



LIGA: a cryptosystem based on the hardness of rank-metric list and interleaved decoding

Julian Renner¹ · Sven Puchinger² · Antonia Wachter-Zeh¹

Received: 15 May 2020 / Revised: 4 December 2020 / Accepted: 25 February 2021 / Published online: 18 April 2021
© The Author(s) 2021

Abstract

We propose the new rank-metric code-based cryptosystem LIGA which is based on the hardness of list decoding and interleaved decoding of Gabidulin codes. LIGA is an improved variant of the *Faure–Loidreau* (FL) system, which was broken in a structural attack by *Gaborit, Otmani, and Talé Kalachi* (GOT, 2018). We keep the FL encryption and decryption algorithms, but modify the insecure key generation algorithm. Our crucial observation is that the GOT attack is equivalent to decoding an interleaved Gabidulin code. The new key generation algorithm constructs public keys for which all polynomial-time interleaved decoders fail—hence LIGA resists the GOT attack. We also prove that the public-key encryption version of LIGA is IND-CPA secure in the standard model and the key encapsulation mechanisms version is IND-CCA2 secure in the random oracle model, both under hardness assumptions of formally defined problems related to list decoding and interleaved decoding of Gabidulin codes. We propose and analyze various exponential-time attacks on these problems, calculate their work factors, and compare the resulting parameters to NIST proposals. The strengths of LIGA are short ciphertext sizes and (relatively) small key sizes. Further, LIGA guarantees correct decryption and has no decryption failure rate. It is *not* based on hiding the structure of a code. Since there are efficient and constant-time algorithms for encoding and decoding Gabidulin codes, timing attacks on the encryption and decryption algorithms can be easily prevented.

Keywords Code-based cryptography · Gabidulin codes · Interleaved decoding · List decoding · Rank-metric codes

Communicated by D. Panario.

✉ Julian Renner
julian.renner@tum.de

Sven Puchinger
svepu@dtu.dk

Antonia Wachter-Zeh
antonia.wachter-zeh@tum.de

¹ Institute for Communications Engineering, Technical University of Munich (TUM), Munich, Germany

² Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Kongens Lyngby, Denmark

Mathematics Subject Classification 11T71**1 Introduction**

Public-key cryptography is the foundation for establishing secure communication between multiple parties. Traditional public-key algorithms such as RSA are based on the hardness of factoring large numbers or the discrete logarithm problem, but can be attacked in polynomial time once a capable quantum computer exists. Code-based public-key cryptosystems are considered to be post-quantum secure, but compared to RSA or elliptic curve cryptography their crucial drawback is the significantly larger key size. Recently, the *National Institute of Standards and Technology* (NIST) has initiated a standardization progress for post-quantum secure public-key algorithms [31]. The currently being evaluated Round 3 of the competition includes 9 code-based and lattice-based public-key encryption algorithms. The NIST competition and its systems attract a lot of attention and show the importance of designing post-quantum secure public-key encryption algorithms.

The *Faure–Loidreau* (FL) code-based cryptosystem [16,28] is based on the problem of reconstructing linearized polynomials and can be seen as linearized equivalent of the (broken) Augot–Finiasz cryptosystem [5]. While the Augot–Finiasz cryptosystem is closely connected to (list) decoding Reed–Solomon codes, the FL cryptosystem is connected to (list) decoding Gabidulin codes, a special class of rank-metric codes [18]. In contrast to McEliece-type (or Niederreiter-type) cryptosystems, where the public key is a *matrix*, in the FL system, the public key is only a *vector*, resulting in a much smaller key size. At the time when the FL cryptosystem was designed, it was only *conjectured* that Gabidulin codes cannot be list decoded efficiently. As this was *proven* in the last years for many families of Gabidulin codes [38,46,48], the FL system could be a very promising post-quantum secure public-key cryptosystem. However, the recent structural attack by Gaborit, Otmani and Talé Kalachi [21] can recover an alternative public key in cubic time complexity.

In this paper, a new system is presented which is based on the original FL system, and therefore relies on the proven hardness of list decoding Gabidulin codes, but makes the attack from [21] impossible. Our contributions are as follows. First, a new coding-theoretic interpretation of the original FL system is given and an alternative decryption algorithm is proposed. Second, we show that the public key can be seen as a corrupted codeword of an *interleaved Gabidulin code*. We prove that the failure condition of the GOT attack [21] on the public key is equivalent to the failure condition of decoding the public key as a corrupted interleaved Gabidulin codeword. This observation enables us to design a new code-based public-key encryption scheme, as well as a corresponding key encapsulation mechanism - data encapsulation mechanism (KEM-DEM), based on the hardness of list and interleaved decoding Gabidulin codes: LIGA. In LIGA, we choose the public key in a way that the corresponding interleaved decoder is guaranteed to fail, and thus, the system is secured against the attack from [21]. We also prove that the public-key encryption version of LIGA is IND-CPA secure in the standard model and the KEM version is IND-CCA2 secure in the random oracle model, both under hardness assumptions on problems related to list and interleaved decoding of Gabidulin codes. We analyze possible (exponential-time) attacks on these hard problems, provide sets of parameters for LIGA, and compare them amongst others to NIST proposals (RQC, ROLLO, BIKE, McEliece).

The structure of this paper is as follows. In Sect. 2, the notation is introduced and definitions are given. In Sect. 3, the key generation of the original FL system is shown and a new coding-

theoretic interpretation of the ciphertext and the public key is derived. After summarizing the attack from [21], we prove its equivalence to decoding the public key as an interleaved Gabidulin code. Based on this equivalence, the new system LIGA is proposed in Sect. 4 and its IND-CPA and IND-CCA2 security are proven in Sect. 5. A security analysis of our system is given in Sect. 6. In Sect. 7, example parameters for security levels 128, 192, and 256 bit are proposed and compared to the NIST proposals RQC [2], ROLLO [1], BIKE [3], ClassicMcEliece [8] and Loidreau’s McEliece-like system from [29]. Conclusions are given in Sect. 8.

Parts of these results have been presented at the *IEEE International Symposium on Information Theory 2018* [50]. The content of this journal paper contains various new results that were not shown in [50]. For instance, in this paper,

- we generalize LIGA’s Key Generation algorithm, i.e., the choice of the \mathbf{z}_i ’s (the interleaved errors in the public key) is more flexible now (in [50], $\mathbf{z}_1 = \mathbf{z}_2 = \dots = \mathbf{z}_u$),
- we present a KEM/DEM version of LIGA,
- we identify formal problems in the rank metric on which the security of LIGA relies and prove the IND-CPA/CCA2 security of the KEM/DEM version under the assumption that some of these problems are hard,
- we analyze new exponential-time attacks on these problems.

2 Preliminaries

2.1 Notations

Let q be a power of a prime and let \mathbb{F}_q denote the finite field of order q . Then, \mathbb{F}_{q^m} and $\mathbb{F}_{q^{mu}}$ denote extension fields of \mathbb{F}_q of order q^m and q^{mu} , respectively. We use $\mathbb{F}_q^{m \times n}$ to denote the set of all $m \times n$ matrices over \mathbb{F}_q and $\mathbb{F}_{q^m}^n = \mathbb{F}_q^{1 \times n}$ for the set of all row vectors of length n over \mathbb{F}_{q^m} . Further, we use another field extension $\mathbb{F}_{q^{mu}}$ with $u > 1$. Thus, $\mathbb{F}_q \subseteq \mathbb{F}_{q^m} \subseteq \mathbb{F}_{q^{mu}}$.

For a field \mathbb{F} , the vector space that is spanned by $\mathbf{v}_1, \dots, \mathbf{v}_l \in \mathbb{F}^n$ is denoted by

$$\langle \mathbf{v}_1, \dots, \mathbf{v}_l \rangle_{\mathbb{F}} := \left\{ \sum_{i=1}^l a_i \mathbf{v}_i : a_i \in \mathbb{F} \right\}.$$

Denote the set of integers $[a, b] = \{i : a \leq i \leq b\}$. Rows and columns of $m \times n$ -matrices are indexed by $1, \dots, m$ and $1, \dots, n$, where $A_{i,j}$ is the element in the i -th row and j -th column of the matrix \mathbf{A} . Further,

$$\mathbf{A}_{[a,b]} := \begin{pmatrix} A_{1,a} & \dots & A_{1,b} \\ \vdots & \ddots & \vdots \\ A_{m,a} & \dots & A_{m,b} \end{pmatrix}.$$

By $\text{rk}_q(\mathbf{A})$ and $\text{rk}_{q^m}(\mathbf{A})$, we denote the rank of a matrix \mathbf{A} over \mathbb{F}_q , respectively \mathbb{F}_{q^m} . Let $(\gamma_1, \gamma_2, \dots, \gamma_u)$ be an ordered basis of $\mathbb{F}_{q^{mu}}$ over \mathbb{F}_{q^m} . By utilizing the vector space isomorphism $\mathbb{F}_{q^{mu}} \cong \mathbb{F}_{q^m}^u$, we can relate each vector $\mathbf{a} \in \mathbb{F}_{q^{mu}}^n$ to a matrix $\mathbf{A} \in \mathbb{F}_{q^m}^{u \times n}$ according to

$$\text{ext}_{\boldsymbol{\gamma}} : \mathbb{F}_{q^m}^n \rightarrow \mathbb{F}_{q^m}^{u \times n}$$

$$\mathbf{a} = (a_1, \dots, a_n) \mapsto \mathbf{A} = \begin{pmatrix} A_{1,1} & \dots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \dots & A_{m,n} \end{pmatrix},$$

where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_u)$ and

$$a_j = \sum_{i=1}^m A_{i,j} \gamma_i, \quad \forall j \in [1, n].$$

The trace operator of a vector $\mathbf{a} \in \mathbb{F}_{q^{mu}}$ to \mathbb{F}_{q^m} is defined by

$$\begin{aligned} \text{Tr} : \mathbb{F}_{q^{mu}} &\rightarrow \mathbb{F}_{q^m} \\ \mathbf{a} = (a_1, a_2, \dots, a_n) &\mapsto \left(\sum_{i=0}^{u-1} a_1^{q^{mi}}, \sum_{i=0}^{u-1} a_2^{q^{mi}}, \dots, \sum_{i=0}^{u-1} a_n^{q^{mi}} \right). \end{aligned}$$

A dual basis $(\gamma_1^*, \gamma_2^*, \dots, \gamma_u^*)$ to $(\gamma_1, \gamma_2, \dots, \gamma_u)$ is a basis that fulfills

$$\text{Tr}(\gamma_i \gamma_j^*) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases},$$

where $i, j \in [1, u]$. Note that a dual basis always exists.

Denote by $\mathcal{M}_{s,q}(\mathbf{a}) \in \mathbb{F}_{q^m}^{s \times n}$ the $s \times n$ Moore matrix for a vector $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{F}_{q^m}^n$, i.e.,

$$\mathcal{M}_{s,q}(\mathbf{a}) := \begin{pmatrix} a_1 & a_2 & \dots & a_n \\ a_1^q & a_2^q & \dots & a_n^q \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{q^{s-1}} & a_2^{q^{s-1}} & \dots & a_n^{q^{s-1}} \end{pmatrix}.$$

If $a_1, a_2, \dots, a_n \in \mathbb{F}_{q^m}$ are linearly independent over \mathbb{F}_q , then $\text{rk}_{q^m}(\mathcal{M}_{s,q}(\mathbf{a})) = \min\{s, n\}$, cf. [26, Lemma 3.15]. This definition can also be extended to matrices by

$$\mathcal{M}_{s,q}(\mathbf{A}) := \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{l,1} & A_{l,2} & \dots & A_{l,n} \\ A_{1,1}^q & A_{1,2}^q & \dots & A_{1,n}^q \\ A_{2,1}^q & A_{2,2}^q & \dots & A_{2,n}^q \\ \vdots & \vdots & \ddots & \vdots \\ A_{l,1}^{q^{s-1}} & A_{l,2}^{q^{s-1}} & \dots & A_{l,n}^{q^{s-1}} \end{pmatrix},$$

where $\mathbf{A} \in \mathbb{F}_{q^m}^{l \times n}$.

The Gaussian binomial coefficient is denoted by

$$\begin{bmatrix} s \\ r \end{bmatrix}_q := \begin{cases} \frac{(1 - q^s)(1 - q^{s-1}) \dots (1 - q^{s-r+1})}{(1 - q)(1 - q^2) \dots (1 - q^r)} & \text{for } r \leq s \\ 0 & \text{for } r > s, \end{cases}$$

where s and r are non-negative integers.

Let \mathcal{X} be a set. When x is drawn uniformly at random from the set \mathcal{X} , we denote it by $x \stackrel{\$}{\leftarrow} \mathcal{X}$. Further, by $x \leftarrow y$ we mean that we assign y to x .

2.2 Rank-metric codes and Gabidulin codes

The rank norm $\text{rk}_q(\mathbf{a})$ is the rank of the matrix representation $\mathbf{A} \in \mathbb{F}_q^{m \times n}$ over \mathbb{F}_q . The rank distance between \mathbf{a} and \mathbf{b} is the rank of the difference of the two matrix representations, i.e.,

$$d_R(\mathbf{a}, \mathbf{b}) := \text{rk}_q(\mathbf{a} - \mathbf{b}) = \text{rk}_q(\mathbf{A} - \mathbf{B}).$$

An $[n, k, d]_q^R$ code \mathcal{C} over \mathbb{F}_{q^m} is a linear rank-metric code, i.e., it is a linear subspace of $\mathbb{F}_{q^m}^n$ of dimension k and minimum rank distance

$$d := \min_{\substack{\mathbf{a}, \mathbf{b} \in \mathcal{C} \\ \mathbf{a} \neq \mathbf{b}}} \{d_R(\mathbf{a}, \mathbf{b}) = \text{rk}_q(\mathbf{a} - \mathbf{b})\}.$$

For linear codes with $n \leq m$, the Singleton-like upper bound [14,18] implies that $d \leq n - k + 1$. If $d = n - k + 1$, the code is called a *maximum rank distance* (MRD) code.

Gabidulin codes [18] are a special class of rank-metric codes and can be defined by their generator matrices.

Definition 1 (Gabidulin Code [18]) A linear $\mathcal{G}(n, k)$ code over \mathbb{F}_{q^m} of length $n \leq m$ and dimension k is defined by its $k \times n$ generator matrix

$$\mathbf{G}_{\mathcal{G}} = \mathcal{M}_{k,q}(g_1, g_2, \dots, g_n),$$

where $\mathbf{g} = (g_1, g_2, \dots, g_n) \in \mathbb{F}_{q^m}^n$ and $\text{rk}_q(\mathbf{g}) = n$.

In [18], it is shown that Gabidulin codes are MRD codes, i.e., $d = n - k + 1$.

For a short description on decoding of Gabidulin codes, denote by $\mathbf{C}_{\mathcal{G}} \in \mathbb{F}_q^{m \times n}$ the transmitted codeword (i.e., the matrix representation of $\mathbf{c}_{\mathcal{G}} \in \mathbb{F}_{q^m}^n$) of a $\mathcal{G}(n, k)$ code that is corrupted by an additive error $\mathbf{E} \in \mathbb{F}_q^{m \times n}$. At the receiver side, only the received matrix $\mathbf{R} \in \mathbb{F}_q^{m \times n}$, where $\mathbf{R} = \mathbf{C}_{\mathcal{G}} + \mathbf{E}$, is known. The channel might provide additional side information in the form of erasures:

- ϱ row erasures (in [45] called “deviations”) and
- γ column erasures (in [45] called “erasures”),

such that the received matrix can be decomposed into

$$\mathbf{R} = \mathbf{C}_{\mathcal{G}} + \underbrace{\mathbf{A}^{(R)}\mathbf{B}^{(R)} + \mathbf{A}^{(C)}\mathbf{B}^{(C)} + \mathbf{E}^{(E)}}_{=\mathbf{E}_{\text{total}}}, \tag{1}$$

where $\mathbf{A}^{(R)} \in \mathbb{F}_q^{m \times \varrho}$, $\mathbf{B}^{(R)} \in \mathbb{F}_q^{\varrho \times n}$, $\mathbf{A}^{(C)} \in \mathbb{F}_q^{m \times \gamma}$, $\mathbf{B}^{(C)} \in \mathbb{F}_q^{\gamma \times n}$ are full-rank matrices, respectively, and $\mathbf{E}^{(E)} \in \mathbb{F}_q^{m \times n}$ is a matrix of rank t . The decoder knows \mathbf{R} and additionally $\mathbf{A}^{(R)}$ and $\mathbf{B}^{(C)}$. Further, t denotes the number of errors without side information. The rank-metric error-erasure decoding algorithms from [20,45,51] can then reconstruct $\mathbf{c}_{\mathcal{G}} \in \mathcal{G}(n, k)$ with asymptotic complexity $\mathcal{O}(n^2)$ operations over \mathbb{F}_{q^m} , or in sub-quadratic complexity using the fast operations described in [36,37], if

$$2t + \varrho + \gamma \leq d - 1 = n - k \tag{2}$$

is fulfilled.

2.3 Interleaved rank-metric codes

Interleaved Gabidulin Codes are a code class for which efficient decoders are known that are able to correct w.h.p. random errors of rank larger than $\lfloor \frac{d-1}{2} \rfloor$.

Definition 2 (Interleaved Gabidulin Codes [27]) A linear (vertically, homogeneous) interleaved Gabidulin code $\mathcal{IG}(u; n, k)$ over \mathbb{F}_{q^m} of length $n \leq m$, dimension $k \leq n$, and interleaving order u is defined by

$$\mathcal{IG}(u; n, k) := \left\{ \begin{pmatrix} \mathbf{c}_{\mathcal{G}}^{(1)} \\ \mathbf{c}_{\mathcal{G}}^{(2)} \\ \vdots \\ \mathbf{c}_{\mathcal{G}}^{(u)} \end{pmatrix} : \mathbf{c}_{\mathcal{G}}^{(i)} \in \mathcal{G}(n, k), \forall i \in [1, u] \right\}.$$

As a short-term notation, we also speak about a u -interleaved Gabidulin code. When considering random errors of rank weight t , the code $\mathcal{IG}(u; n, k)$ can be decoded uniquely with high probability up to $w \leq \lfloor \frac{u}{u+1}(n - k) \rfloor$ errors,¹ cf. [27,43,51]. However, it is well-known that there are many error patterns for which the known efficient decoders fail. In fact, we can explicitly construct a large class of such errors as shown in the following lemma.

Lemma 1 (Interleaved Decoding [27], [43], [49, p. 64]) Let $\mathbf{c}_i = \mathbf{x}_i \cdot \mathbf{G}_{\mathcal{G}}$. All known² efficient decoders for $\mathcal{IG}(u; n, k)$ codes fail to correct an error $\mathbf{z} \in \mathbb{F}_{q^{mu}}^n$ with $\mathbf{z} = \sum_{i=1}^u \mathbf{z}_i \gamma_i^*$ and $\text{rk}_q(\mathbf{z}) = w$ if

$$\text{rk}_{q^m} \begin{pmatrix} \mathcal{M}_{n-w-1,q}(\mathbf{g}) \\ \mathcal{M}_{n-k-w,q}(\mathbf{c}_1 + \mathbf{z}_1) \\ \mathcal{M}_{n-k-w,q}(\mathbf{c}_2 + \mathbf{z}_2) \\ \vdots \\ \mathcal{M}_{n-k-w,q}(\mathbf{c}_u + \mathbf{z}_u) \end{pmatrix} < n - 1.$$

It is widely conjectured that there cannot be a decoder that decodes the error patterns of Lemma 1 *uniquely*. Decoding these failing error patterns has been subject to intensive research since the Loidreau–Overbeck decoder [27] was found in 2006. In the Hamming metric, the equivalent problem for Reed–Solomon codes has been studied since 1997 [25] and more than a dozen papers have dealt with decoding algorithms for these codes. None of these papers was able to give a polynomial-time decoding algorithm for the cases of Lemma 1. It seems that all unique decoders *have to fail* for the error patterns of Lemma 1 since for these cases, there is no unique decision, i.e., more than one interleaved codeword lies in the ball of radius w around the received word.

3 Key generation in the original Faure–Loidreau system

In this section, we recall the key generation algorithm of the original FL cryptosystem, we give a coding-theoretic interpretation of the original public key, and analyze the structural attack from [21].

¹ In this setting, an “error of weight w ” is a matrix in $\mathbb{F}_{q^{mu}}^n$ with \mathbb{F}_q -rank equal to w .

² i.e., the algorithms in [27,43], and [49, p. 64].

Table 1 Summary of the parameters

Name	Use	Restriction
q	Small field size	Prime power
m	Extension degree	$1 \leq m$
n	Code length	$n \leq m$
k	Code dimension	$k < n$
u	Extension degree	$2 \leq u < k$
w	Error weight in public key	$\max \left\{ n - k - \frac{k-u}{u-1}, \left\lfloor \frac{n-k}{2} \right\rfloor + 1 \right\} \leq w < \frac{u}{u+2}(n-k)$
t_{pub}	Error weight in ciphertext	$t_{\text{pub}} = \left\lfloor \frac{n-k-w}{2} \right\rfloor$
ζ	\mathbb{F}_{q^m} -dimension of error vector in the public key	$\zeta < \frac{w}{n-k-w}$ and $\zeta q^{\zeta w-m} \leq \frac{1}{2}$

3.1 The original algorithm

Let $q, m, n, k, u, w, t_{\text{pub}}$ be positive integers that fulfill the restrictions given in Table 1 and are publicly known. In the following, we consider the three finite fields $\mathbb{F}_q, \mathbb{F}_{q^m}$, and $\mathbb{F}_{q^{mu}}$, which are extension fields of each other, i.e.:

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

The original FL key generation is shown in Algorithm 1.

Algorithm 1: Original FL Key Generation

- Input:** Parameters q, m, n, k, u, w as in Table 1
Output: Secret key sk , public key pk
- 1 $\mathbf{g} \xleftarrow{\$} \{ \mathbf{a} \in \mathbb{F}_{q^m}^n : \text{rk}_q(\mathbf{a}) = n \}$
 - 2 $\mathbf{x} \xleftarrow{\$} \{ \mathbf{a} \in \mathbb{F}_{q^{mu}}^k : \dim(\langle a_{k-u+1}, \dots, a_k \rangle_{\mathbb{F}_{q^m}}) = u \}$
 - 3 $\mathbf{s} \xleftarrow{\$} \{ \mathbf{a} \in \mathbb{F}_{q^{mu}}^w : \text{rk}_q(\mathbf{a}) = w \}$
 - 4 $\mathbf{P} \xleftarrow{\$} \{ \mathbf{A} \in \mathbb{F}_q^{n \times n} : \text{rk}_q(\mathbf{A}) = n \}$
 - 5 $\mathbf{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\mathbf{g})$
 - 6 $\mathbf{z} \leftarrow (\mathbf{s} \mid \mathbf{0}) \cdot \mathbf{P}^{-1}$
 - 7 $\mathbf{k}_{\text{pub}} \leftarrow \mathbf{x} \cdot \mathbf{G}_{\mathcal{G}} + \mathbf{z}$
 - 8 $t_{\text{pub}} \leftarrow \left\lfloor \frac{n-w-k}{2} \right\rfloor$
 - 9 **return** $sk = (\mathbf{x}, \mathbf{P}_{[w+1,n]}), pk = (\mathbf{g}, \mathbf{k}_{\text{pub}})$
-

3.2 Coding-theoretic interpretation of the original public key

The public key \mathbf{k}_{pub} of the FL system is a corrupted codeword of a u -interleaved Gabidulin code. To our knowledge, this connection between the public key and interleaved Gabidulin codes has not been known before. This interpretation is central to this paper and will be used

in Sect. 4.1 to define the public key of LIGA such that is not vulnerable against the attacks from [21] and described in Sect. 3.3.

Theorem 1 Fix a basis of $\mathbb{F}_{q^{mu}}$ over \mathbb{F}_{q^m} . Let $*$ be a dual basis to and write $\mathbf{k}_{pub} = \sum_{i=1}^u \mathbf{k}_{pub}^{(i)} \gamma_i^*$. Then,

$$\begin{pmatrix} \mathbf{k}_{pub}^{(1)} \\ \mathbf{k}_{pub}^{(2)} \\ \vdots \\ \mathbf{k}_{pub}^{(u)} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_{\mathcal{G}}^{(1)} \\ \mathbf{c}_{\mathcal{G}}^{(2)} \\ \vdots \\ \mathbf{c}_{\mathcal{G}}^{(u)} \end{pmatrix} + \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_u \end{pmatrix}, \tag{3}$$

where the $\mathbf{c}_{\mathcal{G}}^{(i)} \in \mathbb{F}_{q^m}^n$ are codewords of the Gabidulin code $\mathcal{G}(n, k)$ with generator matrix $\mathbf{G}_{\mathcal{G}}$ and the $\mathbf{z}_i \in \mathbb{F}_{q^m}^n$ are obtained from the vector $\mathbf{z} \in \mathbb{F}_{q^{mu}}^n$ by $\mathbf{z} = \sum_{i=1}^u \mathbf{z}_i \gamma_i^*$.

Proof Recall the definition of the public key

$$\mathbf{k}_{pub} = \mathbf{x} \cdot \mathbf{G}_{\mathcal{G}} + \mathbf{z},$$

where $\mathbf{x} \in \mathbb{F}_{q^{mu}}^k$, $\mathbf{G}_{\mathcal{G}} \in \mathbb{F}_{q^m}^{k \times n}$ is the generator matrix of a $\mathcal{G}(n, k)$ code, and $\mathbf{z} \in \mathbb{F}_{q^{mu}}^n$ with $\text{rk}_q(\mathbf{z}) = w$. Let $\mathbf{x} = \sum_{i=1}^u \mathbf{x}_i \gamma_i^*$, where the \mathbf{x}_i have coefficients in \mathbb{F}_{q^m} .

Then, we obtain the following representation of the public key \mathbf{k}_{pub} as a $u \times n$ matrix in \mathbb{F}_{q^m}

$$\begin{pmatrix} \mathbf{k}_{pub}^{(1)} \\ \mathbf{k}_{pub}^{(2)} \\ \vdots \\ \mathbf{k}_{pub}^{(u)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_u \end{pmatrix} \cdot \mathbf{G}_{\mathcal{G}} + \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_u \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \cdot \mathbf{G}_{\mathcal{G}} \\ \mathbf{x}_2 \cdot \mathbf{G}_{\mathcal{G}} \\ \vdots \\ \mathbf{x}_u \cdot \mathbf{G}_{\mathcal{G}} \end{pmatrix} + \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_u \end{pmatrix}.$$

Since $\mathbf{x}_i \cdot \mathbf{G}_{\mathcal{G}}$ is a codeword of a $\mathcal{G}(n, k)$ code, $\forall i \in [1, u]$, the matrix representation of \mathbf{k}_{pub} can be seen as a codeword from an $\mathcal{IG}(u; n, k)$ code, corrupted by an error. \square

Note that the error $(\mathbf{z}_1^T, \dots, \mathbf{z}_u^T)^T$ in (3) has \mathbb{F}_q -rank at most w due to the structure of $\mathbf{z} = (\mathbf{s}|\mathbf{0})\mathbf{P}^{-1}$.

3.3 Efficient key recovery of the original FL key

The attack by *Gaborit, Otmani and Talé Kalachi* (GOT) on the original FL system in [21] (see Algorithm 2 below) is an efficient structural attack which computes a valid private key of the FL system in cubic time when the public key fulfills certain conditions. We recall this attack in the following and derive an alternative, equally powerful, attack based on interleaved decoding the public key, utilizing the observation of the previous subsection. We prove that the failure conditions of both attacks are equivalent. The interleaved decoding attack does not have any advantage in terms of cryptanalysis compared to [21], but enables us to exactly predict for which public keys both attacks work and for which the attacks fail.

3.3.1 GOT attack

The key recovery in the GOT attack (Algorithm 2) succeeds under the conditions of the following theorem.

Algorithm 2: GOT Attack

- Input:** Public key $(\mathbf{g}, \mathbf{k}_{\text{pub}})$
- Output:** Private key (\mathbf{x}, \mathbf{P})
- 1 Choose $\gamma_1, \dots, \gamma_u$ to be a basis of $\mathbb{F}_{q^{mu}}$ over \mathbb{F}_{q^m}
- 2 $\mathbf{k}_{\text{pub}}^{(i)} \leftarrow \text{Tr}(\gamma_i \mathbf{k}_{\text{pub}})$ for all $i = 1, \dots, u$
- 3 $\mathbf{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\mathbf{g})$
- 4 Pick at random a non-zero vector $\tilde{\mathbf{h}} \in \mathbb{F}_{q^m}^n$ such that

$$\mathcal{M}_{n-w-k,q} \begin{pmatrix} \mathbf{G}_{\mathcal{G}} \\ \mathbf{k}_{\text{pub}}^{(1)} \\ \vdots \\ \mathbf{k}_{\text{pub}}^{(u)} \end{pmatrix} \cdot \tilde{\mathbf{h}}^T = \mathbf{0}.$$

- 5 Choose $\mathbf{P} \in \mathbb{F}_q^{n \times n}$ and $\mathbf{h}' \in \mathbb{F}_{q^m}^{n-w}$ such that $\tilde{\mathbf{h}}(\mathbf{P}^{-1})^T = (\mathbf{0} \mid \mathbf{h}')$
- 6 Choose \mathbf{x} such that $\mathbf{x}\mathbf{G}_{\mathcal{G}}\mathbf{P}' = \mathbf{k}_{\text{pub}}\mathbf{P}'$, where $\mathbf{P}' = \mathbf{P}_{[w+1,n]} \in \mathbb{F}_q^{n \times (n-w)}$
- 7 **return** (\mathbf{x}, \mathbf{P})

Theorem 2 (GOT Attack [21, Thm. 1]) *Let $\gamma_1, \dots, \gamma_u \in \mathbb{F}_{q^{mu}}$ be a basis of $\mathbb{F}_{q^{mu}}$ over \mathbb{F}_{q^m} and let $\mathbf{z}_i = \text{Tr}(\gamma_i \mathbf{z})$, for $i = 1, \dots, u$.*

If the matrix $\mathbf{Z} \in \mathbb{F}_{q^m}^{u \times n}$ with $\mathbf{z}_1, \dots, \mathbf{z}_u$ as rows, satisfies

$$\text{rk}_{q^m}(\mathcal{M}_{n-k-w,q}(\mathbf{Z})) = w,$$

then (\mathbf{x}, \mathbf{z}) can be recovered from $(\mathbf{G}_{\mathcal{G}}, \mathbf{k}_{\text{pub}})$ with $\mathcal{O}(n^3)$ operations in \mathbb{F}_{q^m} by using Algorithm 2.

If the key is generated by Algorithm 1, the GOT attack breaks the original FL system with high probability.

3.3.2 Interleaved decoding attack

Recall from Theorem 1 that the public key \mathbf{k}_{pub} is a corrupted interleaved codeword. Based on this observation we will derive a structural attack on the original FL system to which we refer as *Interleaved Decoding Attack* in the following. We prove that interleaved decoding and the GOT attack *fail* (i.e, do not provide any information) for the *same public keys*. The idea is to decode \mathbf{k}_{pub} in an interleaved Gabidulin code. Since $w \leq \frac{u}{u+1}(n - k)$, such a decoder will return \mathbf{x} with high probability, but fail in certain cases, see Sect. 2.3.

Since $\text{rk}_{q^m}(\mathcal{M}_{n-w-1,q}(\mathbf{g})) = n - w - 1$, the interleaved decoder fails if (compare Lemma 1):

$$\text{rk}_{q^m}(\tilde{\mathbf{Z}}) := \varphi < w, \tag{4}$$

where

$$\tilde{\mathbf{Z}} = \begin{pmatrix} \mathcal{M}_{n-k-w,q}(\mathbf{z}_1) \\ \mathcal{M}_{n-k-w,q}(\mathbf{z}_2) \\ \vdots \\ \mathcal{M}_{n-k-w,q}(\mathbf{z}_u) \end{pmatrix}. \tag{5}$$

3.3.3 Equivalence of GOT attack and interleaved decoding attack

In the following, we prove that the failure condition of the GOT Attack is equivalent to the condition that decoding \mathbf{k}_{pub} in an interleaved Gabidulin code fails.

Theorem 3 *The GOT Attack from [21] fails if and only if the Interleaved Decoding Attack fails. In particular, both fail if (4) holds.*

Proof Rewrite the matrix from Theorem 2 as

$$\mathcal{M}_{n-w-k,q}(\mathbf{Z}) = \begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_u \\ \mathbf{z}_1^q \\ \vdots \\ \mathbf{z}_u^q \\ \vdots \\ \mathbf{z}_1^{q^{n-w-k-1}} \\ \vdots \\ \mathbf{z}_u^{q^{n-w-k-1}} \end{pmatrix} \tag{6}$$

and the matrix from equation (5) as

$$\tilde{\mathbf{Z}} = \begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_1^{q^{n-w-k-1}} \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_2^{q^{n-w-k-1}} \\ \vdots \\ \mathbf{z}_u \\ \vdots \\ \mathbf{z}_u^{q^{n-w-k-1}} \end{pmatrix}. \tag{7}$$

Since the matrix in (6) and in (7) only differ in row permutations, they are row-space equivalent, implying that they have the same rank. Further, the rank of the matrix in (7) cannot become larger than w (since any vector in the right kernel of this matrix has rank weight at least $n - w$ [34, Algorithm 3.2.1]). Thus, the failures of Theorem 2 and Lemma 1 are equivalent. \square

In the next section, we will exploit the observation of Theorem 3, i.e., we propose a new key generation algorithm that avoids public keys that can be efficiently decoded by an interleaved decoder, thereby rendering the GOT attack useless.

4 The new system LIGA

In this section, we propose a public-key code-based encryption scheme $\Pi^{\text{PKE}} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ called LIGA. The system is based on the original FL system [16], where we keep both the original encryption and decryption algorithm, but replace the insecure key-generation algorithm. Further, we present a KEM-DEM version of LIGA denoted by $\Pi^{\text{KEM}} = (\text{KeyGen}, \text{Encaps}, \text{Decaps})$.

Later, in Sect. 5, we will analyze the security of the system. We single out problems from coding theory and we prove that the encryption version is IND-CPA secure and the KEM-DEM version is IND-CCA2 secure under the assumption that the stated problems are hard. Furthermore, we study new and known attacks on these problems and show that they all run in exponential time (see Sect. 6).

4.1 The new key generation algorithm

We introduce a new key generation algorithm that is based on choosing $\mathbf{z} = \sum_{i=1}^u \mathbf{z}_i \gamma_i^*$ in a way that $\varphi < w$, where φ is the rank of the interleaved Moore matrix of the errors \mathbf{z}_i in the public key, see (5). Based on the dimension of the span of the \mathbf{z}_i , we will upper bound φ in the following Theorem 4. Recall that when $\varphi < w$, the GOT attack [21] and interleaved decoding of the public key fail, see Theorem 3. In this case, retrieving any knowledge about the private key from the public key requires to solve Problem 1 (defined later), which basically corresponds to decoding the interleaved codeword when error patterns occur for which all known decoders fail.

Theorem 4 *Let $\dim(\langle \mathbf{z}_1, \dots, \mathbf{z}_u \rangle_{q^m}) = \zeta$. Then*

$$\varphi = \text{rk}_{q^m}(\tilde{\mathbf{Z}}) \leq \min\{\zeta(n - k - w), w\}. \tag{8}$$

Proof The dimension of $\langle \mathbf{z}_1, \dots, \mathbf{z}_u \rangle_{q^m}$ implies that at most $\zeta(n - k - w)$ rows of $\tilde{\mathbf{Z}}$ are linearly independent over \mathbb{F}_{q^m} , meaning that $\varphi \leq \zeta(n - k - w)$.

The definition of $\mathbf{z} = (\mathbf{s} \mid \mathbf{0}) \cdot \mathbf{P}^{-1}$ leads to

$$\begin{aligned} \varphi &= \text{rk}_{q^m}(\tilde{\mathbf{Z}}) \\ &= \text{rk}_{q^m} \left(\left(\begin{array}{c|c} \mathcal{M}_{n-k-w,q}(\mathbf{s}_1) & \mathbf{0} \\ \vdots & \vdots \\ \mathcal{M}_{n-k-w,q}(\mathbf{s}_u) & \mathbf{0} \end{array} \right) \mathbf{P}^{-1} \right) \\ &= \text{rk}_{q^m} \left(\begin{array}{c} \mathcal{M}_{n-k-w,q}(\mathbf{s}_1) \\ \vdots \\ \mathcal{M}_{n-k-w,q}(\mathbf{s}_u) \end{array} \right) \\ &\leq w, \end{aligned}$$

where the last inequality holds since $\mathbf{s}_1, \dots, \mathbf{s}_u$ are vectors of length w . □

We propose the following modification to Line 3 of the **Key Generation**, depending on the parameter ζ :

$$\begin{aligned}
 3 \quad \mathcal{A} &\stackrel{\$}{\leftarrow} \left\{ \text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^w : \dim \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements} \right\} \\
 3' \quad \begin{pmatrix} \mathbf{s}'_1 \\ \vdots \\ \mathbf{s}'_u \end{pmatrix} &\stackrel{\$}{\leftarrow} \left\{ \begin{pmatrix} \mathbf{s}'_1 \\ \vdots \\ \mathbf{s}'_u \end{pmatrix} : \langle \mathbf{s}'_1, \dots, \mathbf{s}'_u \rangle_{\mathbb{F}_{q^m}} = \mathcal{A}, \text{rk}_q(\mathbf{s}'_i) = w, \forall i \right\}
 \end{aligned}$$

Clearly, $\dim(\langle \mathbf{z}_1, \dots, \mathbf{z}_u \rangle_{q^m}) = \zeta$ in this case. To avoid that the GOT attack [21] runs in polynomial time, Theorem 4 implies that the parameter ζ must always be chosen such that $\zeta < \frac{w}{n-k-w}$. In Sect. 6, we will discuss several further exponential-time attacks on LIGA. Some of these attacks have a work factor depending on ζ , which must be considered in the parameter design.

Furthermore, the condition $\text{rk}_q(\mathbf{s}'_i) = w$ ensures that $\text{rk}_q(\mathbf{z}_i) = w$, i.e., as large as possible for a given subspace \mathcal{A} . This choice maximizes the work factor of generic decoding attacks on the rows of the public key (seen as a received word of an interleaved Gabidulin code), see Sect. 6.

The restriction of the choice of \mathcal{A} to subspaces that contain a basis of full- \mathbb{F}_q -rank code-words is to ensure that the set from which we sample in Line 3' is non-empty. Hence, the key generation always works.

Compared to the choice of \mathbf{z} in Line 3 of the original Key Generation algorithm, we restrict the choice of \mathbf{z} , but we will see in Sect. 6 that there are still enough possibilities for \mathbf{z} to prevent an efficient naive brute-force attack.

Appendix 1 contains a more detailed discussion on how to realize Lines 3 and 3' in practice.

4.2 The public key encryption version

The new key generation algorithm KeyGen, the encryption algorithm Encrypt and the decryption algorithm Decrypt are shown in Algorithm 3, Algorithm 4 and Algorithm 5, respectively. Compared to original key generation algorithm, the algorithm KeyGen has one more input parameter ζ (cf. Sect. 4.1).

The proposed system has no decryption failures as proven in the following theorem.

Theorem 5 (Correctness [16]) *Algorithm 5 returns the correct plaintext \mathbf{m} .*

Proof Line 1 computes

$$\mathbf{cP} = (\mathbf{m} + \text{Tr}(\alpha\mathbf{x}))\mathbf{G}_G\mathbf{P} + (\text{Tr}(\alpha\mathbf{s})|\mathbf{0}) + \mathbf{eP},$$

whose last $n - w$ columns are given by

$$\mathbf{c}' = (\mathbf{m} + \text{Tr}(\alpha\mathbf{x}))\mathbf{G}' + \mathbf{e}',$$

where $\mathbf{G}' := \mathbf{G}_G\mathbf{P}_{[w+1,n]} \in \mathbb{F}_{q^m}^{k \times (n-w)}$ and $\mathbf{e}' := \mathbf{eP}_{[w+1,n]}$. By decoding in \mathcal{G}' , we thus obtain the vector

$$\mathbf{m}' = \mathbf{m} + \text{Tr}(\alpha\mathbf{x}).$$

Since the last u positions of the plaintext \mathbf{m} are zero (i.e., $m_i = 0$ for $i = k - u + 1, \dots, k$), we get $\alpha = \sum_{i=k-u+1}^k m'_i x_i^*$, where $\{x_{k-u+1}^*, \dots, x_k^*\}$ is a dual basis to $\{x_{k-u+1}, \dots, x_k\}$. As we know α and \mathbf{x} , we can compute the plaintext \mathbf{m} . □

Algorithm 3: KeyGen (\cdot)

Input: Parameters q, m, n, k, u, w, ζ as in Table 1
Output: Private key sk , public key pk

- 1 $\mathbf{g} \xleftarrow{\$} \{\mathbf{a} \in \mathbb{F}_{q^m}^n : \text{rk}_q(\mathbf{a}) = n\}$
- 2 $\mathbf{x} \xleftarrow{\$} \{\mathbf{a} \in \mathbb{F}_{q^{mu}}^k : \dim(\langle a_{k-u+1}, \dots, a_k \rangle_{\mathbb{F}_{q^m}}) = u\}$
- 3 $\mathcal{A} \xleftarrow{\$} \{\text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^w : \dim \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements}\}$
- 3' $\begin{pmatrix} s_1 \\ \vdots \\ s_u \end{pmatrix} \xleftarrow{\$} \left\{ \begin{pmatrix} s'_1 \\ \vdots \\ s'_u \end{pmatrix} : \langle s'_1, \dots, s'_u \rangle_{\mathbb{F}_{q^m}} = \mathcal{A}, \text{rk}_q(s'_i) = w \forall i \right\}$
- 4 $\mathbf{s} \leftarrow \sum_{i=1}^u s_i \gamma_i^*$
- 5 $\mathbf{P} \xleftarrow{\$} \{\mathbf{A} \in \mathbb{F}_q^{n \times n} : \text{rk}_q(\mathbf{A}) = n\}$
- 6 $\mathbf{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\mathbf{g})$
- 7 $\mathbf{z} \leftarrow (\mathbf{s} \mid \mathbf{0}) \cdot \mathbf{P}^{-1}$
- 8 $\mathbf{k}_{\text{pub}} \leftarrow \mathbf{x} \cdot \mathbf{G}_{\mathcal{G}} + \mathbf{z}$
- 9 $t_{\text{pub}} \leftarrow \lfloor \frac{n-w-k}{2} \rfloor$
- 10 **return** $sk = (\mathbf{x}, \mathbf{P}_{[w+1,n]}), pk = (\mathbf{g}, \mathbf{k}_{\text{pub}})$

Algorithm 4: Encrypt (\cdot)

Input: Plaintext $\mathbf{m} \in \mathbb{F}_{q^m}^{k-u}$, public key $pk = (\mathbf{g}, \mathbf{k}_{\text{pub}})$, randomness θ
Output: Ciphertext \mathbf{c}

- 1 $\alpha \xleftarrow{\$} \mathbb{F}_{q^{mu}}$ using θ
- 2 $\mathbf{e} \xleftarrow{\$} \{\mathbf{a} \in \mathbb{F}_{q^m}^n : \text{rk}_q(\mathbf{a}) = t_{\text{pub}}\}$ using θ
- 3 $\mathbf{G}_{\mathcal{G}} \leftarrow \mathcal{M}_{k,q}(\mathbf{g})$
- 4 **return** $\mathbf{c} \leftarrow (\mathbf{m}, \mathbf{0}_u) \cdot \mathbf{G}_{\mathcal{G}} + \text{Tr}(\alpha \mathbf{k}_{\text{pub}}) + \mathbf{e}$.

Algorithm 5: Decrypt (\cdot)

Input: Ciphertext $\mathbf{c} \in \mathbb{F}_{q^m}^n$, private key $sk = (\mathbf{x}, \mathbf{P}_{[w+1,n]})$
Output: Plaintext $\mathbf{m} \in \mathbb{F}_{q^m}^{k-u}$

- 1 $\mathbf{c}' \leftarrow \mathbf{c} \mathbf{P}_{[w+1,n]}$
- 2 $\mathcal{G}' \leftarrow$ Gabidulin code generated by $\mathbf{G}_{\mathcal{G}} \mathbf{P}_{[w+1,n]}$
- 3 $\mathbf{m}' \leftarrow$ decode \mathbf{c}' in \mathcal{G}'
- 4 $\alpha \leftarrow \sum_{i=k-u+1}^k m'_i x_i^*$
- 5 **return** $\mathbf{m} \leftarrow (\mathbf{m}' - \text{Tr}(\alpha \mathbf{x}))_{[1,k-u]}$

Remark 1 Steps 1 to 3 of Algorithm 5 can be interpreted as an error-erasure decoder of a Gabidulin code. As this observation may have advantages, especially for implementations, we present this connection formally in Appendix 2.

A SageMath v8.8 [47] implementation of the public key encryption version of LIGA can be downloaded from https://bitbucket.org/julianrenner/liga_pke. The purpose of the source code is to clarify the shown algorithms but *not* to provide a secure and efficient instance. Developing an implementation that offers the latter two properties and can serve for a performance

comparison with other schemes is outside of the scope of this paper and is left for future research.

4.3 KEM/DEM version Π^{PKE} and Π^{KEM}

In [23], generic transformations of IND-CPA secure public key encryptions into IND-CCA2 secure KEMs are proposed. In the following, we apply one of the transformations directly to Π^{PKE} to obtain Π^{KEM} . Later, in Sect. 5.2, we will prove that Π^{PKE} fulfills the requirements such that the applied transformation is secure.

Let \mathcal{G} , \mathcal{H} and \mathcal{K} be hash functions, where $\mathcal{G} \neq \mathcal{H}$. In Algorithm 6 and Algorithm 7, we show the encapsulation and decapsulation algorithms of the KEM $\Pi^{\text{KEM}} = (\text{KeyGen}, \text{Encaps}, \text{Decaps})$. The algorithm KeyGen remains Algorithm 3.

Algorithm 6: Encaps (\cdot)

Input: Public key $\text{pk} = (\mathbf{g}, \mathbf{k}_{\text{pub}})$
Output: Ciphertext (\mathbf{c}, \mathbf{d}) , shared key K

- 1 $\mathbf{m} \xleftarrow{\$} \mathbb{F}_q^{k-u}$ (this will serve as shared key)
- 2 $\theta \leftarrow \mathcal{G}(\mathbf{m})$
- 3 $\mathbf{c} \leftarrow \text{Encrypt}(\mathbf{m}, \text{pk}, \theta)$
- 4 $K \leftarrow \mathcal{K}(\mathbf{m}, \mathbf{c})$
- 5 $\mathbf{d} \leftarrow \mathcal{H}(\mathbf{c}, \mathbf{d})$
- 6 **return** $(\mathbf{c}, \mathbf{d}), K$

Algorithm 7: Decaps (\cdot)

Input: Ciphertext (\mathbf{c}, \mathbf{d}) , private key $\text{sk} = (\mathbf{x}, \mathbf{P}_{[w+1, n]})$
Output: Shared key K

- 1 $\mathbf{m}' \leftarrow \text{Decrypt}(\mathbf{c}, \text{sk})$
- 2 $\theta' \leftarrow \mathcal{G}(\mathbf{m}')$
- 3 $\mathbf{c}' \leftarrow \text{Encrypt}(\mathbf{m}', \text{pk}, \theta')$
- 4 **if** $\mathbf{c} \neq \mathbf{c}'$ **or** $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ **then**
- 5 $K \leftarrow \perp$
- 6 **else**
- 7 $K \leftarrow \mathcal{K}(\mathbf{m}', \mathbf{c})$
- 8 **return** K

4.4 Complexity

4.4.1 Timing attacks

Resistance against timing attacks is essential in many applications and systems that do not enable a constant-time implementation are thus considered insecure. Due to the fact that Step 4 of Algorithm 4 can be easily implemented in constant time, the proposed encryption algorithm does not reveal any information about secret knowledge through timing attacks. The same holds for the presented decryption algorithm since there exists an efficient constant-time decoding algorithm for Gabidulin codes [10] and all other steps of Algorithm 5 can be realized in constant time as well.

4.4.2 Asymptotically fastest methods

In some scenarios, a constant-time implementation of the system may not be required but we want that the key generation, encryption, and decryption are as fast as possible. The following results were not known when the original FL system was proposed, but have a major impact on its efficiency.

The complexity of key generation and encryption is dominated by the cost of encoding a Gabidulin code (Line 8 of Algorithm 3 and Line 4 of Algorithm 4).³ The asymptotically fastest-known algorithms [13,36,37] for this require

- $O^{\sim}(n^{\min\{\frac{\omega+1}{2}, 1.635\}})$ operations in \mathbb{F}_{q^m} or $O^{\sim}(n^{\omega-2}m^2)$ operations in \mathbb{F}_q in general⁴ and
- $O^{\sim}(n)$ operations in \mathbb{F}_{q^m} if the entries of \mathbf{g} are a normal basis of $\mathbb{F}_{q^m}/\mathbb{F}_q$,

where ω is the matrix multiplication exponent and O^{\sim} means that log factors are neglected.

The bottleneck of decryption is (error-erasure) decoding of a Gabidulin code (Line 3 of Algorithm 5, see also Appendix 2 below), where the asymptotically fastest algorithm costs

$$O^{\sim}\left(n^{\min\{\frac{\omega+1}{2}, 1.635\}}\right)$$

operations in \mathbb{F}_{q^m} [36,37] or

$$O^{\sim}(n^{\omega-2}m^2)$$

operations in \mathbb{F}_q (decoder in [36] with linearized-polynomial operations in [13]).

For small lengths n , the algorithms from [22,40,44,49], which have quadratic complexity over \mathbb{F}_{q^m} (or cubic complexity over \mathbb{F}_q), might be faster than the mentioned algorithms due to smaller hidden constants in the O -notation.

5 Difficult problems & semantic security of LIGA

In this section, we introduce problems in the rank metric that are considered to be difficult. Furthermore, we prove that the public-key encryption version of LIGA is IND-CPA secure and the KEM version is IND-CCA2 secure under the assumption that there does not exist probabilistic polynomial-time algorithms that can solve them. A detailed complexity analysis of existing and new algorithms solving the stated problems is given in Sect. 6.

5.1 Difficult problems in the rank metric

LIGA is based on several difficult problems which are stated in this section. Note that the search variants of the problems correspond exactly to retrieving information about the private key from the public key (not necessarily a valid private key as explained in the following) or the plaintext from the ciphertext. The decisional problems are equivalent to distinguishing the public key or the ciphertext from random vectors.

³ Note that since \mathbf{x} and \mathbf{z} have coefficients in the large field $\mathbb{F}_{q^{mu}}$, this line can be realized as encoding u messages over \mathbb{F}_{q^m} with the generator matrix $\mathbf{G}_G \in \mathbb{F}_{q^m}^{k \times n}$ and corrupting these codewords with an error (see also Sect. 3.2 below).

⁴ Which of the two algorithms is fastest depends on the relation between n and m , as well as the used working basis of \mathbb{F}_{q^m} over \mathbb{F}_q .

Definition 3 (ResIG-Distribution: Restricted Interleaved Gabidulin Code Distribution)

Input: $q, m, n, k, w > \lfloor \frac{n-k}{2} \rfloor, \zeta < \frac{w}{n-k-w}, u < w$.

Choose uniformly at random

- $\mathbf{G} \xleftarrow{\$} \mathcal{G}$, where \mathcal{G} is the set of all generator matrices of $[n, k]$ Gabidulin codes over \mathbb{F}_{q^m}
- $\mathbf{M} \xleftarrow{\$} \{\mathbf{X} \in \mathbb{F}_{q^m}^{u \times k} : \text{rk}_{q^m}(\mathbf{X}_{[k-u+1, k]}) = u\}$.
- $\mathcal{A} \xleftarrow{\$} \{\text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^w : \dim \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements}\}$
- $\mathbf{E}' \xleftarrow{\$} \left\{ \begin{pmatrix} \mathbf{s}'_1 \\ \vdots \\ \mathbf{s}'_u \end{pmatrix} \in \mathbb{F}_{q^m}^{u \times w} : \langle \mathbf{s}'_1, \dots, \mathbf{s}'_u \rangle_{\mathbb{F}_{q^m}} = \mathcal{A}, \text{rk}_q(\mathbf{s}'_i) = w \forall i \right\}$
- $\mathbf{Q} \xleftarrow{\$} \{\mathbf{A} \in \mathbb{F}_q^{w \times n} : \text{rk}_q(\mathbf{A}) = w\}$
- $\mathbf{E} \leftarrow \mathbf{E}'\mathbf{Q}$

Output: $(\mathbf{G}, \mathbf{M}\mathbf{G} + \mathbf{E})$.

Problem 1 (ResIG-Search: Restricted interleaved Gabidulin Code Search Problem)

Input: (\mathbf{G}, \mathbf{Y}) from ResIG-Distribution with input q, m, n, k, w, ζ, u (Definition 3).

Goal: Find $\mathbf{M} \in \mathbb{F}_{q^m}^{u \times k}$ and $\mathbf{E} \in \{\mathbf{X} \in \mathbb{F}_{q^m}^{u \times n} : \text{rk}_{\mathbb{F}_q}(\mathbf{X}) \leq w\}$ s.t. $\mathbf{M}\mathbf{G} + \mathbf{E} = \mathbf{Y}$.

Problem 1 (ResIG-Search) is equivalent to decoding a codeword of a u -interleaved Gabidulin code that is corrupted by an error \mathbf{E} , see also Sect. 6.1.2 and is therefore the underlying problem of the structural attacks from Sect. 3.3.

Note however that not necessarily every solution of this problem can be used directly as a valid private key since some additional structure on \mathbf{E} is introduced in LIGA (i.e., Problem 1 is easier to solve than retrieving a valid private key of LIGA).

Problem 2 (ResIG-Dec: Restricted Interleaved Gabidulin Code Decisional Problem)

Input: $(\mathbf{G}, \mathbf{Y}) \in \mathbb{F}_{q^m}^{k \times n} \times \mathbb{F}_{q^m}^{u \times n}$.

Goal: Decide with non-negligible advantage whether \mathbf{Y} came from ResIG-Distribution with input q, m, n, k, w, ζ, u (Definition 3) or the uniform distribution over $\mathbb{F}_{q^m}^{u \times n}$.

To solve ResIG-Dec (Problem 2), we do not know a better approach than trying to solve the associated search problem (i.e., ResIG-Search), which is usually done for all decoding-based problems.

Definition 4 (ResErr-Distribution: Restricted Error Distribution)

Input: $q, m, n, k, w, t_{\text{pub}}, u, \gamma, (\mathbf{G}, \mathbf{K})$ from ResIG-Distribution (Definition 3).

Choose uniformly at random

- $\mathbf{e} \xleftarrow{\$} \{\mathbf{x} \in \mathbb{F}_{q^m}^n : \text{rk}_q(\mathbf{x}) = t_{\text{pub}}\}$
- $\alpha \xleftarrow{\$} \mathbb{F}_{q^m}^{mu}$
- $\mathbf{k} \leftarrow \text{ext}^{-1}(\mathbf{K})$
- $\mathbf{y} \leftarrow \text{Tr}(\alpha\mathbf{k}) + \mathbf{e} = \text{Tr}(\alpha\mathbf{m})\mathbf{G} + \text{Tr}(\alpha\mathbf{z}) + \mathbf{e}$

Output: \mathbf{y} .

Problem 3 (*ResG-Search: Restricted Gabidulin Code Search Problem*)

Input: $q, m, n, k, w, t_{pub}, u, \gamma, (\mathbf{G}, \mathbf{K})$ from ResIG-Distribution (Definition 3), \mathbf{y} from ResErr-Distribution (Definition 4) with input (\mathbf{G}, \mathbf{K}) .

Goal: Find $\mathbf{m} \in \mathbb{F}_{q^m}^k$ and $\mathbf{e} \in \{\mathbf{x} \in \mathbb{F}_{q^m}^n : \text{rk}_{\mathbb{F}_q}(\mathbf{x}) \leq t_{pub}\}$ such that $\mathbf{m}\mathbf{G} + \mathbf{e} = \mathbf{y}$.

Problem 3 is equivalent to decoding a codeword of a Gabidulin code that is corrupted by an error that has with high probability a rank weight of $> (n - k)/2$, see Appendix 3.

Problem 4 (*ResG-Dec: Restricted Gabidulin Code Decisional Problem*)

Input: $q, m, n, k, w, t_{pub}, u, \gamma, (\mathbf{G}, \mathbf{K})$ from ResIG-Distribution (Definition 3), $\mathbf{y} \in \mathbb{F}_{q^m}^n$.

Goal: Decide with non-negligible advantage whether \mathbf{y} came from ResErr-Distribution with input $q, m, n, k, w, t_{pub}, u, \gamma, (\mathbf{G}, \mathbf{K})$ or the uniform distribution over $\mathbb{F}_{q^m}^n$.

As before, we are not aware of a faster approach to solve ResG-Dec than through the solution of the associated *search* problem.

We will see in the next subsection that LIGA is IND-CCA2 secure under the assumption that ResG-Dec is a hard problem. As mentioned above, there is an obvious reduction of ResG-Dec to ResG-Search, which can again be efficiently reduced to ResIG-Search. In fact, all relevant attacks studied in Sect. 6 make use of this chain of reduction and aim at solving one of the two search problems.

We are not aware of a reduction of ResIG-Dec to ResIG-Search or one of the other problems. Hence, it might very well be that ResIG-Dec is significantly easier than the other problems. In Sect. 6.3, we show that there is a distinguisher for ResIG-Dec that is efficiently computable if the system parameter ζ is chosen too small. Due to the missing reduction, it is not clear whether or not this distinguisher influences the security of the system.

5.2 Semantic security

In this section, we prove that the public key encryption system Π^{PKE} is semantically secure against chosen plaintext attacks in the standard model under the assumption that ResG-Dec (Problem 4) is difficult. In addition, we show that the IND-CCA2 security of Π^{KEM} reduces tightly to the IND-CPA security of Π^{PKE} in the random oracle model.

5.2.1 IND-CPA security of Π^{PKE}

To show that Π^{PKE} is secure against chosen plaintext attacks, we use the definition of admissibility as in [33].

Definition 5 (Admissibility [33]) The public key encryption scheme $\Pi^{\text{PKE}} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ with a message space \mathcal{M} and a random space \mathcal{R} is called admissible if there is a pair of deterministic polynomial-time algorithms Encrypt_1 and Encrypt_2 satisfying the following property:

- Partible: Encrypt_1 takes as input a public key pk and $r \in \mathcal{R}$, and outputs a $p(\lambda)$ bit-string, where λ is the security parameter. Encrypt_2 takes as input a key pk , and $\mathbf{m} \in \mathcal{M}$ and outputs a $p(\lambda)$ bit-string. Here p is some polynomial in the security parameter λ . Then for any pk given by KeyGen , $r \in \mathcal{R}$, and $\mathbf{m} \in \mathcal{M}$, $\text{Encrypt}_1(\text{pk}, r) \oplus \text{Encrypt}_2(\text{pk}, \mathbf{m}) = \text{Encrypt}(\text{pk}, \mathbf{m}; r)$.

– Pseudorandomness: Let D be a probabilistic algorithm and let

$$\begin{aligned} \text{Adv}_{D, \text{Encrypt}_1}^{\text{ind}}(\lambda) &= \Pr \left[D(\text{pk}, \text{Encrypt}_1(\text{pk}, r)) = 1 \mid r \xleftarrow{\$} \mathcal{R}, (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \right] \\ &\quad - \Pr \left[D(\text{pk}, s) = 1 \mid s \xleftarrow{\$} \mathcal{U}_{p(\lambda)}, (\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(1^\lambda) \right]. \end{aligned}$$

We define the advantage function of the problem as follows. For any t ,

$$\text{Adv}_{\text{Encrypt}_1}^{\text{ind}}(\lambda, t) = \max_D \{ \text{Adv}_{D, \text{Encrypt}_1}^{\text{ind}}(\lambda) \},$$

where the maximum is taken over all D with time-complexity t . Then, the function $\text{Adv}_{\text{Encrypt}_1}^{\text{ind}}(\lambda, t)$ is negligible for every polynomial bounded t and every sufficiently large λ .

In the following we will prove that Π^{PKE} is IND-CPA secure by showing that it fulfills the definition of admissibility.

Theorem 6 *The system $\Pi^{\text{PKE}} = (\text{KeyGen}, \text{Encrypt}, \text{Decrypt})$ is an IND-CPA secure encryption scheme in the standard model under the assumption that the ResG-Dec problem is difficult.*

Proof Let $\text{Encrypt}_1 := \text{Tr}(\alpha \mathbf{k}_{\text{pub}}) + \mathbf{e}$ and $\text{Encrypt}_2 := \mathbf{m} \mathbf{G}_G$. Then, one observes that $\text{Encrypt} = \text{Encrypt}_1 \oplus \text{Encrypt}_2$ and thus Π^{PKE} is *partible*. Since ResG-Dec (Problem 4) is assumed to be difficult, the encryption scheme fulfills *pseudorandomness* and thus, the system is *admissible*. As proven in [33, Lemma 1], if Π^{PKE} fulfills Definition 5, then it is an IND-CPA secure encryption scheme. \square

5.2.2 IND-CCA2 security of Π^{KEM}

We used a transformation proposed in [23] to transform the public key encryption scheme Π^{PKE} into the KEM Π^{KEM} . In the following, we prove that Π^{KEM} is IND-CCA2 secure.

The applied transformation requires that the encryption scheme is γ -spread which is proven to be the case for Π^{PKE} in the following.

Definition 6 (γ -spread, [17,23]) For valid (pk, sk) , the *min-entropy* of $\text{Encrypt}(\mathbf{m}, \text{pk})$ is defined by

$$\gamma(\mathbf{m}, \text{pk}) := -\log_2 \max_{\mathbf{c} \in \hat{\mathcal{C}}} \Pr[\mathbf{c} = \text{Encrypt}(\mathbf{m}, \text{pk}; r)],$$

where $\hat{\mathcal{C}}$ is the set of possible ciphertexts. A public key encryption scheme is called γ -spread if for every valid key pair (pk, sk) and every message $\mathbf{m} \in \mathcal{M}$, $\gamma(\mathbf{m}, \text{pk}) \geq \gamma$. It follows that for all $\mathbf{c} \in \hat{\mathcal{C}}$,

$$\Pr_{r \leftarrow \mathcal{R}}[\mathbf{c} = \text{Encrypt}(\mathbf{m}, \text{pk}; r)] \leq 2^{-\gamma}.$$

Lemma 2 *The public key encryption system Π^{PKE} is γ -spread, where $\gamma = m(t_{\text{pub}} - u) + t_{\text{pub}}(n - t_{\text{pub}} - 1)$.*

Proof We observe that

$$\begin{aligned} \max_{\mathbf{c} \in \hat{\mathcal{C}}} \Pr_{r \leftarrow \mathcal{R}}[\mathbf{c} = \text{Encrypt}(\mathbf{m}, \text{pk}; r)] &= \max_{\mathbf{c} \in \hat{\mathcal{C}}} \Pr_{r \leftarrow \mathcal{R}}[\mathbf{c} = (\mathbf{m}, \mathbf{0}_u) \mathbf{G}_G + \text{Tr}(\alpha \mathbf{k}_{\text{pub}}) + \mathbf{e}] \\ &\stackrel{(i)}{\leq} \max_{\mathbf{c}' \in \hat{\mathcal{C}'}} q^{m u} \Pr_{r \leftarrow \mathcal{R}}[\mathbf{c}' = (\mathbf{m}, \mathbf{0}_u) \mathbf{G}_G + \mathbf{e}] \end{aligned}$$

$$= q^{mu} \frac{1}{|\{e \in \mathbb{F}_{q^m}^n : rk_q(e) = t_{pub}\}|},$$

where \hat{C}' is the set of all vectors in rank distance t_{pub} from $(\mathbf{m}, \mathbf{0}_u) \mathbf{G}_{\mathcal{G}}$ and (i) follows from the fact that there at most q^{mu} choices for α . In [19, Section IV.B], a constructive way of obtaining rank- t_{pub} matrices is given. More precisely, an injective mapping $\varphi : \mathbb{F}_q^{t(n+m-t-1)} \rightarrow \{\mathbf{A} \in \mathbb{F}_q^{n \times m} : rk \mathbf{A} = t\}$ is given. Hence, we have $|\{e \in \mathbb{F}_{q^m}^n : rk_q(e) = t_{pub}\}| \geq q^{t(n+m-t-1)}$. It follows that

$$\begin{aligned} \frac{q^{mu}}{|\{e \in \mathbb{F}_{q^m}^n : rk_q(e) = t_{pub}\}|} &\leq \frac{q^{mu}}{q^{t_{pub}(n+m-t_{pub}-1)}} \\ &= q^{-m(t_{pub}-u)-t_{pub}(n-t_{pub}-1)} \\ &\leq q^{-\gamma}. \end{aligned}$$

□

Theorem 7 *The KEM-DEM $\Pi^{KEM} = (KeyGen, Encaps, Decaps)$ is IND-CCA2 secure in the random oracle model under the assumption that the ResG-Dec problem is difficult.*

Proof Assuming the ResG-Dec is difficult, the encryption Π^{PKE} is IND-CPA secure, see Theorem 6. Further, it is proven in Lemma 2 that Π^{PKE} has γ -spread encryptions. Thus, the system Π^{KEM} can be tightly reduced to Π^{KEM} in the random oracle model as shown in [23].

□

6 Security analysis of LIGA

In this section, we analyze the security of LIGA. As proven in Theorem 6 and 7, the encryption version is IND-CPA secure and the KEM version is IND-CCA2 secure under the assumption that ResG-Dec is difficult. Since there are obvious reductions from ResG-Dec to ResG-Search and from ResG-Dec to ResIG-Search, we will study the hardness of these two search problems in this section (Sect. 6.1 for ResIG-Search and Sect. 6.2 for ResG-Search). In fact, we are not aware of a more efficient method to solve ResG-Dec than through these two search problems.

Although no formal reduction from any of the other three studied problems to ResIG-Dec is known, we study also the hardness of ResIG-Dec (Sect. 6.3). We derive a distinguisher for the public key with exponential complexity in the system parameters, which can be avoided by proper parameter choice.

Due to the nature of the (random) encryption, there are public keys for which the probability that the work factor of some of the ciphertext attacks on ResG-Search (ciphertext attack) is below the designed minimal work factor is not negligible (i.e., $> 2^{-\lambda}$). We show in Sect. 6.4 that these weak keys occur with negligible probability (i.e., $\leq 2^{-\lambda}$) during the random key generation if the parameters are chosen in a suitable way.

6.1 Exponential-time attacks on ResIG-Search

We propose new and summarize known methods that solve ResIG-Search (Problem 1). All studied algorithms have exponential complexity in the code parameters.

Recall that in the decryption algorithm of LIGA, the last u positions of the private key \mathbf{x} have to be a basis of $\mathbb{F}_{q^{mu}}$ over \mathbb{F}_{q^m} . Therefore, not every solution of Problem 1 (ResIG-

Search) can be used as valid private key and Problem 1 is a strictly easier problem than retrieving a valid private key corresponding to a given public key.

6.1.1 Brute-force the vector \mathbf{z} attack

The number of vectors $\mathbf{z} \in \mathbb{F}_{q^{mu}}^n$ that fulfill the conditions stated in Sect. 4.1 is equal to number of possible vectors $\mathbf{s} \in \mathbb{F}_{q^{mu}}^w$ times the number of full rank matrices in $\mathbb{F}_q^{w \times n}$ in reduced row Echelon form. Formally, the number of vectors \mathbf{z} is

$$\underbrace{|\{\mathbf{z} : \mathbf{z} \text{ can occur in Alg. 3}\}|}_{\geq 1} \cdot \underbrace{|\{\mathbf{P} : \mathbf{P} \text{ can occur in Alg. 3}\}|}_{\geq \binom{n}{w}_q} \geq \binom{n}{w}_q.$$

Thus, brute-forcing a vector \mathbf{z} that is a solution to ResIG-Search has work factor

$$WF_{\mathbf{z}} \geq \frac{\binom{n}{w}_q}{\mathcal{N}} \geq \frac{q^{w(n-w)}}{\mathcal{N}},$$

where the latter inequality follows from a lower bound on q -binomials (see [24, Lemma 4]), and

$$\mathcal{N} = \max \left\{ |\mathcal{C}| \cdot \frac{\sum_{i=0}^{w-1} \prod_{j=0}^{i-1} (q^{mu} - q^j) \binom{n}{i}_q}{q^{mun}}, 1 \right\} \tag{9}$$

is the average number of interleaved codewords in a ball of radius w around a uniformly at random chosen interleaved received word.

6.1.2 Interleaved decoding attack

As described in Sect. 3.3, an attacker can apply an interleaved decoder on \mathbf{k}_{pub} to retrieve an alternative private key. A major ingredient of LIGA is that the public key is chosen in a way that this decoding will always fail (i.e., the corresponding linear system of equations does not have a unique solution). However, it is still possible to brute-force search in the solution space of the involved system of equations. This is analyzed in the following. Notice thereby that *any* interleaved codeword in radius at most w is a solution to ResIG-Search.

Problem 1 (ResIG-Search) is equivalent to decoding a codeword of a u -interleaved Gabidulin code that is corrupted by an error \mathbf{E} . The error \mathbf{E} fulfills

$$\left\lfloor \frac{n-k}{2} \right\rfloor < \text{rk}_q(\mathbf{E}) \leq \frac{u}{u+1}(n-k) \quad \text{and} \quad \text{rk}_{q^m}(\mathbf{E}) < \frac{w}{n-k-w} < w$$

and thus, no known algorithm is able to correct it efficiently.

The crucial point of the interleaved decoding algorithms from [27,43] is solving a linear system of equations based on the syndromes with $w+1$ unknowns and φ linearly independent equations which is equivalent to finding the kernel of the matrix in (5), cf. [49, Section 4.1]. For $\zeta \geq \frac{w}{n-k-w}$, the dimension of the solution space is one and all solutions are valid for the remaining decoding steps. For $\zeta < \frac{w}{n-k-w}$, the dimension of the solution space is $w+1-\varphi$ but each valid solution forms only a one-dimensional subspace. An attacker can therefore search in the solution space for a valid solution which requires on average

$$\frac{(q^m)^{w+1-\varphi}}{q^m \cdot \mathcal{N}} = \frac{q^{m(w-\varphi)}}{\mathcal{N}}$$

trials, where \mathcal{N} is the average number of interleaved codewords, see (9).

The size of the solution space is $w + 1 - \varphi$ and clearly maximized for the smallest-possible value of φ , i.e., $\varphi = n - k - w$. In this case, the search through the solution space has work factor

$$WF_{ILD} = \frac{q^{m(w-\zeta(n-k-w))}}{\mathcal{N}}.$$

Since the size of the solution space is maximal for $\varphi = n - k - w$, the repair from Sect. 4.1 with the explicit parameter value $\zeta = 1$ (i.e., $\dim(\langle \mathbf{z}_1, \dots, \mathbf{z}_u \rangle_{\mathbb{F}_{q^m}}) = 1$) is the most secure choice *in this sense*. However, we keep the choice of ζ flexible as the pair-wise linear dependence of the \mathbf{z}_i might decrease the security (we are however not aware of how this fact could be used).

Besides the syndrome-based interleaved decoding algorithms in [27], [43], and [49, p. 64], there is an interpolation-based decoding algorithm [49, Section 4.3 (page 72)]. This interpolation-based algorithm can be interpreted both as a list decoder of interleaved Gabidulin codes with exponential worst-case and average list size or as a probabilistic unique decoder. The probabilistic unique interpolation-based decoder fails if and only if the decoding algorithms in [27], [43], [49, p. 64] fail and therefore the previous analysis applies here as well. For the list decoder, cf. [49, Lemma 4.5], the work factor of the resulting attack is

$$WF_{list, public\ key} \leq \frac{q^{m(u-1)k}}{\mathcal{N}}. \tag{10}$$

Notice that the list of size $q^{m(u-1)k}$ contains many words which are no valid codewords, but we have to go through the whole list to find all valid codewords in radius up to w .

6.1.3 List decoding of the public key attack

Recall that $\mathbf{k}_{pub} = \mathbf{x} \cdot \mathbf{G}_G + \mathbf{z}$. Previously, we have explained why this vector is a corrupted version of a codeword of a u -interleaved Gabidulin code. At the same time, $\mathbf{x} \cdot \mathbf{G}_G$ can be seen as a short Gabidulin code over a large field $\mathbb{F}_{q^{mu}}$ and therefore, if existing, one could apply list decoding algorithms to decode \mathbf{k}_{pub} and obtain \mathbf{x} . The weight of the error \mathbf{z} is larger than the unique decoding radius and therefore a unique decoder cannot be applied to reconstruct \mathbf{x} and a list decoder for radius w is required.

However, such an algorithm has not been found yet. It was even shown in [38,46,48] that for most classes of Gabidulin codes such a polynomial-time list decoding algorithm cannot exist. Note that these results were not known when the original FL cryptosystem was proposed. These results also imply that there is no polynomial-time list decoding algorithm for arbitrary Gabidulin codes beyond the unique decoding radius (such as the Guruswami–Sudan algorithm for Reed–Solomon codes).

6.1.4 Randomized Gabidulin decoding attack on the public key

The public key can be seen as the sum of a Gabidulin codeword over the field $\mathbb{F}_{q^{mu}}$ and an error of weight $w > \frac{n-k}{2}$. Alternatively, as shown in Sect. 3.2, the public key can be seen as an interleaved Gabidulin codeword that is corrupted by an error of weight w (note that this is the reason why all the s_i 's must have full \mathbb{F}_q -rank in Algorithm 3). Each row of (3) is a codeword of a Gabidulin code over \mathbb{F}_{q^m} that is corrupted by an error of rank weight w . Both the corrupted Gabidulin codeword over $\mathbb{F}_{q^{mu}}$ as well as over \mathbb{F}_{q^m} can be decoded using the randomized decoding approach proposed in [39]. Since applying the attack on each

row of the unfolded public key is more efficient, we conclude that the randomized Gabidulin decoding attack on the public key has an average complexity of

$$WF_{\text{RGD}} = \frac{n}{64} \cdot q^{m(n-k)-w(n+m)+w^2+\min\{2\xi(\frac{n+k}{2}-\xi), wk\}}$$

over \mathbb{F}_{q^m} .

6.1.5 Moving to another close error attack

The following attack was suggested by Rosenkilde [41]. It tries to move the vector \mathbf{z} (which we have chosen such that the interleaved decoder fails) to a close vector of the same or smaller rank weight w for which the interleaved decoder for \mathbf{k}_{pub} does not fail.

The idea is to find a vector $\mathbf{y} \in \mathbb{F}_{q^m}^{u \times n}$ such that $\mathbf{z}' := \mathbf{z} + \mathbf{y}$ still has rank weight $\text{rk}_q(\mathbf{z}') \leq w$ and that the rank of the matrix from (5) over \mathbb{F}_{q^m} is at least w . To guarantee the first condition, we want to construct \mathbf{y} such that its extended $um \times n$ matrix over \mathbb{F}_q has a row space $\mathcal{R} := \text{RowSpace}_{\mathbb{F}_q}(\text{ext}_{\mathcal{Y}}(\mathbf{y}))$ that is contained in the one of \mathbf{z} . Since for the original error \mathbf{z} , the matrix (5) has rank $\varphi \leq \zeta(n - k - w)$, \mathcal{R} must have at least \mathbb{F}_q -dimension $w - \varphi \geq w(\zeta + 1) - \zeta(n - k)$. By choosing a random \mathcal{R} with this property and taking a random matrix \mathbf{y} whose extended matrix has \mathbb{F}_q -row space $\text{RowSpace}_{\mathbb{F}_q}(\text{ext}_{\mathcal{Y}}(\mathbf{y})) = \mathcal{R}$, the second condition is fulfilled with high probability.

The complexity of the attack is hence dominated by the complexity of finding a subspace $\mathcal{R} \subseteq \mathbb{F}_q^n$ of dimension $w - \varphi$ that is contained in the w -dimensional \mathbb{F}_q -row space of \mathbf{z} . Since this is unknown, we can find it in a Las-Vegas fashion by repeatedly drawing a subspace uniformly at random. The expected number of iterations until we find a suitable row space is thus one over the probability that a random $(w - \varphi)$ -dimensional subspace of \mathbb{F}_q^n is contained in a given w -dimensional subspace, which is (cf. [15, Proof of Lemma 7]):

$$\frac{\begin{bmatrix} w \\ w-\varphi \end{bmatrix}_q}{\begin{bmatrix} n \\ w-\varphi \end{bmatrix}_q} \approx \frac{q^{\varphi(w-\varphi)}}{q^{(n-w+\varphi)(w-\varphi)}} = q^{-(n-w)(w-\varphi)} \leq q^{-(n-w)(w(\zeta+1)-\zeta(n-k))}.$$

Hence, the attack has work factor

$$WF_{\text{MCE}} = q^{(n-w)(w-\varphi)} \geq q^{(n-w)(w(\zeta+1)-\zeta(n-k))}.$$

6.2 Exponential-time attacks on ResG-Search

Retrieving information about the plaintext from the ciphertext and the public key is equal to solving ResG-Search (Problem 4). In this section, methods that solve this problem are summarized.

6.2.1 Randomized Gabidulin decoding attack on the ciphertext

Each ciphertext of LIGA can be seen as a Gabidulin codeword over \mathbb{F}_{q^m} plus an error:

$$\begin{aligned} \mathbf{c} &= (\mathbf{m}, \mathbf{0}_u) \cdot \mathbf{G}_G + \text{Tr}(\alpha \mathbf{k}_{\text{pub}}) + \mathbf{e} \\ &= \underbrace{[(\mathbf{m}, \mathbf{0}_u) + \text{Tr}(\alpha \mathbf{x})] \cdot \mathbf{G}_G}_{\text{codeword}} + \underbrace{\text{Tr}(\alpha \mathbf{z})}_{\text{error}} + \mathbf{e} \end{aligned}$$

Denote $\tilde{w} := \text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e})$. Then we can use the decoding algorithm proposed in [39], which requires on average at least

$$\frac{n}{64} \cdot q^{m(n-k) - \tilde{w}(n+m) + \tilde{w}^2 + \min\{2\xi(\frac{n+k}{2} - \xi), \tilde{w}k\}} \tag{11}$$

operations in \mathbb{F}_q^m .

Clearly, the complexity of the algorithm strongly depends on the value \tilde{w} , which in turn depends on the generated keys. In general, $\tilde{w} = w + t_{\text{pub}}$, but for some choices of \mathbf{z} , α , and \mathbf{e} , the rank \tilde{w} is smaller. For this issue, see Sect. 6.4 and Appendix 3, where we study the probability that \tilde{w} is small, both for randomness in the encryption (random choice of α and \mathbf{e}) and the key generation (random choice of \mathbf{z}). Some extremely rarely occurring keys (weak keys) thereby result in relatively high probabilities that \tilde{w} is small.

However, we can choose the system parameters such that both, the probability of a weak key as well as the conditional probability that $\tilde{w} < w$ given a non-weak key is below $2^{-\lambda}$. Hence, with overwhelming probability, a random key and ciphertext result in a ciphertext error of rank weight $\tilde{w} \geq w$ and the work factor of this attack is always at least as large as the ‘‘Randomized Gabidulin Decoding Attack on the Public Key’’ in Sect. 6.1.

6.2.2 List decoding of the ciphertext attack

As described in the Randomized Gabidulin Decoding Attack on the Ciphertext above, the ciphertext of LIGA is a codeword of a Gabidulin code, corrupted by an error of rank weight \tilde{w} . Hence, an attacker can try to decode the ciphertext directly. Since τ is always greater than the unique decoding radius $\lfloor \frac{n-k}{2} \rfloor$ of the Gabidulin code, this would require the existence of an efficient (list) decoding algorithm up to radius τ . As explained previously, there is no such algorithm and bounds on the list size prove that there cannot exist a generic list decoding algorithm for all Gabidulin codes which indicates that list decoding is a hard problem.

However, to be secure, we have considered list decoding as follows for the security level of our system. The list size $\mathcal{L}_c, \text{ worst}$ denotes a lower bound on the *worst-case* work factor of list decoding. For example, for a Gabidulin code with parameters $n|m$ and $\text{gcd}(n, n - \tau) \geq 2$, there is a received word such that there are at least

$$\mathcal{L}_c, \text{ worst} \geq \max \left\{ \frac{\lfloor \frac{n/g}{(n-\tau)/g} \rfloor q^g}{q^{n(\tau/g-1)}} : g \geq 2, g|\text{gcd}(n, n - \tau) \right\} \tag{12}$$

codewords in rank distance at most τ to it.

Although $\mathcal{L}_c, \text{ worst}$ does not imply any statement about the average list size/average work factor, it provides an estimate of the order of magnitude of the work factor of a hypothetical list decoding attack. For our suggested parameters, we have ensured that the value of $\mathcal{L}_c, \text{ worst}$ is sufficiently large in the proposed sets of parameters in Sect. 7.

6.2.3 Combinatorial rank syndrome decoding (RSD) attack

The ciphertext can be interpreted as a codeword from a code of dimension k (see [16]), generated by the generator matrix

$$\begin{pmatrix} \mathcal{M}_{k-u, q}(\mathbf{g}) \\ \text{Tr}(\gamma_1 \mathbf{k}_{\text{pub}}) \\ \vdots \\ \text{Tr}(\gamma_u \mathbf{k}_{\text{pub}}) \end{pmatrix}.$$

Since the structure of this code only permits decoding like a random rank-metric code, it can be decoded with the combinatorial syndrome decoding attack from [4] whose complexity is in the order of

$$WF_{CRSD} = (n - k)^3 m^3 q^{t_{pub} \lceil \frac{(k+1)m}{n} \rceil - m}.$$

6.2.4 Algebraic RSD attack

As described in the previous section, the ResG-Search problem can be solved by decoding an error of rank weight t_{pub} in a random $[n, k]$ code. Beside the combinatorial approach, there exist algebraic algorithms to solve the Problem.

In [6], the RSD problem is expressed as a multivariate polynomial system and is solved by computing a Gröbner basis. The complexity of that attack is generally smaller than the combinatorial approach. In case there is a unique solution to the system, then the work factor of the algorithm is

$$WF_{Gr} = \begin{cases} \left[\frac{((m+n)t_{pub})^{t_{pub}}}{t_{pub}!} \right]^\mu & \text{if } m \binom{n-k-1}{t_{pub}} \leq \binom{n}{t_{pub}} \\ \left[\frac{((m+n)t_{pub})^{t_{pub}+1}}{(t_{pub}+1)!} \right]^\mu & \text{otherwise,} \end{cases}$$

where μ is the exponent in the complexity expression of the used matrix multiplication algorithm. The asymptotically fastest known matrix multiplication algorithm has an exponent of $\mu < 2.373$. As the authors of [6], we use $\mu = 2.807$ to compute the work factors in this paper since it corresponds to Strassen’s algorithm, which is (due to the hidden constant) in practice the fastest algorithm for large (but still reasonably small) matrix sizes.

Very recently, a new algebraic algorithm was proposed to solve the RSD problem [7]. It divides the RSD problem instances into two categories. If

$$m \binom{n-k-1}{t_{pub}} \geq \binom{n}{t_{pub}} - 1,$$

we are in the overdetermined case and the proposed algorithm has work factor

$$WF_{Wogr} = m \binom{n-p-k-1}{t_{pub}} \binom{n-p}{t_{pub}}^{\mu-1}$$

in \mathbb{F}_q , where $p = \min\{i : i \in \{1, \dots, n\}, m \binom{n-i-k-1}{t_{pub}} \geq \binom{n-i}{t_{pub}} - 1\}$. Otherwise, we are in the underdetermined case in which the algorithm has work factor

$$WF_{Wogr} = \min\{WF_{Under}, WF_{Hybrid}\}.$$

We have

$$WF_{Hybrid} = q^{at_{pub}} m \binom{n-k-1}{t_{pub}} \binom{n-a}{t_{pub}}^{\mu-1}$$

with $a = \min\{i : i \in \{1, \dots, n\}, m \binom{n-k-1}{t_{pub}} \geq \binom{n-i}{t_{pub}} - 1\}$. Further, for $0 < b < t_{pub} + 2$ and $A_b - 1 \leq B_b + C_b$,

$$WF_{Under} = \frac{B_b \binom{k+t_{pub}+1}{t_{pub}} + C_b(mk+1)(t_{pub}+1)}{B_b + C_b} \left(\sum_{j=1}^b \binom{n}{t_{pub}} \binom{mk+1}{j} \right)^2,$$

where $A_b := \sum_{j=1}^b \binom{n}{t_{\text{pub}}} \binom{mk+1}{j}$, $B_b := \sum_{j=1}^b m \binom{n-k-1}{t_{\text{pub}}} \binom{mk+1}{j}$ and

$$C_b := \sum_{j=1}^b \sum_{i=1}^j \left((-1)^{i+1} \binom{n}{t_{\text{pub}} + i} \binom{m+i-1}{i} \binom{mk+1}{j-i} \right).$$

We denote the minimum of the work factors of the two algorithms as the work factor of the algebraic RSD attack, i.e.,

$$\text{WF}_{\text{ARSD}} = \min\{\text{WF}_{\text{Gr}}, \text{WF}_{\text{Wogr}}\}.$$

Note that for algebraic decoding, it is neither known how to improve the complexity by using the fact that there are multiple solutions, nor it is known how to speed up the algorithm in the quantum world.

6.2.5 Linearization attack

In [16], a message attack was proposed which succeeds for some parameters with high probability in polynomial time.

Lemma 3 (Linearization Attack [16]) *Let $\mathbf{k}_{\text{pub}}^{(i)} = \text{Tr}(\gamma_i \mathbf{k}_{\text{pub}})$ for $i = 1, \dots, u$ and*

$$\mathbf{M} = \begin{pmatrix} \mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{c}) \\ -\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{k}_{\text{pub}}^{(1)}) \\ \vdots \\ -\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{k}_{\text{pub}}^{(u)}) \\ -\mathcal{M}_{k+t_{\text{pub}}-u,q}(\mathbf{g}) \end{pmatrix}. \tag{13}$$

Then, the encrypted message \mathbf{m} can be efficiently recovered if the left kernel of \mathbf{M} has dimension $\dim(\ker(\mathbf{M})) = 1$.

If $(u + 2)t_{\text{pub}} + k > n$, then \mathbf{M} has at least two more rows than columns and we have $\dim(\ker(\mathbf{M})) > 1$. If \mathbf{k}_{pub} is random and $(u + 2)t_{\text{pub}} + k \leq n$, the attack is efficient with high probability [16].

Lemma 4 *Let \mathbf{M} be as in (13). Then,*

$$\text{rk}_{q^m}(\mathbf{M}) \leq \min\{\varphi + k + 2t_{\text{pub}} - u, n\}.$$

Proof We can write

$$\begin{aligned} \mathbf{k}_{\text{pub}}^{(i)} &= \text{Tr}(\gamma_i \mathbf{k}_{\text{pub}}) \\ &= \text{Tr}(\gamma_i \mathbf{x}) \cdot \mathcal{M}_{k,q}(\mathbf{g}) + \mathbf{z}_i, \end{aligned}$$

so by elementary row operations, we can transform \mathbf{M} into

$$\mathbf{M}' = \begin{pmatrix} \mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{c}) \\ -\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{z}_1) \\ \vdots \\ -\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{z}_u) \\ -\mathcal{M}_{k+t_{\text{pub}}-u,q}(\mathbf{g}) \end{pmatrix}.$$

Due to $w + 2t_{\text{pub}} < n - k$, the matrix $\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{z}_i)$ is a sub-matrix of $\mathcal{M}_{n-k-w,q}(\mathbf{z}_i)$, so

$$\begin{aligned} \text{rk}_{q^m}(\mathbf{M}) &= \text{rk}_{q^m}(\mathbf{M}') \\ &\leq \varphi + \text{rk}_{q^m}(\mathcal{M}_{t_{\text{pub}}+1,q}(\mathbf{c})) + \text{rk}_{q^m}(\mathcal{M}_{k+t_{\text{pub}}-u,q}(\mathbf{g})) \\ &= \varphi + k + 2t_{\text{pub}} - u. \end{aligned}$$

Further, since the number of columns of \mathbf{M} is equal to n ,

$$\text{rk}_{q^m}(\mathbf{M}) \leq n.$$

□

The linearization attack is inefficient if the rank of \mathbf{M} is smaller than its number of rows, which implies the following, stronger version of the original statement in [16].

Theorem 8 *If $t_{\text{pub}} > \frac{n-k}{u+2}$ or $\varphi < u(t_{\text{pub}} + 1)$, the linearization attack in [16] is inefficient and its work factor is*

$$WF_{\text{Lin}} = q^{m \cdot \max\{ut_{\text{pub}}+u+1-\varphi, (u+2)t_{\text{pub}}+k+1-n\}}.$$

The first condition in Theorem 8 is again fulfilled by the choice of w in Table 1. The second one reads $t_{\text{pub}} > \frac{\varphi}{u} + 1$, and for any valid φ , there are choices of w such that t_{pub} fulfills this inequality for any $u > 1$.

6.2.6 Algebraic attacks

Faure and Loidreau [16] also described two message attacks of exponential worst-case complexity. The first one is based on computing gcds of polynomials of degrees

$$WF_{\text{GCD}} = q^{m(u-1)} \frac{q^{t_{\text{pub}}+1} - 1}{q - 1}. \tag{14}$$

Since computing the gcd of two polynomials can be implemented in quasi-linear time in the polynomials' degree, (14) gives an estimate on the work factor of this attack. The second algebraic attack is based on finding Gröbner bases of a system of $n_p = \binom{n}{k+2t_{\text{pub}}-u+1}$ many polynomials of degree approximately $d_p = \frac{q^{t_{\text{pub}}+1}-1}{q-1}$. The attack is only efficient for small code parameters, cf. [16, Sec. 5.3]. Since the average-case complexity of Gröbner bases algorithms is hard to estimate, we cannot directly relate n_p and d_p to the attack's work factor. Faure and Loidreau choose the code parameters such that $n_p \approx 2^{32}$ and $d_p = 127$ and claim that the attack is inefficient for these values. Our example parameters in Sect. 7 result in at least these values.

6.2.7 Overbeck-like attack

The key attack described in [28, Ch. 7, Sec. 2.1] is based on a similar principle as Overbeck uses to attack the McEliece cryptosystem based on Gabidulin codes [35]. The attack from [28, Ch. 7, Sec. 2.1] cannot be applied if

$$w \geq n - k - \frac{k - u}{u - 1}.$$

6.2.8 Brute-force attack on the element α

An attacker can brute-force $\alpha \in \mathbb{F}_{q^{mu}}$, which has a complexity of

$$WF_{\alpha} = q^{mu}.$$

By knowing α , he just needs to apply an efficient decoding algorithm on $\tilde{\mathbf{c}} = \mathbf{c} - \text{Tr}(\alpha \mathbf{k}_{\text{pub}})$ to retrieve the secret message.

6.3 Exponential-time attacks on ResIG-Dec

We have seen in Sect. 5 that LIGA is IND-CCA2 secure under the assumption that ResG-Dec is a hard problem. The two previous subsections analyzed all known attacks on the ResG-Search and ResIG-Search problems, which are relevant since there is an obvious reduction of ResG-Dec to these search problems.

In the following, we study Problem ResIG-Dec (which translates to distinguishing the public key from a random vector in $\mathbb{F}_{q^{mu}}^n$), which is different in the sense that we do not know an efficient reduction from ResG-Dec (or one of the search problems) to ResIG-Dec. In other words, even if distinguishing the public key is easy, it might still be hard to distinguish the ciphertext. Nevertheless, we study the hardness of ResIG-Dec in the following and present a distinguisher, which is efficient to compute if ζ is chosen small. The distinguisher is as follows.

Recall the choice of \mathbf{k}_{pub} in Algorithm 3. We have

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

Expand \mathbf{k}_{pub} into a $u \times n$ matrix over \mathbb{F}_{q^m} and choose any $\zeta + 1$ rows. As the \mathbb{F}_{q^m} -expansion of the error \mathbf{z} has \mathbb{F}_{q^m} -rank ζ , there are at least $q^m - 1$ many non-trivial \mathbb{F}_{q^m} -linear combinations of these $\zeta + 1$ rows that are codewords of $G_{\mathcal{G}}$. This is not true with high probability for a random $u \times n$ matrix over \mathbb{F}_{q^m} .

Thus, by repeatedly randomly linearly combining these $\zeta + 1$ rows and checking whether the result is a codeword of $G_{\mathcal{G}}$, we obtain a Monte-Carlo algorithm with an expected work factor of

$$WF_{\mathbf{k}_{\text{pub}}, \text{distinguisher}} = q^{m\zeta},$$

neglecting the cost of checking whether a vector in $\mathbb{F}_{q^m}^n$ is a codeword. Hence, if $m\zeta$ is smaller than the security parameter of the system, this distinguisher is feasible to compute.

6.4 Avoiding Weak Keys

As already discussed in Sect. 6.2, the work factors of the ‘‘Randomized Gabidulin Decoding Attack on the Ciphertext’’ and the ‘‘List Decoding of the Ciphertext Attack’’ depend on the rank of the error part $\text{Tr}(\alpha \mathbf{z}) + \mathbf{e}$ of the ciphertext (seen as codeword plus error). Generically, this error has weight $t_{\text{pub}} + \mathbf{e}$, but due to the trace operation and the addition, the rank might be smaller.

In Appendix 3, we will analyze the probability that for a given key (i.e., \mathbf{z} in this case) and a random encryption (random choices of α and \mathbf{e}) the rank is significantly smaller than expected (we use $< w$ as a threshold, see Sect. 6.2). Briefly summarized, we get the following results.

It turns out that this probability heavily depends on the minimum distance of the code \mathcal{A} used to generate \mathbf{z} in Algorithm 3. The smaller this minimum distance, the larger the probability that the rank is low. More precisely, for a given \mathcal{A} of minimum distance $2 \leq t \leq w - \zeta + 2$

$$\Pr(\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) < w) \leq q^{-m\zeta} + 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)} \left(n + \frac{t-3w-t_{\text{pub}}}{2} \right),$$

cf. Theorem 11 in Appendix 3.

Due to the above discussion, we call a key with $\Pr(\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) < w) > 2^{-\lambda}$ a *weak key*. In Appendix 3, we derive an upper bound on the probability of choosing weak key (i.e., an \mathcal{A} of too small minimum distance) in Algorithm 3. For $\zeta q^{\zeta w - m} \leq \frac{1}{2}$, this bound is roughly

$$\Pr(\text{weak key}) \leq \Theta\left(q^{m[t-(w-\zeta+2)]}\right),$$

cf. Remark 2 (see Theorem 11 for a non-asymptotic bound) in Appendix 3, where t is the smallest minimum distance for which the key is not weak.

It can be seen that the parameters of LIGA can be chosen such that there is a t with $2 \leq t \leq w - \zeta + 2$ such that both $\Pr(\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) < w)$ (for any non-weak key) and $\Pr(\text{weak key})$ are smaller than $2^{-\lambda}$. This is the case for all parameters proposed in Table 1.

6.5 Summary of the work factors

In this section, we recall the conditions on the choice of the parameters such that all known attacks are inefficient and summarize their work factors. Furthermore, we give specific parameters and compare LIGA to other code-based cryptosystems.

In the following, we choose the parameters q, m, n, k, u, w , and t_{pub} as in Table 1. Recall that this choice of w prevents the Overbeck-like attack (Sect. 6.2.7) and results in an exponential work factor of the linearization attack (Sect. 6.2.5).

Furthermore, we choose ζ to be small such that the work factor of searching the exponentially-large output of the interleaved decoding attack (Sect. 6.1.2) is large. Note that the latter attack returns an exponentially-large output if and only if the GOT [21] attack fails, cf. Theorem 3.

The resulting considered work factors are summarized in Table 2. In addition to these work factors, we have considered the following requirements:

- The work factor of the second algebraic attack in [16] (cf. Sect. 6.2.6) is unknown. Hence, we choose the code parameters such that the resulting non-linear system of equations occurring in the attack consists of more than $n_p \approx 2^{32}$ many polynomials of degree at least $d_p = 127$. This is the same choice as in [16].
- Since there is no efficient list decoder for Gabidulin codes, the work factor of the list decoding the public key or the ciphertext in Sect. 6.2.2 is not known. However, we do have a lower bound on the worst-case work factor for some codes, given by the maximal list size \mathcal{L}_c , worst in (12). In all examples for which the bound holds, we chose the parameters such that $\log_2(\mathcal{L}_c, \text{worst})$ is much larger than the claimed security level.
- The probability of generating a weak key should be negligible. Thus, we choose the parameters such that $\zeta q^{\zeta w - m} \leq \frac{1}{2}$ and

$$\Pr(\text{weak key}) \leq \frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \binom{w}{i}_q \prod_{j=0}^{i-1} (q^m - q^j) - 1 \right)$$

Table 2 Summary of the Discussed Attacks' Work Factor

Name of the attack	Work factor
Brute-force \mathbf{z} (Sect. 6.1.1)	$WF_{\mathbf{z}} = \frac{q^{w(n-w)}}{\mathcal{N}}$
Interleaved Decoding (Sect. 6.1.2)	$WF_{ILD} = \frac{q^{m(w-\zeta(n-k-w))}}{\mathcal{N}}$
Randomized Decoding (Sect. 6.1.4)	$WF_{RGD} = \frac{n}{64} q^{m(n-k)-w(n+m)+w^2+\min\{2\xi(\frac{n+k}{2}-\xi), wk\}}$
Moving to Close Error (Sect. 6.1.5)	$WF_{MCE} = q^{(n-w)(w(\zeta+1)-\zeta(n-k))}$
Combinatorial RSD (Sect. 6.2.3)	$WF_{CRSD} = (n-k)^3 m^3 q^{t_{\text{pub}} \lceil \frac{(k+1)m}{n} \rceil - m}$
Algebraic RSD (Sect. 6.2.4)	WF_{ARSD}
Linearization (Sect. 6.2.5)	$WF_{Lin} = q^{m \cdot \max\{ut_{\text{pub}}+u+1-\varphi, (u+2)t_{\text{pub}}+k+1-n\}}$
GCD based attack (Sect. 6.2.6)	$WF_{GCD} = q^{m(u-1)} \frac{q^{t_{\text{pub}}+1}-1}{q-1}$
Brute-force α (Sect. 6.2.8)	$WF_{\alpha} = q^{mu}$
Distinguisher for \mathbf{k}_{pub} (Sect. 6.3)	$WF_{\mathbf{k}_{\text{pub}}, \text{distinguisher}} = q^{m\zeta}$

Table 3 Parameter sets for 128, 192 and 256 bit security

Parameter set	q	u	k	n	m	ζ	w	t_{pub}	R
LIGA-128	2	5	53	92	97	2	27	6	0.52
LIGA-192	2	5	69	120	127	2	35	8	0.53
LIGA-256	2	5	85	148	149	2	43	10	0.54

$$\leq 2^{-\lambda},$$

where λ is the security parameter and

$$t := \min \left\{ t : q^{-m\zeta} + 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)} \left(n + \frac{t-3w-t_{\text{pub}}}{2} \right) \leq 2^{-\lambda} \right\}.$$

7 Parameters and key sizes

We propose parameters for security levels of 128 bit, 192 bit and 256 bit in Table 3, where $R = \frac{k-u}{n}$ denotes the rate. The parameters are chosen in a way that we can send at least 256 bit of information and thus the system can be used as a KEM. Further, we use a security margin of at least 20 bit. For all parameters, the algebraic attack based on computing gcds of polynomials is the most efficient attack.

To evaluate the performance of LIGA, we compare it to the IND-CCA-secure version [42] of Loidreau's system [29] and the NIST proposals RQC [2], ROLLO [1], BIKE [3] and Classic McEliece [8]. We show the sizes of the private key sk , the public key pk and the ciphertext ct in Byte in Table 4 as proposed by the authors.⁵ Note that in LIGA we can use a similar representation of the secret key and public key as in RQC, see [2, Section 2.3.3]. More

⁵ The size of the secret key of Loidreau's system is not shown since the authors of [42] do not state how they represent sk .

Table 4 Comparison of memory costs of sk , pk and the ciphertext ct in Byte with IND-CCA-secure Loidreau [42] and the NIST proposals RQC [2], ROLLO [1], BIKE [3] and Classic McEliece [8]. The entry 'yes' in the column DFR indicates that a scheme has a decryption failure rate larger than 0

System name	sk	pk	ct	Security	DFR
LIGA-128	40	5618	1116	128 bit	No
RQC-I	40	1834	3652	128 bit	No
ROLLO-I-128	40	696	696	128 bit	Yes
Loidreau-128	–	6720	464	128 bit	No
BIKE-2 Level 1	249	1271	1271	128 bit	Yes
McEliece348864	6452	26,1120	128	128 bit	No
LIGA-192	40	9565	1905	192 bit	No
RQC-II	40	2853	5690	192 bit	No
ROLLO-I-192	40	958	958	192 bit	Yes
Loidreau-192	–	11,520	744	192 bit	No
BIKE-2 Level 2	387	2482	2482	192 bit	Yes
McEliece460896	13,568	524,160	188	192 bit	No
LIGA-256	40	13,823	2757	256 bit	No
RQC-III	40	4090	8164	256 bit	No
ROLLO-I-256	40	1371	1371	256 bit	Yes
Loidreau-256	–	16,128	1024	256 bit	No
BIKE-2 Level 3	513	4094	4094	256 bit	Yes
McEliece6688128	13,892	1,044,992	240	256 bit	No

precisely, we just store a seed of size 40 bytes to generate the secret key $sk = (\mathbf{x}, \mathbf{P}_{[w+1,n]})$ which leads to secret key size of 40 bytes. The vector \mathbf{g} in the public key $pk = (\mathbf{g}, \mathbf{k}_{pub})$ can be also stored as a seed of size 40 bytes. Thus, the size of the public key pk is equal to $\lceil \frac{mnu \log_2(q)}{8} \rceil + 40$ bytes. The size of the ciphertext ct is given by $\lceil nm \log_2(q) \rceil$ bytes.

In [11], a generalization of Grover's algorithm is proposed that finds the roots of a function f in $\sqrt{2^b/r}$ function evaluations on average, where r is the number of roots and 2^b is the number of possible inputs of f . Thus, in a post-quantum world, all shown attacks on LIGA may be accelerated using Grover's algorithm except the GCD based attack and the Algebraic RSD attack. Similar to the quantum information-set decoding algorithm described in [9], the mentioned attacks have in common that they guess an element from a large set and then evaluate in polynomial time whether the guess leads to the desired outcome. If the desired outcome is obtained, the system can be broken in polynomial time using exactly this guess. Thus, the work factor of these algorithms is the product of the complexity of checking whether the guess leads to the desired outcome times the inverse of the probability that the guess leads to the desired outcome. Thus, we can easily construct a function f that takes as input a guess and checks in polynomial time whether the guess is as desired. If this is the case, f outputs 0 and otherwise anything except 0. Then, we can apply Grover's algorithm to find a root of that function f . Such a root is then an element of the set that leads to the desired outcome. In this way, Grover's algorithm reduces the work factor to the product of the polynomial time required for checking the guess times the *square-root* of the inverse of the probability that the guess is as desired. For the GCD based attack, we do not know how to improve the work factor by a quantum computer since the stated complexity already assumes a running

time linear in the polynomials' degree. Further, at the current state of research, there is no quantum speed up known for the Algebraic RSD attack [2, Section 6.3]. Using the described work factors, we obtain for LIGA-128, LIGA-192 and LIGA-256 a post-quantum security level of 97.5 bit, 127.5 bit and 157.5 bit, respectively, where Moving to Close Error is the most efficient attack for all three parameter sets.

8 Conclusion

In this paper, we presented a new rank-metric code-based cryptosystem: LIGA. LIGA uses a new coding-theoretic interpretation of the Faure–Loidreau system. We showed that the ciphertext is a corrupted codeword of a Gabidulin code, where to an unauthorized receiver, the error weight is too large to be correctable. The authorized user knows the row space of a part of the error and is thus able to correct the error. Further, we derived that a part of the public key can be seen as a corrupted codeword of an interleaved Gabidulin code and that in the original FL system, an interleaved Gabidulin decoder can efficiently recover the private key from this part of the public key with high probability. We proved that the condition that interleaved Gabidulin decoders fail is equal to the condition that the severe attack by Gaborit, Otmani and Talé Kalachi fails.

Based on the latter observation, we chose LIGA's key generation algorithm such that interleaved Gabidulin decoders fail which in turn implies that the attack by Gaborit *et al.* fails.

We proposed two versions of LIGA and proved that the public key encryption is IND-CPA secure and the KEM is IND-CCA2 secure under the assumption that the ResG-Dec problem is hard. We extensively analyzed the security of this decisional problem by studying attacks on the ResG-Search, ResIG-Search, and ResIG-Dec (recall that there is a reduction of ResG-Dec to each of the two search problems). All studied attacks have an exponential work factor in the proposed parameter ranges and can be avoided by parameter choice.

Finally, we presented parameters for security levels of 128, 192 and 256 bit and compared them to the NIST proposals RQC, ROLLO, BIKE, Classic McEliece and a rank-metric McEliece-like system proposed by Loidreau. It was observed that LIGA has small ciphertext sizes as well as relatively small key sizes. Encryption and decryption correspond to encoding and decoding of Gabidulin codes, for which efficient and constant-time algorithms exist. Further, the proposed system guarantees decryption and is not based on hiding the structure of a code. Hence, the LIGA system should be considered as an alternative of small ciphertext and key size.

Acknowledgements The work of J. Renner and A. Wachter-Zeh was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 801434). S. Puchinger received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No. 713683. We would like to thank Johan Rosenkilde for proposing the “moving to a close error” attack. Also, we are thankful to Michael Schelling for his observation that decryption of the FL system can be seen as error-erasure decoding. Further, we thank Pierre Loidreau for his valuable comments on a previous version of this paper. We are also grateful to Alessandro Neri for fruitful discussions that helped to achieve the results in Appendix 3.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence,

and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A: Practical considerations on the key generation

We discuss practical aspects related to the following lines of the modified key generation algorithm (Algorithm 3).

$$\begin{aligned}
 & \text{3 } \mathcal{A} \stackrel{\$}{\leftarrow} \left\{ \text{subspace } \mathcal{U} \subseteq \mathbb{F}_{q^m}^w : \dim \mathcal{U} = \zeta, \mathcal{U} \text{ has a basis of full-}\mathbb{F}_q\text{-rank elements} \right\} \\
 & \text{3' } \begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_u \end{pmatrix} \stackrel{\$}{\leftarrow} \left\{ \begin{pmatrix} \mathbf{s}'_1 \\ \vdots \\ \mathbf{s}'_u \end{pmatrix} : \langle \mathbf{s}'_1, \dots, \mathbf{s}'_u \rangle_{\mathbb{F}_{q^m}} = \mathcal{A}, \text{rk}_q(\mathbf{s}'_i) = w \forall i \right\}
 \end{aligned}$$

We conjecture that the set from which \mathcal{A} is sampled is almost the entire set of ζ -dimensional subspaces of $\mathbb{F}_{q^m}^w$ (or, equivalently, of linear $[w, \zeta]_{q^m}$ codes). Using a combinatorial argument on the known number of full-rank codewords of MRD codes, we prove in Lemma 8 (Appendix 3) that MRD codes always have a basis consisting of full-rank codewords. Since the weight enumerator is not known in general for non-MRD codes, we cannot give a proof, but we expect that most codes that are close to MRD (i.e., d is close to $n - k + 1$) also have such a basis. The conjecture is then implied by the fact that (close-to) MRD codes constitute the majority of linear codes [12,32] for the parameters considered here.

Since it is hard to check if a randomly drawn code admits a basis of full- \mathbb{F}_q -rank codewords in the worst case, these arguments also imply a practical method on how to implement Lines 3 and 3' in practice: sample uniformly at random from the set of $[w, \zeta]_{q^m}$ codes. With overwhelming probability, the code is close to MRD and a large proportion of its codewords have full \mathbb{F}_q -rank. Randomly choosing u codewords will thus give a generating set consisting of full-rank codewords with high probability. Only if no basis is found after a given number of trials, one needs to formally check if the code does not admit a generating set of full- \mathbb{F}_q -rank codewords. This gives a Las-Vegas-type algorithm with (supposedly) small expected running time.

The worst case of this algorithm (i.e., no suitable generating set is found after a given number of trials) occurs with extremely small probability (provably it is close to the probability of drawing no MRD code at random, it might be even smaller in reality since also “near-MRD” might have suitable bases). Nevertheless, the worst-case complexity is still quite large. Alternatively, one can draw a new code \mathcal{A} if no generating set is found after a given number of trials. This, however, slightly changes the random experiment from which the code \mathcal{A} is drawn. The only part of this paper which is influenced by such a modification is Sect. 6.4, a summary of Appendix 3, which studies weak keys (i.e., keys for which there is a non-negligible probability that the error part of the ciphertext has too low rank and is vulnerable to a feasible ciphertext attack). A key is weak only if the minimum distance of \mathcal{A} is small. By parameter choice, the probability that such a key is generated can be made arbitrarily small (cf. Appendix 3). By the same arguments as above, we conjecture that if the probability of obtaining a generating set of full- \mathbb{F}_q -rank codewords by drawing u codewords uniformly at random is small, then also the minimum distance of the code must be small (i.e.,

far away from MRD). In summary, we expect (but cannot prove) that this change of drawing procedure results in an even smaller weak-key probability than predicted by Theorem 11 (Appendix 3).

B: Decryption as error-erasure decoding

In the following, we give a coding-theoretic interpretation of the ciphertext of the original FL system and of LIGA, which—to the best of our knowledge—has not been observed before.

Lemma 5 Fix a basis of \mathbb{F}_{q^m} over \mathbb{F}_q . Then, the matrix representation of the ciphertext can be written in the form

$$\text{ext}(\mathbf{c}) = \mathbf{C}_G + \mathbf{A}^{(C)}\mathbf{B}^{(C)} + \mathbf{E}, \tag{15}$$

where

- $\mathbf{C}_G = \text{ext}([\mathbf{m} + \text{Tr}(\alpha\mathbf{x})] \cdot \mathbf{G}_G) \in \mathbb{F}_q^{m \times n}$ is unknown and a codeword of a Gabidulin code,
- $\mathbf{A}^{(C)} = \text{ext}(\text{Tr}(\alpha\mathbf{s})) \in \mathbb{F}_q^{m \times w}$ is unknown,
- $\mathbf{B}^{(C)} = (\mathbf{P}^{-1})_{[1, \dots, w]} \in \mathbb{F}_q^{w \times n}$ is known and
- $\mathbf{E} = \text{ext}(\mathbf{e}) \in \mathbb{F}_q^{m \times n}$ is unknown.

Proof Due to the \mathbb{F}_{q^m} -linearity of the trace map Tr and the fact that the entries of the matrices \mathbf{G}_G and \mathbf{P}^{-1} are in \mathbb{F}_{q^m} , we can write the ciphertext as follows.

$$\begin{aligned} \mathbf{c} &= \mathbf{m}\mathbf{G}_G + \text{Tr}(\alpha\mathbf{k}_{\text{pub}}) + \mathbf{e} \\ &= \mathbf{m}\mathbf{G}_G + \text{Tr}(\alpha\mathbf{x}\mathbf{G}_G + \alpha\mathbf{z}) + \mathbf{e} \\ &= [\mathbf{m} + \text{Tr}(\alpha\mathbf{x})] \mathbf{G}_G + \text{Tr}(\alpha\mathbf{z}) + \mathbf{e} \\ &= [\mathbf{m} + \text{Tr}(\alpha\mathbf{x})] \mathbf{G}_G + \text{Tr}(\alpha(\mathbf{s}|\mathbf{0})\mathbf{P}^{-1}) + \mathbf{e} \\ &= [\mathbf{m} + \text{Tr}(\alpha\mathbf{x})] \mathbf{G}_G + \text{Tr}(\alpha\mathbf{s}(\mathbf{P}^{-1})_{[1, w]}) + \mathbf{e} \\ &= [\mathbf{m} + \text{Tr}(\alpha\mathbf{x})] \mathbf{G}_G + \text{Tr}(\alpha\mathbf{s})(\mathbf{P}^{-1})_{[1, w]} + \mathbf{e}. \end{aligned}$$

Since the entries of $(\mathbf{P}^{-1})_{[1, \dots, w]}$ are in \mathbb{F}_q , the expansion of the ciphertext into the \mathbb{F}_q -basis of \mathbb{F}_{q^m} can be written as in (15) above. □

Theorem 9 The message vector \mathbf{m} can be reconstructed by the error-erasure decoders in [20,45,51] (as well as their accelerations in [36,37]) and Steps 4 and 5 of Algorithm 5.

Proof As seen in Lemma 5, we can decompose the matrix representation of the ciphertext into a codeword plus an error that is partially known. In fact, the decomposition is of the form as in (1) (see Sect. 2.2), so $\mathbf{m} + \text{Tr}(\alpha\mathbf{x})$ can be reconstructed by the error-erasure decoders in [20,45,51] since the decoding condition (2) reads as

$$w + 2 \text{rk}_q(\mathbf{E}) = w + 2t_{\text{pub}} \leq n - k$$

in this case and is fulfilled by Table 1.

The message \mathbf{m} can then be recovered from $\mathbf{m} + \text{Tr}(\alpha\mathbf{x})$ using the same steps as in Algorithm 5. □

Theorem 9 leads to the following observation. The ciphertext is a codeword plus an error of rank weight $w + t_{\text{pub}}$, which is beyond the unique decoding radius. The legitimate receiver can only decrypt since she knows the (w -dimensional) row space of a part of the

error. Although the attacker knows the code, she cannot recover the message since she has no further knowledge about the structure of the error. Note the difference to the code-based McEliece cryptosystem, where the security relies on the fact that an attacker does not know the structure of the code. We will turn this observation into an exponential-time message attack in Sect. 6.2.2, which we will consider in our parameter choice.

Furthermore, the procedure implied by Theorem 9 might have a practical advantage compared to the original decryption algorithm. The code \mathcal{G}' used for decoding in Algorithm 5 depends on the private key. In Theorem 9, the code is given by \mathbf{g} , which is public and in fact does not need to be chosen randomly in the key generation.⁶ Depending on the used algorithm and type of implementation (e.g., in hardware), it can be advantageous in terms of complexity or implementation size if the code is fixed.

Probability of large enough ciphertext error weight

In this section, we analyze the probability that the error part $\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}$ of the ciphertext

$$\begin{aligned} \mathbf{c} &= (\mathbf{m}, \mathbf{0}_u) \cdot \mathbf{G}_G + \text{Tr}(\alpha\mathbf{k}_{\text{pub}}) + \mathbf{e} \\ &= \underbrace{[(\mathbf{m}, \mathbf{0}_u) + \text{Tr}(\alpha\mathbf{x})] \cdot \mathbf{G}_G}_{\text{codeword}} + \underbrace{\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}}_{\text{error}} \end{aligned}$$

has large enough rank to avoid the ciphertext attacks discussed in Sect. 6. The results of this appendix are summarized in Sect. 6.4.

Generically (i.e., with probability close to 1 for random choices of $\mathbf{k}_{\text{pub}}, \alpha, \mathbf{e}$), we have $\text{rk}_q(\text{Tr}(\alpha\mathbf{z})) = w$, $\text{rk}_q(\mathbf{e}) = t_{\text{pub}}$, and $\text{rk}_q(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) = w + t_{\text{pub}}$. However, there is a very small probability that the error has significantly smaller rank than the generic case. Our aim is to design the system parameters such that this probability is sufficiently small, e.g., $2^{-\lambda}$, to avoid attacks utilizing this behavior.

As we will see in this section, the choice of \mathbf{z} in the public key influences this probability (fixed \mathbf{z} , randomness in α and \mathbf{e}) significantly. Since \mathbf{z} is itself drawn using a random experiment during the key generation, we study with which probability this key is “strong”, i.e., whether the rank of $\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}$ is large with sufficiently high probability (randomness only in α and \mathbf{e}).

We start with a lemma that shows that the probability mass function of the \mathbb{F}_q -rank of $\text{Tr}(\alpha\mathbf{z})$ for uniformly drawn α only depends on the weight distribution of the code spanned by $\mathbf{a}_1, \dots, \mathbf{a}_\zeta$ (the \mathbb{F}_{q^m} -linearly independent vectors over \mathbb{F}_{q^m} from which \mathbf{z} is constructed).

Lemma 6 *Let \mathbf{z} be constructed from the randomly chosen code $\mathcal{A}[w, \zeta]_{q^m}$ as in Algorithm 3. Denote by A_0, \dots, A_w the rank-weight distribution of \mathcal{A} . For $\alpha \in \mathbb{F}_{q^{m\zeta}}$ chosen uniformly at random, we have*

$$\Pr \left(\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z})) = i \right) = \frac{A_i}{q^{m\zeta}}.$$

Proof We use the notation $(\mathbf{z}, \mathbf{s}, \mathcal{A}, \mathbf{P}$, and $\mathbf{S})$ from Algorithm 3. First observe that $\text{Tr}(\alpha\mathbf{z}) = \text{Tr}(\alpha[\mathbf{s}|\mathbf{0}])\mathbf{P}$. Hence,

$$\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z})) = \text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{s})).$$

⁶ Note that we described the key generation as in [16], where \mathbf{g} is chosen at random, but this is not necessary for the security of the system.

We can expand $\alpha \in \mathbb{F}_{q^{mu}}$ in the dual basis γ_i^* as $\alpha = \sum_{i=1}^u \alpha_i \gamma_i^*$. Then,

$$\text{Tr}(\alpha \mathbf{s}) = \sum_{i=1}^u \alpha_i \mathbf{s}_i = (\alpha_1 \dots \alpha_u) \cdot \begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_u \end{pmatrix} = (\alpha_1 \dots \alpha_u) \mathbf{S} \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_\zeta \end{pmatrix},$$

where $\mathbf{a}_1, \dots, \mathbf{a}_\zeta$ is a basis of \mathcal{A} and $\mathbf{S} \in \mathbb{F}_{q^m}^{u \times \zeta}$ is a matrix of full rank ζ . As α is chosen uniformly at random from $\mathbb{F}_{q^{mu}}$, the α_i are chosen independently and uniformly at random from \mathbb{F}_{q^m} . As $\text{rk}_{q^m} \mathbf{S} = \zeta$, this is equivalent to saying that

$$(\beta_1 \dots \beta_\zeta) := (\alpha_1 \dots \alpha_u) \mathbf{S}$$

is chosen uniformly at random from $\mathbb{F}_{q^m}^\zeta$. Hence, we have

$$\text{Tr}(\alpha \mathbf{s}) = (\beta_1 \dots \beta_\zeta) \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_\zeta \end{pmatrix},$$

i.e., $\text{Tr}(\alpha \mathbf{s})$ is a codeword of \mathcal{A} , chosen uniformly at random. This immediately implies the claim. □

A direct consequence of the lemma above is the following statement.

Corollary 1 *With notation as in Lemma 6, let d be the minimum rank distance of the code $\mathcal{A}[w, \zeta]_{q^m}$. Then,*

$$\Pr \left(\text{rk}_{\mathbb{F}_q} (\text{Tr}(\alpha \mathbf{z})) < d \right) = q^{-m\zeta}.$$

Corollary 1 shows that we can easily bound the probability that $\text{Tr}(\alpha \mathbf{z})$ has small \mathbb{F}_q -rank if the code \mathcal{A} (as defined in Lemma 6) has a large minimum rank distance. Loosely speaking, if the minimum rank distance of the code is small, we can consider this key to be *weak*, and *strong* otherwise. Since the code is chosen uniformly at random from the set of \mathbb{F}_{q^m} -linear $[w, \zeta]_{\mathbb{F}_{q^m}}$ (cf. choice of \mathbf{a}_i in Algorithm 3), we can use the following result from [12] to bound the probability that the key is weak.

Lemma 7 ([12, Corollary 5.4]) *Let $1 \leq k \leq n$ and $2 \leq d \leq n - k + 2$. Choose a code $\mathcal{C} \in \mathbb{F}_{q^m}^n$ uniformly at random from the \mathbb{F}_{q^m} -linear codes of parameters $[n, k]_{\mathbb{F}_{q^m}}$. Then,*

$$\begin{aligned} & \begin{bmatrix} n \\ k \end{bmatrix}_{q^m}^{-1} \sum_{h=1}^{d-1} \begin{bmatrix} d-1 \\ h \end{bmatrix}_{q^m} \sum_{s=h}^{d-1} \begin{bmatrix} d-1-h \\ s-h \end{bmatrix}_{q^m} \begin{bmatrix} n-s \\ n-k \end{bmatrix}_{q^m} (-1)^{s-h} q^{m \binom{s-h}{2}} \leq \\ & \Pr \left(d_{\text{rk}, \min}(\mathcal{C}) < d \right) \leq \frac{q^{mk} - 1}{(q^m - 1)(q^{mn} - 1)} \left(\sum_{i=0}^{d-1} \begin{bmatrix} n \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} (q^m - q^j) - 1 \right) \end{aligned}$$

Since the code in Lemma 7 is chosen uniformly at random, it does not exactly match the distribution of the code \mathcal{A} in Algorithm 3. Hence, we need the following lemma and theorem to estimate the probability of a small minimum distance in our case.

Lemma 8 *An \mathbb{F}_{q^m} -linear MRD code $[n, k]_{q^m}$ has a basis consisting of codewords of \mathbb{F}_q -rank n .*

Proof We show that the number of full-rank codewords is at least $q^{m(k-1)}$. Since these codewords are all non-zero, their \mathbb{F}_{q^m} -span must have cardinality at least q^{mk} and is hence the entire code.

The weight distribution of an MRD code of length n and minimum distance d can be given by (see [18]):

$$A_{d+s} = \begin{bmatrix} n \\ d+s \end{bmatrix}_q \sum_{j=0}^s (-1)^{j+s} \begin{bmatrix} d+s \\ d+j \end{bmatrix}_q q^{(s-j)(s-j-1)/2} (q^{m(j+1)} - 1), \quad s = 0, 1, \dots, n-d, \tag{16}$$

where m is the order of the extension field, $n \leq m$, and A_{d+s} denotes the number of rank- $(d+s)$ codewords.

We are interested in a lower bound for the number of full-rank codewords, i.e., $s = n - d$. The sum in (16) is an alternating sum whose terms get larger, the larger j and therefore can be lower bounded by the case of $j = s$ plus the case of $j = s - 1$. That means:

$$A_{d+s} \geq \begin{bmatrix} n \\ d+s \end{bmatrix}_q \left((q^{m(s+1)} - 1) - \begin{bmatrix} d+s \\ d+s-1 \end{bmatrix}_q (q^{ms} - 1) \right).$$

Hence, for $s = n - d$, we obtain:

$$\begin{aligned} A_n &\geq \begin{bmatrix} n \\ n \end{bmatrix}_q \left((q^{m(n-d+1)} - 1) - \begin{bmatrix} n \\ n-1 \end{bmatrix}_q (q^{m(n-d)} - 1) \right) \\ &= q^{mk} - 1 - \begin{bmatrix} n \\ n-1 \end{bmatrix}_q (q^{m(k-1)} - 1) \\ &= q^{mk} - 1 - \frac{(q^n - 1)q^{m(k-1)}}{q - 1} + \frac{q^n - 1}{q - 1} \\ &\geq q^{mk} - \frac{(q^n - 1)q^{m(k-1)}}{q - 1} \\ &\geq q^{m(k-1)} \underbrace{\left(q^m - \frac{q^n - 1}{q - 1} \right)}_{\geq 1 \text{ since } m \geq n \text{ and } q \geq 2} \\ &\geq q^{m(k-1)}. \end{aligned} \tag{17}$$

Theorem 10 *Let m, ζ , and w be chosen such that*

$$1 - \zeta q^\zeta w^{-m} \geq \frac{1}{2}. \tag{18}$$

Let \mathcal{A} be chosen as in Algorithm 3, i.e., uniformly at random from the set of linear $[w, \zeta]_{q^m}$ codes that have a basis consisting only of codewords with \mathbb{F}_q -rank w . Furthermore, let $2 \leq t \leq w - \zeta + 2$. Then,

$$\Pr \left(d_{\text{rk}, \min}(\mathcal{A}) < t \right) \leq 2 \frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} (q^m - q^j) - 1 \right).$$

Proof We define an alternative random experiment, where a code \mathcal{A}' is chosen uniformly from all linear $[w, \zeta]_{q^m}$. The sought probability is then given by the conditional probability

$$\Pr \left(d_{\text{rk}, \min}(\mathcal{A}') < t \mid \mathcal{S} \right),$$

where \mathcal{S} is the event that \mathcal{A}' has a basis of maximal-rank codewords. We derive the result using the relation

$$\Pr(d_{\text{rk},\min}(\mathcal{A}') < t) \geq \Pr(d_{\text{rk},\min}(\mathcal{A}') < t | \mathcal{S}) \Pr(\mathcal{S}). \tag{19}$$

First note that Lemma 7 gives us

$$\Pr(d_{\text{rk},\min}(\mathcal{A}') < t) \leq \frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} (q^m - q^j) - 1 \right).$$

By Lemma 8, we have

$$\Pr(\mathcal{S}) \geq \Pr(\mathcal{A}' \text{ is MRD}).$$

Using [32, Theorem 21], we can lower-bound this probability by

$$\Pr(\mathcal{A}' \text{ is MRD}) \geq 1 - \zeta q^{\zeta w - m} \geq \frac{1}{2},$$

where the last inequality follows from (18). The claim follows by combining the two bounds with (19). \square

The last building block for a general bound on the probability of $\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}$ having small rank is the following lemma, which gives a bound for this probability conditioned on the event that $\text{Tr}(\alpha\mathbf{z})$ has a given (large) rank.

Lemma 9 *Let $\mathbf{k}_{\text{pub}} = \mathbf{x} \cdot \mathbf{G}_{\mathcal{G}} + \mathbf{z}$ be fixed as in Algorithm 4 and let α be chosen such that $\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z})) = t$. For $\mathbf{e} \xleftarrow{\$} \{\mathbf{a} \in \mathbb{F}_{q^m}^n : \text{rk}_q(\mathbf{a}) = t_{\text{pub}}\}$, drawn uniformly at random, we have*

$$\begin{aligned} & \Pr(\text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) < w \mid \text{rk}_{\mathbb{F}_q}(\text{Tr}(\alpha\mathbf{z})) = t, \text{rk}_{\mathbb{F}_q}(\mathbf{e}) = t_{\text{pub}}) \\ & \leq 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)\left(n+\frac{t-3w-t_{\text{pub}}}{2}\right)} \end{aligned}$$

Proof For simplicity, we write (for some basis $\boldsymbol{\gamma}$ of \mathbb{F}_{q^m} over \mathbb{F}_q)

$$\begin{aligned} \mathbf{e}_1 &:= \text{Tr}(\alpha\mathbf{z}), & \mathbf{E}_1 &:= \text{ext}_{\boldsymbol{\gamma}}(\mathbf{e}_1), & \mathcal{E}_1^{\text{C}} &:= \text{ColumnSpace}(\mathbf{E}_1) \subseteq \mathbb{F}_q^n \\ & & & & \mathcal{E}_1^{\text{R}} &:= \text{RowSpace}(\mathbf{E}_1) \subseteq \mathbb{F}_q^n \\ \mathbf{e}_2 &:= \text{Tr}(\alpha\mathbf{z}), & \mathbf{E}_2 &:= \text{ext}_{\boldsymbol{\gamma}}(\mathbf{e}_2), & \mathcal{E}_2^{\text{C}} &:= \text{ColumnSpace}(\mathbf{E}_2) \subseteq \mathbb{F}_q^n \\ & & & & \mathcal{E}_2^{\text{R}} &:= \text{RowSpace}(\mathbf{E}_2) \subseteq \mathbb{F}_q^n. \end{aligned}$$

It is clear that $\text{rk}_{\mathbb{F}_q}(\mathbf{e}_1 + \mathbf{e}_2) = \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1 + \mathbf{E}_2)$ and, since $\text{rk}_{\mathbb{F}_q}(\mathbf{e}_1) = \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) = t$ and $\text{rk}_{\mathbb{F}_q}(\mathbf{e}_2) = \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) = t_{\text{pub}}$,

$$\begin{aligned} \dim_{\mathbb{F}_q}(\mathcal{E}_1^{\text{C}}) &= \dim_{\mathbb{F}_q}(\mathcal{E}_1^{\text{R}}) = t \\ \dim_{\mathbb{F}_q}(\mathcal{E}_2^{\text{C}}) &= \dim_{\mathbb{F}_q}(\mathcal{E}_2^{\text{R}}) = t_{\text{pub}}. \end{aligned}$$

Note that in our probabilistic model, \mathcal{E}_1^{C} and \mathcal{E}_1^{R} are fixed and it follows easily that \mathcal{E}_2^{C} and \mathcal{E}_2^{R} are random variables that are uniformly distributed on the set of t_{pub} -dimensional subspaces of \mathbb{F}_q^m and \mathbb{F}_q^n , respectively, and stochastically independent. Due to [30, Theorem 1], for

$$\dim(\mathcal{E}_1^{\text{C}} \cap \mathcal{E}_2^{\text{C}}) = i \quad \text{and} \quad \dim(\mathcal{E}_1^{\text{R}} \cap \mathcal{E}_2^{\text{R}}) = j,$$

we have

$$\text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) + \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) - i - j \leq \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1 + \mathbf{E}_2) \leq \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) + \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) - \max\{i, j\}.$$

Since $\text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) + \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) = t + t_{\text{pub}}$, this implies

$$\begin{aligned} & \Pr\left(\text{rk}_{\mathbb{F}_q}(\mathbf{E}_1 + \mathbf{E}_2) < w \mid \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) = t, \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) = t_{\text{pub}}\right) \\ & \leq \Pr\left(\dim(\mathcal{E}_1^{\text{C}} \cap \mathcal{E}_2^{\text{C}}) + \dim(\mathcal{E}_1^{\text{R}} \cap \mathcal{E}_2^{\text{R}}) > t + t_{\text{pub}} - w\right) \\ & = \sum_{\substack{i, j=0 \\ i+j > t+t_{\text{pub}}-w}}^{\min\{t, t_{\text{pub}}\}} \Pr\left(\dim(\mathcal{E}_1^{\text{C}} \cap \mathcal{E}_2^{\text{C}}) = i\right) \Pr\left(\dim(\mathcal{E}_1^{\text{R}} \cap \mathcal{E}_2^{\text{R}}) = j\right). \end{aligned}$$

Due to [15, Proof of Lemma 7], we have

$$\begin{aligned} \Pr(\dim(\mathcal{E}_1^{\text{C}} \cap \mathcal{E}_2^{\text{C}}) = i) &= \frac{\begin{bmatrix} m-w \\ t_{\text{pub}}-i \end{bmatrix}_q \begin{bmatrix} w \\ i \end{bmatrix}_q q^{(w-i)(t_{\text{pub}}-i)}}{\begin{bmatrix} m \\ t_{\text{pub}} \end{bmatrix}_q} \\ &\leq 16 \frac{q^{(t_{\text{pub}}-i)(m-w-t_{\text{pub}}+1)+i(w-i)+(w-i)(t_{\text{pub}}-i)}}{q^{t_{\text{pub}}(m-t_{\text{pub}})}} \\ &= 16q^{-i(m-w-t_{\text{pub}}+i)}. \end{aligned}$$

Likewise, we have

$$\Pr(\dim(\mathcal{E}_1^{\text{R}} \cap \mathcal{E}_2^{\text{R}}) = i) \leq 16q^{-i(n-w-t_{\text{pub}}+i)}.$$

Due to $n \leq n$, we obtain

$$\begin{aligned} & \Pr\left(\text{rk}_{\mathbb{F}_q}(\mathbf{E}_1 + \mathbf{E}_2) < w \mid \text{rk}_{\mathbb{F}_q}(\mathbf{E}_1) = t, \text{rk}_{\mathbb{F}_q}(\mathbf{E}_2) = t_{\text{pub}}\right) \\ & \leq 256 \sum_{\substack{i, j=0 \\ i+j > t+t_{\text{pub}}-w}}^{\min\{t, t_{\text{pub}}\}} q^{-i(n-w-t_{\text{pub}}+i)} q^{-j(m-w-t_{\text{pub}}+j)} \\ & \leq 256 \sum_{\substack{i, j=0 \\ i+j > t+t_{\text{pub}}-w}}^{\min\{t, t_{\text{pub}}\}} q^{-i(n-w-t_{\text{pub}}+i)} q^{-j(n-w-t_{\text{pub}}+j)} \\ & \leq 256 \sum_{\substack{i, j=0 \\ i+j > t+t_{\text{pub}}-w}}^{\min\{t, t_{\text{pub}}\}} q^{-(i+j)(n-w-t_{\text{pub}})-(i^2+j^2)} \\ & \leq 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)(n-w-t_{\text{pub}})-\frac{(t+t_{\text{pub}}-w+1)^2}{2}} \\ & \leq 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)\left(n+\frac{t-3w-t_{\text{pub}}}{2}\right)}. \end{aligned}$$

This proves the claim. □

Summarized, we have the following. The proof follows directly by combining Corollary 1, Lemmas 7 and 9, and a union-bound argument.

Theorem 11 Let m , ζ , and w be chosen such that $1 - \zeta q^{\zeta w - m} \geq \frac{1}{2}$. Choose \mathbf{z} of the public key as in Algorithm 3. Let $2 \leq t \leq w - \zeta + 2$. With probability at least

$$P_{\text{strong, key}}(t) \geq 1 - 2 \frac{q^{m\zeta} - 1}{(q^m - 1)(q^{mw} - 1)} \left(\sum_{i=0}^{t-1} \begin{bmatrix} w \\ i \end{bmatrix}_q \prod_{j=0}^{i-1} (q^m - q^j) - 1 \right)$$

the public key has the following property:

Choose $\alpha \in \mathbb{F}_q^{m\mu}$ and For $\mathbf{e} \xleftarrow{\$} \{\mathbf{a} \in \mathbb{F}_q^m : \text{rk}_q(\mathbf{a}) = t_{\text{pub}}\}$, both uniformly at random. Then the probability that $\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}$ has \mathbb{F}_q -rank at least w is lower-bounded by

$$\begin{aligned} & \Pr \left(\text{rk}_{\mathbb{F}_q} (\text{Tr}(\alpha\mathbf{z}) + \mathbf{e}) \geq w \right) \\ & \geq 1 - q^{-m\zeta} - 256 \min\{t, t_{\text{pub}}\}^2 q^{-(t+t_{\text{pub}}-w+1)} \binom{t-3w-t_{\text{pub}}}{n+\frac{t-3w-t_{\text{pub}}}{2}}. \end{aligned}$$

Remark 2 By the asymptotical analysis in [12], we have

$$P_{\text{strong, key}}(t) \geq 1 - \Theta \left(q^{m[t-(w-\zeta+2)]} \right).$$

Since the hidden constant strongly depends on q , this asymptotic value should only be used for a rough estimation of the strong-key probability and the exact formula in Theorem 11 should be used for parameter design.

Nevertheless, the formula shows that $1 - P_{\text{strong, key}}(t)$ decreases exponentially in m times the difference of t and $w - \zeta + 2$. Hence, usually we can choose t close to the maximal value $w - \zeta + 2$ to achieve a given designed probability for a key to be strong.

For instance, we can choose $t \approx (w - \zeta + 2) - \frac{\lambda}{m} \log_q(2)$ for

$$P_{\text{strong, key}}(t) \geq 1 - 2^{-\lambda},$$

where λ is the security parameter.

References

1. Aguilar Melchor, C., Aragon, N., Bardet, M., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J., Gaborit, P., Hauteville, A., Otmani, A., Ruatta, O., Tillich, J., Zemor, G.: ROLLO - Rank-Ouroboros, LAKE & LOCKER. Second round submission to the NIST post-quantum cryptography call (2019). <https://pqc-rollo.org>
2. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J., Gaborit, P., Zemor, G., Couvreur, A., Hauteville: Rank quasi cyclic (RQC). Second round submission to the NIST post-quantum cryptography call (2019). <https://pqc-rqc.org>
3. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J., Gaborit, P., Gueron, S., Güneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J., Vasseur, V., Zemor, G.: BIKE - bit flipping key encapsulation. Second round submission to the NIST post-quantum cryptography call (2019). <https://pqc-rollo.org>
4. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.P.: A new algorithm for solving the rank syndrome decoding problem. In: IEEE Int. Symp. Inf. Theory (ISIT) (2018)
5. Augot, D., Finiasz, M.: A public key encryption scheme based on the polynomial reconstruction problem. LNCS: Revised selected papers of EUROCRYPT 2003 **2656**, 229–249 (2003)
6. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Neiger, V., Ruatta, O., Tillich, J.P.: An algebraic attack on rank metric code-based cryptosystems. Tech. rep. (2019). [arXiv:1910.00810v1](https://arxiv.org/abs/1910.00810v1)
7. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Algebraic attacks for solving the rank decoding and minrank problems without Gröbner basis (2020)

8. Bernstein, D., Chou, T., Lange, T., Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Wang, W.: Classic McEliece. Second round submission to the NIST post-quantum cryptography call (2019). <https://classic.mceliece.org>
9. Bernstein D.J.: Grover vs. mceliece. In: Sendrier N. (ed.) Post-Quantum Cryptography, pp. 73–80. Springer, Berlin Heidelberg (2010).
10. Bettaieb, S., Bidoux, L., Gaborit, P., Marcotel, E.: Preventing timing attacks against RQC using constant time decoding of Gabidulin codes. In: Int. Conf. on Post-Quantum Cryptography (PQCrypto) (2019)
11. Boyer M., Brassard G., Høyer P., Tapp A.: Tight bounds on quantum searching. *Fortschritte der Physik* **46**(4–5), 493–505 (1998)
12. Byrne E., Ravagnani A.: Partition-balanced families of codes and asymptotic enumeration in coding theory. *J. Comb. Theory A* **171**, 105169 (2020).
13. Caruso, X., Le Borgne, J.: Fast multiplication for skew polynomials. In: ISSAC (2017)
14. Delsarte P.: Bilinear forms over a finite field with applications to coding theory. *J. Comb. Theory Ser. A* **25**(3), 226–241 (1978).
15. Etzion T., Vardy A.: Error-correcting codes in projective space. *IEEE Trans. Inform. Theory* **57**(2), 1165–1173 (2011).
16. Faure C., Loidreau P.: A new public-key cryptosystem based on the problem of reconstructing p-polynomials. *Coding and Cryptography*, pp. 304–315. Springer, Berlin (2006).
17. Fujisaki E., Okamoto T.: Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptol* **26**, 80–101 (2013).
18. Gabidulin E.M.: Theory of codes with maximum rank distance. *Probl. Inf. Transm.* **21**(1), 3–16 (1985).
19. Gabidulin E.M., Ourivski A.V., Honary B., Ammar B.: Reducible rank codes and their applications to cryptography. *IEEE Trans. Inform. Theory* **49**(12), 3289–3293 (2003).
20. Gabidulin E.M., Pilipchuk N.I.: Error and erasure correcting algorithms for rank codes. *Des. Codes Cryptogr.* **49**(1–3), 105–122 (2008).
21. Gaborit P., Otmani A., Talé Kalachi H.: Polynomial-time key recovery attack on the Faure-Loidreau scheme based on gabidulin codes. *Des. Codes Cryptogr.* **86**(7), 1391–1403 (2018).
22. Gadouleau, M., Yan, Z.: Complexity of decoding Gabidulin codes. In: IEEE Annual Conf. Inform. Science and Syst, pp. 1081–1085 (2008)
23. Hofheinz D., Hövelmanns K., Kiltz E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai Y., Reyzin L. (eds.) *Theory of Cryptography*, pp. 341–371. Springer International Publishing, Cham (2017).
24. Koetter R., Kschischang F.R.: Coding for errors and erasures in random network coding. *IEEE Trans. Inform. Theory* **54**(8), 3579–3591 (2008).
25. Krachkovsky V.Y., Lee Y.X.: Decoding for iterative Reed-Solomon coding schemes. *IEEE Trans. Magn.* **33**(5), 2740–2742 (1997).
26. Lidl R., Niederreiter H.: *Finite Fields. Encyclopedia of Mathematics and its Applications* Cambridge University Press, Cambridge (1996).
27. Loidreau, P., Overbeck, R.: Decoding rank errors beyond the error correcting capability. In: Int. Workshop Alg. Combin. Coding Theory (ACCT) (2006)
28. Loidreau, P.: Métrique rang et cryptographie (in French). Mémoire d’habilitation à diriger des recherches, Université Pierre et Marie Curie, Paris 6 (2007)
29. Loidreau, P.: A new rank metric codes based encryption scheme. In: Int. Conf. on Post-Quantum Cryptography (PQCrypto) (2017)
30. Marsaglia, G.: Bounds on the rank of the sum of matrices (1967)
31. National Institute of Standards and Technology (NIST), U.S. Department of Commerce: Post-quantum cryptography standardization (2017), <https://csrc.nist.gov/Projects/post-quantum-cryptography/Post-Quantum-Cryptography-Standardization>
32. Neri A., Horlemann-Trautmann A.L., Randrianarisoa T., Rosenthal J.: On the genericity of maximum rank distance and gabidulin codes. *Des. Codes Cryptogr.* **86**(2), 341–363 (2018).
33. Nojima R., Imai H., Kobara K., Morozov K.: Semantic security for the McEliece cryptosystem without random oracles. *Des. Codes Cryptogr.* **49**, 289–305 (2008).
34. Overbeck, R.: *Public Key Cryptography based on Coding Theory*. Ph.D. thesis, TU Darmstadt, Darmstadt, Germany (2007)
35. Overbeck R.: A new structural attack for GPT and variants. *LNCS MYCRYPT* **3715**, 50–63 (2005).
36. Puchinger, S., Wachter-Zeh, A.: Sub-quadratic decoding of Gabidulin codes. In: IEEE Int. Symp. Inf. Theory (ISIT), pp. 2554–2558 (2016)
37. Puchinger S., Wachter-Zeh A.: Fast operations on linearized polynomials and their applications in coding theory. *J. Symb. Comp* **89**, 194–215 (2018).

38. Raviv N., Wachter-Zeh A.: Some Gabidulin codes cannot be list decoded efficiently at any radius. *IEEE Trans. Inform. Theory* **62**(4), 1605–1615 (2016).
39. Renner, J., Jerkovits, T., Bartz, H., Puchinger, S., Loidreau, P., Wachter-Zeh, A.: Randomized decoding of Gabidulin codes beyond the unique decoding radius. In: *Int. Conf. on Post-Quantum Cryptography (PQCrypto)* (2020)
40. Richter, G., Plass, S.: Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm. In: *International ITG Conference on Systems, Communications and Coding 2004 (SCC)* (2004)
41. Rosenkilde, J.S.H.: Personal Communication (2018)
42. Shehhi H.A., Bellini E., Borba F., Caullery F., Manzano M., Mateu V.: An ind-cca-secure code-based encryption scheme using rank metric. In: Buchmann J., Nitaj A., Rachidi T. (eds.) *Progress in Cryptology: AFRICACRYPT 2019*, pp. 79–96. Springer International Publishing, Cham (2019).
43. Sidorenko V.R., Jiang L., Bossert M.: Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes. *IEEE Trans. Inform. Theory* **57**(2), 621–632 (2011).
44. Silva, D., Kschischang, F.R.: Fast encoding and decoding of gabidulin codes. In: *IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2858–2862 (2009)
45. Silva D., Kschischang F.R., Kötter R.: A rank-metric approach to error control in random network coding. *IEEE Trans. Inform. Theory* **54**(9), 3951–3967 (2008).
46. Trombetti, R., Zullo, F.: On the list decodability of rank metric codes. preprint (2019), <https://arxiv.org/abs/1907.01289>
47. W. A. Stein and *et al.*: SageMath Software (<http://www.sagemath.org>)
48. Wachter-Zeh A.: Bounds on list decoding of rank-metric codes. *IEEE Trans. Inform. Theory* **59**(11), 7268–7277 (2013).
49. Wachter-Zeh, A.: Decoding of Block and Convolutional Codes in Rank Metric. Ph.D. thesis, Ulm University and University of Rennes 1, Ulm, Germany and Rennes, France (2013)
50. Wachter-Zeh, A., Puchinger, S., Renner, J.: Repairing the Faure-Loidreau public-key cryptosystem. In: *IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2426–2430 (2018)
51. Wachter-Zeh A., Zeh A.: List and unique error-erasure decoding of interleaved gabidulin codes with interpolation techniques. *Des. Codes Cryptogr.* **73**(2), 547–570 (2014).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.