Randomized Decoding of Linearized Reed–Solomon Codes Beyond the Unique Decoding Radius

Thomas Jerkovits, Hannes Bartz
Institute of Communication and Navigation
German Aerospace Center (DLR)
{thomas.jerkovits, hannes.bartz}@dlr.de

Antonia Wachter-Zeh
School of Computation, Information and Technology
Technical University of Munich (TUM)
antonia.wachter-zeh@tum.de

Abstract—In this paper we address the problem of decoding linearized Reed–Solomon (LRS) codes beyond their unique decoding radius. We analyze the complexity in order to evaluate if the considered problem is of cryptographic relevance, i.e., can be used to design cryptosystems that are computationally hard to break. We show that our proposed algorithm improves over other generic algorithms that do not take into account the underlying code structure.

I. INTRODUCTION

The sum-rank metric is a generalization of both, the Hamming and the rank metric, and was first introduced for space-time codes in [1]. Since then, several code constructions and decoders have been proposed for the sum-rank metric [2]–[9]. Linearized Reed–Solomon (LRS) codes were later introduced by Martínez-Peñas which include Reed–Solomon codes and Gabidulin codes as special cases [10]. LRS codes are of interest for applications such as multishot network coding [7], [11], locally repairable codes [10], space-time codes [1], and code-based quantum-resistant cryptography [12].

It is well-known that the problem of decoding beyond the unique decoding radius, specifically *maximum-likelihood* decoding, is a difficult problem w.r.t. the complexity. For the Hamming metric, many works have investigated the hardness of this problem [13]–[15]. List decoding is another method to decode beyond the unique decoding radius and the complexity depends on the list size. Bounds on the list size for LRS codes are given in [16] and it was shown that some families of LRS codes cannot be decoded beyond the unique decoding radius. The exponential complexity of list decoding makes it a potentially useful tool for cryptography. Before designing cryptosystems based on the list decoding problem in the sum-rank metric, its computational complexity must be carefully studied and analyzed. For the rank metric the problem of decoding beyond the unique decoding radius was addressed in [17] for Gabidulin codes. Known structural attacks for McEliece-like cryptosystem in the Hamming and Rank metric [18]–[20] have been generalized to the sum-rank metric [21]. This raises the question if the sum-rank metric can be adapted to other cryptosystems that are based on the hardness of decoding beyond the unique decoding radius, such as [22]–[24].

In this paper we propose an algorithm which generalizes the one from [17] to LRS codes (in the sum-rank metric). Note that the work factor, i.e., the computational complexity of the algorithm, derived in [17] can be used to assess the security level of cryptosystems like [22]–[24]. Thus, the work factors derived in this paper might be used to assess the security level of similar cryptosystems in the sum-rank metric. The main idea of the algorithm is to randomly guess parts of the error by introducing so-called erasures and trade errors with erasures. This allows to apply an error-erasure decoder (e.g., [25]) to decode successfully if enough errors were traded with erasures. We analyze the probability of this event for a specific distribution of guessed erasures. The gain comes from the fact, that erasures weigh less than errors with respect to the decoding capability of an LRS code.

Additionally, we demonstrate a method to find the optimal distribution of erasures. We show that the proposed algorithm which exploits the structure of the underlying LRS code improves over the generic decoding algorithm, introduced in [12], in terms of expected computational complexity.

II. PRELIMINARIES

A. Notation

For a prime power q and a positive integer m, let $\mathbb{F}_q$ denote a finite field of order q and $\mathbb{F}_{q^m}$ its extension field of extension degree m. Let $\boldsymbol{b} = (b_1, \ldots, b_m) \in \mathbb{F}_{q^m}^m$ be a fixed (ordered) basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. We denote by $\mathrm{ext}(\alpha)$ the column-wise expansion of an element $\alpha \in \mathbb{F}_{q^m}$ over $\mathbb{F}_q$ w.r.t. to the basis $\boldsymbol{b}$ s.t. $\alpha = \boldsymbol{b} \cdot \mathrm{ext}(\alpha)$. This notation is extended to vectors and matrices by applying $\mathrm{ext}(\cdot)$ in an element-wise manner s.t. $\mathrm{ext} : \mathbb{F}_{q^m}^n \mapsto \mathbb{F}_q^{m \times n}$. For a vector $\boldsymbol{x} \in \mathbb{F}_{q^m}^n$ we define $\mathrm{rk}_q(\boldsymbol{x}) := \mathrm{rk}_q(\mathrm{ext}(\boldsymbol{x}))$. The $\mathbb{F}_q$-linear row space of a matrix $\boldsymbol{B} \in \mathbb{F}_q^{m \times n}$ is denoted as $\mathcal{R}_q(\boldsymbol{B})$ and the Grassmanian $\mathcal{G}_q(\mathcal{V}, i)$ of an $\mathbb{F}_q$-vector space $\mathcal{V}$ is the set of all i-dimensional subspaces of $\mathcal{V}$. We use the notation $a \xleftarrow{\$} \mathcal{A}$ to denote an element a drawn uniformly at random from a set $\mathcal{A}$.

B. Sum-Rank Weight and Linearized Reed–Solomon Codes

Let $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{F}_{q^m}^n$ be a vector, that is partitioned into blocks $\boldsymbol{x}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$ w.r.t. to a *length partition*

$\boldsymbol{n} = (n_1, \dots, n_\ell) \in \mathbb{N}^\ell$. The *sum-rank weight* of $\boldsymbol{x}$ w.r.t. to the length partition $\boldsymbol{n}$ is then defined as

$$\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x}) := \sum_{i=1}^{\ell} \mathrm{rk}_q(\boldsymbol{x}^{(i)}).$$

The *sum-rank distance* of two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_{q^m}^n$ is then defined by the sum-rank weight $d_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x}, \boldsymbol{y}) := \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{x} - \boldsymbol{y})$.

LRS codes [6] are a special class of sum-rank-metric codes which are maximum sum-rank distance (MSRD). This means that the *minimum sum-rank distance* is $n - k + 1$ where n is the code length and k is the code dimension. Hence, LRS codes can uniquely decode errors of weight up to $\tau := \lfloor \frac{d-1}{2} \rfloor$. Throughout this paper we consider LRS codes of length n, with length partition $\boldsymbol{n}$ and dimension k over $\mathbb{F}_{q^m}$ which we denote as $\mathcal{C}_{\mathrm{LRS}}$. Note that LRS codes are restricted to $\ell \leq q-1$ and $n_i \leq m$ for all $i = 1, \dots, \ell$ (see [6]).

C. Channel Model

Let $\boldsymbol{c} \in \mathcal{C}_{\mathrm{LRS}}$ and let $\boldsymbol{c}$ be corrupted by an error $\boldsymbol{e}$ of sum-rank weight w, i.e., the received word is $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$.

Any error $\boldsymbol{e} = (\boldsymbol{e}^{(1)} \mid \dots \mid \boldsymbol{e}^{(\ell)}) \in \mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = w$ can be decomposed into a vector-matrix product as

$$\boldsymbol{e} = \boldsymbol{a}\boldsymbol{B} \qquad (1)$$

with $\boldsymbol{a} := (\boldsymbol{a}^{(1)} \mid \dots \mid \boldsymbol{a}^{(\ell)})$ and $\boldsymbol{B} := \mathrm{diag}(\boldsymbol{B}^{(1)}, \dots, \boldsymbol{B}^{(\ell)})$ and with $\boldsymbol{a}^{(i)} \in \mathbb{F}_{q^m}^{w_i}$ and $\boldsymbol{B} \in \mathbb{F}_q^{w_i \times n_i}$ s.t. $\mathrm{rk}_q(\boldsymbol{a}^{(i)}) = \mathrm{rk}_q(\boldsymbol{B}^{(i)}) = w_i$ and $w = \sum_{i=1}^{\ell} w_i$ for all $i = 1, \dots, \ell$. It follows that $\boldsymbol{e}^{(i)} = \boldsymbol{a}^{(i)} \boldsymbol{B}^{(i)}$ for $i = 1, \dots, \ell$ and we have that the entries of $\boldsymbol{a}^{(i)}$ form a basis of the column space of $\boldsymbol{e}^{(i)}$ and the rows of $\boldsymbol{B}^{(i)}$ form a basis of its row space.

The error $\boldsymbol{e}$ can be further decomposed into a sum of three types of error vectors, namely $\boldsymbol{e}_F$, $\boldsymbol{e}_R$ and $\boldsymbol{e}_C$ s.t.

$$\boldsymbol{e} = \boldsymbol{e}_F + \boldsymbol{e}_R + \boldsymbol{e}_C$$

with $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}_F) = t$, $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}_R) = \rho$ and $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}_C) = \gamma$, respectively [25]. Each of the three vectors can then be decomposed again as in (1), with $\boldsymbol{a}_F$, $\boldsymbol{B}_F$, $\boldsymbol{a}_R$, $\boldsymbol{B}_R$ and $\boldsymbol{a}_C$, $\boldsymbol{B}_C$, respectively. Assuming neither $\boldsymbol{a}_F$ nor $\boldsymbol{B}_F$ are known, the term $\boldsymbol{a}_F \boldsymbol{B}_F$ is called *full rank errors*. If $\boldsymbol{a}_R$ is known but $\boldsymbol{B}_R$ is unknown, the vector $\boldsymbol{a}_R \boldsymbol{B}_R$ is called *row erasures* and assuming $\boldsymbol{a}_C$ is unknown but $\boldsymbol{B}_C$ is known the product $\boldsymbol{a}_C \boldsymbol{B}_C$ is called *column erasures*.

An efficient algorithm for LRS codes has been proposed in [25] that is able to correct a combination of *full rank errors*, *row erasures* and *column erasures* up to

$$2t + \gamma + \rho \leq n - k \qquad (2)$$

with a complexity of $O(n^2)$ operations over $\mathbb{F}_{q^m}$. We denote by $\mathrm{DEC}(\boldsymbol{y}, \boldsymbol{a}_R, \boldsymbol{B}_C)$ such an error-erasure decoder that takes as input the received word $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$, and a basis $\boldsymbol{a}_R$ of parts of the column spaces (row erasures) and a basis $\boldsymbol{B}_C$ for parts of the row spaces (column erasures) of the error $\boldsymbol{e}$ and outputs a valid codeword $\hat{\boldsymbol{c}}$ if (2) is fulfilled or returns $\emptyset$ else.

Definition 1 (Row and Column Support) *Let* $\boldsymbol{e} \in \mathbb{F}_{q^m}^n$ *be of sum-rank weight* w.

- **Row Support:** *The row support of* $\boldsymbol{e}$ *is defined as*
$$\boldsymbol{\mathcal{E}}_R := \mathcal{E}_R^{(1)} \times \mathcal{E}_R^{(2)} \times \cdots \times \mathcal{E}_R^{(\ell)},$$
where $\mathcal{E}_R^{(i)} \subseteq \mathbb{F}_q^{n_i}$ *is the* $\mathbb{F}_q$*-row space of* $\boldsymbol{B}^{(i)} \in \mathbb{F}_q^{w_i \times n_i}$ *and thus of* $\boldsymbol{e}^{(i)}$ *as in* (1) *for all* $i = 1, \dots, \ell$.
- **Column Support:** *The column support of* $\boldsymbol{e}$ *is defined as*
$$\boldsymbol{\mathcal{E}}_C := \mathcal{E}_C^{(1)} \times \mathcal{E}_C^{(2)} \times \cdots \times \mathcal{E}_C^{(\ell)},$$
where $\mathcal{E}_C^{(i)} \subseteq \mathbb{F}_q^m$ *is the column space of* $\mathrm{ext}(\boldsymbol{a}^{(i)}) \in \mathbb{F}_q^{m \times w_i}$ *and thus of* $\mathrm{ext}(\boldsymbol{e}^{(i)})$ *as in* (1) *for all* $i = 1, \dots, \ell$.

Assume $\boldsymbol{\mathcal{E}}$ to be either a row or column support of the error. We denote by $\dim_\Sigma(\boldsymbol{\mathcal{E}})$ the *sum dimension* of an error support:

$$\dim_\Sigma(\boldsymbol{\mathcal{E}}) := \sum_{i=1}^{\ell} \dim(\mathcal{E}^{(i)}).$$

The intersection of two supports $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$ is defined as

$$\boldsymbol{\mathcal{E}}_1 \cap \boldsymbol{\mathcal{E}}_2 := \mathcal{E}_1^{(1)} \cap \mathcal{E}_2^{(1)} \times \cdots \times \mathcal{E}_1^{(\ell)} \cap \mathcal{E}_2^{(\ell)}.$$

D. Combinatorics

Definition 2 (Weight Composition) *Let* w, ℓ *and* μ *be nonnegative integers s.t.* $w \leq \ell\mu$. *We define the set*

$$\mathcal{T}_{w,\ell,\mu} = \left\{ \boldsymbol{w} \in \{0, \dots, \mu\}^\ell : \sum_{i=1}^{\ell} w_i = w \right\},$$

which contains all possible weight compositions of a vector consisting of ℓ *blocks and sum-rank weight* w. *This notion is also known in combinatorics as* **weak integer composition**.

Definition 3 *Let* μ *be a positive integer and* $0 \leq s \leq \ell\mu$. *For* $\boldsymbol{s} \in \mathcal{T}_{s,\ell,\mu}$, *we define the set of all supports as*

$$\Xi_\mu(\boldsymbol{s}) := \left\{ \mathcal{F}_1 \times \cdots \times \mathcal{F}_\ell : \mathcal{F}_i \subseteq \mathbb{F}_q^\mu \text{ s.t. } \dim(\mathcal{F}_i) = s_i \right\}.$$

The number of all matrices in $\mathbb{F}_q^{a \times b}$ with $\mathbb{F}_q$-rank j is

$$\mathrm{NM}_q(a, b, j) := \begin{bmatrix} a \\ j \end{bmatrix}_q \cdot \prod_{i=0}^{j-1} (q^b - q^i)$$

where $\begin{bmatrix} a \\ j \end{bmatrix}_q$ is the Gaussian binomial coefficient defined as

$$\begin{bmatrix} a \\ j \end{bmatrix}_q := \prod_{i=1}^{j} \frac{q^{a-j+i} - 1}{q^i - 1}.$$

Let $\boldsymbol{n} \in \mathbb{N}^\ell$ be a length partition and let μ and w be nonnegative integers s.t. $w \leq \ell\mu$, we denote by $\mathrm{NSR}_q(m, \boldsymbol{n}, j)$ the number of vectors in $\mathbb{F}_{q^m}^n$ of sum-rank weight exactly j. It is easy to see that we have

$$\mathrm{NSR}_q(m, \boldsymbol{n}, j) = \sum_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \prod_{i=1}^{\ell} \mathrm{NM}_q(m, n_i, w_i).$$

Note that there are efficient ways to compute $\mathrm{NSR}_q(m, \boldsymbol{n}, j)$ (cf. [12]).

III. GENERIC DECODING

A generic sum-rank-metric decoding (SRMD) algorithm is an algorithm solving Problem 1.

Problem 1 (Search-SRMD)
- **Instance:** *Linear sum-rank metric code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$, $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ and an integer $t > 0$.*
- **Objective:** *Find a codeword $\boldsymbol{c} \in \mathcal{C}$, s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \boldsymbol{c}) \leq t$.*

For $t \leq \tau = \left\lfloor \frac{d-1}{2} \right\rfloor$ at most one solution to Problem 1 exists. In general, for decoding beyond the unique decoding radius there might be many solutions to Problem 1 but for our consideration, and as stated in Problem 1, it is sufficient to find one of them. A generic decoder for sum-rank metric codes for Problem 1 with $t \leq \tau$ is presented in [12] and several bounds on the computational complexity of the proposed algorithm are given. We denote the lower and upper bound on the work factor of [12] for solving Problem 1 as $\tilde{W}_{\mathrm{gen}}^{(\mathrm{LB})}$ and $\tilde{W}_{\mathrm{gen}}^{(\mathrm{UB})}$, respectively. Note that in contrast to [12] we consider a constant complexity of a single iteration for the lower bound.

Problem 2 (Search-LRS)
- **Instance:** *Linearized Reed–Solomon code $\mathcal{C}_{\mathrm{LRS}} \subseteq \mathbb{F}_{q^m}^n$, $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$ and an integer $t > 0$.*
- **Objective:** *Find $\boldsymbol{c} \in \mathcal{C}_{\mathrm{LRS}}$, s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \boldsymbol{c}) \leq t$.*

Problem 2 is a special instance of Problem 1, where the linear code is an LRS code. Currently, the generic decoder from [12] has the smallest known complexity to solve Problem 2. In this paper we show how to reduce the complexity of solving Problem 2 compared to the generic decoder for errors with weight $w = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \boldsymbol{c})$, s.t. $w > \tau$. In particular, we assume that the excess of the error over the unique decoding radius τ is larger than zero, i.e. $w - \tau > 0$.

IV. RANDOMIZED DECODING

The proposed approach is a generalization of the *randomized decoding algorithm* presented in [17] from Gabidulin codes (rank metric) to LRS codes (sum-rank metric). In the considered problem we assume an error $\boldsymbol{e}$ of weight $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = w > \tau$ with row support $\mathcal{E}_R$ and column support $\mathcal{E}_C$. The main idea is to guess parts of the error support. This is done by first drawing a weight composition $\boldsymbol{u} = (u_1, \ldots, u_\ell) \in \mathbb{N}^\ell$ of the guessed error support, according to a probability mass function (PMF) $p_{\boldsymbol{u}}$ and then a guessed support is drawn uniformly at random from $\Xi_\mu(\boldsymbol{u})$. This means, that for each block a row and/or column space is guessed independently with dimension u_i for $i = 1, \ldots, \ell$. If the sum of the dimensions of the intersections of the guessed spaces with the spaces of the actual error is large enough (that means enough errors were traded for erasures) an error-erasure decoder can decode successfully.

In [17] for Gabidulin codes it was shown, that the algorithm cannot be improved by guessing a combination of row spaces and column spaces. Therefore, we restrict to guessing only parts of the row support. Let $\mathcal{U}$ with $u := \dim_\Sigma(\mathcal{U})$ be the guessed row support then $\gamma = u$ and $\rho = 0$. The corresponding weight composition of $\mathcal{U}$ is $\boldsymbol{u}$. For simplicity, we restrict to the case of LRS codes with constant block sizes, that means that $n_1 = n_2 = \cdots = n_\ell$ and we denote $\eta = n/\ell$ for all $i = 1, \ldots, \ell$. In this case the maximum rank of a single block is at most $\mu := \min\{\eta, m\}$. The adaptation to variable block sizes is straightforward. The proposed algorithm for guessing only row spaces is presented in Algorithm 1.

In this section we analyze an upper bound on the expected number of operations over $\mathbb{F}_{q^m}$ of Algorithm 1 which solves Problem 2. Further, we show a method how to evaluate this bound without knowing the actual distribution $p_{\boldsymbol{u}}$. A lower bound on the expected number of operations $\mathbb{F}_{q^m}$ is given in Section IV-B. Finally, we present a method to compute the optimal distribution $p_{\boldsymbol{u}}$ that minimizes the worst-case number of iterations of the proposed algorithm.

Algorithm 1: Randomized LRS Decoder

Input : Parameters $q, m, k, \boldsymbol{n}, \ell, w, u$

Received word $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$

LRS error-erasure decoder $\mathrm{DEC}(\cdot, \cdot, \cdot)$

Output: $\hat{\boldsymbol{c}} \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \hat{\boldsymbol{c}}) \leq w$

1   $\mu \leftarrow \min\{\eta, m\}$

2   **while True do**

3   Draw $\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}$ according to the distribution $p_{\boldsymbol{u}}$

4     $\mathcal{U} \xleftarrow{\$} \Xi_\mu(\boldsymbol{u})$

5   **for** $j = 1, \ldots, \ell$ **do**

6       $\boldsymbol{B}^{(j)} \leftarrow$ full-rank matrix in $\mathbb{F}_q^{u_j \times \eta}$ s.t
       $\mathcal{R}_q(\boldsymbol{B}^{(j)}) = \mathcal{U}_j$ with $\dim(\mathcal{U}_j) = u_j$

7     $\boldsymbol{B} = \mathrm{diag}(\boldsymbol{B}^{(i)}, \ldots, \boldsymbol{B}^{(\ell)})$

8     $\hat{\boldsymbol{c}} \leftarrow \mathrm{DEC}(\boldsymbol{y}, \boldsymbol{0}, \boldsymbol{B})$

9   **if** $\hat{\boldsymbol{c}} \neq \emptyset$ and $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \hat{\boldsymbol{c}}) \leq w$ **then**

10   **return** $\hat{\boldsymbol{c}}$

A. Upper Bound on the Work Factor of the Algorithm

Define by ϵ the sum dimension of the guessed error support $\mathcal{U}$ and the actual error support $\mathcal{E}$, i.e. $\epsilon := \dim_\Sigma(\mathcal{U} \cap \mathcal{E})$. This means if ϵ is large enough, we trade errors for erasures and the decoding condition for an error-erasure decoder such as in [25] is $2(w - \epsilon) + u \leq n - k$ which implies that we should have

$$\epsilon \geq \left\lceil w + \frac{u - (n-k)}{2} \right\rceil.$$

Lemma 1 *[17, Lemma 1] Let the error space $\mathcal{E}^{(i)}$ of the i-th component of the error support $\mathcal{E}$ with $\dim(\mathcal{E}^{(i)}) = w_i$ and u_i, the dimension of the i-th component $\mathcal{U}^{(i)}$ of the guessed support be given. Choose $\mathcal{U}^{(i)}$ uniformly at random from $\mathcal{G}_q(\mathbb{F}_q^\mu, u_i)$ for all $i = 1, \ldots, \ell$. Then, the conditional probability $p_{w_i,u_i}^{(i)}(j) := \Pr[\dim(\mathcal{E}_i \cap \mathcal{U}_i) = j | \mathcal{E}^{(i)}, u_i]$, that the intersection of $\mathcal{E}^{(i)}$ and $\mathcal{U}^{(i)}$ is exactly j is*

$$p_{w_i,u_i}^{(i)}(j) := \frac{\begin{bmatrix} \mu - w_i \\ u_i - j \end{bmatrix}_q \begin{bmatrix} w_i \\ j \end{bmatrix}_q q^{(w_i-j)(u_i-j)}}{\begin{bmatrix} \mu \\ u_i \end{bmatrix}_q}.$$

Lemma 2 *Let μ be a non-negative integer. For a fixed error $e \in \mathbb{F}_{q^m}^n$ and given the weight composition $\boldsymbol{u} = (u_i, \ldots, u_\ell)$ of the guessed space $\mathcal{U}$ with $\dim(\mathcal{U}) = u$, choose $\mathcal{U}$ uniformly at random from $\Xi_\mu(\boldsymbol{u})$. Further let S_j be the event that $\dim_\Sigma(\mathcal{E} \cap \mathcal{U}) = j$. The probability of S_j conditioned on e and $\boldsymbol{u}$ is then*

$$\Pr[S_j|\boldsymbol{e},\boldsymbol{u}] = \left(\underset{i=1}{\overset{\ell}{\circledast}}\, p_{w_i,u_i}^{(i)} \right)(j)$$

with

$$\left(\underset{i=1}{\overset{\ell}{\circledast}}\, p_{w_i,u_i}^{(i)} \right)(j) := \left(p_{w_1,u_1}^{(1)} \circledast \cdots \circledast p_{w_\ell,u_\ell}^{(\ell)} \right)(j)$$

being the ℓ-fold discrete convolution of the PMFs $p_{w_i,u_i}^{(i)}$ evaluated at j for all $i = 1, \ldots, \ell$.

Proof: Given the error e with weight composition $\boldsymbol{w}$ and given the weight composition $\boldsymbol{u}$ of the guessed support, let V_i be a random variable that corresponds to the rank of the intersection of the i-th guessed space $\mathcal{U}^{(i)}$ with the i-th actual error space $\mathcal{E}^{(i)}$ for $i = 1, \ldots, \ell$. By Lemma 1 we have that $p_{w_i,u_i}^{(i)}(j)$ is the PMF of that event, i.e.

$$\Pr[V_i = j|\boldsymbol{e},\boldsymbol{u}] = p_{w_i,u_i}^{(i)}(j).$$

Since we are interested in the sum of random variables, i.e. $V = \sum_{i=1}^{\ell} V_i$ the resulting PMF is given by the ℓ-fold discrete convolution of the PMFs of the random variables V_i for $i = 1, \ldots, \ell$. Thus

$$\Pr[V = j|\boldsymbol{e},\boldsymbol{u}] = \left(\underset{i=1}{\overset{\ell}{\circledast}}\, p_{w_i,u_i}^{(i)} \right)(j)$$

with

$$\left(p_{w_1,u_1}^{(1)} \circledast p_{w_2,u_2}^{(2)} \right)(j) := \sum_{r=-\infty}^{\infty} p_{w_1,u_1}^{(1)}(r)\, p_{w_2,u_1}^{(2)}(j-r).$$

Finally we have that S_j is the event that $V = j$ and this proves the claim. ∎

For a given weight composition $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ of the error vector $\boldsymbol{e}$, each block $\boldsymbol{e}^{(i)}$ is drawn uniformly at random for $i = 1, \ldots, \ell$ and we have that $\Pr[S_j|\boldsymbol{e},\boldsymbol{u}] = \Pr[S_j|\boldsymbol{w},\boldsymbol{u}]$ for any non-negative integer j.

For further analysis we consider the worst-case expected number of iterations of Algorithm 1 until an appropriate guessed error support $\mathcal{U}$ is drawn s.t. the error-erasure de-coder can successfully decode. For given weight compositions $\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}$ and $\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}$ of the actual error and the guessed spaces, respectively, we define

$$\varphi_\mu(\boldsymbol{u},\boldsymbol{w}) := \sum_{j=\lceil w + \frac{u-(n-k)}{2} \rceil}^{\min[u,w]} \Pr[S_j|\boldsymbol{u},\boldsymbol{w}]$$

and for a given PMF $p_{\boldsymbol{u}}$ of $\boldsymbol{u}$ we have

$$\varphi_{\mu,u}(\boldsymbol{w}) := \sum_{\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}} p_{\boldsymbol{u}}\, \varphi_\mu(\boldsymbol{u},\boldsymbol{w}). \qquad (3)$$

The worst-case probability $\varphi_{\mu,u}(w)$ that maximizes the number of iterations over all possible weight compositions $\boldsymbol{w}$ is

$$\varphi_{\mu,u}(w) := \min_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{\mu,u}(\boldsymbol{w}) \qquad (4)$$

which implies that

$$\max_{\boldsymbol{w} \in \mathcal{T}_{w,\ell,\mu}} \mathbb{E}[\#\text{iterations}] = \varphi_{\mu,u}(w)^{-1}. \qquad (5)$$

Theorem 1 *Let $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ with $\boldsymbol{c} \in \mathcal{C}_{\mathrm{LRS}}$ and $w = \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) > \tau$. Then, Algorithm 1 with input $\boldsymbol{y}$ returns $\hat{\boldsymbol{c}} \in \mathcal{C}_{\mathrm{LRS}}$ s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y} - \hat{\boldsymbol{c}}) = w$ and the expected number of operations over $\mathbb{F}_{q^m}$ to output $\hat{\boldsymbol{c}} \in \mathcal{C}_{\mathrm{LRS}}$ for $u = \dim_\Sigma(\mathcal{U})$ is at most*

$$\mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})} = \frac{n^2 \ell^u}{\varphi_{\mu,u}(w)} \qquad (6)$$

with $\varphi_{\mu,u}(w)$ as in (4).

Proof: Drawing from the distribution $p_{\boldsymbol{u}}$ in Line 3 draws from the set $\mathcal{T}_{u,\ell,\mu}$ which according to the bound introduced in [12] has cardinality at most $|\mathcal{T}_{u,\ell,\mu}| \leq \binom{\ell+u-1}{\ell-1}$. For a fixed u, this means that the set size $|\mathcal{T}_{u,\ell,\mu}|$ is in $O(\ell^u)$. One iteration of Algorithm 1 (Line 8) costs $O(n^2)$ operations over $\mathbb{F}_{q^m}$ (see [25]). This means in total we have a complexity of $n^2\ell^u$ for a single iteration in Algorithm 1. Since we have that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ with $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = w$ we know that there is at least one valid codeword s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}}) \leq w$ and since $\varphi_{\mu,u}(w)$ is by definition the smallest probability over all $\boldsymbol{w}$ for the algorithm to succeed, we have that the expected number of iterations is at most $\mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})}$ as in (6). ∎

In order to evaluate $\mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})}$, the PMF $p_{\boldsymbol{u}}$ must be known for $\boldsymbol{u} \in \mathcal{T}_{u,\ell,\mu}$. Theorem 2 gives a lower and upper bound on $\mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})}$ which both do not depend on $p_{\boldsymbol{u}}$.

Theorem 2 *Let the same conditions hold as in Theorem 1 The work factor $\mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})}$ on the expected complexity can then be bounded from below and above as follows:*

$$\tilde{\mathrm{W}}_{\mathrm{RD}}^{(\mathrm{LB})} \leq \mathrm{W}_{\mathrm{RD}}^{(\mathrm{UB})} \leq \tilde{\mathrm{W}}_{\mathrm{RD}}^{(\mathrm{UB})}$$

with

$$\tilde{\mathrm{W}}_{\mathrm{RD}}^{(\mathrm{LB})} = n^2 \ell^u \cdot \frac{Q_{\mu,w,u}}{|\mathcal{T}_{w,\ell,\mu}|}$$

and

$$\tilde{\mathrm{W}}_{\mathrm{RD}}^{(\mathrm{UB})} = n^2 \ell^u \cdot Q_{\mu,w,u}$$

where

$$Q_{\mu,w,u} := \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \frac{1}{\max\limits_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \varphi_\mu(\boldsymbol{u},\boldsymbol{w})}.$$

Proof: First, define

$$\hat{\boldsymbol{u}} = \xi_{\mu,u}(\boldsymbol{w}) := \arg\max_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \varphi_\mu(\boldsymbol{u},\boldsymbol{w})$$

i.e. $\xi_{\mu,u}(\boldsymbol{w})$ returns the weight composition $\hat{\boldsymbol{u}}$ that maximizes $\varphi_\mu(\boldsymbol{u},\boldsymbol{w})$ for a given $\boldsymbol{w}$ over all $\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}$. We have that

$$\varphi_\mu(\xi_{\mu,u}(\boldsymbol{w}),\boldsymbol{w}) = \max_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \varphi_\mu(\boldsymbol{u},\boldsymbol{w}). \quad (7)$$

Consider, that instead of choosing a vector $\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}$ directly, we draw a vector $\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}$ at random according to a designed probability distribution, defined as

$$\tilde{p}_{\boldsymbol{w}} := \frac{1}{\varphi_\mu(\xi_{\mu,u}(\boldsymbol{w}),\boldsymbol{w})} \cdot Q_{\mu,w,u}^{-1}.$$

Denote by $\tilde{p}_{\boldsymbol{u}}$ the resulting probability distribution of $\boldsymbol{u}$, for a fixed error $\boldsymbol{w}_e$. By (3) we have that

$$\begin{aligned}
\varphi_{\mu,u}(\boldsymbol{w}_e) &= \sum_{\boldsymbol{u}\in\mathcal{T}_{u,\ell,\mu}} \tilde{p}_{\boldsymbol{u}}\,\varphi_\mu(\boldsymbol{u},\boldsymbol{w}_e) \\
&= \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \tilde{p}_{\boldsymbol{w}}\,\varphi_\mu(\xi_\mu(\boldsymbol{w},u),\boldsymbol{w}_e) \\
&\geq \tilde{p}_{\boldsymbol{w}_e}\,\varphi_\mu(\xi_\mu(\boldsymbol{w}_e,u),\boldsymbol{w}_e) \\
&= Q_{\mu,w,u}^{-1}.
\end{aligned}$$

The value of $Q_{\mu,w,u}$ does not depend on $\boldsymbol{w}_e$ anymore and thus holds for all $\varphi_{\mu,u}(\boldsymbol{w}_e)$ with any $\boldsymbol{w}_e\in\mathcal{T}_{w,\ell,\mu}$ and therefore $\varphi_{\mu,u}(w)\geq Q_{\mu,w,u}^{-1}$. Considering the same costs of one iteration in Algorithm 1 as in Theorem 1 proves the upper bound. By (7) and assuming that $\boldsymbol{w}_e$ is the weight composition of the worst-case error vector that minimizes (4) we have that

$$\begin{aligned}
\varphi_{\mu,u}(w) &= \varphi_{\mu,u}(\boldsymbol{w}_e) \\
&= \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \tilde{p}_{\boldsymbol{w}}\,\varphi_\mu(\xi_\mu(\boldsymbol{w},u),\boldsymbol{w}_e) \\
&\leq \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \tilde{p}_{\boldsymbol{w}}\,\varphi_\mu(\xi_\mu(\boldsymbol{w},u),\boldsymbol{w}) \\
&= \sum_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} Q_{\mu,w,u}^{-1} = |\mathcal{T}_{w,\ell,\mu}|Q_{\mu,w,u}^{-1},
\end{aligned}$$

which proves the claim for the lower bound. ∎

B. Lower Bound on the Work Factor of the Algorithm

In the previous section we obtained an upper bound on the worst-case number of iterations needed for Algorithm 1 to output a valid codeword $\hat{\boldsymbol{c}}\in\mathcal{C}_{\mathrm{LRS}}$ s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}})=w$ where we assumed $\boldsymbol{y}=\boldsymbol{c}+\boldsymbol{e}\in\mathbb{F}_{q^m}^n$ with $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e})=w$. In this setting, there is at least one codeword in distance w around the received word $\boldsymbol{y}$. Neither Problem 1 nor Problem 2 make any assumptions on the received word $\boldsymbol{y}$. Since the decoder is limited to a maximum radius w, in general there can be potentially many more solutions to our decoding problem or none at all. In this section we consider a lower bound on the number of iterations needed for Algorithm 1 to output a valid codeword $\hat{\boldsymbol{c}}\in\mathcal{C}_{\mathrm{LRS}}$ s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}})\leq w$ and we assume that $\boldsymbol{y}$ is drawn uniformly at random from $\mathbb{F}_{q^m}^n$.

Theorem 3 *Let $\boldsymbol{y}$ be uniformly drawn at random from $\mathbb{F}_{q^m}^n$. Then the average work factor of Algorithm 1 to output $\hat{\boldsymbol{c}}\in\mathcal{C}_{\mathrm{LRS}}$ s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}})\leq w$ is at least*

$$\mathrm{W}_{\mathrm{RD}}^{(\mathrm{LB})} = \frac{n^2\ell^u}{\sum_{j=0}^w \bar{A}_j\hat{\varphi}_{\mu,u}(j)}$$

with

$$\bar{A}_j := q^{m(k-n)}\,\mathrm{NSR}(m,\boldsymbol{n},j) \quad (16)$$

and

$$\hat{\varphi}_{\mu,u}(w) := \max_{\boldsymbol{w}\in\mathcal{T}_{w,\ell,\mu}} \varphi_{\mu,u}(\boldsymbol{w}). \quad (17)$$

Proof: Let $\hat{\mathcal{C}}$ be the set of codewords that have rank distance at most w from the received word, i.e.,

$$\hat{\mathcal{C}} := \left\{\boldsymbol{c}\in\mathcal{C}_{\mathrm{LRS}}: \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\boldsymbol{c})\leq w\right\} = \{\hat{\boldsymbol{c}}_1,\ldots,\hat{\boldsymbol{c}}_N\}.$$

Further, let X_i be the event that the error-erasure decoder outputs $\hat{\boldsymbol{c}}_i$ for any $i=1,\ldots,N$ and let

$$\mathcal{A}_j := \{\hat{\boldsymbol{c}}_i\in\hat{\mathcal{C}}: \mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}}_i)=j\}.$$

The probability of success over all $\boldsymbol{y}\in\mathbb{F}_{q^m}^n$ is

$$\sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n} p_{\boldsymbol{y}}\Pr\left[\bigcup_{i=1}^N X_i|\boldsymbol{y}\right] \leq \sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n} p_{\boldsymbol{y}}\sum_{i=1}^N \Pr[X_i|\boldsymbol{y}]$$

with $p_{\boldsymbol{y}} = |\mathbb{F}_{q^m}^n|^{-1} = q^{-mn}$. Denote with $\psi(\cdot)$ the function that returns the error weight composition $\boldsymbol{w}\in\mathbb{N}^\ell$ of a given error vector $\boldsymbol{e}\in\mathbb{F}_{q^m}^n$ s.t. $\boldsymbol{w}=\psi(\boldsymbol{e})$. We then have that

$$\begin{aligned}
\sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n}\sum_{i=1}^N p_{\boldsymbol{y}}\Pr[X_i|\boldsymbol{y}] &= \sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n}\sum_{i=1}^N p_{\boldsymbol{y}}\varphi_{\mu,u}(\psi(\boldsymbol{y}-\hat{\boldsymbol{c}}_i)) \\
&\leq \sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n}\sum_{i=1}^N p_{\boldsymbol{y}}\hat{\varphi}_{\mu,u}(\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}}_i)).
\end{aligned}$$

Since $\hat{\varphi}_{\mu,u}(\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}}_i))$ with $\hat{\varphi}_{\mu,u}(\cdot)$ as defined in (17) is the same for all $\hat{\boldsymbol{c}}_i\in\mathcal{A}_{\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{y}-\hat{\boldsymbol{c}}_i)}$ we have that

$$\sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n}\sum_{i=1}^N p_{\boldsymbol{y}}\Pr[X_i|\boldsymbol{y}] = \sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n}\sum_{i=0}^w p_{\boldsymbol{y}}\cdot|\mathcal{A}_i|\cdot\hat{\varphi}_{\mu,u}(i).$$

For the last step, we have that on average it holds that $\sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n} p_{\boldsymbol{y}}|\mathcal{A}_i| = \bar{A}_i$ with $\bar{A}_i$ as in (16) and thus

$$\sum_{\boldsymbol{y}\in\mathbb{F}_{q^m}^n} p_{\boldsymbol{y}}\Pr\left[\bigcup_{i=1}^N X_i|\boldsymbol{y}\right] \leq \sum_{i=0}^w \hat{\varphi}_{\mu,u}(i)\bar{A}_i.$$

The bound for the work factor then follows by considering the complexity of a single iteration divided by the probability of success. ∎

C. Finding the Optimal Drawing Distribution

Similar to [12], the problem of minimizing (5) over all distributions $p_{\boldsymbol{u}}$ on $\mathcal{T}_{u,\ell,\mu}$ can be formulated as a *linear*

program and solved numerically for small parameters ℓ, μ and u.

Theorem 4 *Let $N_u = |\mathcal{T}_{u,\ell,\mu}|$ and $N_w = |\mathcal{T}_{w,\ell,\mu}|$ and fix arbitrary orders $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_u}$ and $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{N_w}$ of all elements in $\mathcal{T}_{u,\ell,\mu}$ and $\mathcal{T}_{w,\ell,\mu}$, respectively. Further, let*

$$\boldsymbol{c} = (0, 0, \ldots, 0, 1)^\top \in \mathbb{R}^{(N_u+1)\times 1}$$
$$\boldsymbol{b} = (0, 0, \ldots, 0, 1, -1)^\top \in \mathbb{R}^{(N_w+2)\times 1}$$

and

$$\boldsymbol{A} = \begin{pmatrix} -\varphi_\mu(\boldsymbol{u}_1, \boldsymbol{w}_1) & \ldots & -\varphi_\mu(\boldsymbol{u}_{N_u}, \boldsymbol{w}_1) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ -\varphi_\mu(\boldsymbol{u}_1, \boldsymbol{w}_{N_w}) & \cdots & -\varphi_\mu(\boldsymbol{u}_{N_u}, \boldsymbol{w}_{N_w}) & 1 \\ 1 & \cdots & 1 & 0 \\ -1 & \cdots & -1 & 0 \end{pmatrix}$$

with $\boldsymbol{A} \in \mathbb{R}^{(N_w+2)\times(N_u+1)}$. If $\boldsymbol{x} = (x_1, \ldots, x_{N_u+1})$ with $\boldsymbol{x} \in \mathbb{R}^{(N_u+1)\times 1}$ is a solution to the linear program

- *Maximize $\boldsymbol{c}^\top \boldsymbol{x}$*
- *subject to $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$,*

then $\tilde{p}_{\boldsymbol{u}} = x_i$, for all $i = 1, \ldots, N_u$, is a distribution that minimizes (5) and we have

$$x_{N_u+1} = \max_{\boldsymbol{p_u}} \varphi_{\mu,u}(w) \tag{20}$$

with $\varphi_{\mu,u}(w)$ as defined in (4) and $\boldsymbol{p_u} = (p_{\boldsymbol{u}_1}, \ldots, p_{\boldsymbol{u}_{N_u}})$ with $\boldsymbol{p_u} \in [0,1]^{N_u}$ s.t. $\sum_{i=1}^{N_u} p_{\boldsymbol{u}_i} = 1$.

Proof: Let $\tilde{p}_{\boldsymbol{u}_i} = x_i$ then the last two rows of $\boldsymbol{A}$ and the last two entries of $\boldsymbol{b}$ correspond to $\sum_{i=1}^{N_u} \tilde{p}_{\boldsymbol{u}_i} = 1$. Together with $\boldsymbol{x} \geq \boldsymbol{0}$, we get that $\tilde{p}_{\boldsymbol{u}_i}$ is a valid PMF. The first N_w rows of $\boldsymbol{A}$ correspond to the constraints

$$\sum_{i=1}^{N_u} \tilde{p}_{\boldsymbol{u}_i} \varphi_\mu(\boldsymbol{u}_i, \boldsymbol{w}_j) \geq x_{N_u+1} \quad \forall j = 1, \ldots, N_w.$$

Since x_{N_u+1} is the maximal positive value for which this constraint is fulfilled for all $j = 1, \ldots, N_w$ and all solutions $\tilde{p}_{\boldsymbol{u}_i}$, we have

$$x_{N_u+1} = \max_{\boldsymbol{p_u}}\left\{ \min_{j=1,\ldots,N_w}\left\{ \sum_{i=1}^{N_u} p_{\boldsymbol{u}_i} \varphi_\mu(\boldsymbol{u}_i, \boldsymbol{w}_j) \right\} \right\}$$

which is equivalent to (20) due to the definitions in (4) and (3). ∎

The worst-case complexity using the PMF $\tilde{p}_{\boldsymbol{u}}$ obtained via the linear program is then given by

$$\mathrm{W}_{\mathrm{opt}}^{(\mathrm{UB})} := \frac{n^2 \ell^u}{x_{N_u+1}}$$

where $n^2 \ell^u$ is the approximate cost of a single iteration in Algorithm 1 as stated in Theorem 1 and $x_{N_u+1}^{-1}$ is the worst-case number of iterations using $\tilde{p}_{\boldsymbol{u}}$ as stated in Theorem 4.

V. NUMERICAL RESULTS

In this section we evaluate the tightness of the bounds on the work factor of Algorithm 1 given in Section IV. Figure 1

shows the comparison of the bounds on the worst-case number of operations over $\mathbb{F}_{q^m}$ for both the generic decoder from [12] and the proposed algorithm with the assumption, that the received word $\boldsymbol{y}$ is $\boldsymbol{y} = \boldsymbol{c}+\boldsymbol{e}$ with an error $\boldsymbol{e}$ s.t. $\mathrm{wt}_{\Sigma R}^{(\boldsymbol{n})}(\boldsymbol{e}) = w$. We also give the upper bound $\mathrm{W}_{\mathrm{opt}}^{(\mathrm{UB})}$ w.r.t. the worst-case number of iterations derived from the optimal distribution, discussed in Section IV-B. We observe that the bounds as given in Theorem 2 which can be computed without any knowledge of the distribution $p_{\boldsymbol{u}}$ of the weight composition $\boldsymbol{u}$ of the guessed supports $\mathcal{U}$ are relatively tight and the work factor for the optimal distribution $\mathrm{W}_{\mathrm{opt}}^{(\mathrm{UB})}$ lies in between those bounds.

Further, we back up the correctness of the lower bound for the scenario that $\boldsymbol{y} \overset{\$}{\leftarrow} \mathbb{F}_{q^m}^n$ by simulations. The lower bound and the simulation is shown in Figure 2 for small code parameters of $q = 11$, $n = 10$ and $k = 5$. The simulations were performed using the error-erasure decoder for LRS codes from [25] and running for a maximum samples size of 10^7 vectors $\boldsymbol{y}$ drawn uniformly at random from $\mathbb{F}_{q^m}^n$. For reference we also depict the upper bound $\mathrm{W}_{\mathrm{opt}}^{(\mathrm{UB})}$ for the optimal distribution obtained from the linear program discussed in Section IV-B as well.
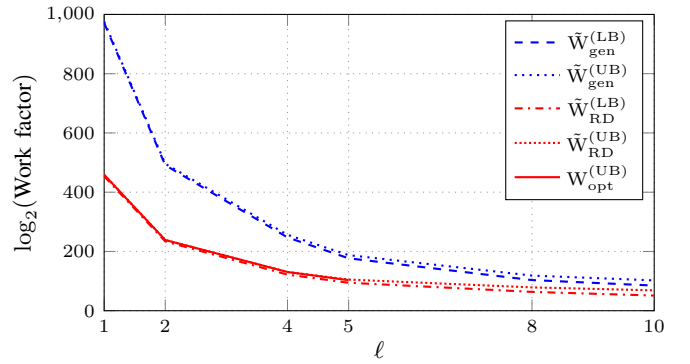

Fig. 1. For LRS codes with parameters $q = 2^4$, $n = 40$, $k = 20$, and $m = \eta = n/\ell$, with errors of weight $w = 12$ we compare the bounds for the generic decoder proposed in [12] for $s = 20$ and the bounds given in Theorem 2 with $u = 4$ for Algorithm 1.
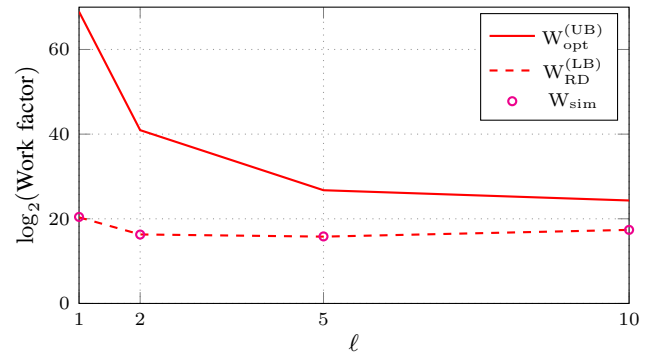

Fig. 2. Simulation results of Algorithm 1 for LRS codes with small parameters $q = 11$, $n = 10$, $k = 5$, and $m = \eta = n/\ell$, with errors of weight $w = 4$ with $u = 3$ for different values of ℓ.

VI. Conclusion

We presented a randomized decoding algorithm for LRS codes that can correct errors beyond the unique decoding radius and analyzed its theoretical expected complexity. We showed that the algorithm improves upon the generic decoding approach from [12] by exploiting the structure of the underlying LRS code. The problem of decoding LRS codes beyond their unique decoding radius has exponential complexity and thus it can be of interest to analyze future code-based cryptosystems in the sum-rank metric that are based on the hardness of decoding beyond the unique decoding radius. Future work will include a more detailed analysis of the bounds on the expected complexity of the algorithm as well as complexity analysis of the evaluation of the bounds.

References

[1] H.-f. Lu and P. V. Kumar, "A Unified Construction of Space-Time Codes with Optimal Rate-Diversity Tradeoff," *IEEE Transactions on Information Theory*, vol. 51, no. 5, pp. 1709–1730, 2005.

[2] A. Wachter, V. R. Sidorenko, M. Bossert, and V. V. Zyablov, "On (Partial) Unit Memory Codes Based on Gabidulin Codes," *Problems of Information Transmission*, vol. 47, no. 2, pp. 117–129, Jun 2011. [Online]. Available: https://doi.org/10.1134/S0032946011020049

[3] A. Wachter-Zeh and V. Sidorenko, "Rank Metric Convolutional Codes for Random Linear Network Coding," in *2012 International Symposium on Network Coding (NetCod)*, 2012, pp. 1–6.

[4] A. Wachter-Zeh, M. Stinner, and V. Sidorenko, "Convolutional Codes in Rank Metric With Application to Random Network Coding," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3199–3213, 2015.

[5] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori, "MRD Rank Metric Convolutional Codes," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2766–2770.

[6] U. Martínez-Peñas, "Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes over any Division Ring," *Journal of Algebra*, vol. 504, pp. 587–612, 2018.

[7] U. Martínez-Peñas and F. R. Kschischang, "Reliable and Secure Multishot Network Coding using Linearized Reed-Solomon Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 4785–4803, 2019.

[8] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde, "Fast Decoding of Codes in the Rank, Subspace, and Sum-Rank Metric," *IEEE Transactions on Information Theory*, vol. 67, no. 8, pp. 5026–5050, 2021.

[9] H. Bartz and S. Puchinger, "Decoding of Interleaved Linearized Reed-Solomon Codes with Applications to Network Coding," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 160–165.

[10] U. Martínez-Peñas and F. R. Kschischang, "Universal and Dynamic Locally Repairable Codes With Maximal Recoverability via Sum-Rank Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7790–7805, 2019.

[11] R. W. Nóbrega and B. F. Uchôa-Filho, "Multishot Codes for Network Coding using Rank-Metric Codes," in *2010 Third IEEE International Workshop on Wireless Network Coding*. IEEE, 2010, pp. 1–6.

[12] S. Puchinger, J. Renner, and J. Rosenkilde, "Generic Decoding in the Sum-Rank Metric," *IEEE Transactions on Information Theory*, vol. 68, no. 8, pp. 5075–5097, 2022.

[13] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the Inherent Intractability of Certain Coding Problems (Corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[14] J. Stern, "Approximating the Number of Error Locations Within a Constant Ratio is NP-complete," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, G. Cohen, T. Mora, and O. Moreno, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 325–331.

[15] A. Vardy, "The Intractability of Computing the Minimum Distance of a Code," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1757–1766, 1997.

[16] S. Puchinger and J. Rosenkilde, "Bounds on List Decoding of Linearized Reed-Solomon Codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 154–159.

[17] J. Renner, T. Jerkovits, H. Bartz, S. Puchinger, P. Loidreau, and A. Wachter-Zeh, "Randomized Decoding of Gabidulin Codes Beyond the Unique Decoding Radius," in *Post-Quantum Cryptography*, J. Ding and J.-P. Tillich, Eds. Cham: Springer International Publishing, 2020, pp. 3–19.

[18] R. Overbeck, "Structural attacks for public key cryptosystems based on gabidulin codes," *Journal of cryptology*, vol. 21, no. 2, pp. 280–301, 2008.

[19] C. Wieschebrink, "An attack on a modified Niederreiter encryption scheme," in *International Workshop on Public Key Cryptography*. Springer, 2006, pp. 14–26.

[20] V. M. Sidelnikov and S. O. Shestakov, "On insecurity of cryptosystems based on generalized Reed–Solomon codes," *Discrete Mathematics and Applications*, vol. 2, no. 4, 1992.

[21] F. Hörmann, H. Bartz, and A.-L. Horlemann, "Distinguishing and recovering generalized linearized reed–solomon codes," in *Code-Based Cryptography*, J.-C. Deneuville, Ed. Cham: Springer Nature Switzerland, 2023, pp. 1–20.

[22] C. Faure and P. Loidreau, "A New Public-Key Cryptosystem Based on the Problem of Reconstructing p-Polynomials," in *Coding and Cryptography*. Springer, 2006, pp. 304–315.

[23] A. Wachter-Zeh, S. Puchinger, and J. Renner, "Repairing the Faure-Loidreau Public-Key Cryptosystem," in *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 2426–2430.

[24] J. Renner, S. Puchinger, and A. Wachter-Zeh, "LIGA: A Cryptosystem Based on the Hardness of Rank-Metric List and Interleaved Decoding," *Designs, Codes, and Cryptography*, vol. 89, p. 1279–1319, 2021.

[25] F. Hörmann, H. Bartz, and S. Puchinger, "Error-Erasure Decoding of Linearized Reed-Solomon Codes in the Sum-Rank Metric," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 7–12.