Decoding supercodes of Gabidulin codes and applications to cryptanalysis

Maxime Bombar[1,2] and Alain Couvreur[2,1]

[1] LIX, CNRS UMR 7161, École Polytechnique,
1 rue Honoré d'Estienne d'Orves
91120 Palaiseau Cedex
[2] Inria
{maxime.bombar, alain.couvreur}@inria.fr

Abstract. This article discusses the decoding of Gabidulin codes and shows how to extend the usual decoder to any supercode of a Gabidulin code at the cost of a significant decrease of the decoding radius. Using this decoder, we provide polynomial time attacks on the rank–metric encryption schemes Ramesses and Liga.

Keywords: Code–based cryptography, Gabidulin codes, decoding, rank–metric, cryptanalysis

Introduction

It is well–known that error correcting codes lie among the possible candidates for post quantum cryptographic primitives. For codes endowed with the Hamming metric the story begins in 1978 with McEliece's proposal [McE78]. The security of this scheme relies on two difficult problems: the hardness of distinguishing classical Goppa codes from arbitrary codes and the hardness of the syndrome decoding problem. To instantiate McEliece scheme, the only requirement is to have a family of codes whose structure can be hidden and benefiting from an efficient decoder. In particular, this paradigm does not require the use of codes endowed with the Hamming metric. Hence other metrics may be considered such as the rank metric, as proposed by Gabidulin, Paramonov and Tretjakov in [GPT91].

Besides McEliece's paradigm, another approach to perform encryption with error correcting codes consists in using codes whose structure is no longer hidden but where encryption is performed so that decryption without the knowledge of the secret key would require to decode the public code far beyond the decoding radius. This principle has been first instantiated in Hamming metric by Augot and Finiasz in [AF03] using Reed–Solomon codes. Later, a rank metric counterpart is designed by Faure and Loidreau in [FL05]. Both proposals have been subject to attacks, by Coron [Cor03] for the Hamming metric proposal and by Gaborit, Otmani and Talé–Kalachi [GOTK18] for the rank metric one. More recently, two independent and different repairs of Faure–Loidreau scheme resisting

to the attack of Gaborit, Otmani and Talé–Kalachi appeared. The first one, due to Renner, Puchinger and Wachter–Zeh and is called Liga [WPR18,RPWZ20]. The second one, due to Lavauzelle, Loidreau and Pham is called Ramesses [LLP20].

Our contribution

In the present article, we show how to extend the decoding of a Gabidulin code to a supercode at the cost of a significant decrease of the decoding radius. With this decoder in hand we perform a polynomial time message recovery attack on Ramesses and Liga.

1 Notation and prerequisites

In this article, we work over a finite field $\mathbb{F}_q$ and we frequently consider two nested extensions denoted $\mathbb{F}_{q^m}$ and $\mathbb{F}_{q^{mu}}$. Rank metric codes will be subspaces $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$, the code length and dimension are respectively denoted as n and k.

Vectors are represented by lower case bold face letters such as $\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{e}$ and matrices by upper case letters $\boldsymbol{A}, \boldsymbol{G}, \boldsymbol{M}$. The space of $m \times n$ matrices with entries in a field $\mathbb{K}$ is denoted by $\mathcal{M}_{m,n}(\mathbb{K})$. When $m = n$ we denote it by $\mathcal{M}_n(\mathbb{K})$ and the group of non-singular matrices is denoted by $\mathrm{GL}_n(\mathbb{K})$. Given a matrix $\boldsymbol{M} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ its rank is denoted by $\mathbf{rank}_q(\boldsymbol{M})$. Similarly, given a vector $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$, the $\mathbb{F}_q$*–rank* or *rank* of $\boldsymbol{c}$ is defined as the dimension of the subspace of $\mathbb{F}_{q^m}$ spanned by the entries of $\boldsymbol{c}$. Namely,

$$\mathbf{rank}_q(\boldsymbol{c}) \stackrel{\text{def}}{=} \dim_{\mathbb{F}_q} \left(\mathbf{Span}_{\mathbb{F}_q}\{c_1, \ldots, c_n\} \right).$$

We will consider two notions of *support* in the rank metric. Inspired by the Hamming metric, the most natural one which we will denote by the *column support* of a vector $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$ is the linear subspace spanned by its coordinates:

$$\mathrm{Supp}(\boldsymbol{c}) \stackrel{\text{def}}{=} \mathbf{Span}_{\mathbb{F}_q}\{c_1, \ldots, c_n\}.$$

But we can define another notion, namely the *row support*. Let $\mathscr{B}$ be a basis of the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$. Then, we define the extension of $\boldsymbol{c}$ with respect to $\mathscr{B}$ as the matrix $\mathrm{Ext}_{\mathscr{B}}(\boldsymbol{c}) \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ whose columns are the entries of $\boldsymbol{c}$ represented in the basis $\mathscr{B}$. The row space of $\mathrm{Ext}_{\mathscr{B}}(\boldsymbol{c})$ with respect to $\mathscr{B}$ will be called the *row support* of $\boldsymbol{c}$, *i.e.*

$$\mathrm{RowSupp}(\boldsymbol{c}) \stackrel{\text{def}}{=} \{\boldsymbol{x}\mathrm{Ext}_{\mathscr{B}}(\boldsymbol{c}) \mid \boldsymbol{x} \in \mathbb{F}_q^m\} \subset \mathbb{F}_q^n.$$

Notice that the above definition does not depend on the choice of the basis $\mathscr{B}$.

Given $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{F}_{q^m}^n$, and $j \in \{0, \ldots, m-1\}$, we denote

$$\boldsymbol{c}^{[j]} \stackrel{\text{def}}{=} (c_1^{q^j}, \ldots, c_n^{q^j}).$$

Similarly, for a code $\mathscr{C} \subseteq \mathbb{F}_{q^m}^n$, we denote

$$\mathscr{C}^{[j]} \stackrel{\text{def}}{=} \left\{ \boldsymbol{c}^{[j]} \mid \boldsymbol{c} \in \mathscr{C} \right\}.$$

Let $n \leqslant m$, $k \leqslant n$ and $\boldsymbol{g} \in \mathbb{F}_{q^m}^n$ with $\mathbf{rank}_q(\boldsymbol{g}) = n$, the *Gabidulin code of dimension k supported by $\boldsymbol{g}$* is defined as

$$\mathscr{G}_k(\boldsymbol{g}) \stackrel{\text{def}}{=} \left\{ \boldsymbol{g}^{[i]} \mid 0 \leqslant i \leqslant k-1 \right\}.$$

A *q–polynomial* is a polynomial $P \in \mathbb{F}_{q^m}[X]$ whose monomials are only q–th powers of X, *i.e.* a polynomial of the form

$$P(X) = p_0 X + p_1 X^q + \cdots + p_r X^{q^r}.$$

Assuming that $p_r \neq 0$ then the integer r is called the *q–degree* of P. Such a polynomial induces an $\mathbb{F}_q$–linear map $P : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$ and we call the *rank* of the q–polynomial, the rank of the induced map. A well–known fact on q–polynomials is that the $\mathbb{F}_q$–dimension of the kernel of the induced endomorphism is bounded from above by their q-degree. Conversely, any $\mathbb{F}_q$–linear endomorphism of $\mathbb{F}_{q^m}$ is uniquely represented by a q–polynomial of degree $< m$. Denote by $\mathcal{L}$ the space of q–polynomials, this space equipped with the composition law is a non commutative ring which is left and right Euclidean [Gos96, § 1.6] and the two–sided ideal $(X^{q^m} - X)$ is the kernel of the canonical map

$$\mathcal{L} \longrightarrow \mathrm{Hom}_{\mathbb{F}_q}(\mathbb{F}_{q^m}, \mathbb{F}_{q^m})$$

inducing an isomorphism :

$$\mathcal{L}/(X^{q^m} - X) \simeq \mathrm{Hom}_{\mathbb{F}_q}(\mathbb{F}_{q^m}, \mathbb{F}_{q^m}).$$

Finally, given a positive integer $k < m$, we denote by $\mathcal{L}_{<k}$ (resp. $\mathcal{L}_{\leqslant k}$) the space of q–polynomials of q–degree less than (resp. less than or equal to) k. The Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$ is canonically isomorphic to $\mathcal{L}_{<k}$ under the isomorphism:

$$\begin{cases} \mathcal{L}_{<k} \longrightarrow & \mathscr{G}_k(\boldsymbol{g}) \\ P \longmapsto & (P(g_1), \ldots, P(g_n)). \end{cases}$$

The above map is actually an isometry: it is rank preserving. In this article, we will extensively use this isometry and Gabidulin codes will be represented either as an evaluation code or as a space of q–polynomials of bounded degree $\mathcal{L}_{<k}$, when one representation is more suitable than the other. In particular, given two $\mathbb{F}_{q^m}$–linear subspaces $\mathscr{A}, \mathscr{B}$ of $\mathcal{L}/(X^{q^m} - X)$ we define their *composition* as

$$\mathscr{A} \circ \mathscr{B} \stackrel{\text{def}}{=} \mathbf{Span}_{\mathbb{F}_{q^m}}\{P \circ Q \mid P \in \mathscr{A},\ Q \in \mathscr{B}\}.$$

This definition may be regarded as a rank–metric analogue of the so–called Schur product of codes in Hamming metric.

Another notion which is very useful in the sequel is the notion of *adjoint* of a class of q–polynomial $P = \sum_{i=0}^{m-1} p_i X^{q^i}$ in $\mathcal{L}/(X^{q^m} - X)$, which is defined as

$$\langle \cdot, \cdot \rangle : P^{\vee}(X) \stackrel{\text{def}}{=} \sum_{i=0}^{m-1} X^{q^{m-i}} p_i = \sum_{i=0}^{m-1} p_i^{q^{m-i}} X^{q^{m-i}}. \tag{1}$$

Regarding P as an $\mathbb{F}_q$–linear endomorphism of $\mathbb{F}_{q^m}$, the notion of adjoint is nothing but the usual notion of *adjoint* or *transposed* endomorphism with respect to the inner product

$$\begin{cases} \mathbb{F}_{q^m} \times \mathbb{F}_{q^m} & \longrightarrow & \mathbb{F}_q \\ (x, y) & \longmapsto & \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xy) \end{cases}.$$

Details are given in a more general context in [ACLN20, Section 4.2], we give them here in the context of q–polynomials for the sake of self–containedness.

With respect to this bilinear form, the multiplication map by a scalar $x \mapsto ax$ for $a \in \mathbb{F}_{q^m}^{\times}$ is *symmetric* (or *self-adjoint*) since,

$$\langle x, ay \rangle = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xay) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(axy) = \langle ax, y \rangle.$$

Next, the Frobenius endomorphism is *orthogonal*, that is to say its adjoint $(X^q)^{\vee}$ is its inverse $X^{q^{m-1}}$ because:

$$\langle x, y^q \rangle = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(xy^q) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}\left((xy^q)^{q^{m-1}}\right) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(x^{q^{m-1}} y) = \langle x^{q^{m-1}}, y \rangle.$$

Finally, using that adjunction is anticommutative, *i.e.* for any $\mathbb{F}_q$–endomorphisms f, g of $\mathbb{F}_{q^m}$ we have $(fg)^{\vee} = g^{\vee} f^{\vee}$ we can prove that Definition 1 coincides with that of adjoint endomorphism. In particular, for any $P \in \mathcal{L}/(X^{q^m} - X)$, we have $\mathbf{rank}_q(P) = \mathbf{rank}_q(P^{\vee})$.

2 Two rank metric proposals with short keys

2.1 Liga encryption scheme

In this section, we recall Faure–Loidreau cryptosystem [FL05] and the repaired version [WPR18] recently extended to proposal Liga [RPWZ20].

Parameters Let q, m, n, k, u, w be positive integers such that q is a prime power, $u < k < n$ and

$$n - k > w > \lfloor \tfrac{n-k}{2} \rfloor.$$

In the following, we consider the three finite fields

$$\mathbb{F}_q \subseteq \mathbb{F}_{q^m} \subseteq \mathbb{F}_{q^{mu}}.$$

Let $t_{pub} \stackrel{\text{def}}{=} \lfloor \frac{n-k-w}{2} \rfloor$ be the public $\mathbb{F}_q$–rank of the error in the ciphertext, and let $\boldsymbol{G}$ be a generator matrix of a public Gabidulin code of length n and dimension k over $\mathbb{F}_{q^m}$.

Key generation. Alice picks uniformly at random a vector $\boldsymbol{x} \in \mathbb{F}_{q^{mu}}^k$ whose last u entries form an $\mathbb{F}_{q^m}$–basis of $\mathbb{F}_{q^{mu}}$, and a vector $\boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n$ of $\mathbb{F}_q-$rank w. In order to do that, she chooses a full-rank vector $\boldsymbol{s} \in \mathbb{F}_{q^{mu}}^w$ and a non-singular matrix $\boldsymbol{P} \in \mathrm{GL}_n(\mathbb{F}_q)$ and sets

$$\boldsymbol{z} = (\boldsymbol{s} \mid \boldsymbol{0}) \cdot \boldsymbol{P}^{-1}.$$

The *private key* is then $(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{P})$ and the *public key* is the *vector*

$$\boldsymbol{k}_{pub} \stackrel{\text{def}}{=} \boldsymbol{x} \cdot \boldsymbol{G} + \boldsymbol{z} \in \mathbb{F}_{q^{mu}}^n.$$

The key generation is summarised by Algorithm 1.

Algorithm 1: Original Faure–Loidreau Key Generation

Input: Parameters $q, \boldsymbol{G}, m, n, k, u, w$.
Output: Private key $\mathbf{sk}$, and public key $\mathbf{pk}$

$\boldsymbol{x} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^k \mid \dim(\mathbf{Span}_{\mathbb{F}_{q^m}}\{a_{k-u+1}, \ldots, a_k\}) = u\}$
$\boldsymbol{s} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^w \mid \mathbf{rank}_q(\boldsymbol{a}) = w\}$
$\boldsymbol{P} \xleftarrow{\$} \mathrm{GL}_n(\mathbb{F}_q)$
$\boldsymbol{z} \leftarrow (\boldsymbol{s} \mid \boldsymbol{0}) \cdot \boldsymbol{P}^{-1}$
$\boldsymbol{k}_{pub} \leftarrow \boldsymbol{x} \cdot \boldsymbol{G} + \boldsymbol{z}$
$\mathbf{sk} \leftarrow (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{P})$
$\mathbf{pk} \leftarrow \boldsymbol{k}_{pub}$
return $(\mathbf{sk}, \mathbf{pk})$

Encryption. Let $\boldsymbol{m} = (m_1, \ldots, m_{k-u}, 0, \ldots, 0) \in \mathbb{F}_{q^m}^k$ be the plaintext. Note that the last u entries are chosen to be zero in order to be able to decrypt. The encryption of $\boldsymbol{m}$ works as follows:

1. Pick $\alpha \in \mathbb{F}_{q^{mu}}$ at random.
2. Pick $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ such that $\mathbf{rank}_q(\boldsymbol{e}) \leqslant t_{pub}$ at random.

The ciphertext is then $\boldsymbol{c} \in \mathbb{F}_{q^m}^n$:

$$\boldsymbol{c} = \boldsymbol{m} \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub}) + \boldsymbol{e}.$$

As shown in (2) below, the public key acts on the one hand as a one-time pad on the message $\boldsymbol{m}$, and on the other hand adds a random error of large weight. The ciphertext can indeed be seen as a codeword of the Gabidulin code corrupted by a two-part errors formed by the private key $\boldsymbol{z}$ and the random error vector $\boldsymbol{e}$:

$$\boldsymbol{c} = (\boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G} + (\mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{z}) + \boldsymbol{e}). \tag{2}$$

With very high probability, the error in the *ciphertext* is of rank-weight $w + t_{pub}$. See [RPWZ20] for a detailed discussion about the parameters in order to avoid so-called weak keys.

Decryption. The *receiver* first computes

$$\boldsymbol{c} \cdot \boldsymbol{P} = (\boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G}\boldsymbol{P} + (\mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{s}) \mid 0) + \boldsymbol{e}\boldsymbol{P}.$$

whose last $n - w$ entries are given by

$$\boldsymbol{c}' = (\boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G}' + \boldsymbol{e}',$$

where $\boldsymbol{G}'$ is the generator matrix of a Gabidulin code of length $n - w$ and dimension k and $\boldsymbol{e}'$ is an error vector of rank-weight at most $t_{pub} = \lfloor \frac{n-w-k}{2} \rfloor$. By decoding in this new Gabidulin code, the *receiver* obtains the vector

$$\boldsymbol{m}' = \boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{x}).$$

Since by construction $\boldsymbol{m}$ is chosen such that its last u components are 0 and the last u components of $\boldsymbol{x}$ form a basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$, the *receiver* can compute

$$\alpha = \sum_{i=k-u+1}^{k} m'_i x_i^*$$

where $(x_{k-u+1}^*, \ldots, x_k^*)$ is the dual basis of $(x_{k-u+1}, \ldots, x_k)$ with respect to the on degenerate bilinear form $(x, y) \mapsto \mathrm{Tr}_{q^{mu}/q^m}(xy)$. Knowing both α and $\boldsymbol{x}$, the *receiver* can finally recover the plaintext $\boldsymbol{m}$. This encryption scheme has no decryption failure.

A key recovery attack. In [GOTK18], Gaborit, Otmani and Talé–Kalachi showed that a valid private key for this system could be efficiently computed from $\boldsymbol{k}_{pub}$, and later in [WPR18] and [RPWZ20], Renner, Puchinger and Wachter–Zeh introduced a coding-theoretic interpretation of the public key as a corrupted codeword of an u–interleaved Gabidulin code. They derived an equally powerful key recovery attack, and proved that the failure conditions of both attacks were equivalent.

Based on this interpretation, Renner et. al. proposed to change the key generation algorithm to resist previous attacks. More precisely, they proved that if ζ denotes the dimension of the $\mathbb{F}_{q^m}$–support of $\boldsymbol{z}$, all then known attacks were inefficient when $\zeta < \frac{w}{n-k-w}$. The new key generation can be summarised in the following Algorithm 2.

2.2 Ramesses

In this section, we present the proposal Ramesses [LLP20] which is another repair of the Faure–Loidreau scheme. We chose to describe the scheme in a rather different manner which turns out to be completely equivalent to the original proposal. The connection between this point of view and that of the original article is detailed in Appendix A. Our presentation rests only on q–polynomials. As explained in Section 1, the space $\mathcal{L}_{<k}$ will be regarded as a Gabidulin code of

Algorithm 2: LIGA Key Generation

Input: Parameters $q, \boldsymbol{G}, m, n, k, u, w, \zeta$.
Output: Private key $\mathbf{sk}$, and public key $\mathbf{pk}$

$\gamma \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^{u} \mid \mathbf{rank}_{q^m}(\boldsymbol{a}) = u\}$
$\boldsymbol{x} \xleftarrow{\$} \{\boldsymbol{a} \in \mathbb{F}_{q^{mu}}^{k} \mid \dim(\mathbf{Span}_{\mathbb{F}_{q^m}}\{a_{k-u+1}, \ldots, a_k\}) = u\}$
$\mathcal{A} \xleftarrow{\$} \{\text{subspaces } \mathcal{U} \subseteq \mathbb{F}_{q^m}^{w} \mid \dim \mathcal{U} = \zeta,\ \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements}\}$
$\begin{pmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_u \end{pmatrix} \xleftarrow{\$} \left\{ \begin{pmatrix} \boldsymbol{s}'_1 \\ \vdots \\ \boldsymbol{s}'_u \end{pmatrix} \mid \langle \boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_u \rangle_{\mathbb{F}_{q^m}} = \mathcal{A},\ \mathbf{rank}_q(\boldsymbol{s}'_i) = w\ \forall i \right\}$
$\boldsymbol{s} \leftarrow \sum_{i=1}^{u} \boldsymbol{s}_i \gamma_{\mathbf{i}}^{*}$
$\boldsymbol{P} \xleftarrow{\$} \mathrm{GL}_n(\mathbb{F}_q)$
$\boldsymbol{z} \leftarrow (\boldsymbol{s} \mid \boldsymbol{0}) \cdot \boldsymbol{P}^{-1}$
$\boldsymbol{k}_{pub} \leftarrow \boldsymbol{x}\boldsymbol{G} + \boldsymbol{z}$
$\mathbf{sk} \leftarrow (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{P})$
$\mathbf{pk} \leftarrow \boldsymbol{k}_{pub}$
return $(\mathbf{sk}, \mathbf{pk})$

dimension k. We also fix an $\mathbb{F}_q$–basis $\mathscr{B}$ of $\mathbb{F}_{q^m}$, which permits to have an $m \times m$ matrix representation of q–polynomials (modulo $(X^{q^m} - X)$) and conversely provides a description of any $m \times m$ matrix with entries in $\mathbb{F}_q$ as a q–polynomial of q–degree less than m.

Parameters The public parameters are integers $1 \leqslant w, k, \ell, t \leqslant m$ and should satisfy

$$t \leqslant \frac{n - k - \ell - w}{2}. \tag{3}$$

Key generation Alice picks a uniformly random q–polynomial K_{sec} of rank w. The public key is the affine space:

$$\mathscr{C}_{\mathrm{pub}} \stackrel{\mathrm{def}}{=} K_{\mathrm{sec}} + \mathcal{L}_{<k}.$$

Encryption The plaintext $\boldsymbol{m}$ is a t–dimensional $\mathbb{F}_q$–subspace of $\mathbb{F}_{q^m}$. It is encrypted as follows:

- Pick a uniformly random $T \in \mathcal{L}$ of q–degree ℓ
- Pick a uniformly random $E \in \mathcal{L}_{<m}$ whose matrix representation admits $\boldsymbol{m}$ as its row space, equivalently E is such that $\boldsymbol{m}$ is the image of $E^{\vee}$.
- Pick a uniformly random $C \in \mathcal{L}_{<k}$
- Pick a uniformly random $C_0 \in \mathcal{L}_{<k}$, yielding a uniformly random

$$C' = C_0 + K_{\mathrm{sec}} \in \mathscr{C}_{\mathrm{pub}}.$$

The ciphertext is

$$Y \stackrel{\mathrm{def}}{=} C + C' \circ T + E. \tag{4}$$

Note that, this cipher text satisfies

$$Y = C_1 + K_{\text{sec}} \circ T + E, \tag{5}$$

where $C_1 = C + C_0 \circ T \in \mathcal{L}_{<k} + \mathcal{L}_{<k} \circ T \subseteq \mathcal{L}_{<k+\ell}$. This C_1 is *a prioiri* unknown by anyone.

Decryption The owner of K_{sec} knows a q–polynomial $V \in \mathcal{L}_{\leqslant w}$ such that $V \circ K_{\text{sec}} \equiv 0 \mod (X^{q^m} - X)$. Hence she can compute

$$V \circ Y \equiv V \circ C_1 + V \circ E \mod (X^{q^m} - X).$$

Now, $V \circ C_1 \in \mathcal{L}_{<k+\ell+w}$, *i.e.* lies in a Gabidulin code, while $\mathbf{rank}_q(V \circ E) \leqslant \mathbf{rank}_q(E) = t$. Hence, thanks to (3), one can deduce $V \circ E$ and as soon as $\mathbf{rank}_q(V \circ E) = t$, the row space of the matrix representation of E is that of $V \circ E$ which can be recovered.

3 Decoding of Gabidulin codes on the right

In this section, we assume that $n = m$, *i.e.* $\boldsymbol{g} \stackrel{\text{def}}{=} (g_1, \dots, g_n)$ forms a basis of the extension field $\mathbb{F}_{q^m}/\mathbb{F}_q$. Let $\mathscr{C}$ be a Gabidulin code of dimension k and support $\boldsymbol{g}$. Suppose we receive a vector $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ where $\boldsymbol{c} \in \mathscr{C}$ and $\boldsymbol{e}$ has rank $t \leqslant \lfloor \frac{n-k}{2} \rfloor$. There exist three q–polynomials $C \in \mathcal{L}_{<k}$ and $Y, E \in \mathcal{L}_{<m}$ with $\mathbf{rank}_q(E) = t$ such that

$$Y = C + E \tag{6}$$

and the polynomial Y can be deduced from the knowledge of $\boldsymbol{y}$ and the basis $\boldsymbol{g}$ by interpolation (see for instance [WZ13, Chapter 3]).

Remark 1. Note that the requirement $n = m$ is necessary. Indeed, if $n < m$ the choice of the interpolating polynomial $Y \in \mathcal{L}_{<m}$ is not unique and a wrong choice for Y yields an $E = Y - C$ of too large rank. It is not clear to us how to weaken this condition.

In a nutshell, our approach can be explained as follows. Starting from the decoding problem (6) and applying the adjunction operator we have to solve the problem

$$Y^{\vee} = C^{\vee} + E^{\vee},$$

where $\mathbf{rank}_q E^{\vee} = \mathbf{rank}_q E \leqslant \lfloor \frac{n-k}{2} \rfloor$ and $C^{\vee}$ is contained in a code which is equivalent to a Gabidulin code. Hence, $C^{\vee}$ can be recovered by applying the decoding algorithm of [Loi06].

In what follows, we give a detailed and self–contained presentation of how to apply and implement this algorithm practically. We believe that this algorithm might be folklore, but we weren't able to find it in the literature.

Starting from the decoding problem $Y = C + E$, the decoding problem can be thought as finding the q–polynomial C, given Y. Using the analogy with Reed–Solomon codes, Loidreau introduced in [Loi06] a Welch–Berlekamp like algorithm

to decode Gabidulin codes that consists in finding the unique q–polynomial V of q–degree less than or equal to t such that V vanishes on the column support of $\boldsymbol{e}$, which is equivalent to $V \circ E = 0$, *i.e.* V is a left annihilator of the error. Using a linearisation technique, this leads to the resolution of a linear system that can be efficiently solved provided that t is less than half the minimum distance. It then suffices to compute a left Euclidean division to recover C and therefore the codeword $\boldsymbol{c}$.

The core of the algorithm to follow consists in searching a right–hand side annihilator of E instead of a left–hand one. Due to the non commutativity of the ring $\mathcal{L}$, working on the right–hand side is not directly equivalent to working on the left–hand side, and as we will see in the sequel.

We begin to state the existence of a right annihilator.

Proposition 1. *Let E be a q–polynomial of rank t. Then there exists a unique monic q–polynomial V with $\deg_q(V) \leqslant t$ such that $E \circ V = 0$ modulo $(X^{q^m} - X)$.*

Proof. Let $Q \stackrel{\text{def}}{=} \sum_{i=0}^{t} a_i X^{q^i}$ be the unique monic q–polynomial of q–degree less than or equal to t that vanishes exactly on $\mathrm{Im}(E^{\vee})$, *i.e.* $\mathrm{Im}(E^{\vee}) = \mathrm{Ker} Q$. Such a q–polynomial is guaranteed to exist (see for instance [Ber15] or [Ore33].) It follows that $\mathrm{Ker}(E) = \mathrm{Im}(Q^{\vee})$. Moreover,

$$Q^{\vee} = \sum_{i=0}^{t} a_i^{q^{m-i}} X^{q^{m-i}} = \left(\sum_{i=0}^{t} a_{t-i}^{q^{m-t+i}} X^{q^i} \right) \circ X^{q^{m-t}}.$$

Let V be the left–hand factor of $Q^{\vee}$ in the above decomposition. It is a q–polynomial of q–degree t, and $E \circ Q^{\vee} = 0$ leads to $E \circ V \circ X^{q^{m-t}} = 0$. Since $X^{q^{m-t}}$ is invertible in $\mathcal{L}/(X^{q^m} - X)$, we get $E \circ V = 0 \mod (X^{q^m} - X)$. □

The goal is to compute this right annihilator V. It satisfies

$$Y \circ V = C \circ V + E \circ V \equiv C \circ V \mod (X^{q^m} - X). \tag{7}$$

Equation (7) leads to a non linear system of n equations whose variables are the $t + k + 1$ unknown coefficients of C and V.

$$\begin{cases} (Y \circ V)(g_i) = C \circ V(g_i) \\ \deg_q V \leqslant t \\ \deg_q C \leqslant k - 1. \end{cases} \tag{8}$$

Due to the non linearity, it is not clear how this can efficiently be solved. That is why we consider instead the following linearised system

$$\begin{cases} (Y \circ V)(g_i) = N(g_i) \\ \deg_q V \leqslant t \\ \deg_q N \leqslant k + t - 1, \end{cases} \tag{9}$$

whose unknowns are the $k + 2t + 1$ coefficients of N and V. The latter is *a priori* more general than the former. But we can link the set of solutions of the two systems. This is specified in the following two propositions.

Proposition 2. *Any solution* (V, C) *of* (8) *gives a solution* $(V, N = C \circ V)$ *of* (9).

Proof. This is the direct analogue of [Loi06, Proposition 1]. □

Proposition 3. *Assume that* E *is of rank* $t \leqslant \lfloor \frac{n-k}{2} \rfloor$. *If* (V, N) *is a nonzero solution of* (9) *then* $N = C \circ V$ *where* $C = Y - E$ *is the interpolating* q*–polynomial of the codeword.*

Proof. Let $(V, N) \neq (0, 0)$ be a solution of (9), and let C be the q–polynomial of q–degree strictly less than k that interpolates the codeword. Let $R \stackrel{\text{def}}{=} N - C \circ V$. It is a q–polynomial, of q–degree at most $k - 1 + t$. Assume that $R \neq 0$. Then,

$$(Y - C) \circ V = Y \circ V - C \circ V = N - C \circ V \equiv R \mod (X^{q^m} - X)$$

i.e.

$$E \circ V \equiv R \mod (X^{q^m} - X). \tag{10}$$

Hence, $\mathbf{rank}_q(R) \leqslant \mathbf{rank}_q(E) \leqslant t$. Since $R \neq 0$, $\deg_q R \geqslant \dim \mathrm{Ker} R$. Therefore, by the rank–nullity theorem,

$$n = \dim \mathrm{Ker} R + \mathbf{rank}_q(R) \leqslant \deg_q R + \mathbf{rank}_q(R) \leqslant k - 1 + 2t \leqslant n - 1 < n$$

which is a contradiction. Therefore, R must be zero, *i.e.* $N = C \circ V$. □

Thenceforth, whenever $t \leqslant \lfloor \frac{n-k}{2} \rfloor$, any non zero solution of (9) allows to recover the codeword by simply computing a right Euclidean division, which can be done efficiently (see [Ore33]). The decoding process boils down to solving the system of equations (9). However, despite the transformation, the system is only semi-linear over $\mathbb{F}_{q^m}$. To solve the problem, we will again use the adjoint of a (class of) q–polynomial. Let ${y_i}^\vee \stackrel{\text{def}}{=} Y^\vee(g_i)$, for all $i = 1, \ldots, n$. Using the anticommutativity of the adjoint operator, system (9) is equivalent to

$$V^\vee({y_i}^\vee) = N^\vee(g_i) \text{ for } i = 1, \ldots, n. \tag{11}$$

which is now an $\mathbb{F}_{q^m}$–linear system of n equations whose unknowns are the coefficients of $V^\vee$ and $N^\vee$, that are in explicit one-to-one correspondence with the coefficients of V and N.

An implementation of this algorithm using SageMath v9.2 [S+20] can be found on Github: `https://github.com/mbombar/Attack_on_LIGA`.

Remark 2. This right–hand side algorithm can be generalised in order to decode an u–interleaved Gabidulin code (see for instance [WZ14] for further reference about interleaved Gabidulin codes and their decoding algorithms). Indeed, let $\boldsymbol{Y} = \boldsymbol{C} + \boldsymbol{E}$ be a corrupted codeword of an u–interleaved Gabidulin code, with $\boldsymbol{E} \in \mathbb{F}_{q^m}^{u \times n}$ being an error matrix of $\mathbb{F}_q$-rank equal to t. Then, the rows of $\boldsymbol{E}$ as seen as vectors of $\mathbb{F}_{q^m}^n$ share a common *row* support, namely the *row* support

Algorithm 3: Right–hand side variant of Welch–Berlekamp

Input: q a prime power, k, n, m integers, $\boldsymbol{g} = (g_1, \ldots, g_n)$ a basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$, $\mathscr{C}$ a Gabidulin code of dimension k and support $\boldsymbol{g}$, $t \leqslant \lfloor \frac{n-k}{2} \rfloor$ an integer, $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$.

Output: $\boldsymbol{c} \in \mathscr{C}$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ for some $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ with $\mathbf{rank}_q(\boldsymbol{e}) \leqslant t$.

1 Find Y the q–polynomial of q–degree strictly less than n such that $Y(g_i) = y_i$

2 Compute $Y^{\vee}$ and evaluate in $\boldsymbol{g}$ to get $\boldsymbol{y}^{\vee} \stackrel{\text{def}}{=} Y^{\vee}(\boldsymbol{g}) \in \mathbb{F}_{q^m}^n$

3 Find a non zero solution (V_0, N_0) of the linear system (11)

4 Compute $V \stackrel{\text{def}}{=} {V_0}^{\vee}$ and $N \stackrel{\text{def}}{=} {N_0}^{\vee}$

5 Recover C by computing the right–hand side Euclidean division of N by V

6 **return** $\boldsymbol{c} \stackrel{def}{=} C(\boldsymbol{g})$

of $\boldsymbol{E}$, of dimension t. Hence, they also share a common right annihilator, of q–degree at most t. The algorithm from [LO06, §4] where the errors shared a common *column* support can be adapted straightforwardly in this setting, and allows to decode almost all error matrix $\boldsymbol{E}$ of rank-weight $t \leqslant \lfloor \frac{u}{u+1}(n-k) \rfloor$. This can be used in order to attack the original Faure–Loidreau cryptosystem in the same fashion as [RPWZ20, §3].

4 Decoding supercodes of Gabidulin codes

A common feature of the cryptanalyses to follow can be understood as the decoding of a supercode of a Gabidulin code. Consider a code (represented as a subspace of $\mathcal{L}_{<m}$)

$$\mathscr{C} \stackrel{\text{def}}{=} \mathcal{L}_{<k} \oplus \mathscr{T},$$

where $\mathscr{T} \subseteq \mathcal{L}_{<m}$, the code $\mathscr{C}$ benefits from a decoding algorithm in a similar manner to that of [Loi06]. Indeed, given a received word

$$Y = C + E$$

where $C \in \mathscr{C}$ and $E \in \mathcal{L}_{<m}$ with $\mathbf{rank}_q E = t$, one can look for the left annihilator of E. Let $\Lambda \in \mathcal{L}_{\leqslant t}$ be the left annihilator of E. We have to solve

$$\Lambda \circ Y \equiv \Lambda \circ C \mod (X^{q^m} - X),$$

where the unknowns are Λ, C. Then, similarly to the decoding of Gabidulin codes, one may linearise the system. For this sake, recall that $C = C_0 + T$ for $C_0 \in \mathcal{L}_{<k}$ and $T \in \mathscr{T}$. Therefore, we are looking for the solutions of a system

$$\Lambda \circ Y \equiv N \mod (X^{q^m} - X) \tag{12}$$

where $N \in (\mathcal{L}_{\leqslant t} \circ \mathcal{L}_{<k}) + \mathcal{L}_{\leqslant t} \circ \mathscr{T} = \mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}$.

Lemma 1. *Under the assumption that* $(\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}) \cap (\mathcal{L}_{\leqslant t} \circ E) = \{0\}$, *any nonzero solution* (Λ, N) *of the system (12) satisfies* $\Lambda \circ E = 0$.

Proof. Let (Λ, N) be such a nonzero solution. Then,

$$\Lambda \circ Y - \Lambda \circ C \equiv \Lambda \circ E \mod (X^{q^m} - X).$$

Since $\Lambda \circ Y \equiv N \mod (X^{q^m} - X)$, the left–hand side is contained in $(\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T})$ while the right–hand one is contained in $\mathcal{L}_{\leqslant t} \circ E$. Therefore, by assumption, both sides are zero. This yields the result. □

Under the hypotheses of Lemma 1, decoding can be performed as follows.

1. Solve System (12).
2. Take a nonzero solution (Λ, N) of the system. Compute the right kernel of Λ. This kernel contains the image of E and hence the support of the error.
3. Knowing the support of E, one can recover it by solving a linear system. See for instance [GRS13, §3] or [AGHT17, §1.4].

Remark 3. Note that for the decoding of Gabidulin codes, once a solution (Λ, N) is computed, one can recover C by left Euclidean division of N by Λ. In the present situation, this calculation is no longer efficient. Indeed, the proof of Lemma 1 permits only to assert that $N \equiv \Lambda \circ C \mod (X^{q^m} - X)$. In the Gabidulin case, the fact that $\deg_q C < k$ permits to assert that $\deg_q \Lambda \circ C < m$ and hence that $N = \Lambda \circ C$. This is no longer true in our setting since, there is *a priori* no upper bound on the q–degree of C. For this reason, we need to use the knowledge of the support of the error to decode.

For the decoding to succeed, the condition $(\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}) \cap (\mathcal{L}_{\leqslant t} \circ E) = \{0\}$ needs to be satisfied. In the case of Gabidulin codes (*i.e.* $\mathscr{T} = \{0\}$), this is guaranteed by a minimum distance argument entailing that $\mathcal{L}_{<k+t} \cap \mathcal{L}_{\leqslant t} \circ E$ is zero as soon as $t \leqslant \frac{n-k}{2}$. In our situation, estimating the minimum distance of $\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}$ is difficult. However, one can expect that in the typical case, the intersection $(\mathcal{L}_{<k+t} + \mathcal{L}_{\leqslant t} \circ \mathscr{T}) \cap (\mathcal{L}_{\leqslant t} \circ E)$ is 0 when the sums of the dimensions of the codes is less than that of the ambient space. Therefore, one can reasonably expect to correct almost any error of rank t as soon as

$$k + 2t + \dim(\mathcal{L}_{\leqslant t} \circ \mathscr{T}) \leqslant n. \tag{13}$$

In the case $\mathscr{T} = \{0\}$, we find back the decoding radius of Gabidulin codes.

Remark 4. Note that the previous approach applies *mutatis mutandis* to the decoding of supercodes of Reed–Solomon codes.

The right–hand side version. In the spirit of Section 3, a similar approach using right–hand side decoding shows that decoding is also possible when

$$k + 2t + \dim(\mathscr{T} \circ \mathcal{L}_{\leqslant t}) \leqslant n. \tag{14}$$

We will use this decoding algorithm to attack Liga and Ramesses. For Liga, the code $\mathscr{T}$ is a random code of low dimension, while for Ramesses, $\mathscr{T}$ is a Gabidulin code.

5 Application to cryptanalysis

5.1 Ramesses

The decoder introduced in Section 4 is the key of our cryptanalysis of Ramesses. Using the notation of Section 2.2, suppose we have a ciphertext as in (5):

$$Y = C + E \quad \text{with} \quad C = C_1 + K_{\text{sec}} \circ T,$$

where $C_1 \in \mathcal{L}_{<k+\ell}$, $T \in \mathcal{L}_\ell$ and $E \in \mathcal{L}_{<m}$ of rank t. Recall that the plaintext is the row space of E (equivalently, the image of $E^\vee$). We perform the right–hand side version of the decoding algorithm of Section 4. Here the code $\mathscr{T}$ is $K_{\text{sec}} \circ \mathcal{L}_{\leqslant \ell}$ and the supercode $\mathscr{C}$ is $\mathcal{L}_{<k} + \mathscr{T}$. We compute the solutions (Λ, N) of the system

$$Y \circ \Lambda \equiv N \mod (X^{q^m} - X),$$

where

$$N \in \mathscr{C} \circ \mathcal{L}_{\leqslant t} = \mathcal{L}_{<k+t} + K_{\text{sec}} \circ \mathcal{L}_{\leqslant \ell+t}.$$

According to Lemma 1, the algorithm will very likely return pairs of the form $(\Lambda, C \circ \Lambda)$ with $E \circ \Lambda = 0$ as soon as

$$k + 2t + \dim(\mathscr{T} \circ \mathcal{L}_{\leqslant t}) = k + 2t + \dim(K_{\text{sec}} \circ \mathcal{L}_{\leqslant t+\ell}) = k + 3t + \ell + 1 \leqslant n. \quad (15)$$

Once such a Λ is obtained, one recovers E and the image of $E^\vee$ yields the plaintext.

A comparison of (15) with the proposed parameters for Ramesses in [LLP20, Section 4] is given in Table 1. As observed, inequality (15) is satisfied for any proposed parameter set.

$m\ (=n)$	k	w	ℓ	t	Security (bits)	$k+3t+\ell+1$
64	32	19	3	5	141	51
80	40	23	3	7	202	65
96	48	27	3	9	265	79
164	116	27	3	9	256	147

Table 1. This table compares the values of the formula (15) with the parameters proposed for Ramesses. The first three rows are parameters for Ramesses as a KEM and the last one are parameters for Ramesses as a PKE. Note that for any proposed parameter set, we have $m = n$.

5.2 A message recovery attack against Liga cryptosystem

In this section, we show that it is possible to recover the plaintext from a ciphertext. Notice that Liga cryptosystem has been proven IND-CCA2 in [RPWZ20],

under some computational assumption, namely the Restricted Gabidulin Code Decision Problem ([RPWZ20], Problem 4). We are not disproving this claim here, however our attack can be precisely used as a distinguisher, and hence this problem is not as hard as supposed.

Recall that $\boldsymbol{G}$ is a generator matrix of a public Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$, the public key is a noisy vector $\boldsymbol{k}_{pub} = \boldsymbol{x} \cdot \boldsymbol{G} + \boldsymbol{z}$ and the encryption of a message $\boldsymbol{m}$ is $\boldsymbol{c} = \boldsymbol{m} \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub}) + \boldsymbol{e}$ for some uniformly random element $\alpha \in \mathbb{F}_{q^{mu}}$ and a uniformly random error $\boldsymbol{e}$ of small rank weight $t_{pub} = \lfloor \frac{n-k-w}{2} \rfloor$ both chosen by Alice. See Section 2.1 for further details.

The attack works in two parts. First, we introduce a supercode of the public Gabidulin code, in which we are able to decode the ciphertext and get rid of the small error. Then, we recover the plaintext.

Step 1: Get rid of the small error. Let $\zeta \stackrel{\text{def}}{=} \mathbf{rank}_{q^m}(\boldsymbol{z})$, so that $\boldsymbol{z} = \sum_{i=1}^{\zeta} \mu_i \boldsymbol{z}_i$ where the μ_i's $\in \mathbb{F}_{q^{mu}}$ and the $\boldsymbol{z}_i$'s $\in \mathbb{F}_{q^m}^n$ are both linearly independent over $\mathbb{F}_{q^m}$. The ciphertext can now be written as

$$\boldsymbol{c} = \boldsymbol{m} \cdot \boldsymbol{G} + \sum_{i=1}^{\zeta} \mathrm{Tr}_{q^{mu}/q^m}(\alpha \mu_i) \boldsymbol{z}_i + \boldsymbol{e} \tag{16}$$

Let

$$\mathscr{C} \stackrel{\text{def}}{=} \mathscr{G}_k(\boldsymbol{g}) + \mathbf{Span}_{\mathbb{F}_{q^m}}\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_\zeta\} \subseteq \mathbb{F}_{q^m}^n.$$

The ciphertext can be seen as a codeword of $\mathscr{C}$ corrupted by a small rank weight error $\boldsymbol{e}$. Moreover, $\mathscr{C}$ can be computed from public data as suggested by the following statement.

Theorem 1. *Let $\mathscr{C}$ be the code defined in (16) and $\mathscr{C}_{pub}$ be the code generated by $\mathscr{G}_k(\boldsymbol{g})$ and $\mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{k}_{pub})$ for $i \in \{1, \ldots, \zeta\}$, where the γ_i's denote ζ elements of $\mathbb{F}_{q^{mu}}$ linearly independent over $\mathbb{F}_{q^m}$. Then, for a uniformly random choice of $(\gamma_1, \ldots, \gamma_\zeta)$,*

$$\mathbb{P}(\mathscr{C} = \mathscr{C}_{pub}) = 1 - e^{O\left(\frac{1}{q^m}\right)}.$$

The proof of Theorem 1 rests on the following technical lemma.

Lemma 2. *Let F be a linear subspace of dimension m in a linear space E of dimension n over a finite field $\mathbb{F}_q$. Then, $\#\{G \mid F \oplus G = E\} = q^{m(n-m)}$.*

Proof. Let $\mathrm{Stab}(F)$ denote the stabiliser of F under the action of $\mathrm{GL}(E)$. It is isomorphic to the group of the matrices of the form $\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{C} \end{pmatrix}$ with $\boldsymbol{A} \in \mathrm{GL}_m(\mathbb{F}_q)$, $\boldsymbol{C} \in \mathrm{GL}_{n-m}(\mathbb{F}_q)$ and $\boldsymbol{B} \in \mathcal{M}_{m,n-m}(\mathbb{F}_q)$, *i.e.*

$$\mathrm{Stab}(F) \cong (\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q)) \ltimes \mathcal{M}_{m,n-m}(\mathbb{F}_q).$$

This group acts transitively on the complement spaces of F. Indeed, let G and G' be such that $F \oplus G = F \oplus G' = E$. Let $(f_1, \ldots, f_m)$ be a basis of F and

$(g_1, \ldots, g_{n-m})$ (respectively $(g'_1, \ldots, g'_{n-m})$) be a basis of G (resp. G'). Then the linear map that stabilises F and maps g_i onto g'_i is an element of $\mathrm{Stab}(F)$ that maps G onto G'. The stabiliser of a complement G under this action is simply $\mathrm{GL}_m(\mathbb{F}_q) \times \mathrm{GL}_{n-m}(\mathbb{F}_q)$. Hence,

$$\#\{G \mid F \oplus G = E\} = \frac{(\#\mathrm{GL}_m(\mathbb{F}_q)) \times (\#\mathrm{GL}_{n-m}(\mathbb{F}_q)) \times q^{m(n-m)}}{(\#\mathrm{GL}_m(\mathbb{F}_q)) \times (\#\mathrm{GL}_{n-m}(\mathbb{F}_q))} = q^{m(n-m)}.$$

□

Proof of Theorem 1. Inclusion $\supseteq$ is always satisfied. Indeed, let $\boldsymbol{c} \in \mathscr{C}_{pub}$. Then, there exist $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$ and $\lambda_1, \ldots, \lambda_\zeta \in \mathbb{F}_{q^m}$ such that

$$\begin{aligned}\boldsymbol{c} &= \boldsymbol{m}\boldsymbol{G} + \sum_{i=1}^{\zeta} \lambda_i \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{k}_{pub}) \\ &= \left(\boldsymbol{m} + \sum_{i=1}^{\zeta} \lambda_i \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{x})\right) \boldsymbol{G} + \sum_{i=1}^{\zeta}\sum_{j=1}^{\zeta} \lambda_j \mathrm{Tr}_{q^{mu}/q^m}(\gamma_j \mu_i) \boldsymbol{z}_i\end{aligned}$$

and $\boldsymbol{c} \in \mathscr{C}$.

Now, let us discuss inclusion $\subseteq$. Let $\boldsymbol{c} \in \mathscr{C}$. There exists $\boldsymbol{m} \in \mathbb{F}_{q^m}^k$ and $\lambda_1, \ldots, \lambda_\zeta \in \mathbb{F}_{q^m}$ such that

$$\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G} + \sum_{i=1}^{\zeta} \lambda_i \boldsymbol{z}_i.$$

If we can find $\alpha \stackrel{\text{def}}{=} (\alpha_1, \ldots, \alpha_\zeta) \in \mathbb{F}_{q^m}^\zeta$ such that $\boldsymbol{c} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{k}_{pub}) \in \mathscr{G}_k(\boldsymbol{g})$, then we are done.

$$\boldsymbol{c} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{k}_{pub}) =$$
$$\left(\boldsymbol{m} - \sum_{i=1}^{\zeta} \alpha_i \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{x})\right) \boldsymbol{G} + \sum_{i=1}^{\zeta} \left(\lambda_i - \sum_{j=1}^{\zeta} \alpha_j \mathrm{Tr}_{q^{mu}/q^m}(\gamma_j \mu_i)\right) \boldsymbol{z}_i.$$

It suffices to choose α such that $\lambda_i - \sum_{j=1}^{\zeta} \alpha_j \mathrm{Tr}_{q^{mu}/q^m}(\gamma_j \mu_i) = 0$ for $i \in \{1, \ldots, \zeta\}$, *i.e.*

$$(\lambda_1, \ldots, \lambda_\zeta) = (\alpha_1, \ldots, \alpha_\zeta) \begin{pmatrix} \mathrm{Tr}_{q^{mu}/q^m}(\gamma_1\mu_1) & \cdots & \mathrm{Tr}_{q^{mu}/q^m}(\gamma_1\mu_\zeta) \\ \vdots & \ddots & \vdots \\ \mathrm{Tr}_{q^{mu}/q^m}(\gamma_\zeta\mu_1) & \cdots & \mathrm{Tr}_{q^{mu}/q^m}(\gamma_\zeta\mu_\zeta) \end{pmatrix}.$$

Let $\boldsymbol{M}$ denote this last matrix. We can prove that $\boldsymbol{M}$ is non singular with overwhelming probability over the choice of $\gamma_1, \ldots, \gamma_\zeta$. Indeed, let

$$\Gamma \stackrel{\text{def}}{=} \mathbf{Span}(\gamma_1, \ldots, \gamma_\zeta) \quad \text{and} \quad \mathscr{M} \stackrel{\text{def}}{=} \mathbf{Span}(\mu_1, \ldots, \mu_\zeta).$$

Then, $\boldsymbol{M}$ is singular if and only if $\Gamma \cap \mathscr{M}^\perp \neq \{0\}$. Since Γ and $\mathscr{M}$ have the same dimension ζ over $\mathbb{F}_{q^m}$, $\Gamma \cap \mathscr{M}^\perp = \{0\}$ if and only if $\Gamma \oplus \mathscr{M}^\perp = \mathbb{F}_{q^{mu}}$. Therefore,

$$\mathbb{P}(\boldsymbol{M} \text{ is non singular }) = \frac{\#\{\Gamma \mid \mathscr{M}^\perp \oplus \Gamma = \mathbb{F}_{q^{mu}}\}}{\#\{\Gamma \mid \dim_{\mathbb{F}_{q^m}}(\Gamma) = \zeta\}}.$$

Recall the Gaussian binomial coefficient $\begin{bmatrix} u \\ \zeta \end{bmatrix}_{q^m}$ denotes the number of $\mathbb{F}_{q^m}$–linear subspaces of dimension ζ in an $\mathbb{F}_{q^m}$–vector space of dimension u. Applying Lemma 2, we have

$$\mathbb{P}(\boldsymbol{M} \text{ is non singular}) = \frac{q^{m\zeta(u-\zeta)}}{\begin{bmatrix} u \\ \zeta \end{bmatrix}_{q^m}} \geqslant \left(1 - \frac{1}{q^m}\right)^{\frac{q^m}{q^m-1}},$$

where the inequality on the right–hand side can be found for instance in [CC19, Appendix A]. □

Set

$$\mathscr{T} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{\zeta} \mathbf{Span}_{\mathbb{F}_{q^m}} \left\{ \mathrm{Tr}_{q^{mu}/q^m}(\gamma_i \boldsymbol{k}_{pub}) \right\}.$$

By interpolation, it can be regarded as a subspace of $\mathcal{L}_{<m}$, and $\mathscr{C}_{pub} = \mathcal{L}_{<k} \oplus \mathscr{T}$. In order to remove the error $\boldsymbol{e}$ we just need to decode in this public supercode. Notice that

$$\dim(\mathcal{L}_{\leqslant t} \circ \mathscr{T}) \leqslant \zeta(t+1).$$

Therefore, using the algorithm of Section 4, one can expect to decode in $\mathscr{C}_{pub}$ whenever

$$k + 2t + \zeta(t+1) \leqslant n. \tag{17}$$

Table 2 compares (17) with the proposed parameters for Liga in [RPWZ20, Section 7]. As observed, Inequality (17) is satisfied for any proposed parameter set. Moreover, if one tries to increase ζ in order to avoid this attack, one also needs to increase w to resist the key recovery attack from [GOTK18], which decreases $t \stackrel{\text{def}}{=} \lfloor \frac{n-k-w}{2} \rfloor$ that must be greater than 1.

Name	n	k	t	ζ	Security (bits)	$k + 2t + \zeta(t+1)$
Liga-128	92	53	6	2	128	79
Liga-192	120	69	8	2	192	103
Liga-256	148	85	10	2	256	127

Table 2. This table compares the values of the formula (17) with the parameters proposed for Liga.

Step 1 is summed up in the following proposition.

Proposition 4. *If $\boldsymbol{c} = \boldsymbol{m} \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub}) + \boldsymbol{e}$ is the encryption of a plaintext $\boldsymbol{m}$, then we can recover the support of the error $\boldsymbol{e}$ and the corrupted codeword $\boldsymbol{m} \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub})$ in polynomial time using only the knowledge of the public key.*

Step 2: Remove the $\boldsymbol{z}$ dependency. From now on, since we got rid of the small error term $\boldsymbol{e}$, we can do as if the ciphertext was

$$\begin{aligned} \boldsymbol{c}' &\stackrel{\text{def}}{=} \boldsymbol{m} \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub}) \\ &= (\boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{x})) \cdot \boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{z}). \end{aligned} \tag{18}$$

This is a codeword of a Gabidulin code $\mathscr{G} \stackrel{\text{def}}{=} \mathscr{G}_k(\boldsymbol{g})$, corrupted by an error of rank $w > \lfloor \frac{n-k}{2} \rfloor$. Hence, we cannot decode in $\mathscr{G}$ to recover the plaintext. However, thanks to the knowledge of the public key, one can easily recover the affine space

$$A \stackrel{\text{def}}{=} \{\beta \in \mathbb{F}_{q^{mu}} \mid \boldsymbol{c}' - \mathrm{Tr}_{q^{mu}/q^m}(\beta \boldsymbol{k}_{pub}) \in \mathscr{G}\}$$

using linear algebra.

Lemma 3. *Let $\beta \in \mathbb{F}_{q^{mu}}$. Then $\boldsymbol{c}' - \mathrm{Tr}_{q^{mu}/q^m}(\beta \boldsymbol{k}_{pub}) \in \mathscr{G}$ if and only if $\mathrm{Tr}_{q^{mu}/q^m}((\alpha - \beta)\boldsymbol{z}) = 0$.*

Proof. Note that for any $\lambda \in \mathbb{F}_{q^{mu}}$,

$$\mathbf{rank}_q(\mathrm{Tr}_{q^{mu}/q^m}(\lambda \boldsymbol{z})) \leqslant \mathbf{rank}_q(\boldsymbol{z}) = w < n - k.$$

Indeed, let $\mathscr{B}$ be a basis of the extension field $\mathbb{F}_{q^{mu}}/\mathbb{F}_q$. Then, if $\lambda \neq 0$, the extension of $\lambda \boldsymbol{z}$ in $\mathscr{B}$ is the extension of $\boldsymbol{z}$ in the basis $\lambda \mathscr{B}$. Therefore,

$$\mathrm{RowSupp}(\lambda \boldsymbol{z}) = \mathrm{RowSupp}(\boldsymbol{z})$$

and the trace cannot increase the rank.
Let $\beta \in \mathbb{F}_{q^{mu}}$. Then

$$\boldsymbol{c}' - \mathrm{Tr}_{q^{mu}/q^m}(\beta \boldsymbol{k}_{pub}) = (\boldsymbol{m} + \mathrm{Tr}_{q^{mu}/q^m}((\alpha - \beta)\boldsymbol{x}))\boldsymbol{G} + \mathrm{Tr}_{q^{mu}/q^m}((\alpha - \beta)\boldsymbol{z}).$$

Therefore, $\beta \in A$ if and only if $\mathrm{Tr}_{q^{mu}/q^m}((\alpha - \beta)\boldsymbol{z}) \in \mathscr{G}$. Since it has rank weight less than the minimum distance of $\mathscr{G}$, it follows that $\mathrm{Tr}_{q^{mu}/q^m}((\alpha - \beta)\boldsymbol{z}) = 0$. □

Lemma 4. *Let $\mathscr{E} \stackrel{def}{=} \bigcap_{i=1}^{\zeta} \langle \mu_i \rangle^{\perp}$. Then A is the affine space $\alpha + \mathscr{E}$.*

Proof. Let $\beta \in A$. Then,

$$\mathrm{Tr}_{q^{mu}/q^m}((\alpha-\beta)\boldsymbol{z}) = \sum_{i=1}^{\zeta} \mathrm{Tr}_{q^{mu}/q^m}((\alpha-\beta)\mu_i)\boldsymbol{z}_i = 0.$$

By the linear independence of the $\boldsymbol{z}_i$'s, it follows that $\mathrm{Tr}_{q^{mu}/q^m}((\alpha-\beta)\mu_i) = 0$ for all i, *i.e.*

$$A = \alpha + \bigcap_{i=1}^{\zeta} \langle \mu_i \rangle^{\perp}.$$

□

We are now able to remove the $\boldsymbol{z}$ dependency in the ciphertext. Indeed, let $\mathscr{F} \stackrel{\text{def}}{=} \{\mathrm{Tr}_{q^{mu}/q^m}(\gamma \boldsymbol{x}) \mid \gamma \in \mathscr{E}\}$. The knowledge of A gives finally access to the affine space $\boldsymbol{m} + \mathscr{F}$.

Step 3: Recover the plaintext. Let $\boldsymbol{s}$ be some random element of $\boldsymbol{m} + \mathscr{F}$. Notice that from a description of the affine space $\boldsymbol{m}+\mathscr{F}$ it is possible to recover a generating set $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{u-1})$ of the parallel linear space $\mathscr{F}$ using linear algebra. Then, $\boldsymbol{s}$ can be decomposed as

$$\boldsymbol{s} \stackrel{\text{def}}{=} \boldsymbol{m} + \sum_{i=1}^{u-1} \lambda_i \boldsymbol{e}_i$$

for some unknown coefficients $\lambda_i \in \mathbb{F}_{q^m}$. Furthermore, recall that the last u positions of $\boldsymbol{m}$ are 0. Then, $\boldsymbol{m}$ is a solution of the following linear system of $u+k-1$ unknowns and $u+k$ equations:

$$\begin{cases} \boldsymbol{m} + \sum_{i=1}^{u-1} \lambda_i \boldsymbol{e}_i = \boldsymbol{s} \\ \boldsymbol{m}_{k-u+1} = \cdots = \boldsymbol{m}_k = 0 \end{cases} \tag{19}$$

Finally, the following lemma shows that $\boldsymbol{m}$ can be recovered from *any* solution of (19).

Lemma 5. *Let $(\boldsymbol{m}', \lambda')$ be another solution of (19). Then $\boldsymbol{m}' = \boldsymbol{m}$.*

Proof. Since $\boldsymbol{m} - \boldsymbol{m}' = \sum_{i=1}^{u-1}(\lambda_i' - \lambda_i)\boldsymbol{e}_i \in \mathscr{F}$, it is of the form $\mathrm{Tr}_{q^{mu}/q^m}(\gamma\boldsymbol{x})$ for some $\gamma \in \mathscr{E}$. Moreover, its last u positions are 0. Recall that $(\boldsymbol{x}_{k-u+1}, \ldots, \boldsymbol{x}_k)$ is a basis of $\mathbb{F}_{q^{mu}}/\mathbb{F}_{q^m}$. Then, the last u positions of $\mathrm{Tr}_{q^{mu}/q^m}(\gamma\boldsymbol{x})$ are the coefficients of γ in the dual basis $\{\boldsymbol{x}^*_{k-u+1}, \ldots, \boldsymbol{x}^*_k\}$. Hence, $\gamma = 0$ and $\boldsymbol{m} = \boldsymbol{m}'$. □

Summary of the attack

- Decode in a public supercode of a Gabidulin code to get rid of the small error $\boldsymbol{e}$ and recover $\boldsymbol{c}' = \boldsymbol{mG} + \mathrm{Tr}_{q^{mu}/q^m}(\alpha \boldsymbol{k}_{pub})$.
- Using linear algebra, deduce the affine space

$$A = \{\beta \in \mathbb{F}_{q^{mu}} \mid \boldsymbol{c}' - \mathrm{Tr}_{q^{mu}/q^m}(\beta \boldsymbol{k}_{pub}) \ \in \ \mathscr{G}\}.$$

- Recover the affine space $\boldsymbol{m} + \mathscr{F}$ where $\mathscr{F} = \{\mathrm{Tr}_{q^{mu}/q^m}(\gamma \boldsymbol{x}) \mid \alpha + \gamma \in A\}$.
- Deduce a basis of $\mathscr{F}$.
- Solve linear system (19) to recover the plaintext $\boldsymbol{m}$.

Implementation. Tests have been done using SageMath v9.2 [S+20] on an Intel® Core™ i5-10310U CPU. We are able to recover the plaintext on the three LIGA proposals. The average running times are listed in Table 3. Our implementation is available on Github `https://github.com/mbombar/Attack_on_LIGA`.

Name	Parameters $(\mathbf{q}, \mathbf{n}, \mathbf{m}, \mathbf{k}, \mathbf{w}, \mathbf{u}, \zeta)$	Claimed security level	Average running time
LIGA-128	$(2, 92, 92, 53, 27, 5, 2)$	128 bits	8 minutes
LIGA-192	$(2, 120, 120, 69, 35, 5, 2)$	192 bits	27 minutes
LIGA-256	$(2, 148, 148, 85, 43, 5, 2)$	256 bits	92 minutes

Table 3. Average running times for the attack on LIGA.

References

ACLN20. Daniel Augot, Alain Couvreur, Julien Lavauzelle, and Alessandro Neri. Rank-metric codes over arbitrary Galois extensions and rank analogues of Reed-Muller codes. 26 pages, 1 figure, `https://hal.archives-ouvertes.fr/hal-02882019`, June 2020.

AF03. Daniel Augot and Matthieu Finiasz. A public key encryption scheme based on the polynomial reconstruction problem. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 229–240. Springer, 2003.

AGHT17. Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. Improvement of Generic Attacks on the Rank Syndrome Decoding Problem. working paper or preprint, October 2017.

Ber15. Elwyn R. Berlekamp. *Algebraic Coding Theory (Revised Edition)*, volume 10.1142/9407. World Scientific, 2015.

CC19. Daniel Coggia and Alain Couvreur. On the security of a Loidreau's rank metric code based encryption scheme. In *WCC 2019 - Workshop on Coding Theory and Cryptography*, Saint Jacut de la mer, France, March 2019.

Cor03. Jean-Sébastien Coron. Cryptanalysis of the repaired public-key encryption scheme based on the polynomial reconstruction problem. *IACR Cryptology ePrint Archive*, 2003:219, 2003.

FL05. Cédric Faure and Pierre Loidreau. A new public-key cryptosystem based on the problem of reconstructing p-polynomials. In *Coding and Cryptography, International Workshop, WCC 2005, Bergen, Norway, March 14-18, 2005. Revised Selected Papers*, pages 304–315, 2005.

Gos96. David Goss. *Basic Structures of Function Field arithmetic*, volume 35 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1996.

GOTK18. Philippe Gaborit, Ayoub Otmani, and Hervé Talé-Kalachi. Polynomial-time key recovery attack on the Faure-Loidreau scheme based on Gabidulin codes. *Des. Codes Cryptogr.*, 86(7):1391–1403, 2018.

GPT91. Ernst M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their applications to cryptography. In *Advances in Cryptology - EUROCRYPT'91*, number 547 in LNCS, pages 482–489, Brighton, April 1991.

GRS13. Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *CoRR*, abs/1301.1026, 2013.

LLP20. Julien Lavauzelle, Pierre Loidreau, and Ba-Duc Pham. RAMESSES, a Rank Metric Encryption Scheme with Short Keys. Working paper or preprint. Available online at ArXiv:1911.13119, January 2020.

LO06. Pierre Loidreau and Raphael Overbeck. Decoding rank errors beyond the error-correction capability. In *Proceedings of the Tenth International Workshop on Algebraic and Combinatorial Coding Theory, ACCT-10*, pages 168–190, 2006.

Loi06. Pierre Loidreau. A Welch–Berlekamp like algorithm for decoding Gabidulin codes. In Øyvind Ytrehus, editor, *Coding and Cryptography*, pages 36–45, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

McE78. Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.

Ore33. Oystein Ore. On a special class of polynomials. *Trans. Amer. Math. Soc.*, 35(3):559–584, 1933.

RPWZ20. Julian Renner, Sven Puchinger, and Antonia Wachter-Zeh. LIGA: A cryptosystem based on the hardness of rank–metric list and interleaved decoding. Available online at ArXiv:1812.04892, 2020.

S$^+$20. W. A. Stein et al. *Sage Mathematics Software (Version 9.2)*. The Sage Development Team, 2020. `http://www.sagemath.org`.

WPR18. Antonia Wachter-Zeh, Sven Puchinger, and Julian Renner. Repairing the Faure-Loidreau public-key cryptosystem. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT*, pages 2426–2430, 2018.

WZ13. Antonia Wachter-Zeh. *Decoding of block and convolutional codes in rank metric*. Theses, Université Rennes 1, October 2013.

WZ14. Antonia Wachter-Zeh and Alexander Zeh. List and unique error-erasure decoding of interleaved gabidulin codes with interpolation techniques. *Des. Codes Cryptogr.*, 73(2):547–570, 2014.

A Further details about RAMESSES' specifications

As explained in Section 2.2, our presentation of RAMESSES may seem to differ from the original proposal [LLP20]. Indeed in Section 2.2, we present the scheme using only q–polynomials, while the original publication prefers using matrices

and vectors. The purpose of the present appendix is to prove that our way to present RAMESSES is equivalent to that of [LLP20].

Caution. In the present article we use q to denote the cardinality of the ground field $\mathbb{F}_q$. In [LLP20] the ground field is always supposed to be $\mathbb{F}_2$ and q refers to some power of 2, *i.e.* $q = 2^n$ for some positive n. Moreover, the exponent of q is denoted n while it is denoted m in the present article. This might be confusing while reading both papers in parallel.

The other notations w, k, ℓ, t are the same in the two articles. Finally, in [LLP20] a public $\mathbb{F}_q$–basis $\boldsymbol{g} = (g_1, \ldots, g_m)$ of $\mathbb{F}_{q^m}$ is fixed once for all. Our presentation does not require such a setting.

A.1 Key generation.

Recall that [LLP20] fixes a vector $\boldsymbol{g} \in \mathbb{F}_{q^m}^m$ of rank m (*i.e.* an $\mathbb{F}_q$–basis of $\mathbb{F}_{q^m}$). This data together with a parity–check matrix $\boldsymbol{H}$ of the Gabidulin code $\mathscr{G}_k(\boldsymbol{g})$ are public.

Original presentation The key generation consists in picking a uniformly random $\boldsymbol{k}_{\text{priv}} \in \mathbb{F}_{q^m}^m$ of weight w and the public key is its syndrome $\boldsymbol{k}_{pub} \stackrel{\text{def}}{=} \boldsymbol{H}\boldsymbol{k}_{\text{priv}}^\top$ with respect to the public Gabidulin code.

Our presentation Since the code $\mathscr{G}_k(\boldsymbol{g})$ is public, any of its elements may be associated to an element of $\mathcal{L}_{<k}$. The transition from codewords to q–polynomials is nothing but interpolation. Similarly, the choice of a vector $\boldsymbol{k}_{\text{priv}} \in \mathbb{F}_{q^m}^m$ of rank w is (again by interpolation) equivalent to that of a q–polynomial K of rank w. Finally, publishing its syndrome $\boldsymbol{H}\boldsymbol{k}_{\text{priv}}^\top$ is equivalent to publish the coset $\boldsymbol{k}_{\text{priv}} + \mathscr{G}_k(\boldsymbol{g})$, which in our setting is nothing but publishing the affine space $K_{\text{sec}} + \mathcal{L}_{<k}$.

A.2 Encryption

Original presentation The plain text is encoded into a matrix $\boldsymbol{P} \in \mathbb{F}_q^{m\times m}$ in row echelon form and of rank t.

- Compute $\boldsymbol{y} \in \mathbb{F}_{q^m}^m$ such that $\boldsymbol{H}\boldsymbol{y}^\top = \boldsymbol{k}_{pub}$
- Pick a uniformly random $\boldsymbol{T} \in \mathbb{F}_q^{m\times m}$ of $\boldsymbol{g}$–degree ℓ *i.e.* representing a q–polynomial of q–degree ℓ in the basis $\boldsymbol{g}$;
- Pick a uniformly random $\boldsymbol{S} \in \mathrm{GL}_m(\mathbb{F}_q)$.

The ciphertext is

$$\boldsymbol{u}^\top \stackrel{\text{def}}{=} \boldsymbol{H}(\boldsymbol{y}\boldsymbol{T} + \boldsymbol{g}\boldsymbol{S}\boldsymbol{P})^\top.$$

Our presentation The vector $\boldsymbol{u}$ is a syndrome of any word of the form:

$$\boldsymbol{yT} + \boldsymbol{gSP} + \boldsymbol{c},$$

where $\boldsymbol{c}$ ranges over $\mathscr{G}_k(\boldsymbol{g})$. From a q–polynomial point of view, such a word corresponds to:

$$(K_{\text{sec}} + C_0) \circ T + G \circ S \circ P + C,$$

where

- C, C_0 are arbitrary elements of $\mathcal{L}_{<k}$;
- $T \in \mathcal{L}_\ell$ is the interpolating polynomial of $\boldsymbol{T}$;
- and G, S, P are the respective interpolating polynomials of $\boldsymbol{g}, \boldsymbol{S}, \boldsymbol{P}$.

Note that, since $\boldsymbol{g}$ has rank m and $\boldsymbol{S}$ is supposed to be nonsingular, then their interpolating polynomials are invertible in $\mathcal{L}/(X^{q^m} - X)$. Hence, setting $E = G \circ S \circ P$, we get a q–polynomial whose matrix representation in basis $\boldsymbol{g}$ has the same row space as the matrix representation of P. Thus, we get the ciphertext description in (4).

A.3 Decryption

Original presentation Start by computing $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ such that $\boldsymbol{H}\boldsymbol{x}^\top = \boldsymbol{u}^\top$. Next, knowing $\boldsymbol{k}_{\text{priv}}$, one can compute an annihilator polynomial $V_{\boldsymbol{k}_{\text{priv}}} \in \mathcal{L}_w$ of the support of $\boldsymbol{k}_{\text{priv}}$. Then, compute $\boldsymbol{z} \stackrel{\text{def}}{=} V_{\boldsymbol{k}_{\text{priv}}}(\boldsymbol{x}) = (V_{\boldsymbol{k}_{\text{priv}}}(x_1), \ldots, V_{\boldsymbol{k}_{\text{priv}}}(x_m))$ and decode $\boldsymbol{z}$ as a corrupted codeword of $\mathscr{G}_{k+\ell+w}(\boldsymbol{g})$. If succeeds, it returns an error vector $\boldsymbol{a}$. If its rank equals t, then the row echelon form of $\text{Ext}_{\boldsymbol{g}}(\boldsymbol{a})$ yields $\boldsymbol{P}$.

Our presentation Similarly, the approach is based on applying $V_{\boldsymbol{k}_{\text{priv}}}$ and performing Gabidulin codes decoding. Indeed, starting from ciphertext (4), we apply $V = V_{\boldsymbol{k}_{\text{priv}}}$ and get

$$V \circ Y \equiv V \circ C_1 + V \circ E$$

and a decoding procedure returns $V \circ E$. If this q–polynomial has rank t, then the row echelon form of its matrix representation yields the plaintext $\boldsymbol{P}$.