, KEM)

Alice		Bob
$(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(d_1,d_2)}$, $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y}$		
	$\xrightarrow{\quad \boldsymbol{h} \quad}$	
		$(\boldsymbol{e}_1, \boldsymbol{e}_2) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(r_1,r_2)}$ $E_1 = \mathrm{Supp}(\boldsymbol{e}_1), E_2 = \mathrm{Supp}(\boldsymbol{e}_2)$ $E = E_1 + E_2$ $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$
	$\xleftarrow{\quad \boldsymbol{c} \quad}$	
$\boldsymbol{x}\boldsymbol{c} = \boldsymbol{x}\boldsymbol{e}_1 + \boldsymbol{y}\boldsymbol{e}_2$ $(E_1, E_2) = \text{2-RSR}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}\boldsymbol{c}, r_1, r_2)$ $E = E_1 + E_2$		
$\boxed{\mathrm{Hash}(E)}$	Shared Secret	$\boxed{\mathrm{Hash}(E)}$

$$\boldsymbol{x}\boldsymbol{c} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix}$$

Improved ROLLO-II (Locker, PKE)

- Locker.KGen(λ): $(\boldsymbol{x}, \boldsymbol{y}) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(d_1,d_2)}$, $\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y}$,
  $pk = \boldsymbol{h}$, $sk = (\boldsymbol{x}, \boldsymbol{y})$.
- Locker.Enc(pk, M): $\boldsymbol{h}$, M,
  $(\boldsymbol{e}_1, \boldsymbol{e}_2) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(r_1,r_2)}$, $\boldsymbol{c} = \boldsymbol{e}_1 + \boldsymbol{h}\boldsymbol{e}_2$,
  $E_1 = \text{Supp}(\boldsymbol{e}_1)$, $E_2 = \text{Supp}(\boldsymbol{e}_2)$, $E = E_1 + E_2$,
  $C = (\boldsymbol{c}, M \oplus \text{Hash}(E)) = (\boldsymbol{c}, \boldsymbol{c}')$.
- Locker.Dec(sk, C): $(\boldsymbol{x}, \boldsymbol{y})$, C,
  $\boldsymbol{x}\boldsymbol{c} = \boldsymbol{x}\boldsymbol{e}_1 - \boldsymbol{y}\boldsymbol{e}_2$, $(E_1, E_2) \leftarrow$ 2-RSR$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}\boldsymbol{c}, r_1, r_2)$, $E = E_1 + E_2$,
  $M = \boldsymbol{c}' \oplus \text{Hash}(E)$.

$$\boldsymbol{x}\boldsymbol{c} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{y} \end{pmatrix} \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix}.$$

The security of improved ROLLO-I and ROLLO-II

Theorem 16

If the decisional 2-IRSD problems are hard, then our Lake KEM and Locker PKE are IND-CPA secure in the random oracle model.

Proof.

The 2-IRSD instances are

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

□

$$\boldsymbol{h} = \boldsymbol{x}^{-1}\boldsymbol{y} \Longleftrightarrow \boldsymbol{y} - \boldsymbol{x}\boldsymbol{h} = \mathbf{0} \Longleftrightarrow \mathbf{0} = \begin{pmatrix} 1 & \boldsymbol{h} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ -\boldsymbol{x} \end{pmatrix}$$

Improved ROLLO-III (Ouroboros-R, KEM)

Alice		Bob
$\text{seed}_{f_1} \xleftarrow{\$} \{0,1\}^\lambda$, $\boldsymbol{f}_1 \xleftarrow{\text{seed}_{f_1}} \mathbb{F}_{q^m}^n$		
$(\boldsymbol{h}_0, \boldsymbol{h}_1) \xleftarrow{\$} \mathcal{S}^{(n,n)}_{(d_1,d_2)}$		
$\boldsymbol{f}_0 = \boldsymbol{h}_1 + \boldsymbol{f}_1 \boldsymbol{h}_0$		
	$\xrightarrow{\boldsymbol{f}_0, \boldsymbol{f}_1}$	
		$(\boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}) \xleftarrow{\$} \mathcal{S}^{(n,n,n)}_{(r_1,r_2,r_3)}$
		$E_1 = \text{Supp}(\boldsymbol{e}_0)$, $E_2 = \text{Supp}(\boldsymbol{e}_1)$
		$E = E_1 + E_2$
		$\boldsymbol{c}_0 = \boldsymbol{f}_0 \boldsymbol{e}_1 + \boldsymbol{e}$, $\boldsymbol{c}_1 = \boldsymbol{f}_1 \boldsymbol{e}_1 + \boldsymbol{e}_0$
	$\xleftarrow{\boldsymbol{c}_0, \boldsymbol{c}_1}$	
$\boldsymbol{s} = \boldsymbol{c}_0 - \boldsymbol{h}_0 \boldsymbol{c}_1 = \boldsymbol{h}_1 \boldsymbol{e}_1 - \boldsymbol{h}_0 \boldsymbol{e}_0 + \boldsymbol{e}$		
$(E_1, E_2) = \text{3-RSR}(\boldsymbol{h}_0, \boldsymbol{h}_1, \boldsymbol{1}, \boldsymbol{s}, r_1, r_2)$		
$E = E_1 + E_2$		
$\text{Hash}(E)$	Shared Secret	$\text{Hash}(E)$

$$\boldsymbol{s} = \begin{pmatrix} \boldsymbol{1} & \boldsymbol{h}_0 & \boldsymbol{h}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{e} \\ -\boldsymbol{e}_0 \\ \boldsymbol{e}_1 \end{pmatrix}.$$

The security of ROLLO-III

Theorem 17

If decisional 2-IRSD and 3-IRSD problems are hard, then our Ouroboros-R is IND-CPA secure in the random oracle model.

Proof.

The 2-IRSD and 3-IRSD instances are

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

□

Size and DFR (HQC, BIKE, Classic McEliece)

Tradeoff between the hardness of the ℓ-RD problem and the decoding capacity of the ℓ-LRPC codes.

The gain of using ℓ-errors and ℓ-LRPC codes in decoding capacity outweighs the complexity loss in solving ℓ-RD.

Schemes		pks (bytes)	cts (bytes)	total (bytes)	DFR
RQC	**Our**	860	1704	**2564**	-
	NIST	1834	3652	5486	-
ROLLO-I	**Our**	511	511	**1022**	2^{-31}
	NIST	696	696	1392	2^{-28}
ROLLO-II	**Our**	1814	1942	**3756**	2^{-131}
	NIST	1941	2089	4030	2^{-134}
ROLLO-III	**Our**	623	1166	**1789**	2^{-33}
	TIT 2022 [7]	736	1431	2167	2^{-28}
HQC	NIST 4	2249	4497	6746	-
BIKE	NIST 4	1541	1573	3114	2^{-128}
Classic McEliece	NIST 4	261120	96	261216	-
Ouroboros	TIT 2022 [7]	1566	3100	4666	2^{-128}

128-security, public key size (pks), ciphertext size (cts), total = pks+cts.

The original Rollo-I and Rollo-II do not achieve the claimed security level due to structural attack.

Scheme - claimed security level	Structural attack on ideal LRPC codes
Lake (Rollo-I) - 192	2^{180}
Locker (Rollo-II) - 128	2^{103}
Locker (Rollo-II) - 192	2^{142}
Locker (Rollo-II) - 256	2^{173}

We will update their parameters and comparison at eprint and the potential structure attack do not influence the advantage of blockwise struture.

Performance（ROLLO，SageMath 9.0）

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)	995	109	391	128
Lake (NIST)	1220	134	525	192
Lake (NIST)	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)	2760	258	446	128
Locker (NIST)	3410	314	583	192
Locker (NIST)	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 2022) [7]	130	153	368	128
Ouroboros-R (TIT 2022) [7]	275	308	1040	192
Ouroboros-R (TIT 2022) [7]	504	614	2560	256

Sage scripts for the implementation and the complexity of the MM modeling:
https://github.com/YCSong232431/NH-ROLLO

Perspectives

In the future,

1. More analysis on the ℓ-RD problem to obtain the gain of a factor ℓ for the combinatorial attacks
2. More analysis on the indistinguishability of the ℓ-LRPC codes
3. Analyze blockwise rank support learning problem
4. Decrease DFR of Lake, Locker, and Ouroboros-R to $2^{-\lambda}$ for the security level λ, achieve the bandwidth of about 2 KB for Lake and Locker.
5. Construct blockwise cryptosystems without ideal structure

New Progresses – at last night

Combining our blockwise structure, the new improvements on RQC and ROLLO have appeared at https://eprint.iacr.org/2023/1875

8.3 Comparison with other schemes

For comparison, we compare our sizes with those of other encryption schemes, see Figure 16. We can see that our scheme has very competitive performances for 128 bits of security,by getting slightly smaller sizes than the lattice-based scheme KYBER.

Scheme	128 bits	192 bits
RQC-Block-MS-AG (this paper)	**1.4**	**2.8**
ILRPC-Block-MS (this paper)	**1.7**	**3.3**
KYBER [11]	1.5	2.2
BIKE [6]	3.1	6.2
HQC [2]	6.7	13.5
Classic McEliece [5]	261.2	624.3

Fig. 16: Comparaison of different schemes, the sizes represent the sum of the key and the ciphertext, expressed in kB

References

[1] Gaborit, P., Ruatta, O., Schrek, J.. On the complexity of the rank syndrome decod- ing problem. IEEE Transactions on Information Theory 62(2), 1006–1019 (2016)

[2] 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)

[3] 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)

[4] Bardet, M., Bros, M., Cabarcas, D., et al. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Advances in Cryptology - ASIACRYPT. vol. 12491, pp. 507–536. Springer (2020)

[5] Bardet, M., Briaud, P., Bros, M., et al. An algebraic attack on rank metric code-based cryptosystems. In: Advances in Cryptology - EUROCRYPT. vol. 12107, pp. 64–93. Springer (2020)

[6] Aragon, N., Gaborit, P., Hauteville, A., et al. Low rank parity check codes: New decoding algorithms and applications to cryptography. IEEE Transactions on Information Theory 65(12), 7697–7717 (2019)

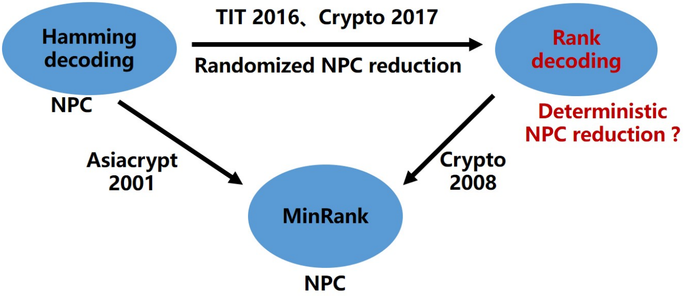
[7] Aragon, N., Blazy, O., Deneuville, J., et al. Ouroboros: An efficient and provably secure KEM family. IEEE Transactions on Information Theory 68(9), 6233–6244 (2022)

Thank you for your attention !

https://eprint.iacr.org/2023/1387

Backup slides

The hardness of the Rank Decoding problem



Recall Reduction

Theorem 18

Solving ℓ-$RD(q,m,n,k,r,\ell)$ problem defined by the $[n,k]_{q^m}$ linear code $\mathcal{C}$ $\Longrightarrow$ Finding an ℓ-codeword (i.e., ℓ-error) of weight r in $[n,k+1]_{q^m}$ linear code $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$.

The generator matrix of $\mathcal{C}_{\boldsymbol{y}} = \mathcal{C} + \langle \boldsymbol{y} \rangle$ is $\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix} \in \mathbb{F}_{q^m}^{(k+1)\times n}$.

$\boldsymbol{e} = \begin{pmatrix} 1 & -\boldsymbol{m} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{G} \end{pmatrix}$ is an ℓ-codeword (i.e., ℓ-error) of weight r of $\mathcal{C}_{\boldsymbol{y}}$.

$\boldsymbol{G}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(k+1)\times n}$: generator matrix of $\mathcal{C}_{\boldsymbol{y}}$.

$\boldsymbol{H}_{\boldsymbol{y}} \in \mathbb{F}_{q^m}^{(n-k-1)\times n}$: parity-check matrix of $\mathcal{C}_{\boldsymbol{y}}$.

Solving ℓ-$RD(q,m,n,k,r,\ell)$ problem $\Longrightarrow$

– Finding $\boldsymbol{u} \in \mathbb{F}_{q^m}^{k+1}$, s.t.,

$$\boldsymbol{u}\boldsymbol{G}_{\boldsymbol{y}} = \boldsymbol{e}, \tag{3}$$

– Finding ℓ-error $\boldsymbol{e}$ of weight r, s.t.,

$$\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^{\top} = \boldsymbol{0}. \tag{4}$$

Combinatorial attack ——The AGHT attack

- Guess randomly a t-dimensional subspace $F \supset \mathrm{Supp}(\boldsymbol{e})$.
- 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

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

By Equation (4): $\boldsymbol{H}_{\boldsymbol{y}} \boldsymbol{e}^\top = \boldsymbol{0}$, let $\boldsymbol{h}_j$ is the j-th row of $\boldsymbol{H}_{\boldsymbol{y}}$, we have

$$\boldsymbol{H}_{\boldsymbol{y}} \boldsymbol{e}^\top = \begin{pmatrix} \boldsymbol{h}_1 f_1 & \boldsymbol{h}_1 f_2 & \cdots & \boldsymbol{h}_1 f_t \\ \boldsymbol{h}_2 f_1 & \boldsymbol{h}_2 f_2 & \cdots & \boldsymbol{h}_2 f_t \\ \vdots & \vdots & & \vdots \\ \boldsymbol{h}_{n-k-1} f_1 & \boldsymbol{h}_{n-k-1} f_2 & \cdots & \boldsymbol{h}_{n-k-1} f_t \end{pmatrix} \begin{pmatrix} \overline{\boldsymbol{e}}_1^\top \\ \overline{\boldsymbol{e}}_2^\top \\ \vdots \\ \overline{\boldsymbol{e}}_t^\top \end{pmatrix} = \boldsymbol{0}_{n-k-1}. \quad (5)$$

- Express Equation (5) 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 $F \supset \mathrm{Supp}(\boldsymbol{e})$ is estimated as $\frac{\begin{bmatrix} t \\ r \end{bmatrix}_q}{\begin{bmatrix} m \\ r \end{bmatrix}_q} \approx q^{-r(m-t)}$.
- 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 (4): $\boldsymbol{H}_{\boldsymbol{y}} \boldsymbol{e}^\top = \boldsymbol{0}$, the complexity is divided by about q^m.

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

Let $\overline{e}_1$ and $\overline{e}_2$ 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 $\boldsymbol{e} = (\overline{e}_1, \overline{e}_2) = \boldsymbol{\varepsilon}(\boldsymbol{A}_1, \boldsymbol{A}_2) = (\boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{A}_1, \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{A}_2)$. Then

$$\boldsymbol{u}\boldsymbol{G}_{\boldsymbol{y}} = \boldsymbol{e} \iff (\boldsymbol{u}\ \boldsymbol{u}\boldsymbol{R}) = (\overline{e}_1, \overline{e}_2) \iff \overline{e}_1\boldsymbol{R} = \overline{e}_2 \iff \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{R} = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{A}_2. \tag{6}$$

For the 2-RD problem, by Equation (6), 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

$$\boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{r}_j = \boldsymbol{\alpha}\boldsymbol{S}\boldsymbol{a}_j \iff \boldsymbol{\alpha}\boldsymbol{S}\,(\boldsymbol{A}_1\ \ \boldsymbol{a}_j)\begin{pmatrix}\boldsymbol{r}_j\\ -1\end{pmatrix} = 0. \tag{7}$$

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 (7) can be written $\boldsymbol{\alpha}\boldsymbol{S}\,(\boldsymbol{A}_1\ \ \boldsymbol{a}_j)\,\boldsymbol{T}_j\boldsymbol{\alpha}^\top = 0$. This means

$$\boldsymbol{S}(\boldsymbol{A}_1\ \ \boldsymbol{a}_j)\,\boldsymbol{T}_j = \boldsymbol{0}_{m\times m}. \tag{8}$$

The entries of $\boldsymbol{S}(\boldsymbol{A}_1\ \ \boldsymbol{a}_j)\,\boldsymbol{T}_j$ are quadratic polynomials. Then Equation (8) gives a quadratic multivariate system over $\mathbb{F}_q$ with m^2 quadratic polynomials in the entries of $\boldsymbol{S}$ and $\boldsymbol{C}$.

$\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_\ell)$; $\boldsymbol{G} = (\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_\ell)$. Then

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

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 $j \in \{1..n_\nu\}$

$$P^{(\nu)}\left(y_j - \sum_{i=1}^{k} x_i g_{ij}\right) = \sum_{\delta=0}^{r_\nu}\left(p_\delta^{(\nu)} y_j^{q^\delta} - \sum_{i=1}^{k} p_\delta^{(\nu)} x_i^{q^\delta} g_{ij}^{q^\delta}\right) = P^{(\nu)}(e_j) = 0. \tag{9}$$

Equation (9) 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.

Standard errors: Coefficient matrix $\boldsymbol{C} \in \mathbb{F}_q^{r\times n}$

$\boldsymbol{e}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{0}$ and $\boldsymbol{e} = \boldsymbol{\varepsilon}\boldsymbol{C} \implies \boldsymbol{\varepsilon}\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top = \boldsymbol{0} \implies \boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top \in \mathbb{F}_{q^m}^{r\times(n-k-1)}$ is not row full rank.

The maximal minors $|\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top|_{*,J}$ equal to 0 ($J \subset \{1..n-k-1\}$, $\#J = r$).

By the Cauchy-Binet formula, each $|\boldsymbol{C}\boldsymbol{H}_{\boldsymbol{y}}^\top|_{*,J}$ can be expressed as the linear combination of $c_T = |\boldsymbol{C}|_{*,T}$ ($T \subset \{1..n\}$, $\#T = r$).

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

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.

Equations $<$ unknowns （underdetermined）, then unfold $P_J = 0$ over $\mathbb{F}_q$ for more equations

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

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.

Blockwise errors: Coefficient matrix $\boldsymbol{C} = \text{diag}(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_\ell) \in \mathbb{F}_q^{r\times n}$

The unknowns number of MM-$\mathbb{F}_q$: $\prod_{i=1}^{\ell} \binom{n_i}{r_i}$.