

Fast Gao-like Decoding of Horizontally Interleaved Linearized Reed–Solomon Codes^{*}

Felicitas Hörmann^{1,2}  and Hannes Bartz¹ 

¹ Institute of Communications and Navigation, German Aerospace Center (DLR), Oberpfaffenhofen–Wessling, Germany

{felicitas.hoermann, hannes.bartz}@dlr.de

² School of Computer Science, University of St. Gallen, St. Gallen, Switzerland

Abstract. Both horizontal interleaving as well as the sum-rank metric are currently attractive topics in the field of code-based cryptography, as they could mitigate the problem of large key sizes. In contrast to vertical interleaving, where codewords are stacked vertically, each codeword of a horizontally s -interleaved code is the horizontal concatenation of s codewords of s component codes. In the case of horizontally interleaved linearized Reed–Solomon (HILRS) codes, these component codes are chosen to be linearized Reed–Solomon (LRS) codes.

We provide a Gao-like decoder for HILRS codes that is inspired by the respective works for non-interleaved Reed–Solomon and Gabidulin codes. By applying techniques from the theory of minimal approximant bases, we achieve a complexity of $\tilde{O}(s^{2.373}n^{1.635})$ operations in \mathbb{F}_{q^m} , where $\tilde{O}(\cdot)$ neglects logarithmic factors, s is the interleaving order and n denotes the length of the component codes. For reasonably small interleaving order $s \ll n$, this is subquadratic in the component-code length n and improves over the only known syndrome-based decoder for HILRS codes with quadratic complexity. Moreover, it closes the performance gap to vertically interleaved LRS codes for which a decoder of complexity $\tilde{O}(s^{2.373}n^{1.635})$ is already known.

We can decode beyond the unique-decoding radius and handle errors of sum-rank weight up to $\frac{s}{s+1}(n-k)$ for component-code dimension k . We also give an upper bound on the failure probability in the zero-derivation setting and validate its tightness via Monte Carlo simulations.

Keywords: Gao-like Decoding · Horizontal Interleaving · Linearized Reed–Solomon Codes · Sum-Rank Metric · Code-Based Cryptography · Minimal Approximant Bases

1 Introduction

The American National Institute of Standards and Technology (NIST) started a competition for post-quantum cryptography (PQC) in 2016. After three rounds,

^{*} F. Hörmann and H. Bartz acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of “Souverän. Digital. Vernetzt.” Joint project 6G-RIC, project identification number: 16KISK022.

the lattice-based key-encapsulation mechanism (KEM) CRYSTALS-Kyber [8] was standardized in July 2022 [3]. Moreover, NIST announced a fourth round to which four KEM candidates advanced: BIKE [4], Classic McEliece [14], HQC [1], and SIKE [9]. SIKE is the only candidate based on hard problems in the area of isogenies and was broken by [17] shortly after NIST’s round-4 announcement. The remaining three candidates in this round rely on coding-theoretical problems in the Hamming metric.

In his seminal paper [32] in 1978, McEliece proposed the first code-based cryptosystem, which still serves as a blueprint for most of the recent proposals. The McEliece framework essentially resisted the cryptanalytic effort of 45 years. However, it suffers from large key sizes and is thus not usable in many practical applications.

Rank and Sum-Rank Metric As the syndrome-decoding problem in the rank metric is harder than its Hamming-metric counterpart [7,10], many McEliece-like schemes based on rank-metric codes as e.g. [18,19,28,27] were considered. Unfortunately, most of them were broken by structural attacks. A new approach is to consider the sum-rank metric which covers both the Hamming and the rank metric as special cases. Even though the gain in terms of key size might not be as large as for the rank metric, it is reasonable to hope that rank-metric attacks cannot be adapted to the sum-rank-metric case [21] and the corresponding systems will remain secure.

Interleaved Codes Another way to reduce the key size is to use codes with higher error-correction capability. An increased error weight will result in higher complexities for generic attacks like [37] and thus require smaller parameter sizes to achieve the same level of security. One well-known code construction to improve the (burst) error-correction capability is interleaving, where each codeword of the s -interleaved code consists of s vertically or horizontally stacked codewords of s component codes, respectively.

Metzner and Kapturowski [33] showed that vertically interleaved Hamming-metric codes can be efficiently decoded with negligible failure probability as soon as their interleaving order s is high compared to the error weight t . This result was generalized to the rank metric [40,38] and recently also to the sum-rank metric [24]. As no knowledge about the code structure is needed for Metzner–Kapturowski-like decoders, this is a direct generic attack on any code-based cryptosystems based on vertically interleaved codes with high interleaving order. Thus, horizontal interleaving appears to be better suited for cryptographic purposes. This is also reflected in recent proposals as for example in the KEM LowMS [6] that is based on horizontally interleaved Gabidulin codes, in the signature scheme Durandal [5] based on the closely related rank-support-learning (RSL) problem [10], and in the cryptosystem [2] that makes use of horizontally interleaved low-rank parity-check (LRPC) codes [39].

The cryptanalysis of the underlying hard problems ensures reliable security-level estimates. However, also performance improvements for decoding horizontally interleaved codes have a significant impact as they directly speed up de-

ryption and verification within the corresponding cryptosystems and digital signatures.

HILRS Codes Horizontally interleaved linearized Reed–Solomon (HILRS) codes combine the usage of an alternative decoding metric for higher generic-decoding complexity and the interleaving construction for higher error-correction capability. Both approaches promise to reduce the key size in a McEliece-like setup. The component codes of an HILRS code are linearized Reed–Solomon (LRS) codes which were introduced by Martínez-Peñas in 2018 [29]. Up to now, LRS codes are one of the most studied code families in the sum-rank metric. They are evaluation codes with respect to skew polynomials and form the natural generalization of Reed–Solomon (RS) codes in the Hamming metric and Gabidulin codes in the rank metric.

As the performance of code-based cryptosystems strongly depends on the decoding speed for the underlying codes, fast decoders for HILRS codes are crucial. Currently, the only known decoder for HILRS codes is syndrome-based and has a quadratic complexity in the length sn of the interleaved code (ongoing work [23] extending [22]). It can handle a combination of errors, row erasures, and column erasures.

In contrast, vertically interleaved linearized Reed–Solomon (VILRS) codes, which are constructed by vertically stacking s LRS codewords, allow for decoding with lower complexity $\tilde{O}(s^\omega \mathcal{M}(n)) \subseteq \tilde{O}(s^{2.373} n^{1.635})$ [12,13]. Here, ω and $\mathcal{M}(n)$ denote the matrix-multiplication coefficient and the cost of multiplying two skew polynomials of degree at most n , respectively, and $\tilde{O}(\cdot)$ neglects logarithmic factors.

Contributions This paper presents a Gao-like decoder for HILRS codes. It is based on the original Gao decoder for Reed–Solomon codes in the Hamming metric [20] as well as on its known extensions to Gabidulin codes [46,45] and their horizontally interleaved version [36] in the rank metric. We consider probabilistic unique decoding beyond the unique-decoding radius and derive an upper bound on the decoding-failure probability in the zero-derivation case. We achieve a decoding radius of $\frac{s}{s+1}(n - k)$ for the interleaving order s and for n and k denoting the length and the dimension of the component codes, respectively.

We further show how a major speedup can be obtained by using the theory of minimal approximant bases [11]. The fast variant of the Gao-like decoder achieves subquadratic complexity in the length n of the component codes for a fixed interleaving order s . Particularly, we obtain $\tilde{O}(s^\omega \mathcal{M}(n)) \subseteq \tilde{O}(s^{2.373} n^{1.635})$ and thus close the performance gap with respect to the decoding of VILRS codes.

Our conceptually new approach to solving the Gao-like key equation results in the fastest known decoder for HILRS codes in the sum-rank metric. Moreover, the special case obtained for the rank metric yields the fastest decoder for horizontally interleaved Gabidulin codes in the rank metric, improving on [41,42,36].

Outline We start the paper in Section 2 by giving basic preliminaries on skew polynomials, on HILRS codes in the sum-rank metric, and on the channel model

we consider. Then, we present a Gao-like decoder for HILRS codes in Section 3 and analyze its decoding radius, complexity, and failure probability. Section 4 deals with a speedup for the shown decoder that is based on the theory of minimal approximant bases. Finally, we summarize the main results of the paper in Section 5 and give an outlook on future work.

2 Preliminaries

We denote the finite field of order q by \mathbb{F}_q and refer to its degree- m extension field by \mathbb{F}_{q^m} . We often consider vectors $\mathbf{x} \in \mathbb{F}_{q^m}^n$ that are divided into blocks. More precisely, we define a *length partition* of $n \in \mathbb{N}^*$ as the vector $\mathbf{n} = (n_1, \dots, n_\ell) \in \mathbb{N}^\ell$ with $\sum_{i=1}^\ell n_i = n$ and $n_i > 0$ for all $i = 1, \dots, \ell$. We write $\mathbf{x} = (\mathbf{x}^{(1)} \mid \dots \mid \mathbf{x}^{(\ell)})$, where the blocks $\mathbf{x}^{(i)}$ belong to $\mathbb{F}_{q^m}^{n_i}$ for all $i = 1, \dots, \ell$. Similarly, we write $\mathbf{X} = (\mathbf{X}^{(1)} \mid \dots \mid \mathbf{X}^{(\ell)})$ for a subdivided matrix $\mathbf{X} \in \mathbb{F}_{q^m}^{k \times n}$ with $\mathbf{X}^{(i)} \in \mathbb{F}_{q^m}^{k \times n_i}$ for all $i = 1, \dots, \ell$. The \mathbb{F}_{q^m} -linear row space of \mathbf{X} is denoted by $\langle \mathbf{X} \rangle_{q^m}$.

Further choose an \mathbb{F}_{q^m} -automorphism θ with fixed field \mathbb{F}_q . Note that θ is \mathbb{F}_q -linear and satisfies both $\theta(a+b) = \theta(a) + \theta(b)$ and $\theta(a \cdot b) = \theta(a) \cdot \theta(b)$ for arbitrary $a, b \in \mathbb{F}_{q^m}$. Moreover, we consider a map $\delta : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_{q^m}$ for which the equalities $\delta(a+b) = \delta(a) + \delta(b)$ and $\delta(ab) = \delta(a)b + \theta(a)\delta(b)$ hold for all $a, b \in \mathbb{F}_{q^m}$. In the finite-field setting, all such θ -derivations δ are inner derivations [29], i.e., they have the form $\delta = \gamma(\text{Id} - \theta)$ for a parameter $\gamma \in \mathbb{F}_{q^m}$ and the identity Id .

The automorphism θ and the derivation δ give rise to a partition of \mathbb{F}_{q^m} with respect to (θ, δ) -conjugacy [25]. Namely, two elements $a, b \in \mathbb{F}_{q^m}$ are conjugate if there is a nonzero $c \in \mathbb{F}_{q^m}^*$ with

$$a^c := \theta(c)ac^{-1} + \delta(c)c^{-1}.$$

The conjugacy class of an element $a \in \mathbb{F}_{q^m}$ is denoted by $\mathcal{C}(a) := \{a^c : c \in \mathbb{F}_{q^m}^*\}$ and $\mathcal{C}(0)$ is called the trivial conjugacy class. There are $q-1$ distinct nontrivial (θ, δ) -conjugacy classes. In the zero-derivation case, each of the first $q-1$ powers of any primitive element of \mathbb{F}_{q^m} belongs to another nontrivial class.

2.1 Skew-Polynomial Rings

Skew polynomials were first studied by Ore in 1933 [34,35] and are used e.g. for the construction of LRS codes [29]. The skew-polynomial ring $\mathbb{F}_{q^m}[x; \theta, \delta]$ contains all formal polynomials $\sum_i f_i x^{i-1}$ with finitely many nonzero coefficients $f_i \in \mathbb{F}_{q^m}$. The notion of the degree $\deg(f) := \max\{i-1 : f_i \neq 0\}$ of a skew polynomial $f(x) = \sum_i f_i x^{i-1}$ carries over from $\mathbb{F}_{q^m}[x]$. The set of skew polynomials forms a non-commutative ring with respect to conventional polynomial addition and a multiplication that is determined by the non-commutative rule $xa = \theta(a)x + \delta(a)$ for any $a \in \mathbb{F}_{q^m}$. By $\mathbb{F}_{q^m}[x; \theta, \delta]_{<k}$ we denote the subset of $\mathbb{F}_{q^m}[x; \theta, \delta]$ containing all skew polynomials of degree less than k . For simplicity, we refer to the skew-polynomial ring with zero derivation by $\mathbb{F}_{q^m}[x; \theta] := \mathbb{F}_{q^m}[x; \theta, 0]$.

$\mathbb{F}_{q^m}[x; \theta, \delta]$ is Euclidean which ensures the existence of skew polynomials $q, r \in \mathbb{F}_{q^m}[x; \theta, \delta]$ with $f(x) = q(x)g(x) + r(x)$ and $\deg(r) < \deg(g)$ for each pair $f, g \in \mathbb{F}_{q^m}[x; \theta, \delta]$ with $\deg(f) \geq \deg(g)$. We denote the remainder r of this right-hand division by $f \bmod_r g$.

The literature provides two meaningful ways to evaluate skew polynomials, namely, the remainder evaluation [25] and the generalized operator evaluation [29]. The former corresponds to the idea of enforcing a remainder theorem similar to the one in conventional polynomial rings and will not be of interest for this paper. The latter is e.g. used for the construction of LRS codes that we heavily rely on. For defining the generalized operator evaluation of skew polynomials we first introduce the operator $\mathcal{D}_a(b) := \theta(b)a + \delta(b)$ and its i -th power $\mathcal{D}_a^i(b) := \mathcal{D}_a(\mathcal{D}_a^{i-1}(b))$ for $i \in \mathbb{N}^*$ and any $a, b \in \mathbb{F}_{q^m}$. The operator simplifies to $\mathcal{D}_a(b) = \theta(b)a$ for all $a, b \in \mathbb{F}_{q^m}$ in the case of zero derivation. In this case, its i -th power $\mathcal{D}_a^i(b)$ for $i \in \mathbb{N}^*$ can be written as $\mathcal{D}_a^i(b) = \theta^i(b) \cdot \mathcal{N}_i(a)$, where $\mathcal{N}_i(a) := \prod_{k=0}^{i-1} \theta^k(a)$ is the i -th truncated norm of a .

The *generalized operator evaluation* of a skew polynomial $f(x) = \sum_{i=1}^d f_i x^{i-1} \in \mathbb{F}_{q^m}[x; \theta, \delta]$ at a point $b \in \mathbb{F}_{q^m}$ and with respect to an evaluation parameter $a \in \mathbb{F}_{q^m}$ is defined as

$$f(b)_a := \sum_{i=1}^d f_i \mathcal{D}_a^{i-1}(b).$$

We use the notation $f(\mathbf{b})_a := (f(b_1)_a, \dots, f(b_n)_a)$ to denote the vector containing the evaluations of f at every entry of $\mathbf{b} \in \mathbb{F}_{q^m}^n$. Moreover, if $\mathbf{b} = (\mathbf{b}^{(1)} \mid \dots \mid \mathbf{b}^{(\ell)}) \in \mathbb{F}_{q^m}^n$ is subdivided according to a length partition \mathbf{n} and $\mathbf{a} = (a_1, \dots, a_\ell) \in \mathbb{F}_{q^m}^\ell$, we use the shorthand $f(\mathbf{b})_{\mathbf{a}} := (f(\mathbf{b}^{(1)})_{a_1}, \dots, f(\mathbf{b}^{(\ell)})_{a_\ell})$ to evaluate f at the elements of the i -th block $\mathbf{b}^{(i)}$ with respect to the evaluation parameter a_i for every $i = 1, \dots, \ell$.

The evaluation of a product of two skew polynomials $f, g \in \mathbb{F}_{q^m}[x; \theta, \delta]$ satisfies the product rule $(f \cdot g)(b)_a = f(g(b)_a)_a$ for all $a, b \in \mathbb{F}_{q^m}$ [25].

For a vector $\mathbf{x} = (\mathbf{x}^{(1)} \mid \dots \mid \mathbf{x}^{(\ell)}) \in \mathbb{F}_{q^m}^n$, a vector $\mathbf{a} \in \mathbb{F}_{q^m}^\ell$, and a parameter $d \in \mathbb{N}^*$ the *generalized Moore matrix* $\mathfrak{M}_d(\mathbf{x})_{\mathbf{a}}$ is defined as

$$\mathfrak{M}_d(\mathbf{x})_{\mathbf{a}} := \left(\mathfrak{m}_d(\mathbf{x}^{(1)})_{a_1} \mid \dots \mid \mathfrak{m}_d(\mathbf{x}^{(\ell)})_{a_\ell} \right) \in \mathbb{F}_{q^m}^{d \times n},$$

$$\text{with } \mathfrak{m}_d(\mathbf{x}^{(i)})_{a_i} := \begin{pmatrix} x_1^{(i)} & \dots & x_{n_i}^{(i)} \\ \mathcal{D}_{a_i}(x_1^{(i)}) & \dots & \mathcal{D}_{a_i}(x_{n_i}^{(i)}) \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{a_i}^{d-1}(x_1^{(i)}) & \dots & \mathcal{D}_{a_i}^{d-1}(x_{n_i}^{(i)}) \end{pmatrix} \quad \text{for all } i = 1, \dots, \ell.$$

If \mathbf{a} contains representatives of pairwise distinct nontrivial conjugacy classes of \mathbb{F}_{q^m} and $\text{rk}_q(\mathbf{x}^{(i)}) = n_i$ for all $i = 1, \dots, \ell$, it holds $\text{rk}_{q^m}(\mathfrak{M}_d(\mathbf{x})_{\mathbf{a}}) = \min(d, n)$ [25,29].

Consider $\mathbf{b} = (\mathbf{b}^{(1)} \mid \dots \mid \mathbf{b}^{(\ell)}) \in \mathbb{F}_{q^m}^n$ and $\mathbf{a} = (a_1, \dots, a_\ell) \in \mathbb{F}_{q^m}^\ell$. The minimal skew polynomial that vanishes on the entries of $\mathbf{b}^{(i)}$ with respect to

the evaluation parameter a_i for each $i = 1, \dots, \ell$ is denoted by $\text{mpol}_{(\mathbf{b})_a}(x)$ and characterized by

$$\text{mpol}_{(\mathbf{b})_a}(\mathbf{b}^{(i)})_{a_i} = \mathbf{0} \quad \text{for all } i = 1, \dots, \ell.$$

According to [15], it can be computed as a least common left multiple (lclm) via

$$\text{mpol}_{(\mathbf{b})_a}(x) = \text{lclm} \left\{ x - \frac{\mathcal{D}_{a_i}(b_\iota^{(i)})}{b_\iota^{(i)}} : b_\iota^{(i)} \neq 0, \begin{array}{l} \iota = 1, \dots, n_i, \\ i = 1, \dots, \ell \end{array} \right\}. \quad (1)$$

The degree satisfies $\deg(\text{mpol}_{(\mathbf{b})_a}) \leq n$ with equality if and only if the entries of $\mathbf{b}^{(i)}$ are \mathbb{F}_q -linearly independent for all $i = 1, \dots, \ell$ and the evaluation parameters a_1, \dots, a_ℓ belong to distinct nontrivial conjugacy classes of \mathbb{F}_{q^m} .

Now consider an additional vector $\mathbf{c} = (\mathbf{c}^{(1)} \mid \dots \mid \mathbf{c}^{(\ell)}) \in \mathbb{F}_{q^m}^n$. Then there exists a unique skew interpolation polynomial $\text{intpol}_{(\mathbf{b})_a}^{\mathbf{c}}(x) \in \mathbb{F}_{q^m}[x; \theta, \delta]$ with $\deg(\text{intpol}_{(\mathbf{b})_a}^{\mathbf{c}}) < n$ and

$$\text{intpol}_{(\mathbf{b})_a}^{\mathbf{c}}(\mathbf{b}^{(i)})_{a_i} = \mathbf{c}^{(i)} \quad \text{for all } i = 1, \dots, \ell \text{ [16].}$$

For the complexity analysis of the Gao-like decoder, we will use $\mathcal{O}(\cdot)$ to state asymptotic costs in terms of the usual big-O notation. Moreover, the notation $\tilde{\mathcal{O}}(\cdot)$ indicates that logarithmic factors in the input parameter are neglected. The complexity of skew-polynomial operations in the zero-derivation setting was summarized in [11, Section II.D.]. Particularly, left and right division of skew polynomials with degree at most n as well as the computation of a minimal or an interpolation polynomial of degree at most n can be achieved in $\tilde{\mathcal{O}}(\mathcal{M}_{q,m}(n))$ operations in \mathbb{F}_{q^m} . Here, $\mathcal{M}_{q,m}(n)$ denotes the cost of multiplying two skew polynomials of degree n from $\mathbb{F}_{q^m}[x; \theta]$ and it holds $\mathcal{O}(\mathcal{M}_{q,m}(n)) \subseteq \mathcal{O}(n^{\min(\frac{\omega+1}{2}, 1.635)}) \subseteq \mathcal{O}(n^{1.635})$. The exponent $\omega \geq 2$ denotes the matrix-multiplication coefficient for which the currently best known upper bound is $\omega < 2.3728639$ [26].

2.2 The Sum-Rank Metric and the Corresponding Interleaved Channel Model

The *sum-rank weight* of a vector $\mathbf{x} = (\mathbf{x}^{(1)} \mid \dots \mid \mathbf{x}^{(\ell)}) \in \mathbb{F}_{q^m}^n$ with respect to the length partition \mathbf{n} is

$$\text{wt}_{\Sigma R, \mathbf{n}}(\mathbf{x}) = \sum_{i=1}^{\ell} \text{rk}_q(\mathbf{x}^{(i)})$$

where $\text{rk}_q(\mathbf{x}^{(i)})$ is the maximum number of \mathbb{F}_q -linearly independent entries of the block $\mathbf{x}^{(i)}$ for each $i = 1, \dots, \ell$. The *sum-rank metric* is induced by the sum-rank weight via $d_{\Sigma R, \mathbf{n}}(\mathbf{x}, \mathbf{y}) = \text{wt}_{\Sigma R, \mathbf{n}}(\mathbf{x} - \mathbf{y})$ for all vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_{q^m}^n$.

Note that we omit the index \mathbf{n} and simply write $\text{wt}_{\Sigma R}$ and $d_{\Sigma R}$ when the length partition is clear from the context.

The sum-rank metric coincides with the Hamming metric for $\ell = n$, i.e., when every block has length one, and with the rank metric for $\ell = 1$, i.e., when the vector is considered as a single block.

Let now $\mathbf{x} = (\mathbf{x}_1 \mid \cdots \mid \mathbf{x}_s) \in \mathbb{F}_{q^m}^{sn}$ with $\mathbf{x}_j \in \mathbb{F}_{q^m}^n$ for all $j = 1, \dots, s$ be a horizontally s -interleaved vector for an interleaving order $s \in \mathbb{N}^*$. Let us further assume for simplicity that all component vectors $\mathbf{x}_j = (\mathbf{x}_j^{(1)} \mid \cdots \mid \mathbf{x}_j^{(\ell)}) \in \mathbb{F}_{q^m}^n$ for $j = 1, \dots, s$ are equipped with the same length partition \mathbf{n} . The natural way to define the sum-rank weight of $\mathbf{x} \in \mathbb{F}_{q^m}^{sn}$ is with respect to the *block-ordered* length partition $\tilde{\mathbf{n}} = (sn_1, \dots, sn_\ell)$, i.e., as

$$\text{wt}_{\Sigma R, \tilde{\mathbf{n}}}(\mathbf{x}) := \sum_{i=1}^{\ell} \text{rk}_q(\mathbf{x}^{(i)}) \quad \text{for } \mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)} \mid \cdots \mid \mathbf{x}_s^{(i)}).$$

As for the conventional sum-rank metric, we often omit the length partition in the index and simply write $\text{wt}_{\Sigma R}(\mathbf{x})$ when $\tilde{\mathbf{n}}$ is clear from the context. Figure 1 illustrates how the sum-rank weight of horizontally interleaved vectors is computed by grouping the same-indexed blocks of the component vectors. It shows how the block-ordered length partition arises naturally in this setting.

$$\mathbf{x} = \left(\underbrace{\mathbf{x}_1^{(1)} \mid \mathbf{x}_1^{(2)} \mid \cdots \mid \mathbf{x}_1^{(\ell)}}_{\mathbf{x}_1 \in \mathbb{F}_{q^m}^n} \mid \underbrace{\mathbf{x}_2^{(1)} \mid \mathbf{x}_2^{(2)} \mid \cdots \mid \mathbf{x}_2^{(\ell)}}_{\mathbf{x}_2 \in \mathbb{F}_{q^m}^n} \mid \cdots \mid \underbrace{\mathbf{x}_s^{(1)} \mid \mathbf{x}_s^{(2)} \mid \cdots \mid \mathbf{x}_s^{(\ell)}}_{\mathbf{x}_s \in \mathbb{F}_{q^m}^n} \right) \in \mathbb{F}_{q^m}^{sn}$$

$$\text{wt}_{\Sigma R}(\mathbf{x}) = \text{rk}_q \left(\underbrace{\mathbf{x}_1^{(1)} \mid \mathbf{x}_2^{(1)} \mid \cdots \mid \mathbf{x}_s^{(1)}} \right) + \text{rk}_q \left(\underbrace{\mathbf{x}_1^{(2)} \mid \mathbf{x}_2^{(2)} \mid \cdots \mid \mathbf{x}_s^{(2)}} \right) + \cdots + \text{rk}_q \left(\underbrace{\mathbf{x}_1^{(\ell)} \mid \mathbf{x}_2^{(\ell)} \mid \cdots \mid \mathbf{x}_s^{(\ell)}} \right)$$

Fig. 1. Illustration of the sum-rank weight for a horizontally s -interleaved vector $\mathbf{x} = (\mathbf{x}_1 \mid \cdots \mid \mathbf{x}_s) \in \mathbb{F}_{q^m}^{sn}$.

We now consider the transmission of an interleaved vector $\mathbf{x} \in \mathbb{F}_{q^m}^{sn}$ over a sum-rank error channel with output

$$\mathbf{y} = \mathbf{x} + \mathbf{e} \quad (2)$$

where the error vector \mathbf{e} is understood as a horizontally s -interleaved vector $\mathbf{e} = (\mathbf{e}_1 \mid \cdots \mid \mathbf{e}_s) \in \mathbb{F}_{q^m}^{sn}$ of sum-rank weight $\text{wt}_{\Sigma R, \tilde{\mathbf{n}}}(\mathbf{e}) = t$. We further assume a uniform channel distribution, that is, that the error \mathbf{e} is drawn uniformly at random from the set

$$\{\mathbf{x} = (\mathbf{x}_1 \mid \cdots \mid \mathbf{x}_s) \in \mathbb{F}_{q^m}^{sn} : \text{wt}_{\Sigma R, \tilde{\mathbf{n}}}(\mathbf{x}) = t\}. \quad (3)$$

The described channel is illustrated in Figure 2.

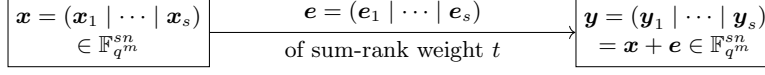


Fig. 2. The additive sum-rank channel for horizontally interleaved vectors.

Let $\mathbf{t} = (t_1, \dots, t_\ell) \in \mathbb{N}^\ell$ with $t_i = \text{rk}_q(\mathbf{e}^{(i)}) := \text{rk}_q(\mathbf{e}_1^{(i)} \mid \cdots \mid \mathbf{e}_s^{(i)})$ for all $i = 1, \dots, \ell$ denote the rank partition of \mathbf{e} . Then, we obtain for each $i = 1, \dots, \ell$ a decomposition of the form $(\mathbf{e}_1^{(i)} \mid \cdots \mid \mathbf{e}_s^{(i)}) = \mathbf{a}^{(i)} \cdot (\mathbf{B}_1^{(i)} \mid \cdots \mid \mathbf{B}_s^{(i)})$, where $\mathbf{a}^{(i)} \in \mathbb{F}_q^{t_i m}$ with $\text{rk}_q(\mathbf{a}^{(i)}) = t_i$ and $\mathbf{B}_j^{(i)} \in \mathbb{F}_q^{t_i \times n_j}$ with $\text{rk}_q(\mathbf{B}_1^{(i)} \mid \cdots \mid \mathbf{B}_s^{(i)}) = t_i$ for all $j = 1, \dots, s$. After reordering the components, the error vector \mathbf{e} can thus be decomposed as

$$\mathbf{e} = \mathbf{a} \cdot \mathbf{B} \quad (4)$$

with $\mathbf{a} = (\mathbf{a}^{(1)} \mid \cdots \mid \mathbf{a}^{(\ell)}) \in \mathbb{F}_q^{tm}$ and

$$\mathbf{B} = \left(\begin{array}{c|c|c} \mathbf{B}_1^{(1)} & & \mathbf{B}_s^{(1)} \\ & \ddots & \\ & & \mathbf{B}_1^{(\ell)} \end{array} \right) \in \mathbb{F}_q^{t \times sn}, \quad (5)$$

where for any $i = 1, \dots, \ell$ and any $j = 1, \dots, s$

$$\begin{aligned} \mathbf{a}^{(i)} &\in \mathbb{F}_q^{t_i m} \text{ with } \text{rk}_q(\mathbf{a}^{(i)}) = t_i \\ \text{and } \mathbf{B}_j^{(i)} &\in \mathbb{F}_q^{t_i \times n_j} \text{ with } \text{rk}_q(\mathbf{B}_1^{(i)} \mid \cdots \mid \mathbf{B}_s^{(i)}) = t_i. \end{aligned}$$

Note that the decomposition in (4) is not unique. Moreover, the uniform distribution of \mathbf{e} among all vectors of sum-rank weight t implies that, for fixed rank partition \mathbf{t} , both \mathbf{a} and \mathbf{B} are also chosen uniformly at random from the sets

$$\begin{aligned} &\{\mathbf{x} \in \mathbb{F}_q^{tm} : \text{wt}_{\Sigma R, \mathbf{t}}(\mathbf{x}) = t\} \\ \text{and } &\{\mathbf{X} \in \mathbb{F}_q^{tm \times sn} \text{ of the form (5)} : \text{wt}_{\Sigma R, \tilde{\mathbf{n}}}(\mathbf{X}) = t\}, \end{aligned} \quad (6)$$

respectively.

The elements in $\mathbf{a}^{(i)}$ form a basis of the column space of $\mathbf{e}^{(i)}$ and are called *error values*. Similarly, the rows of $\mathbf{B}_j^{(i)}$ form a basis of the row space of $\mathbf{e}_j^{(i)}$ and are referred to as *error locations*. For horizontal interleaving, the error values in \mathbf{a} are common for all component errors.

2.3 Horizontally Interleaved Linearized Reed–Solomon (HILRS) Codes

We first introduce LRS codes [29, Definition 31], which are one of the most prominent families of sum-rank-metric codes.

Definition 1 (Linearized Reed–Solomon Codes). Let $\boldsymbol{\xi} = (\xi_1, \dots, \xi_\ell) \in \mathbb{F}_{q^m}^\ell$ contain elements of distinct nontrivial conjugacy classes of \mathbb{F}_{q^m} . Further denote by $\mathbf{n} = (n_1, \dots, n_\ell) \in \mathbb{N}^\ell$ a length partition of n , i.e., $n = \sum_{i=1}^\ell n_i$. Let the vectors $\boldsymbol{\beta}^{(i)} = (\beta_1^{(i)}, \dots, \beta_{n_i}^{(i)}) \in \mathbb{F}_{q^m}^{n_i}$ contain \mathbb{F}_q -linearly independent \mathbb{F}_{q^m} -elements for all $i = 1, \dots, \ell$ and write $\boldsymbol{\beta} = (\boldsymbol{\beta}^{(1)} \mid \dots \mid \boldsymbol{\beta}^{(\ell)}) \in \mathbb{F}_{q^m}^n$. A linearized Reed–Solomon (LRS) code of length n and dimension k is defined as

$$\text{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}; \mathbf{n}, k] = \left\{ \left(f(\boldsymbol{\beta}^{(1)})_{\xi_1} \mid \dots \mid f(\boldsymbol{\beta}^{(\ell)})_{\xi_\ell} \right) : f \in \mathbb{F}_{q^m}[x; \theta, \delta]_{<k} \right\} \subseteq \mathbb{F}_{q^m}^n.$$

Every codeword $\mathbf{c} \in \text{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}; \mathbf{n}, k]$ corresponds to a skew polynomial $f \in \mathbb{F}_{q^m}[x; \theta, \delta]_{<k}$. We sometimes write $\mathbf{c} = \mathbf{c}(f)$ to emphasize this and call f the *message polynomial* of \mathbf{c} .

The minimum distance d of an LRS code satisfies the Singleton-like bound $d \leq n - k + 1$ with equality. Thus, LRS codes are maximum sum-rank distance (MSRD) codes.

Similar to RS and Gabidulin codes, LRS codes have a generator matrix \mathbf{G} of a particularly useful form. Namely, the matrix $\mathbf{G} = (\mathbf{G}^{(1)} \mid \dots \mid \mathbf{G}^{(\ell)}) = \mathfrak{M}_k(\boldsymbol{\beta})_{\boldsymbol{\xi}} \in \mathbb{F}_{q^m}^{k \times n}$ with

$$\mathbf{G}^{(i)} = \mathfrak{m}_k(\boldsymbol{\beta}^{(i)})_{\xi_i} = \begin{pmatrix} \beta_1^{(i)} & \dots & \beta_{n_i}^{(i)} \\ \mathcal{D}_{\xi_i}(\beta_1^{(i)}) & \dots & \mathcal{D}_{\xi_i}(\beta_{n_i}^{(i)}) \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{\xi_i}^{k-1}(\beta_1^{(i)}) & \dots & \mathcal{D}_{\xi_i}^{k-1}(\beta_{n_i}^{(i)}) \end{pmatrix} \in \mathbb{F}_{q^m}^{k \times n_i}$$

for all $i = 1, \dots, \ell$ generates the code $\text{LRS}[\boldsymbol{\beta}, \boldsymbol{\xi}; \mathbf{n}, k]$.

We obtain an HILRS code with interleaving order $s \in \mathbb{N}^*$ by combining s LRS component codes. Namely, each codeword of the HILRS code is the horizontal concatenation of s codewords of the chosen component codes.

Definition 2 (Horizontally Interleaved LRS Codes). Fix an interleaving order $s \in \mathbb{N}^*$ and pick for each $j = 1, \dots, \ell$ an LRS code $\text{LRS}[\boldsymbol{\beta}_j, \boldsymbol{\xi}; \mathbf{n}, k]$ according to Definition 1. We define the horizontally interleaved linearized Reed–Solomon (HILRS) code with interleaving order s , code locators $\boldsymbol{\beta} := (\boldsymbol{\beta}_1 \mid \dots \mid \boldsymbol{\beta}_s)$, evaluation parameters $\boldsymbol{\xi}$, and length partition $s\mathbf{n} := (sn_1, \dots, sn_\ell)$ as

$$\text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; s\mathbf{n}, sk] = \left\{ (\mathbf{c}_1 \mid \dots \mid \mathbf{c}_s) : \begin{array}{l} \mathbf{c}_j \in \text{LRS}[\boldsymbol{\beta}_j, \boldsymbol{\xi}; \mathbf{n}, k] \\ \text{for all } j = 1, \dots, s \end{array} \right\} \subseteq \mathbb{F}_{q^m}^{sn}.$$

The code $\text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; s\mathbf{n}, sk]$ has length sn and dimension sk over \mathbb{F}_{q^m} . Its minimum distance d equals the minimum distance of its component codes, i.e., $d = n - k + 1$. HILRS codes are hence *not* MSRD. Similar to LRS codes, we write $\mathbf{c}(\mathbf{f}) = (\mathbf{c}_1(f_1) \mid \dots \mid \mathbf{c}_s(f_s)) \in \text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; s\mathbf{n}, sk]$ with $\mathbf{f} = (f_1, \dots, f_s)$ and $f_j \in \mathbb{F}_{q^m}[x; \theta, \delta]_{<k}$ for each $j = 1, \dots, s$ to emphasize the relation to the message polynomials of the component codewords $\mathbf{c}_1, \dots, \mathbf{c}_s$. We call \mathbf{f} the *message-polynomial vector* corresponding to \mathbf{c} .

Remark 1. It is straightforward to generalize Definition 2 and all concepts of this paper to component codes with different length partitions, lengths, and dimensions. However, we assume that the component codes only have different code locators β_j for $j = 1, \dots, s$ for simplicity of notation. \square

3 A Gao-like Decoder for HILRS Codes

We now derive a Gao-like decoder in the spirit of [20,45,36] for HILRS codes and the interleaved sum-rank-channel model described in (2). Let $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_{q^m}^{sn}$ denote the received vector after the codeword $\mathbf{c} = \mathbf{c}(\mathbf{f}) \in \text{HILRS}[\beta, \xi, s; sn, sk]$ was corrupted by the error $\mathbf{e} \in \mathbb{F}_{q^m}^{sn}$ of sum-rank weight $\text{wt}_{\Sigma R}(\mathbf{e}) = t$ during transmission. Recall that we assume a uniform error distribution, that is, that \mathbf{e} is chosen uniformly at random from the set of all vectors of sum-rank weight t as given in (3).

The main ingredient of the decoder is the Gao-like key equation that exploits the relation between certain polynomials to recover the error values as zeros of the error-span polynomial. Then, the message-polynomial vector \mathbf{f} that corresponds to \mathbf{c} can be retrieved.

The *error span polynomial (ESP)* $\sigma \in \mathbb{F}_{q^m}[x; \theta, \delta]$ makes use of the error decomposition shown in (4). It is the skew polynomial that vanishes at all error values, i.e.,

$$\sigma(\mathbf{a}^{(i)})_{\xi_i} = \mathbf{0} \quad \text{for all } i = 1, \dots, \ell.$$

For horizontal interleaving, the component errors \mathbf{e}_j share the same error values \mathbf{a} for all $j = 1, \dots, s$ according to (4). This implies that the ESP is common for all component errors.

Next let $G_j \in \mathbb{F}_{q^m}[x; \theta, \delta]$ for each $j = 1, \dots, s$ be the minimal skew polynomial for the code locators β_j with respect to generalized operator evaluation. Namely,

$$G_j(x) := \text{mpol}_{(\beta_j)_\xi}(x) \quad \text{for all } j = 1, \dots, s.$$

Remark that these polynomials only depend on code parameters and can thus be precomputed. Further, define $R_j \in \mathbb{F}_{q^m}[x; \theta, \delta]$ for each $j = 1, \dots, s$ as the interpolation polynomial whose evaluation at the code locators β_j yields the channel observation \mathbf{y}_j . That means that $R_j(x) := \text{intpol}_{(\beta_j)_\xi}^{\mathbf{y}_j}(x)$ satisfies

$$R_j(\beta_j)_\xi = \mathbf{y}_j \quad \text{for all } j = 1, \dots, s.$$

Note that the polynomials R_j can be computed directly from the channel observation $\mathbf{y} = (\mathbf{y}_1 \mid \dots \mid \mathbf{y}_s)$.

Theorem 1 (Gao-like Key Equation for HILRS Codes). *Let $\mathbf{c} = \mathbf{c}(\mathbf{f}) \in \text{HILRS}[\beta, \xi, s; sn, sk]$ be a codeword corresponding to the message-polynomial vector $\mathbf{f} = (f_1, \dots, f_s)$ with $f_j \in \mathbb{F}_{q^m}[x; \theta, \delta]_{<k}$ for all $j = 1, \dots, s$. Let further*

$\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_{q^m}^{sn}$ denote a channel observation according to (2). For the ESP $\sigma \in \mathbb{F}_{q^m}[x; \theta, \delta]$ and the polynomials

$$G_j(x) = \text{mpol}_{(\beta_j)_\xi}(x) \quad \text{and} \quad R_j(x) = \text{intpol}_{(\beta_j)_\xi}^{\mathbf{y}_j}(x) \quad \text{for each } j = 1, \dots, s,$$

it holds

$$\sigma \cdot R_j \equiv \sigma \cdot f_j \pmod{G_j} \quad \text{for all } j = 1, \dots, s. \quad (7)$$

Proof. Consider a fixed $j = 1, \dots, s$ and let us show the equivalent formulation

$$\sigma \cdot (R_j - f_j) \equiv 0 \pmod{G_j}$$

of the key equation. By definition, we know that the evaluation of $R_j - f_j$ at β_j is $(R_j - f_j)(\beta_j)_\xi = \mathbf{y}_j - \mathbf{c}_j = \mathbf{e}_j$. Thus,

$$(\sigma \cdot (R_j - f_j))(\beta_j)_\xi \stackrel{(\Delta)}{=} \sigma((R_j - f_j)(\beta_j)_\xi)_\xi = \sigma(\mathbf{e}_j)_\xi = \mathbf{0}$$

applies, where (Δ) follows from the product rule for generalized operator evaluation and the other equalities hold by definition. Together with the fact that G_j is the minimal polynomial of the code locators, we conclude that G_j divides $\sigma \cdot (R_j - f_j)$ on the right. Since this argument is true for every $j = 1, \dots, s$, the statement follows. \square

As can be seen from the proof of Theorem 1, the Gao-like key equation (7) is in fact equivalent to

$$(\sigma \cdot (R_j - f_j))(\beta_j)_\xi = \mathbf{0} \quad \text{for all } j = 1, \dots, s.$$

By rewriting it in terms of a system of \mathbb{F}_{q^m} -linear equations, we obtain

$$\underbrace{\begin{pmatrix} (\mathfrak{M}_{t+k}(\beta_1)_\xi)^\top & & -(\mathfrak{M}_{t+1}(\mathbf{y}_1)_\xi)^\top \\ & \ddots & \vdots \\ & & (\mathfrak{M}_{t+k}(\beta_s)_\xi)^\top - (\mathfrak{M}_{t+1}(\mathbf{y}_s)_\xi)^\top \end{pmatrix}}_{=: \mathbf{M}^\top} \cdot \begin{pmatrix} \sigma \mathbf{f}_1 \\ \vdots \\ \sigma \mathbf{f}_s \\ \sigma \end{pmatrix} = \mathbf{0}. \quad (8)$$

Here, the vectors σ and $\sigma \mathbf{f}_j$ for $j = 1, \dots, s$ contain the coefficients of the respective polynomials, i.e.,

$$\begin{aligned} (\sigma \mathbf{f}_j)^\top &:= ((\sigma \cdot f_j)_1, \dots, (\sigma \cdot f_j)_{t+k}) \in \mathbb{F}_{q^m}^{t+k} \quad \text{for all } j = 1, \dots, s \\ \text{and} \quad \sigma^\top &:= (\sigma_1, \dots, \sigma_{t+1}) \in \mathbb{F}_{q^m}^{t+1}. \end{aligned}$$

Equation (8) displays a homogeneous system of sn equations in $s(t+k) + t + 1 = (s+1)t + sk + 1$ unknowns. It can be solved by Gaussian elimination with a complexity of $\mathcal{O}(\max(sn, (s+1)t + sk + 1)^\omega)$ operations in \mathbb{F}_{q^m} [44, Proposition 2.15].

As soon as the Gao-like key equation is solved, we have access to a candidate $\tilde{\sigma}$ for the ESP $\sigma \in \mathbb{F}_{q^m}[x; \theta, \delta]$ as well as to candidates p_j for the products $\sigma \cdot f_j \in \mathbb{F}_{q^m}[x; \theta, \delta]_{<t+k}$ for all $j = 1, \dots, s$. Thus, for any $j = 1, \dots, s$, left division of p_j by $\tilde{\sigma}$ recovers a candidate \tilde{f}_j for the j -th message polynomial f_j . If the remainder r_j of the left division of p_j by $\tilde{\sigma}$ is nonzero for any $j = 1, \dots, s$ or if any of the $\tilde{f}_1, \dots, \tilde{f}_s$ has degree at least k , we declare a decoding failure. Otherwise, the decoding was correct and $\tilde{f}_j = f_j$ applies for all $j = 1, \dots, s$. Algorithm 1 summarizes all steps of the Gao-like decoder.

Algorithm 1: Gao-like Decoder for HILRS Codes

Input : received vector $\mathbf{y} \in \mathbb{F}_{q^m}^{sn}$ with $\mathbf{y} = \mathbf{c}(\mathbf{f}) + \mathbf{e}$ according to (2) and with $\mathbf{c}(\mathbf{f}) \in \text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; sn, sk]$
precomputed G_1, \dots, G_s with $G_j := \text{mpol}_{(\boldsymbol{\beta}_j)\boldsymbol{\xi}}(x)$ for all $j = 1, \dots, s$

Output : $\mathbf{f} = (f_1, \dots, f_s)$ or "decoding failure"

- 1 $R_j := \text{intpol}_{(\boldsymbol{\beta}_j)\boldsymbol{\xi}}^{\mathbf{y}_j}(x) \in \mathbb{F}_{q^m}[x; \theta, \delta]$ for all $j = 1, \dots, s$
/* use $\sigma \cdot R_j \equiv \sigma \cdot f_j \pmod{G_j}$ to find $p_j \triangleq \sigma \cdot f_j$ and $\tilde{\sigma} \triangleq \sigma$ */
- 2 $(p_1, \dots, p_s, \tilde{\sigma}) := \text{solveKeyEquation}(R_1, \dots, R_s, G_1, \dots, G_s, n, k, s)$
- 3 **forall** $j = 1, \dots, s$ **do**
- 4 $(\tilde{f}_j, r_j) := \text{leftDivide}(p_j, \tilde{\sigma})$
- 5 **if** $r_j \neq 0$ **or** $\deg(\tilde{f}_j) \geq k$ **then**
- 6 **return** "decoding failure"
- 7 **return** $\mathbf{f} := (\tilde{f}_1, \dots, \tilde{f}_s)$

Let us now further investigate the structure of \mathbf{M}^\top , which gives rise to the decoding-failure probability \Pr_{fail} . Remark that the system (8) has a nontrivial solution by definition, which implies $\text{rk}_{q^m}(\mathbf{M}) \leq (s+1)t + sk$. Moreover, a decoding failure can only occur if the solution space of (8) has dimension greater than one. In other words, $\text{rk}_{q^m}(\mathbf{M}^\top) = \text{rk}_{q^m}(\mathbf{M}) < (s+1)t + sk$ must apply and we obtain the inequality

$$\Pr_{\text{fail}} \leq \Pr(\text{rk}_{q^m}(\mathbf{M}) < (s+1)t + sk).$$

The following lemma gives a characterization of when the solution space of (8) is one-dimensional. Recall that this case implies correct decoding.

Lemma 1. *Consider a vector $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_{q^m}^{sn}$ that was received after transmitting $\mathbf{c} \in \text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; sn, sk]$ over the channel (2). Assume that the error has weight $\text{wt}_{\Sigma R}(\mathbf{e}) = t \leq n - k$ and can be decomposed into $\mathbf{e} = \mathbf{a} \cdot \mathbf{B}$ according to (4). Further, define \mathbf{M} as in (8) and let $\mathbf{H} = \text{diag}(\mathbf{H}_1, \dots, \mathbf{H}_s) \in \mathbb{F}_{q^m}^{s(n-k-t) \times sn}$ be a parity-check matrix of the code $\text{HILRS}[\boldsymbol{\beta}, \boldsymbol{\xi}, s; sn, s(k+t)]$. Then,*

$$\text{rk}_{q^m}(\mathbf{M}) = (s+1)t + sk \quad \text{if and only if} \quad \text{rk}_{q^m}(\mathbf{B}\mathbf{H}^\top) = t.$$

Proof. First note that the upper part of \mathbf{M} is a generator matrix of the code HILRS $[\boldsymbol{\beta}, \boldsymbol{\xi}, s; s\mathbf{n}, s(k+t)]$. In other words, the j -th block on its diagonal generates LRS $[\boldsymbol{\beta}_j, \boldsymbol{\xi}; \mathbf{n}, k+t]$ for all $j = 1, \dots, s$. For any $j = 1, \dots, s$, the additivity of the generalized operator evaluation yields $\mathfrak{M}_{t+1}(\mathbf{y}_j)_\xi = \mathfrak{M}_{t+1}(\mathbf{c}_j)_\xi + \mathfrak{M}_{t+1}(\mathbf{e}_j)_\xi$. Further, $\mathbf{c}_j \in \text{LRS}[\boldsymbol{\beta}_j, \boldsymbol{\xi}; \mathbf{n}, k] = \langle \mathfrak{M}_k(\boldsymbol{\beta}_j)_\xi \rangle_{q^m}$ implies $\mathcal{D}_\xi^t(\mathbf{c}_j) \in \langle \mathfrak{M}_{k+t}(\boldsymbol{\beta}_j)_\xi \rangle_{q^m}$ for all $\iota = 1, \dots, t$. We can hence consider the matrix

$$\widetilde{\mathbf{M}} = \begin{pmatrix} \mathfrak{M}_{t+k}(\boldsymbol{\beta}_1)_\xi & & \\ & \ddots & \\ & & \mathfrak{M}_{t+k}(\boldsymbol{\beta}_s)_\xi \\ \hline \mathfrak{M}_{t+1}(\mathbf{e}_1)_\xi & \dots & \mathfrak{M}_{t+1}(\mathbf{e}_s)_\xi \end{pmatrix} =: \begin{pmatrix} \mathbf{U} \\ \mathbf{L} \end{pmatrix}$$

which has the same \mathbb{F}_{q^m} -linear row space, and thus the same \mathbb{F}_{q^m} -rank, as \mathbf{M} . In the following, we denote the upper $s(t+k)$ rows of \mathbf{M} by \mathbf{U} and the lower part by \mathbf{L} for convenience. The error decomposition and the \mathbb{F}_q -linearity of the generalized operator evaluation let us write $\mathbf{L} = \mathfrak{M}_{t+1}(\mathbf{a})_\xi \cdot \mathbf{B}$. Therefore,

$$\widetilde{\mathbf{M}} = \begin{pmatrix} \mathbf{I}_{s(t+k)} & \mathbf{0} \\ \mathbf{0} & \mathfrak{M}_{t+1}(\mathbf{a})_\xi \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U} \\ \mathbf{B} \end{pmatrix}$$

applies, where $\mathbf{I}_{s(t+k)}$ denotes the identity matrix of size $s(t+k) \times s(t+k)$. Since the left matrix has full column rank over \mathbb{F}_{q^m} , [31, Theorem 2] yields

$$\text{rk}_{q^m}(\widetilde{\mathbf{M}}) = \text{rk}_{q^m} \begin{pmatrix} \mathbf{U} \\ \mathbf{B} \end{pmatrix}.$$

Define $\mathbf{H} := \text{diag}(\mathbf{H}_1, \dots, \mathbf{H}_s) \in \mathbb{F}_{q^m}^{s(n-k-t) \times sn}$ with \mathbf{H}_j being a parity-check matrix of the code LRS $[\boldsymbol{\beta}_j, \boldsymbol{\xi}; \mathbf{n}, k+t]$ for all $j = 1, \dots, s$. Then, \mathbf{H} is a parity-check matrix of HILRS $[\boldsymbol{\beta}, \boldsymbol{\xi}, s; s\mathbf{n}, s(k+t)]$ and satisfies $\mathbf{U}\mathbf{H}^\top = \mathbf{0}$. Since

$$\begin{aligned} \text{rk}_{q^m}(\mathbf{M}) &= \text{rk}_{q^m}(\mathbf{U}) + \text{rk}_{q^m}(\mathbf{B}) - \dim_{q^m}(\langle \mathbf{U} \rangle_{q^m} \cap \langle \mathbf{B} \rangle_{q^m}) \\ &\leq (s+1)t + sk - \dim_{q^m}(\langle \mathbf{U} \rangle_{q^m} \cap \langle \mathbf{B} \rangle_{q^m}) \end{aligned}$$

holds, the equality $\text{rk}_{q^m}(\mathbf{M}) = (s+1)t + sk$ is equivalent to $\langle \mathbf{U} \rangle_{q^m} \cap \langle \mathbf{B} \rangle_{q^m} = \{\mathbf{0}\}$ and thus to $\langle \mathbf{H} \rangle_{q^m}^\perp \cap \langle \mathbf{B} \rangle_{q^m} = \{\mathbf{0}\}$. This is equivalent to $\text{rk}_{q^m}(\mathbf{B}\mathbf{H}^\top) = t$, which proves the lemma. \square

This equivalent reformulation gives a condition on the error weight t and thus determines the decoding radius. In fact, the matrix $\mathbf{B}\mathbf{H}^\top$ has t rows and $s(n-k-t)$ columns and can achieve $\text{rk}_{q^m}(\mathbf{B}\mathbf{H}^\top) = t$ only if $t \leq s(n-k-t)$ applies. Since we obtain a decoding failure in all other cases, we obtain the necessary condition

$$t \leq t_{\max} := \frac{s}{s+1}(n-k)$$

for successful decoding.

We now focus on the zero-derivation case and derive an upper bound on the probability that $\text{rk}_{q^m}(\mathbf{B}\mathbf{H}^\top) < t$ which will also bound the decoding-failure

probability according to Lemma 1. Recall that we can choose \mathbf{H} such that $\mathbf{H}_1, \dots, \mathbf{H}_s$ are generalized Moore matrices, as the dual of an LRS code is again an LRS code in the zero-derivation setting [30, Theorem 4]. For such a choice of \mathbf{H} , the product $\mathbf{B}\mathbf{H}^\top = (\mathbf{B}_1\mathbf{H}_1^\top \mid \dots \mid \mathbf{B}_s\mathbf{H}_s^\top)$ is the transpose of vertically stacked generalized Moore matrices because $\mathbf{B} = (\mathbf{B}_1 \mid \dots \mid \mathbf{B}_s)$ contains only \mathbb{F}_q -elements and $\mathcal{D}_\xi(\cdot)$ is \mathbb{F}_q -linear for a fixed $\xi \in \mathbb{F}_{q^m}$. Namely,

$$\mathbf{H}\mathbf{B}^\top = \begin{pmatrix} \mathfrak{M}_{t+k}(\mathbf{h}_1\mathbf{B}_1^\top)\xi \\ \dots \\ \mathfrak{M}_{t+k}(\mathbf{h}_s\mathbf{B}_s^\top)\xi \end{pmatrix},$$

where \mathbf{h}_j denotes the first row of \mathbf{H}_j for each $j = 1, \dots, s$.

Further recall that, for a fixed rank partition \mathbf{t} , the matrix \mathbf{B} is uniformly distributed among the set of all matrices of a particular form having fixed sum-rank weight as described in (6). As $\text{wt}_{\Sigma R}(\mathbf{h}_j) = n$ applies for every $j = 1, \dots, s$, the $(s \times t)$ -matrix containing the vectors $\mathbf{h}_j\mathbf{B}_j^\top$ as rows is chosen uniformly at random from all matrices in $\mathbb{F}_{q^m}^{s \times t}$ with sum-rank weight t . This allows us to apply parts of the proof of [13, Lemma 7].

In the zero-derivation setting, we thus obtain the upper bound

$$\Pr_{\text{fail}} \leq \Pr(\text{rk}_{q^m}(\mathbf{B}\mathbf{H}^\top) < t) \leq \kappa_q^{\ell+1} q^{-m((s+1)(t_{\max}-t)+1)} \quad (9)$$

on the decoding-failure probability \Pr_{fail} , where $t_{\max} := \frac{s}{s+1}(n-k)$ and $\kappa_q < 3.5$ is defined as $\kappa_q := \prod_i \frac{1}{1-q^{-i}}$ for any prime power q .

We implemented the proposed decoder in SageMath [43] and ran a Monte Carlo simulation to heuristically verify the tightness of the upper bound on the decoding-failure probability given in (9). Note that the actual failure probability is hard to simulate for reasonable parameter sizes, as even the upper bound decreases exponentially. To obtain observable results, we chose $\mathbb{F}_{q^m} = \mathbb{F}_{3^8}$, $\mathbb{F}_q = \mathbb{F}_3$, and an HILRS code of length $n = 16$ and dimension $k = 4$ with respect to the Frobenius automorphism. We considered $\ell = 2$ blocks of the same length, namely $\mathbf{n} = (8, 8)$, interleaving order $s = 3$, and randomly chosen errors of sum-rank weight $t = t_{\max} = 9$. The failure probability that we observed for 100 Monte Carlo errors is $1.569 \cdot 10^{-4}$ while the bound yields $6.535 \cdot 10^{-3}$.

We finish this section with a summary of the results we have obtained so far and give a complexity analysis of the Gao-like decoder for HILRS codes.

Theorem 2 (Gao-like Decoding of HILRS Codes). *Consider the transmission of a codeword $\mathbf{c} \in \text{HILRS}[\beta, \xi, s; \mathbf{sn}, sk]$ over the channel (2). Let $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_{q^m}^{sn}$ denote the received word and assume that the error \mathbf{e} has bounded sum-rank weight*

$$\text{wt}_{\Sigma R}(\mathbf{e}) = t \leq \frac{s}{s+1}(n-k). \quad (10)$$

Then, the Gao-like decoder from Algorithm 1 can recover \mathbf{c} with a failure probability \Pr_{fail} that is bounded by

$$\Pr_{\text{fail}} \leq \kappa_q^{\ell+1} q^{-m((s+1)(t_{\max}-t)+1)} < 3.5^{\ell+1} q^{-m((s+1)(t_{\max}-t)+1)}$$

in the zero-derivation setting. If the key equation (7) is solved via Gaussian elimination in the formulation of (8), the overall complexity of the decoder is in the order of $\tilde{\mathcal{O}}((sn)^\omega) \subseteq \tilde{\mathcal{O}}((sn)^{2.373})$ operations in \mathbb{F}_{q^m} .

Proof. The decoding radius and the bound on the failure probability were derived above. Let us thus focus on the complexity analysis.

- The computation of a minimal or an interpolation polynomial of degree at most n can be done with complexity $\tilde{\mathcal{O}}(\mathcal{M}_{q,m}(n))$ according to [11, Section II.D.], e.g. by using the recursive formula (1). Thus, the computation of G_1, \dots, G_s and R_1, \dots, R_s takes $\tilde{\mathcal{O}}(s\mathcal{M}_{q,m}(n))$ operations in \mathbb{F}_{q^m} .
- Finding the solution of the key equation via Gaussian elimination has complexity $\mathcal{O}(\max(sn, (s+1)t + sk + 1)^\omega)$ as stated above. Since equation (10) ensures $sn \geq (s+1)t + sk + 1$, we obtain $\mathcal{O}((sn)^\omega)$.
- The for-loop runs in $\tilde{\mathcal{O}}(s\mathcal{M}_{q,m}(n))$ operations in \mathbb{F}_{q^m} because the left division in line 4 has complexity $\tilde{\mathcal{O}}(\mathcal{M}_{q,m}(n))$ for each $j = 1, \dots, s$ according to [11, Section II.D.]. Checking the conditions for a decoding failure is essentially for free.

Note that $\tilde{\mathcal{O}}(s\mathcal{M}_{q,m}(n)) \subseteq \tilde{\mathcal{O}}(sn^{\min(\frac{\omega+1}{2}, 1.635)}) \subseteq \tilde{\mathcal{O}}(sn^{1.635})$. Thus, solving the Gao-like key equation determines the overall complexity of $\tilde{\mathcal{O}}((sn)^\omega)$ operations in \mathbb{F}_{q^m} . \square

4 A Fast Variant of the Gao-like Decoder for HILRS Codes

We now present a fast variant of the decoder from Algorithm 1. As we have seen in its complexity analysis in the proof of Theorem 2, the complexity-dominating task is the solution of the Gao-like key equation. Thus, we focus on this problem and obtain a performance gain by reformulating it in terms of minimal approximant bases.

Note that we restrict ourselves to the zero-derivation case in this section, even though the used concepts and algorithms generalize straightforwardly to nonzero derivations. The reason is that the complexity analysis of algorithms involving skew-polynomial operations with nonzero derivations is more involved and was e.g. not conducted for the minimal-approximant-basis algorithm [11, Algorithm 5] that we use for the speedup.

4.1 Minimal Approximant Bases

Let us give some definitions and basic properties of minimal approximant bases. Note that we will only discuss left/row approximant bases and leave out their right/column counterparts, as we are only concerned with these.

Let $\mathbf{v} \in \mathbb{Z}^a$ be a shifting vector. Then, the \mathbf{v} -shifted row degree of a vector $\mathbf{b} \in \mathbb{F}_{q^m}[x; \theta]^a$ is

$$\text{rdeg}_{\mathbf{v}}(\mathbf{b}) := \max_{j=1, \dots, a} \{\deg(b_j + v_j)\}.$$

For $\mathbf{b} \in \mathbb{F}_{q^m}[x; \theta]^a \setminus \{\mathbf{0}\}$ and $\mathbf{v} = (v_1, \dots, v_b) \in \mathbb{Z}^a$, the \mathbf{v} -pivot index of \mathbf{b} is the largest index $i \in \{1, \dots, a\}$ with $\deg(b_i) + v_i = \text{rdeg}_{\mathbf{v}}(\mathbf{b})$.

A matrix $\mathbf{W} \in \mathbb{F}_{q^m}[x; \theta]^{a \times b}$ with $a \leq b$ is in \mathbf{v} -ordered row weak-Popov form if the \mathbf{v} -pivot indices of its rows are strictly increasing in the row index.

A vector $\mathbf{b} \in \mathbb{F}_{q^m}[x; \theta]^a$ is a left approximant of order $d \in \mathbb{N}$ of a matrix $\mathbf{W} \in \mathbb{F}_{q^m}[x; \theta]^{a \times b}$ if

$$\mathbf{b}\mathbf{W} \equiv \mathbf{0} \pmod{x^d}.$$

A left \mathbf{v} -ordered weak-Popov approximant basis of \mathbf{A} of order $d \in \mathbb{N}$ is a full-rank matrix $\mathbf{B} \in \mathbb{F}_{q^m}[x; \theta]^{a \times a}$ in \mathbf{v} -ordered row weak-Popov form whose rows are a basis of all left approximants of \mathbf{A} of order d .

4.2 Solving the Gao-like Key Equation via Minimal Approximant Bases

The Gao-like key equation (7) can also be written as

$$\sigma \cdot f_j = \chi_j \cdot G_j + \sigma \cdot R_j \quad \text{for all } j = 1, \dots, s, \quad (11)$$

where $\chi_j \in \mathbb{F}_{q^m}[x; \theta]$ exists according to the Euclidean algorithm and has degree at most $k + t$ for each $j = 1, \dots, s$. Observe that (11) implies that the vector

$$(\sigma \cdot f_1, \dots, \sigma \cdot f_s, \sigma, \chi_1, \dots, \chi_s) \in \mathbb{F}_{q^m}[x; \theta]^{2s+1}$$

is in the left kernel of the matrix

$$\mathbf{W} = \begin{pmatrix} -\mathbf{I}_s \\ \mathbf{R} \\ \mathbf{G} \end{pmatrix} \in \mathbb{F}_{q^m}[x; \theta]^{(2s+1) \times s} \quad (12)$$

where $\mathbf{R} := (R_1, \dots, R_s)$ and $\mathbf{G} := \text{diag}(G_1, \dots, G_s)$.

The following result based on [11, Lemma 21] is fundamental for reformulating the Gao-like key equation as a minimal-approximant-bases problem.

Lemma 2. *Consider the same setting as in Theorem 2 and let \mathbf{W} be defined as in (12). Further write*

$$\boldsymbol{\rho} := (\sigma \cdot f_1, \dots, \sigma \cdot f_s, \sigma) \quad \text{and} \quad \boldsymbol{\chi} := (\chi_1, \dots, \chi_s)$$

for simplicity. Further define the shifting vectors $\mathbf{w} := (\mathbf{0}_s, k-1) \in \mathbb{Z}^{s+1}$ and $\mathbf{v} := (\mathbf{0}_s, k-1, \mathbf{0}_s) \in \mathbb{Z}^{2s+1}$, as well as the degree constraints $D := t_{\max} = \frac{s}{s+1}(n-k)$ and $d := D + n$. Then,

$$(\boldsymbol{\rho} \mid \boldsymbol{\chi}) \cdot \mathbf{W} = \mathbf{0} \quad \text{and} \quad \text{rdeg}_{\mathbf{w}}(\boldsymbol{\rho}) < D \quad (13)$$

if and only if

$$(\boldsymbol{\rho} \mid \boldsymbol{\chi}) \cdot \mathbf{W} \equiv \mathbf{0} \pmod{x^d} \quad \text{and} \quad \text{rdeg}_{\mathbf{v}}(\boldsymbol{\rho} \mid \boldsymbol{\chi}) < D. \quad (14)$$

Proof. We start with showing that (13) implies (14). The left-hand side of (14) clearly follows from (13) and it remains to show that $\deg(\chi_j) < D$ holds for all $j = 1, \dots, s$. With (11), we get

$$\begin{aligned} \deg(\chi_j) &\leq \max\{\deg(\sigma \cdot f_j), \deg(\sigma \cdot R_j)\} - \deg(G_j) \\ &\leq \max\{t + k - 1, t + n - 1\} - n < t \leq t_{\max} = D. \end{aligned}$$

For the other implication, note that the right-hand side of (14) directly implies the right-hand side of (13). In order to see that the left-hand side of (13) holds, we show that all entries of the vector $(\boldsymbol{\rho} \mid \boldsymbol{\chi}) \cdot \mathbf{W}$ have degree less than d . With the help of the right-hand side of (14) and (11), we obtain:

- $\deg(\sigma \cdot f_j) < D < d$,
- $\deg(\sigma \cdot R_j) \leq \deg(\sigma) + \deg(R_j) \leq t + n - 1 = D + n - 1 < d$,
- $\deg(\chi_j \cdot G_j) < t + n = D + n = d$.

□

Hence, we can solve the Gao-like key equation (7) by computing a left \mathbf{v} -ordered weak-Popov approximant basis \mathbf{B} of \mathbf{W} . This can be accomplished by [11, Algorithm 5] requiring $\tilde{\mathcal{O}}(\mathcal{M}(n)) \subseteq \tilde{\mathcal{O}}(n^{\min\{\frac{s+1}{2}, 1.635\}}) \subseteq \tilde{\mathcal{O}}(n^{1.635})$ operations in \mathbb{F}_{q^m} .

We then obtain candidates p_j for the products $\sigma \cdot f_j$ for each $j = 1, \dots, s$ and a candidate $\tilde{\sigma}$ for the σ by choosing the row \mathbf{b}_{\min} of \mathbf{B} having minimal \mathbf{v} -weighted degree. This choice makes sure to satisfy the degree constraint in (14) to get a proper solution as described in Lemma 2. The subroutine for solving the Gao-like key equation via the presented minimal-approximant-bases approach is summarized in Algorithm 2.

Algorithm 2: Subroutine solveKEviaMAB(\cdot) for Solving the Gao-like Key Equation via a Minimal Approximant Basis

Input : $R_1, \dots, R_s, G_1, \dots, G_s, n, k, s$
Output : $p_1, \dots, p_s, \tilde{\sigma}$

- 1 $\mathbf{v} := (\mathbf{0}_s, k - 1, \mathbf{0}_s)$
- 2 $D := \frac{s}{s+1}(n - k)$ and $d := D + n$
- 3 $\mathbf{W} := \begin{pmatrix} -\mathbf{I}_s \\ \mathbf{R} \\ \mathbf{G} \end{pmatrix} \in \mathbb{F}_{q^m}[x; \theta]^{(2s+1) \times s}$ with $\mathbf{R} := (R_1, \dots, R_s)$ and
 $\mathbf{G} := \text{diag}(G_1, \dots, G_s)$
/* left \mathbf{v} -ordered weak Popov approximant basis of \mathbf{W} of order d */
- 4 $\mathbf{B} := \text{LeftSkewPMBasis}(d, \mathbf{W}, \mathbf{v}) \in \mathbb{F}_{q^m}[x; \theta]^{(2s+1) \times (2s+1)}$
- 5 Define $\mathbf{b}_{\min} = (b_{\min,1}, \dots, b_{\min,2s+1})$ as the minimal row of \mathbf{B} with respect to the \mathbf{v} -weighted degree
- 6 **return** $b_{\min,1}, \dots, b_{\min,s}, b_{\min,s+1}$

Theorem 3. *Algorithm 2 solves the Gao-like key equation (7) in $\tilde{O}(s^\omega n^{1.635}) \subseteq \tilde{O}(s^{2.373} n^{1.635})$ \mathbb{F}_{q^m} -operations.*

Proof. The complexity of Algorithm 2 is dominated by finding a minimal approximant basis in line 4. This can be achieved using [11, Algorithm 5] whose complexity is $\tilde{O}(s^\omega n^{1.635}) \subseteq \tilde{O}(s^{2.373} n^{1.635})$ [11, Theorem 11].

This directly implies the following complexity improvement for Theorem 2:

Corollary 1. *When the Gao-like key equation (7) is solved by Algorithm 2, the complexity of the Gao-like decoder from Algorithm 1 decreases to $\tilde{O}(s^\omega n^{1.635}) \subseteq \tilde{O}(s^{2.373} n^{1.635})$ operations in \mathbb{F}_{q^m} .*

With Corollary 1, the Gao-like decoder is the fastest known decoder for HILRS codes in the sum-rank metric as well as for horizontally interleaved Gabidulin codes in the rank metric. Its complexity is essentially subquadratic in the component-code length n , as the interleaving order s is usually much smaller than the code length n . Remark in particular that the gain in the error-correcting capacity increases fast for increasing s , as $\frac{s}{s+1}$ quickly tends to one.

5 Conclusion

We studied HILRS codes and their fast decoding which has promising potential applications in code-based cryptography. As a starting point, we presented a Gao-like decoder that features probabilistic unique decoding for an error of sum-rank weight at most $\frac{s}{s+1}(n-k)$, where s is the interleaving order, and n and k are the length and the dimension of the component codes. We gave a bound on the failure probability and achieved a complexity of $\tilde{O}((sn)^{2.373})$ operations in \mathbb{F}_{q^m} by solving the Gao-like key equation conventionally via Gaussian elimination.

Techniques from the area of minimal approximant bases allowed us to speed up the decoder significantly and obtain a complexity of $\tilde{O}(s^{2.373} n^{1.635})$ operations in \mathbb{F}_{q^m} . Under the reasonable assumption that the interleaving order s is small compared to the component-code length n , this is subquadratic. Overall, this results in the fastest known decoders for both HILRS codes in the sum-rank metric and for horizontally interleaved Gabidulin codes in the rank metric.

Further work can include the generalization of the presented decoder to the error-erasure case. Next to errors, this error model includes row and column erasures, for which either the row space or the column space is known. Moreover, other techniques could give bounds on the failure probability for nonzero derivations or yield tighter ones for the zero-derivation setting.

References

1. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J.C., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J.M., Véron, P., Zémor, G.: Hamming Quasi-Cyclic (HQC) (2023), http://pqc-hqc.org/download.php?file=hqc-specification_2023-04-30.pdf

2. Aguilar-Melchor, C., Aragon, N., Dyseryn, V., Gaborit, P., Zémor, G.: LRPC Codes With Multiple Syndromes: Near Ideal-Size KEMs Without Ideals. In: Post-Quantum Cryptography. pp. 45–68 (2022)
3. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process (2022). <https://doi.org/10.6028/NIST.IR.8413-upd1>
4. Aragon, N., Barreto, P.S.L.M., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Ghosh, S., Gueron, S., Güneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Richter-Brockmann, J., Sendrier, N., Tillich, J.P., Vasseur, V., Zémor, G.: BIKE: Bit Flipping Key Encapsulation (2022), https://bikesuite.org/files/v5.0/BIKE_Spec.2022.10.10.1.pdf
5. Aragon, N., Blazy, O., Gaborit, P., Hauteville, A., Zémor, G.: Durandal: A Rank Metric Based Signature Scheme. In: Advances in Cryptology – EUROCRYPT 2019. pp. 728–758 (2019)
6. Aragon, N., Dyseryn, V., Gaborit, P., Loidreau, P., Renner, J., Wachter-Zeh, A.: LowMS: A New Rank Metric Code-Based KEM Without Ideal Structure. Cryptology ePrint Archive, Paper 2022/1596 (2022)
7. Aragon, N., Gaborit, P., Hauteville, A., Tillich, J.P.: A New Algorithm for Solving the Rank Syndrome Decoding Problem. In: IEEE International Symposium on Information Theory (ISIT). pp. 2421–2425 (2018)
8. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber: Algorithm Specifications and Supporting Documentation (Version 3.02) (2021), <https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf>
9. Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Hutchinson, A., Jalali, A., Karabina, K., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., Urbanik, D.: Supersingular Isogeny Key Encapsulation (2022), <https://sike.org/files/SIDH-spec.pdf>
10. Bardet, M., Briaud, P.: An Algebraic Approach to the Rank Support Learning Problem. In: Post-Quantum Cryptography. pp. 442–462 (2021)
11. Bartz, H., Jerkovits, T., Puchinger, S., Rosenkilde, J.: Fast Decoding of Codes in the Rank, Subspace, and Sum-Rank Metric. IEEE Transactions on Information Theory **67**(8), 5026–5050 (2021)
12. Bartz, H., Puchinger, S.: Decoding of Interleaved Linearized Reed-Solomon Codes with Applications to Network Coding. In: IEEE International Symposium on Information Theory (ISIT). pp. 160–165 (2021)
13. Bartz, H., Puchinger, S.: Fast Decoding of Interleaved Linearized Reed-Solomon Codes and Variants. submitted to: IEEE Transactions on Information Theory (2023), available at <https://arxiv.org/abs/2201.01339v3>
14. Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Sendrier, N., Szefer, J., Tjhai, C.J., Tomlinson, M., Wang, W.: Classic McEliece: Conservative Code-Based Cryptography: Cryptosystem Specification (2022), <https://classic.mceliece.org/mceliece-spec-20221023.pdf>
15. Boucher, D.: An Algorithm for Decoding Skew Reed-Solomon Codes with respect to the Skew Metric. Designs, Codes and Cryptography **88**(9), 1991–2005 (2020)
16. Caruso, X.: Residues of Skew Rational Functions and Linearized Goppa Codes. arXiv preprint arXiv:1908.08430v1 (2019)
17. Castryck, W., Decru, T.: An Efficient Key Recovery Attack on SIDH. Cryptology ePrint Archive, Paper 2022/975 (2022)

18. Gabidulin, E.M., Paramonov, A., Tretjakov, O.: Ideals over a Non-Commutative Ring and Their Application in Cryptology. In: Workshop on the Theory and Application of Cryptographic Techniques. pp. 482–489. Springer (1991)
19. Gabidulin, E.M., Rashwan, H., Honary, B.: On Improving Security of GPT Cryptosystems. In: IEEE International Symposium on Information Theory. pp. 1110–1114 (2009)
20. Gao, S.: A New Algorithm for Decoding Reed–Solomon Codes. In: Communications, Information and Network Security, pp. 55–68. Springer (2003)
21. Hörmann, F., Bartz, H., Horlemann, A.L.: Distinguishing and Recovering Generalized Linearized Reed–Solomon Codes. In: Code-Based Cryptography: CBCrypto 2022, pp. 1–20 (2023)
22. Hörmann, F., Bartz, H., Puchinger, S.: Error-Erasure Decoding of Linearized Reed–Solomon Codes in the Sum-Rank Metric. In: IEEE International Symposium on Information Theory (ISIT). pp. 7–12 (2022)
23. Hörmann, F., Bartz, H., Puchinger, S.: Syndrome-Based Error-Erasure Decoding of Interleaved Linearized Reed–Solomon Codes. to be submitted to: IEEE Transactions on Information Theory (2023)
24. Jerkovits, T., Hörmann, F., Bartz, H.: On Decoding High-Order Interleaved Sum-Rank-Metric Codes. In: Code-Based Cryptography: CBCrypto 2022. pp. 90–109 (2023)
25. Lam, T.Y., Leroy, A.: Vandermonde and Wronskian Matrices over Division Rings. *Journal of Algebra* **119**(2), 308–336 (1988)
26. Le Gall, F.: Powers of Tensors and Fast Matrix Multiplication. In: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation. pp. 296–303 (2014)
27. Loidreau, P.: An Evolution of GPT Cryptosystem. In: International Workshop on Algebraic and Combinatorial Coding Theory (ACCT) (2016)
28. Loidreau, P.: Designing a Rank Metric Based McEliece Cryptosystem. In: International Workshop on Post-Quantum Cryptography. pp. 142–152 (2010)
29. Martínez-Peñas, U.: Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes over any Division Ring. *Journal of Algebra* **504**, 587–612 (2018)
30. Martínez-Peñas, U., Kschischang, F.R.: Reliable and Secure Multishot Network Coding using Linearized Reed-Solomon Codes. *IEEE Transactions on Information Theory* **65**(8), 4785–4803 (2019)
31. Matsaglia, G., Styan, G.P.H.: Equalities and Inequalities for Ranks of Matrices. *Linear and Multilinear Algebra* **2**(3), 269–292 (1974)
32. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. *The Deep Space Network Progress Report* **42-44**, 114–116 (1978)
33. Metzner, J., Kapturowski, E.: A General Decoding Technique Applicable to Replicated File Disagreement Location and Concatenated Code Decoding. *IEEE Transactions on Information Theory* **36**(4), 911–917 (1990)
34. Ore, O.: On a Special Class of Polynomials. *Transactions of the American Mathematical Society* **35**(3), 559–584 (1933)
35. Ore, O.: Theory of Non-Commutative Polynomials. *Annals of Mathematics* pp. 480–508 (1933)
36. Puchinger, S., Rosenkilde né Nielsen, J., Li, W., Sidorenko, V.: Row Reduction Applied to Decoding of Rank-Metric and Subspace Codes. *Designs, Codes and Cryptography* **82**(1-2), 389–409 (2017)

37. Puchinger, S., Renner, J., Rosenkilde, J.: Generic Decoding in the Sum-Rank Metric. In: IEEE International Symposium on Information Theory (ISIT). pp. 54–59 (2020)
38. Puchinger, S., Renner, J., Wachter-Zeh, A.: Decoding High-Order Interleaved Rank-Metric Codes. arXiv preprint arXiv:1904.08774 (2019)
39. Renner, J., Jerkovits, T., Bartz, H.: Efficient Decoding of Interleaved Low-Rank Parity-Check Codes. In: 2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems"(REDUNDANCY). pp. 121–126 (2019)
40. Renner, J., Puchinger, S., Wachter-Zeh, A.: Decoding High-Order Interleaved Rank-Metric Codes. In: IEEE International Symposium on Information Theory (ISIT). pp. 19–24 (2021)
41. Sidorenko, V., Bossert, M.: Decoding Interleaved Gabidulin Codes and Multisequence Linearized Shift-Register Synthesis. In: IEEE International Symposium on Information Theory. pp. 1148–1152 (2010)
42. Sidorenko, V., Jiang, L., Bossert, M.: Skew-Feedback Shift-Register Synthesis and Decoding Interleaved Gabidulin Codes. *IEEE Transactions on Information Theory* **57**(2), 621–632 (2011)
43. Stein, W.A., et al.: Sage Mathematics Software (Version 9.6). The Sage Development Team (2022), <http://www.sagemath.org>
44. Storjohann, Arne: Algorithms for Matrix Canonical Forms. Ph.D. thesis (2000)
45. 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)
46. Wachter-Zeh, A., Afanassiev, V., Sidorenko, V.: Fast Decoding of Gabidulin Codes. *Designs, Codes and Cryptography* **66**(1), 57–73 (2013)