

A Rank-Metric Approach to Error Control in Random Network Coding

Danilo Silva, *Student Member, IEEE*, Frank R. Kschischang, *Fellow, IEEE*,
and Ralf Kötter, *Senior Member, IEEE*

Abstract

The problem of error control in random linear network coding is addressed from a matrix perspective that is closely related to the subspace perspective of Kötter and Kschischang. A large class of constant-dimension subspace codes is investigated. It is shown that codes in this class can be easily constructed from rank-metric codes, while preserving their distance properties. Moreover, it is shown that minimum distance decoding of such subspace codes can be reformulated as a generalized decoding problem for rank-metric codes where partial information about the error is available. This partial information may be in the form of erasures (knowledge of an error location but not its value) and *deviations* (knowledge of an error value but not its location). Taking erasures and deviations into account (when they occur) strictly increases the error correction capability of a code: if μ erasures and δ deviations occur, then errors of rank t can always be corrected provided that $2t \leq d - 1 + \mu + \delta$, where d is the minimum rank distance of the code. For Gabidulin codes, an important family of maximum rank distance codes, an efficient decoding algorithm is proposed that can properly exploit erasures and deviations. In a network coding application where n packets of length M over \mathbb{F}_q are transmitted, the complexity of the decoding algorithm is given by $O(dM)$ operations in an extension field \mathbb{F}_{q^n} .

Index Terms

Constant-dimension codes, error correction, linearized polynomials, random network coding, rank-metric codes.

This work was supported by CAPES Foundation, Brazil, and by the Natural Sciences and Engineering Research Council of Canada. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Nice, France, June 2007 and at the IEEE Information Theory Workshop, Bergen, Norway, July 2007.

D. Silva and F. R. Kschischang are with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: danilo@comm.utoronto.ca, frank@comm.utoronto.ca).

R. Kötter is with the Institute for Communications Engineering, Technical University of Munich, D-80333 Munich, Germany (e-mail: ralf.koetter@tum.de).

I. INTRODUCTION

While random linear network coding [1]–[3] is an effective technique for information dissemination in communication networks, it is highly susceptible to errors. The insertion of even a single corrupt packet has the potential, when linearly combined with legitimate packets, to affect all packets gathered by an information receiver. The problem of error control in random network coding is therefore of great interest.

In this paper, we focus on end-to-end error control coding, where only the source and destination nodes apply error control techniques. Internal network nodes are assumed to be unaware of the presence of an outer code; they simply create outgoing packets as random linear combinations of incoming packets in the usual manner of random network coding. In addition, we assume that the source and destination nodes have no knowledge—or at least make no effort to exploit knowledge—of the topology of the network or of the particular network code used in the network. This is in contrast to the pioneering approaches [4]–[6], which have considered the design of a network code as part of the error control problem.

In the basic transmission model for end-to-end coding, the source node produces n packets, which are length- M vectors in a finite field \mathbb{F}_q , and the receiver gathers N packets. Additive packet errors may occur in any of the links. The channel equation is given by $Y = AX + BZ$, where X , Y and Z are matrices whose rows represent the transmitted, received and (possibly) corrupting packets, respectively, and A and B are the (unknown) corresponding transfer matrices induced by linear network coding.

There have been three previous quite different approaches to reliable communication under this model. In [7], Zhang characterizes the error correction capability of a network code under a brute-force decoding algorithm. He shows that network codes with good error-correcting properties exist if the field size is sufficiently large. His approach can be applied to random network coding if an extended header is included in each packet in order to allow for the matrix B (as well as A) to be estimated at a sink node. A drawback of this approach is that the extended header has size equal to the number of network edges, which may incur excessive overhead. In addition, no efficient decoding algorithm is provided for errors occurring according to an adversarial model.

Jaggi et al. [8] propose a different approach specifically targeted to combat Byzantine adversaries. They provide rate-optimal end-to-end codes that do not rely on the specific network code used and that can be decoded in polynomial time. However, their approach is based on probabilistic arguments that require both the field size and the packet length to be sufficiently large.

In contrast, Kötter and Kschischang [9] take a more combinatorial approach to the problem, which

provides correction guarantees against adversarial errors and can be used with any given field and packet size. Their key observation is that, under the unknown linear transformation applied by random network coding, the only property of the matrix X that is preserved is its row space. Thus, information should be encoded in the choice of a subspace rather than a specific matrix. The receiver observes a subspace, given by the row space of Y , which may be different from the transmitted space when packet errors occur. A metric is proposed to account for the discrepancy between transmitted and received spaces, and a new coding theory based on this metric is developed. In particular, nearly-optimal Reed-Solomon-like codes are proposed that can be decoded in $O(nM)$ operations in an extension field \mathbb{F}_{q^n} .

Although the approach in [9] seems to be the appropriate abstraction of the error control problem in random network coding, one inherent difficulty is the absence of a natural group structure on the set of all subspaces of the ambient space \mathbb{F}_q^M . As a consequence, many of the powerful concepts of classical coding theory such as group codes and linear codes do not naturally extend to codes consisting of subspaces.

In this paper, we explore the close relationship between subspace codes and codes for yet another distance measure: the rank metric. Codewords of a rank metric code are $n \times m$ matrices and the rank distance between two matrices is the rank of their difference. The rank metric was introduced in coding theory by Delsarte [10]. Codes for the rank metric were largely developed by Gabidulin [11] (see also [10], [12]). An important feature of the coding theory for the rank metric is that it supports many of the powerful concepts and techniques of classical coding theory, such as linear and cyclic codes and corresponding decoding algorithms [11]–[14].

One main contribution of this paper is to show that codes in the rank metric can be naturally “lifted” to subspace codes in such a way that the rank distance between two codewords is reflected in the subspace distance between their lifted images. In particular, nearly-optimal subspace codes can be obtained directly from optimal rank-metric codes. Conversely, when lifted rank-metric codes are used, the decoding problem for random network coding can be reformulated purely in rank-metric terms, allowing many of the tools from the theory of rank-metric codes to be applied to random network coding.

In this reformulation, we obtain a generalized decoding problem for rank-metric codes that involves not only ordinary rank errors, but also two additional phenomena that we call *erasures* and *deviations*. Erasures and deviations are dual to each other and correspond to partial information about the error matrix, akin to the role played by symbol erasures in the Hamming metric. Here, an erasure corresponds to the knowledge of an error location but not its value, while a deviation corresponds to the knowledge of an error value but not its location. These concepts generalize similar concepts found in the rank-metric literature under the terminology of “row and column erasures” [13], [15]–[18]. Although with a different

terminology, the concept of a deviation (and of a code that can correct deviations) has appeared before in [19].

Our second main contribution is an efficient decoding algorithm for rank-metric codes that takes into account erasures and deviations. Our algorithm is applicable to Gabidulin codes [11], a class of codes, analogous to conventional Reed-Solomon codes, that attain maximum distance in the rank metric. We show that our algorithm fully exploits the correction capability of Gabidulin codes; namely, it can correct any pattern of ϵ errors, μ erasures and δ deviations provided $2\epsilon + \mu + \delta \leq d - 1$, where d is the minimum rank distance of the code. Moreover, the complexity of our algorithm is $O(dM)$ operations in \mathbb{F}_{q^n} , which is smaller than that of the algorithm in [9], especially for practical high-rate codes.

In the course of setting up the problem, we also prove a result that can be seen as complementary to [9]; namely, we relate the performance guarantees of a subspace code with more concrete network parameters such as the maximum number of corrupting packets that can be injected in the network. This result provides a tighter connection between the subspace approach of [9] and previous approaches that deal with link errors.

The remainder of this paper is organized as follows. In Section II, we provide a brief review of rank-metric codes and subspace codes. In Section III, we describe in more detail the problem of error control in random network coding, along with Kötter and Kschischang's approach to this problem. In Section IV, we present our code construction and show that the resulting error control problem can be replaced by a generalized decoding problem for rank-metric codes. At this point, we turn our attention entirely to rank-metric codes. The generalized decoding problem that we introduce is developed in more detail in Section V, wherein the concepts of erasures and deviations are described and compared to related concepts in the rank-metric literature. In Section VI, we present an efficient algorithm for decoding Gabidulin codes in the presence of errors, erasures and deviations. Finally, Section VII contains our conclusions.

II. PRELIMINARIES

A. Notation

Let $q \geq 2$ be a power of a prime. In this paper, all vectors and matrices have components in the finite field \mathbb{F}_q , unless otherwise mentioned. We use $\mathbb{F}_q^{n \times m}$ to denote the set of all $n \times m$ matrices over \mathbb{F}_q and we set $\mathbb{F}_q^n = \mathbb{F}_q^{n \times 1}$. In particular, $v \in \mathbb{F}_q^n$ is a column vector and $v \in \mathbb{F}_q^{1 \times m}$ is a row vector.

If v is a vector, then the symbol v_i denotes the i th entry of v . If A is a matrix, then the symbol A_i denotes either the i th row or the i th column of A ; the distinction will always be clear from the way

in which A is defined. In either case, the symbol A_{ij} always refers to the entry in the i th row and j th column of A .

For clarity, the $k \times k$ identity matrix is denoted by $I_{k \times k}$. If we set $I = I_{n \times n}$, then the notation I_i will denote the i th column of I . More generally, if $\mathcal{U} \subseteq \{1, \dots, n\}$, then $I_{\mathcal{U}} = [I_i, i \in \mathcal{U}]$ will denote the sub-matrix of I consisting of the columns indexed by \mathcal{U} .

The linear span of a set of vectors v_1, \dots, v_k is denoted by $\langle v_1, \dots, v_k \rangle$. The row space, the rank and the number of nonzero rows of a matrix X are denoted by $\langle X \rangle$, $\text{rank } X$ and $\text{wt}(X)$, respectively. The reduced row echelon (RRE) form of a matrix X is denoted by $\text{RRE}(X)$.

B. Properties of Matrix Rank and Subspace Dimension

Let $X \in \mathbb{F}_q^{n \times m}$. By definition, $\text{rank } X = \dim \langle X \rangle$; however, there are many useful equivalent characterizations. For example, $\text{rank } X$ is the smallest r for which there exist matrices $A \in \mathbb{F}_q^{n \times r}$ and $B \in \mathbb{F}_q^{r \times m}$ such that $X = AB$, i.e.,

$$\text{rank } X = \min_{\substack{r, A \in \mathbb{F}_q^{n \times r}, B \in \mathbb{F}_q^{r \times m}: \\ X = AB}} r. \quad (1)$$

It is well-known that, for any $X, Y \in \mathbb{F}_q^{n \times m}$, we have

$$\text{rank}(X + Y) \leq \text{rank } X + \text{rank } Y \quad (2)$$

and that, for $X \in \mathbb{F}_q^{n \times m}$ and $A \in \mathbb{F}_q^{N \times n}$, we have

$$\text{rank}(AX) \geq \text{rank } A + \text{rank } X - n. \quad (3)$$

Recall that if U and V are subspaces of some fixed vector space, then the sum

$$U + V = \{u + v : u \in U, v \in V\}$$

is the smallest subspace that contains both U and V . Recall also that

$$\dim(U + V) = \dim U + \dim V - \dim(U \cap V). \quad (4)$$

We will make extensive use of the fact that

$$\left\langle \begin{bmatrix} X \\ Y \end{bmatrix} \right\rangle = \langle X \rangle + \langle Y \rangle \quad (5)$$

and therefore

$$\begin{aligned} \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} &= \dim(\langle X \rangle + \langle Y \rangle) \\ &= \text{rank } X + \text{rank } Y - \dim(\langle X \rangle \cap \langle Y \rangle). \end{aligned} \quad (6)$$

C. Rank-Metric Codes

A *matrix code* is defined as any nonempty subset of $\mathbb{F}_q^{n \times m}$. A matrix code is also commonly known as an *array code* when it forms a linear space over \mathbb{F}_q [12].

A natural and useful distance measure between elements of $\mathbb{F}_q^{n \times m}$ is given in the following definition.

Definition 1: For $X, Y \in \mathbb{F}_q^{n \times m}$, the *rank distance* between X and Y is defined as $d_r(X, Y) \triangleq \text{rank}(Y - X)$.

As observed in [11], rank distance is indeed a *metric*. In particular, the triangle inequality for the rank metric follows directly from (2). In the context of the rank metric, a matrix code is called a *rank-metric code*. The minimum (rank) distance of a rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ is defined as

$$d_r(\mathcal{C}) \triangleq \min_{\substack{\mathbf{x}, \mathbf{x}' \in \mathcal{C} \\ \mathbf{x} \neq \mathbf{x}'}} d_r(\mathbf{x}, \mathbf{x}').$$

Associated with every rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ is the *transposed code* $\mathcal{C}^T \subseteq \mathbb{F}_q^{m \times n}$, whose codewords are obtained by transposing the codewords of \mathcal{C} , i.e., $\mathcal{C}^T = \{\mathbf{x}^T : \mathbf{x} \in \mathcal{C}\}$. We have $|\mathcal{C}^T| = |\mathcal{C}|$ and $d_r(\mathcal{C}^T) = d_r(\mathcal{C})$. Observe the symmetry between rows and columns in the rank metric; the distinction between a code and its transpose is in fact transparent to the metric.

A minimum distance decoder for a rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ takes a word $\mathbf{r} \in \mathbb{F}_q^{n \times m}$ and returns a codeword $\hat{\mathbf{x}} \in \mathcal{C}$ that is closest to \mathbf{r} in rank distance, that is,

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{C}}{\text{argmin}} \text{rank}(\mathbf{r} - \mathbf{x}). \quad (7)$$

Note that if $d_r(\mathbf{x}, \mathbf{r}) < d_r(\mathcal{C})/2$, then a minimum distance decoder is guaranteed to return $\hat{\mathbf{x}} = \mathbf{x}$.

Throughout this paper, problem (7) will be referred to as the *conventional* rank decoding problem.

There is a rich coding theory for rank-metric codes that is analogous to the classical coding theory in the Hamming metric. In particular, we mention the existence of a Singleton bound [10], [11] (see also [20] [21]), which states that every rank metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ with minimum distance $d = d_r(\mathcal{C})$ must satisfy

$$\begin{aligned} \log_q |\mathcal{C}| &\leq \min\{n(m - d + 1), m(n - d + 1)\} \\ &= \max\{n, m\}(\min\{n, m\} - d + 1). \end{aligned} \quad (8)$$

Codes that achieve this bound are called *maximum-rank-distance* (MRD) codes. An extensive class of MRD codes with $n \leq m$ was presented by Gabidulin in [11]. By transposition, MRD codes with $n > m$ can also be obtained. Thus, MRD codes exist for all n and m and all $d \leq \min\{n, m\}$, irrespectively of the field size q .

D. Subspace Codes

Let $\mathcal{P}(\mathbb{F}_q^M)$ denote the set of all subspaces of \mathbb{F}_q^M . We review some concepts of the coding theory for subspaces developed in [9].

Definition 2: Let $V, V' \in \mathcal{P}(\mathbb{F}_q^M)$. The *subspace distance* between V and V' is defined as

$$\begin{aligned} d_s(V, V') &\triangleq \dim(V + V') - \dim(V \cap V') \\ &= 2 \dim(V + V') - \dim V - \dim V' \end{aligned} \quad (9)$$

$$= \dim V + \dim V' - 2 \dim(V \cap V'). \quad (10)$$

It is shown in [9] that the subspace distance is indeed a metric on $\mathcal{P}(\mathbb{F}_q^M)$.

A *subspace code* is defined as a nonempty subset of $\mathcal{P}(\mathbb{F}_q^M)$. The minimum (subspace) distance of a subspace code $\Omega \subseteq \mathcal{P}(\mathbb{F}_q^M)$ is defined as

$$d_s(\Omega) \triangleq \min_{\substack{V, V' \in \Omega \\ V \neq V'}} d_s(V, V').$$

The minimum distance decoding problem for a subspace code is to find a subspace $\hat{V} \in \Omega$ that is closest to a given subspace $U \in \mathcal{P}(\mathbb{F}_q^M)$, i.e.,

$$\hat{V} = \operatorname{argmin}_{V \in \Omega} d_s(V, U). \quad (11)$$

A minimum distance decoder is guaranteed to return $\hat{V} = V$ if $d_s(V, U) < d_s(\Omega)/2$.

Let $\mathcal{P}(\mathbb{F}_q^M, n)$ denote the set of all n -dimensional subspaces of \mathbb{F}_q^M . A subspace code Ω is called a constant-dimension code if $\Omega \subseteq \mathcal{P}(\mathbb{F}_q^M, n)$. It follows from (9) or (10) that the minimum distance of a constant-dimension code is always an even number.

Let $A_q[M, 2d, n]$ be denote the maximum number of codewords in a constant-dimension code with minimum subspace distance $2d$. Many bounds on $A_q[M, 2d, n]$ were developed in [9], in particular the Singleton-like bound

$$A_q[M, 2d, n] \leq \left[\begin{array}{c} M - d + 1 \\ \max\{n, M - n\} \end{array} \right]_q \quad (12)$$

where

$$\left[\begin{array}{c} M \\ n \end{array} \right]_q \triangleq \frac{(q^M - 1) \cdots (q^{M-n+1} - 1)}{(q^n - 1) \cdots (q - 1)}$$

denotes the *Gaussian coefficient*. It is well known that the Gaussian coefficient gives the number of distinct n -dimensional subspaces of an M -dimensional vector space over \mathbb{F}_q , i.e., $\left[\begin{array}{c} M \\ n \end{array} \right]_q = |\mathcal{P}(\mathbb{F}_q^M, n)|$.

A useful bound on $\begin{bmatrix} M \\ n \end{bmatrix}_q$ is given by [9, Lemma 5]

$$\begin{bmatrix} M \\ n \end{bmatrix}_q < 4q^{n(M-n)}. \quad (13)$$

Combining (12) and (13) gives

$$A_q[M, 2d, n] < 4q^{\max\{n, M-n\}(\min\{n, M-n\}-d+1)}. \quad (14)$$

There exist also bounds on $A_q[M, 2d, n]$ that are tighter than (12), namely the Wang-Xing-Safavi-Naini bound [22] and a Johnson-type bound [23].

For future reference, we define the *sub-optimality* of a constant-dimension code $\Omega \subseteq \mathcal{P}(\mathbb{F}_q^M, n)$ with $d_s(\Omega) = 2d$ to be

$$\alpha(\Omega) \triangleq \frac{\log_q A_q[M, 2d, n] - \log_q |\Omega|}{\log_q A_q[M, 2d, n]}. \quad (15)$$

III. ERROR CONTROL IN RANDOM NETWORK CODING

A. Channel Model

We start by reviewing the basic model for single-source generation-based random linear network coding [2], [3]. Consider a point-to-point communication network with a single source node and a single destination node. Each link in the network is assumed to transport, free of errors, a packet of M symbols in a finite field \mathbb{F}_q . Links are directed, incident *from* the node transmitting the packet and incident *to* the node receiving the packet. A packet transmitted on a link incident to a given node is said to be an *incoming packet* for that node, and similarly a packet transmitted on a link incident from a given node is said to be an *outgoing packet* for that node.

During each transmission generation, the source node formats the information to be transmitted into n packets $X_1, \dots, X_n \in \mathbb{F}_q^{1 \times M}$, which are regarded as incoming packets for the source node. Whenever a node (including the source) has a transmission opportunity, it produces an outgoing packet as a random \mathbb{F}_q -linear combination of all the incoming packets it has until then received. The destination node collects N packets $Y_1, \dots, Y_N \in \mathbb{F}_q^{1 \times M}$ and tries to recover the original packets X_1, \dots, X_n .

Let X be an $n \times M$ matrix whose rows are the transmitted packets X_1, \dots, X_n and, similarly, let Y be an $N \times M$ matrix whose rows are the received packets Y_1, \dots, Y_N . Since all packet operations are linear over \mathbb{F}_q , then, regardless of the network topology, the transmitted packets X and the received packets Y can be related as

$$Y = AX, \quad (16)$$

where A is an $N \times n$ matrix corresponding to the overall linear transformation applied by the network.

Before proceeding, we remark that this model encompasses a variety of situations:

- The network may have cycles or delays. Since the overall system is linear, expression (16) will be true regardless of the network topology.
- The network could be wireless instead of wired. Broadcast transmissions in wireless networks may be modeled by constraining each intermediate node to send exactly the same packet on each of its outgoing links.
- The source node may transmit more than one *generation* (a set of n packets). In this case, we assume that each packet carries a label identifying the generation to which it corresponds and that packets from different generations are processed separately throughout the network [2].
- The network topology may be time-varying as nodes join and leave and connections are established and lost. In this case, we assume that each network link is the instantiation of an actual successful packet transmission.
- The network may be used for multicast, i.e., there may be more than one destination node. Again, expression (16) applies; however, the matrix A may be different for each destination.

Let us now extend this model to incorporate packet errors. Following [4]–[6], we consider that packet errors may occur in any of the links of the network. Suppose the links in the network are indexed from 1 to ℓ , and let Z_i denote the error packet applied at link $i \in \{1, \dots, \ell\}$. The application of an error packet is modeled as follows. We assume that, for each link i , the node transmitting on that link first creates a prescribed packet $P_{\text{in},i} \in \mathbb{F}_q^{1 \times M}$ following the procedure described above. Then, an error packet $Z_i \in \mathbb{F}_q^{1 \times M}$ is added to $P_{\text{in},i}$ in order to produce the outgoing packet on this link, i.e., $P_{\text{out},i} = P_{\text{in},i} + Z_i$. Note that any arbitrary packet $P_{\text{out},i}$ can be formed simply by choosing $Z_i = P_{\text{out},i} - P_{\text{in},i}$.

Let Z be an $\ell \times M$ matrix whose rows are the error packets Z_1, \dots, Z_ℓ . By linearity of the network, we can write

$$Y = AX + BZ, \tag{17}$$

where B is an $N \times \ell$ matrix corresponding to the overall linear transformation applied to Z_1, \dots, Z_ℓ on route to the destination. Note that $Z_i = 0$ means that no corrupt packet was injected at link i . Thus, the number of nonzero rows of Z , $\text{wt}(Z)$, gives the total number of (potentially) corrupt packets injected in the network. Note that it is possible that a nonzero error packet happens to be in the row space of X , in which case it is not really a corrupt packet.

Observe that this model can represent not only the occurrence of random link errors, but also the action of malicious nodes. A malicious node can potentially transmit erroneous packets on all of its

outgoing links. A malicious node may also want to disguise itself and transmit correct packets on some of these links, or may simply refuse to transmit some packet (i.e., transmitting an all-zero packet), which is represented in the model by setting $Z_i = -P_{\text{in},i}$. In any case, $\text{wt}(Z)$ gives the total number of “packet interventions” performed by all malicious nodes and thus gives a sense of the total adversarial “power” employed towards jamming the network.

Equation (17) is our basic model of a channel induced by random linear network coding, and we will refer to it as the *random linear network coding channel* (RLNCC). The channel input and output alphabets are given by $\mathbb{F}_q^{n \times M}$ and $\mathbb{F}_q^{N \times M}$, respectively. To give a full probabilistic specification of the channel, we would need to specify the joint probability distribution of A , B and Z given X . We will not pursue this path in this paper, taking, instead, a more combinatorial approach.

B. Transmission via Subspace Selection

Let $\Omega \subseteq \mathcal{P}(\mathbb{F}_q^M)$ be a subspace code with maximum dimension n . In the approach in [9], the source node selects a subspace $V \in \Omega$ and transmits this subspace over the RLNCC as some matrix $X \in \mathbb{F}_q^{n \times M}$ such that $V = \langle X \rangle$. The destination node receives $Y \in \mathbb{F}_q^{N \times M}$ and computes $U = \langle Y \rangle$, from which the transmitted subspace can be inferred using a minimum distance decoder (11).

In this paper, it will be convenient to view the above approach from a matrix perspective. In order to do that, we simply replace Ω by an (arbitrarily chosen) matrix code that generates Ω . More precisely, let $[\Omega] \triangleq \{X \in \mathbb{F}_q^{n \times M} : X = \text{RRE}(X), \langle X \rangle \in \Omega\}$ be a matrix code consisting of all the $n \times M$ matrices in RRE form whose row space is in Ω . Now, the above setup can be reinterpreted as follows. The source node selects a matrix $X \in [\Omega]$ to transmit over the RLNCC. Upon reception of Y , the destination node tries to infer the transmitted matrix using the minimum distance decoding rule

$$\hat{X} = \underset{X \in [\Omega]}{\text{argmin}} d_s(\langle X \rangle, \langle Y \rangle). \quad (18)$$

Note that the decoding is guaranteed to be successful if $d_s(\langle X \rangle, \langle Y \rangle) < d_s(\Omega)/2$.

C. Performance Guarantees

In this subsection, we wish to relate the performance guarantees of a subspace code with more concrete network parameters. Still, we would like these parameters to be sufficiently general so that we do not need to take the whole network topology into account.

We make the following assumptions:

- The column-rank deficiency of the transfer matrix A is never greater than ρ , i.e., $\text{rank } A \geq n - \rho$.

- The adversarial nodes together can inject at most t corrupting packets, i.e., $\text{wt}(Z) \leq t$.

The following result characterizes the performance guarantees of a subspace code under our assumptions.

Theorem 1: Suppose $\text{rank } A \geq n - \rho$ and $\text{wt}(Z) \leq t$. Then, decoding according to (18) is guaranteed to be successful provided $2t + \rho < d_s(\Omega)/2$.

In order to prove Theorem 1, we need a few results relating rank and subspace distance.

Proposition 2: Let $X, Y \in \mathbb{F}_q^{N \times M}$. Then

$$\text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} \leq \text{rank}(Y - X) + \min\{\text{rank } X, \text{rank } Y\}.$$

Proof: We have

$$\begin{aligned} \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} &= \text{rank} \begin{bmatrix} X \\ Y - X \end{bmatrix} \leq \text{rank}(Y - X) + \text{rank } X \\ \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} &= \text{rank} \begin{bmatrix} Y - X \\ Y \end{bmatrix} \leq \text{rank}(Y - X) + \text{rank } Y. \end{aligned}$$

■

Corollary 3: Let $X, Z \in \mathbb{F}_q^{N \times M}$ and $Y = X + Z$. Then

$$d_s(\langle X \rangle, \langle Y \rangle) \leq 2 \text{rank } Z - |\text{rank } X - \text{rank } Y|.$$

Proof: From Proposition 2, we have

$$\begin{aligned} d_s(\langle X \rangle, \langle Y \rangle) &= 2 \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} - \text{rank } X - \text{rank } Y \\ &\leq 2 \text{rank } Z + 2 \min\{\text{rank } X, \text{rank } Y\} \\ &\quad - \text{rank } X - \text{rank } Y \\ &= 2 \text{rank } Z - |\text{rank } X - \text{rank } Y|. \end{aligned}$$

■

We can now give a proof of Theorem 1.

Proof of Theorem 1: From Corollary 3, we have that

$$d_s(\langle AX \rangle, \langle Y \rangle) \leq 2 \text{rank } BZ \leq 2 \text{rank } Z \leq 2 \text{wt}(Z) \leq 2t.$$

Using (3), we find that

$$d_s(\langle X \rangle, \langle AX \rangle) = \text{rank } X - \text{rank } AX \leq n - \text{rank } A \leq \rho.$$

Since $d_s(\cdot, \cdot)$ satisfies the triangle inequality, we have

$$\begin{aligned} d_s(\langle X \rangle, \langle Y \rangle) &\leq d_s(\langle X \rangle, \langle AX \rangle) + d_s(\langle AX \rangle, \langle Y \rangle) \\ &\leq \rho + 2t \\ &< \frac{d_s(\Omega)}{2} \end{aligned}$$

and therefore the decoding is guaranteed to be successful. ■

Theorem 1 is analogous to Theorem 2 in [9], which states that minimum subspace distance decoding is guaranteed to be successful if $2(\mu + \delta) < d_s(\Omega)$, where δ and μ are, respectively, the number of “insertions” and “deletions” of dimensions that occur in the channel [9]. Intuitively, since one corrupted packet injected at a network min-cut can effectively replace a dimension of the transmitted subspace, we see that t corrupted packets can cause t deletions and t insertions of dimensions. Combined with possible ρ further deletions caused by a row-rank deficiency of A , we have that $\delta = t$ and $\mu = t + \rho$. Thus,

$$\delta + \mu < \frac{d_s(\Omega)}{2} \implies 2t + \rho < \frac{d_s(\Omega)}{2}.$$

In other words, under the condition that corrupt packets may be injected in any of the links in network (which must be assumed if we do not wish to take the network topology into account), the performance guarantees of a minimum distance decoder are essentially given by Theorem 1.

It is worth to mention that, according to recent results [24], minimum subspace distance decoding may not be the optimal decoding rule when the subspaces in Ω have different dimensions. For the remainder of this paper, however, we focus on the case of a constant-dimension code and therefore we use the minimum distance decoding rule (18). Our goal will be to construct constant-dimension subspace codes with good performance and efficient encoding/decoding procedures.

IV. CODES FOR THE RANDOM LINEAR NETWORK CODING CHANNEL BASED ON RANK-METRIC CODES

In this section, we show how a constant-dimension subspace code can be constructed from any rank-metric code. In particular, this construction will allow us to obtain nearly-optimal subspace codes that possess efficient encoding and decoding algorithms.

A. Lifting Construction

From now on, assume that $M = n + m$, where $m > 0$. Let $I = I_{n \times n}$.

Definition 3: Let $\mathcal{I}: \mathbb{F}_q^{n \times m} \rightarrow \mathcal{P}(\mathbb{F}_q^{n+m})$, given by $\mathbf{x} \mapsto \mathcal{I}(\mathbf{x}) = \left\langle \begin{bmatrix} I & \mathbf{x} \end{bmatrix} \right\rangle$. The subspace $\mathcal{I}(\mathbf{x})$ is called the *lifting* of the matrix \mathbf{x} . Similarly, if $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ is a rank-metric code, then the subspace code $\mathcal{I}(\mathcal{C})$, obtained by lifting each codeword of \mathcal{C} , is called the *lifting* of \mathcal{C} .

Definition 3 provides an injective mapping between rank-metric codes and subspace codes. Note that a subspace code constructed by lifting is always a constant-dimension code (with codeword dimension n).

Although the lifting construction is a particular way of constructing subspace codes, it can also be seen as a generalization of the standard approach to random network coding [2], [3]. In the latter, every transmitted matrix has the form $X = \begin{bmatrix} I & \mathbf{x} \end{bmatrix}$, where the payload matrix $\mathbf{x} \in \mathbb{F}_q^{n \times m}$ corresponds to the raw data to be communicated. In our approach, each transmitted matrix is also of the form $X = \begin{bmatrix} I & \mathbf{x} \end{bmatrix}$, but the payload matrix $\mathbf{x} \in \mathcal{C}$ is restricted to be a codeword of a rank-metric code rather than uncoded data.

Our reasons for choosing \mathcal{C} to be a rank-metric code will be made clear from the following proposition.

Proposition 4: Let $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ and $\mathbf{x}, \mathbf{x}' \in \mathcal{C}$. Then

$$d_s(\mathcal{I}(\mathbf{x}), \mathcal{I}(\mathbf{x}')) = 2d_r(\mathbf{x}, \mathbf{x}')$$

$$d_s(\mathcal{I}(\mathcal{C})) = 2d_r(\mathcal{C}).$$

Proof: Since $\dim \mathcal{I}(\mathbf{x}) = \dim \mathcal{I}(\mathbf{x}') = n$, we have

$$\begin{aligned} d_s(\mathcal{I}(\mathbf{x}), \mathcal{I}(\mathbf{x}')) &= 2 \dim(\mathcal{I}(\mathbf{x}) + \mathcal{I}(\mathbf{x}')) - 2n \\ &= 2 \operatorname{rank} \begin{bmatrix} I & \mathbf{x} \\ I & \mathbf{x}' \end{bmatrix} - 2n \\ &= 2 \operatorname{rank} \begin{bmatrix} I & \mathbf{x} \\ 0 & \mathbf{x}' - \mathbf{x} \end{bmatrix} - 2n \\ &= 2 \operatorname{rank}(\mathbf{x}' - \mathbf{x}). \end{aligned}$$

The second statement is immediate. ■

Proposition 4 shows that a subspace code constructed by lifting inherits the distance properties of its underlying rank-metric code. The question of whether such lifted rank-metric codes are “good” compared to the whole class of constant-dimension codes is addressed in the following proposition.

Proposition 5: Let $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ be an MRD code with $d_r(\mathcal{C}) = d$. Then $d_s(\mathcal{I}(\mathcal{C})) = 2d$ and

$$A_q[n + m, 2d, n] < 4|\mathcal{I}(\mathcal{C})| = 4|\mathcal{C}|.$$

Moreover, for any code parameters, the sub-optimality of $\mathcal{I}(\mathcal{C})$ in $\mathcal{P}(\mathbb{F}_q^{n+m}, n)$ satisfies

$$\alpha(\mathcal{I}(\mathcal{C})) < \frac{4}{(n + m) \log_2 q}.$$

Proof: Using (14) and the fact that \mathcal{C} achieves the Singleton bound for rank-metric codes (8), we have

$$\begin{aligned} A_q[n + m, 2d, n] &< 4q^{\max\{n, m\}(\min\{n, m\} - d + 1)} \\ &= 4|\mathcal{C}|. \end{aligned}$$

Applying this result in (15), we obtain

$$\begin{aligned} \alpha(\mathcal{I}(\mathcal{C})) &< \frac{\log_q 4}{\max\{n, m\}(\min\{n, m\} - d + 1)} \\ &\leq \frac{\log_q 4}{\max\{n, m\}} \\ &\leq \frac{\log_q 4}{(n + m)/2} \\ &= \frac{4}{(n + m) \log_2 q}. \end{aligned}$$

■

Proposition 5 shows that, for all practical purposes, lifted MRD codes are essentially optimal as constant-dimension codes. Indeed, the rate loss in using a lifted MRD code rather than an optimal constant-dimension code is smaller than $4/P$, where $P = (n + m) \log_2 q$ is the packet size in bits. In particular, for packet sizes of 50 bytes or more, the rate loss is smaller than 1%.

In this context, it is worth mentioning that the nearly-optimal Reed-Solomon-like codes proposed in [9] correspond exactly to the lifting of the class of MRD codes proposed by Gabidulin [11]. The latter will be discussed in more detail in Section VI.

B. Decoding

We now specialize the decoding problem (18) to the specific case of lifted rank-metric codes. We will see that it is possible to reformulate such a problem in a way that resembles the conventional decoding problem for rank-metric codes, but with additional side-information presented to the decoder.

Let the transmitted matrix be given by $X = [I \ x]$, where $x \in \mathcal{C}$ and $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ is a rank-metric code. Write the received matrix as

$$Y = [\hat{A} \ y]$$

where $\hat{A} \in \mathbb{F}_q^{N \times n}$ and $y \in \mathbb{F}_q^{N \times m}$. In accordance with the formulation of Section III-B, we assume that $\text{rank } Y = N$, since any linearly dependent received packets do not affect the decoding problem and may be discarded by the destination node. Now, define

$$\mu \triangleq n - \text{rank } \hat{A} \quad \text{and} \quad \delta \triangleq N - \text{rank } \hat{A}.$$

Here μ measures the rank deficiency of \hat{A} with respect to columns, while δ measures the rank deficiency of \hat{A} with respect to rows.

Before examining the general problem, we study the simple special case that arises when $\mu = \delta = 0$.

Proposition 6: If $\mu = \delta = 0$, then

$$d_s(\langle X \rangle, \langle Y \rangle) = 2d_r(x, r)$$

where $r = \hat{A}^{-1}y$.

Proof: Since $\mu = \delta = 0$, \hat{A} is invertible. Thus, $\bar{Y} = [I \ \hat{A}^{-1}y]$ is row equivalent to Y , i.e., $\langle \bar{Y} \rangle = \langle Y \rangle$. Applying Proposition 4, we get the desired result. ■

The above proposition shows that, whenever \hat{A} is invertible, a solution to (18) can be found by solving the conventional rank decoding problem. This case is illustrated by the following example.

Example 1: Let $n = 4$ and $q = 5$. Let x_1, \dots, x_4 denote the rows of a codeword $x \in \mathcal{C}$. Suppose that

$$A = \begin{bmatrix} 2 & 4 & 2 & 4 \\ 0 & 0 & 3 & 3 \\ 1 & 0 & 4 & 3 \\ 0 & 4 & 1 & 4 \end{bmatrix},$$

$B = \begin{bmatrix} 4 & 0 & 1 & 0 \end{bmatrix}^T$ and $Z = \begin{bmatrix} 1 & 2 & 3 & 4 & z \end{bmatrix}$. Then

$$Y = \begin{bmatrix} 1 & 2 & 4 & 0 & 2x_1 + 4x_2 + 2x_3 + 4x_4 + 4z \\ 0 & 0 & 3 & 3 & 3x_3 + 3x_4 \\ 2 & 2 & 2 & 2 & x_1 + 4x_3 + 3x_4 + z \\ 0 & 4 & 1 & 4 & 4x_2 + x_3 + 4x_4 \end{bmatrix}.$$

Converting Y to RRE form, we obtain

$$\bar{Y} = \begin{bmatrix} I & r \end{bmatrix} \tag{19}$$

where

$$\mathbf{r} = \begin{bmatrix} 3x_2 + 2x_3 + x_4 + z \\ 3x_1 + 2x_2 + 4x_3 + 2x_4 + 2z \\ 4x_1 + 3x_2 + 3x_3 + x_4 + z \\ x_1 + 2x_2 + 3x_3 + 4z \end{bmatrix}.$$

Note that, if no errors had occurred, we would expect to find $\mathbf{r} = \mathbf{x}$.

Now, observe that we can write

$$\mathbf{r} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 1 \\ 4 \end{bmatrix} \begin{bmatrix} 4x_1 + 3x_2 + 2x_3 + x_4 + z \end{bmatrix}.$$

Thus, $\text{rank}(\mathbf{r} - \mathbf{x}) = 1$. We can think of this as an error word $\mathbf{e} = \mathbf{r} - \mathbf{x}$ of rank 1 applied to \mathbf{x} . This error can be corrected if $d_r(\mathcal{C}) \geq 3$. ■

Let us now proceed to the general case, where \hat{A} is not necessarily invertible. We first examine a relatively straightforward approach that, however, leads to an unattractive decoding problem.

Similarly to the proof of Proposition 6, it is possible to show that

$$d_s(\langle X \rangle, \langle Y \rangle) = 2 \text{rank}(\mathbf{y} - \hat{A}\mathbf{x}) + \mu - \delta$$

which yields the following decoding problem:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{C}}{\text{argmin}} \text{rank}(\mathbf{y} - \hat{A}\mathbf{x}). \quad (20)$$

If we define a new code $\mathcal{C}' = \hat{A}\mathcal{C} = \{\hat{A}\mathbf{x}, \mathbf{x} \in \mathcal{C}\}$, then a solution to (20) can be found by first solving

$$\hat{\mathbf{x}}' = \underset{\mathbf{x}' \in \mathcal{C}'}{\text{argmin}} \text{rank}(\mathbf{y} - \mathbf{x}')$$

using a conventional rank decoder for \mathcal{C}' and then choosing any $\hat{\mathbf{x}} \in \{\mathbf{x} \mid \hat{A}\mathbf{x} = \hat{\mathbf{x}}'\}$ as a solution. An obvious drawback of this approach is that it requires a new code \mathcal{C}' to be used at each decoding instance. This is likely to increase the decoding complexity, since the existence of an efficient algorithm for \mathcal{C} does not imply the existence of an efficient algorithm for $\mathcal{C}' = \hat{A}\mathcal{C}$ for all \hat{A} . Moreover, even if efficient algorithms are known for all \mathcal{C}' , running a different algorithm for each received matrix may be impractical or undesirable from an implementation point-of-view.

In the following, we seek an expression for $d_s(\langle X \rangle, \langle Y \rangle)$ where the structure of \mathcal{C} can be exploited. In order to motivate our approach, we consider the following two examples, which generalize Example 1.

Example 2: Let us return to Example 1, but now suppose

$$A = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 1 & 3 & 0 & 3 \\ 1 & 4 & 0 & 3 \\ 2 & 0 & 4 & 0 \\ 1 & 1 & 2 & 4 \end{bmatrix},$$

$B = \begin{bmatrix} 4 & 0 & 1 & 0 & 0 \end{bmatrix}^T$ and $Z = \begin{bmatrix} 1 & 2 & 3 & 4 & z \end{bmatrix}$. Then

$$Y = \begin{bmatrix} 0 & 3 & 4 & 4 & x_1 + 2x_3 + 3x_4 + 4z \\ 1 & 3 & 0 & 3 & x_1 + 3x_2 + 3x_4 \\ 2 & 1 & 3 & 2 & x_1 + 4x_2 + 3x_4 + z \\ 2 & 0 & 4 & 0 & 2x_1 + 4x_3 \\ 1 & 1 & 2 & 4 & x_1 + x_2 + 2x_3 + 4x_4 \end{bmatrix} = \begin{bmatrix} \hat{A} & \mathbf{y} \end{bmatrix}.$$

Although \hat{A} is not invertible, we can nevertheless convert Y to RRE form to obtain

$$\bar{Y} = \begin{bmatrix} I & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \quad (21)$$

where

$$\mathbf{r} = \begin{bmatrix} 2x_1 + 2x_2 + 3x_3 + 4x_4 + 4z \\ 4x_1 + 4x_2 + 2x_3 + x_4 + z \\ 2x_1 + 4x_2 + 2x_3 + 3x_4 + 3z \\ 3x_1 + x_2 + 4x_3 + 3x_4 + 2z \end{bmatrix}$$

and

$$\hat{E} = 2x_1 + 4x_2 + x_3 + 3x_4 + 3z.$$

Observe that

$$\mathbf{e} = \mathbf{r} - \mathbf{x} = \begin{bmatrix} x_1 + 2x_2 + 3x_3 + 4x_4 + 4z \\ 4x_1 + 3x_2 + 2x_3 + x_4 + z \\ 2x_1 + 4x_2 + x_3 + 3x_4 + 3z \\ 3x_1 + x_2 + 4x_3 + 2x_4 + 2z \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix} \hat{E}.$$

Thus, we see not only that $\text{rank } \mathbf{e} = 1$, but we have also recovered part of its decomposition as an outer product, namely, the vector \hat{E} . ■

Example 3: Consider again the parameters of Example 1, but now let

$$A = \begin{bmatrix} 3 & 2 & 1 & 1 \\ 0 & 4 & 3 & 2 \\ 2 & 1 & 0 & 4 \end{bmatrix}$$

and suppose that there are no errors. Then

$$Y = \begin{bmatrix} 3 & 2 & 1 & 1 & 3x_1 + 2x_2 + x_3 + x_4 \\ 0 & 4 & 3 & 2 & 4x_2 + 3x_3 + 2x_4 \\ 2 & 1 & 0 & 4 & 2x_1 + x_2 + 4x_4 \end{bmatrix} = \begin{bmatrix} \hat{A} & \mathbf{y} \end{bmatrix}.$$

Once again we cannot invert \hat{A} ; however, after converting Y to RRE form and inserting an all-zero row in the third position, we obtain

$$\begin{aligned} \hat{Y} &= \begin{bmatrix} 1 & 0 & 4 & 0 & x_1 + 4x_3 \\ 0 & 1 & 2 & 0 & x_2 + 2x_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_4 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 4 & 0 & x_1 + 4x_3 \\ 0 & 1 & 2 & 0 & x_2 + 2x_3 \\ 0 & 0 & 1 - 1 & 0 & x_3 - x_3 \\ 0 & 0 & 0 & 1 & x_4 \end{bmatrix} \\ &= \begin{bmatrix} I + \hat{L}I_3^T & \mathbf{x} + \hat{L}x_3 \end{bmatrix} \\ &= \begin{bmatrix} I + \hat{L}I_3^T & \mathbf{r} \end{bmatrix} \end{aligned} \tag{22}$$

where

$$\hat{L} = \begin{bmatrix} 4 \\ 2 \\ -1 \\ 0 \end{bmatrix}.$$

Once again we see that the error word has rank 1, and that we have recovered part of its decomposition as an outer product. Namely, we have

$$\mathbf{e} = \mathbf{r} - \mathbf{x} = \hat{L}x_3$$

where this time \hat{L} is known. ■

Having seen from these two examples how side information (partial knowledge of the error matrix) arises at the output of the RLNCC, we address the general case in the following proposition.

Proposition 7: Let Y , μ and δ be defined as above. There exist a tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{m \times \mu} \times \mathbb{F}_q^{\delta \times m}$ and a set $\mathcal{U} \subseteq \{1, \dots, n\}$ satisfying

$$|\mathcal{U}| = \mu \quad (23)$$

$$I_{\mathcal{U}}^T \mathbf{r} = 0 \quad (24)$$

$$I_{\mathcal{U}}^T \hat{L} = -I_{\mu \times \mu} \quad (25)$$

$$\text{rank } \hat{E} = \delta \quad (26)$$

such that

$$\left\langle \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \right\rangle = \langle Y \rangle. \quad (27)$$

Proof: See the Appendix. ■

Proposition 7 shows that every matrix Y is row equivalent to a matrix

$$\bar{Y} = \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix}$$

which is essentially the matrix Y in reduced row echelon form. Equations (19), (21) and (22) are examples of matrices in this form. We can think of the matrices \mathbf{r} , \hat{L} and \hat{E} and the set \mathcal{U} as providing a compact description of the received subspace $\langle Y \rangle$. The set \mathcal{U} is in fact redundant and can be omitted from the description, as we show in the next proposition.

Proposition 8: Let $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{m \times \mu} \times \mathbb{F}_q^{\delta \times m}$ be a tuple and $\mathcal{U} \subseteq \{1, \dots, n\}$ be a set that satisfy (23)–(26). For any $\mathcal{S} \subseteq \{1, \dots, n\}$, $T \in \mathbb{F}_q^{\mu \times \mu}$ and $R \in \mathbb{F}_q^{\delta \times \delta}$ such that $(\mathbf{r}, \hat{L}T, R\hat{E})$ and \mathcal{S} satisfy (23)–(26), we have

$$\left\langle \begin{bmatrix} I + \hat{L}T I_{\mathcal{S}}^T & \mathbf{r} \\ 0 & R\hat{E} \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \right\rangle.$$

Proof: See the Appendix. ■

Proposition 8 shows that, given a tuple $(\mathbf{r}, \hat{L}, \hat{E})$ obtained from Proposition 7, the set \mathcal{U} can be found as *any* set satisfying (23)–(25). Moreover, the matrix \hat{L} can be multiplied on the right by any nonsingular matrix (provided that the resulting matrix satisfies (23)–(25) for some \mathcal{U}), and the matrix \hat{E} can be multiplied on the left by any nonsingular matrix; none of these operations change the subspace described by $(\mathbf{r}, \hat{L}, \hat{E})$. The notion of a concise description of a subspace $\langle Y \rangle$ is captured in the following definition.

Definition 4: A tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_q^{\delta \times m}$ that satisfies (23)–(27) for some $\mathcal{U} \subseteq \{1, \dots, n\}$ is said to be a *reduction* of the matrix Y .

Remark 1: It would be enough to specify, besides the matrix \mathbf{r} , only the column space of \hat{L} and the row space of \hat{E} in the definition of a reduction. For simplicity we will, however, not use this notation here.

Note that if Y is a lifting of \mathbf{r} , then $(\mathbf{r}, [], [])$ is a reduction of Y (where $[]$ denotes an empty matrix). Thus, reduction can be interpreted as the inverse of lifting.

We can now prove the main theorem of this section.

Theorem 9: Let $(\mathbf{r}, \hat{L}, \hat{E})$ be a reduction of Y . Then

$$d_s(\langle X \rangle, \langle Y \rangle) = 2 \operatorname{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} - (\mu + \delta).$$

Proof: See the Appendix. ■

A consequence of Theorem 9 is that, under the lifting construction, the decoding problem (18) for random network coding can be abstracted to a generalized decoding problem for rank-metric codes. More precisely, if we cascade an RLNCC, at the input, with a device that takes \mathbf{x} to its lifting $X = \begin{bmatrix} I & \mathbf{x} \end{bmatrix}$ and, at the output, with a device that takes Y to its reduction $(\mathbf{r}, \hat{L}, \hat{E})$, then the decoding problem (18) reduces to the following problem:

Generalized Decoding Problem for Rank-Metric Codes: Let $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ be a rank-metric code. Given a received tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_q^{\delta \times m}$ with $\operatorname{rank} \hat{L} = \mu$ and $\operatorname{rank} \hat{E} = \delta$, find

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{C}} \operatorname{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix}. \quad (28)$$

The problem above will be referred to as the generalized decoding problem for rank-metric codes, or generalized rank decoding for short. Note that the conventional rank decoding problem (7) corresponds to the special case where $\mu = \delta = 0$.

The remainder of this paper is devoted to the study of the generalized rank decoding problem and to its solution in the case of MRD codes.

V. A GENERALIZED DECODING PROBLEM FOR RANK-METRIC CODES

In this section, we develop a perspective on the generalized rank decoding problem that will prove useful to the understanding of the correction capability of rank-metric codes, as well as to the formulation of an efficient decoding algorithm.

A. Error Locations and Error Values

Let $\mathcal{C} \in \mathbb{F}_q^{n \times m}$ be a rank-metric code. For a transmitted codeword \mathbf{x} and a received word \mathbf{r} , define $\mathbf{e} \triangleq \mathbf{r} - \mathbf{x}$ as the error word.

Note that if an error word \mathbf{e} has rank τ , then we can write $\mathbf{e} = \mathbf{L}\mathbf{E}$ for some full-rank matrices $\mathbf{L} \in \mathbb{F}_q^{n \times \tau}$ and $\mathbf{E} \in \mathbb{F}_q^{\tau \times m}$, as in (1). Let $L_1, \dots, L_\tau \in \mathbb{F}_q^n$ denote the columns of \mathbf{L} and let $E_1, \dots, E_\tau \in \mathbb{F}_q^{1 \times m}$ denote the rows of \mathbf{E} . Then we can expand \mathbf{e} as a summation of outer products

$$\mathbf{e} = \mathbf{L}\mathbf{E} = \sum_{j=1}^{\tau} L_j E_j. \quad (29)$$

We will now borrow some terminology from classical coding theory. Recall that an error vector $\mathbf{e} \in \mathbb{F}_q^n$ of Hamming weight τ can be expanded uniquely as a sum of products

$$\mathbf{e} = \sum_{j=1}^{\tau} I_{i_j} e_j$$

where $1 \leq i_1 < \dots < i_\tau \leq n$ and $e_1, \dots, e_\tau \in \mathbb{F}_q$. The index i_j (or the unit vector I_{i_j}) specifies the *location* of the j th error, while e_j specifies the *value* of the j th error.

Analogously, in the sum-of-outer-products expansion (29) we will refer to L_1, \dots, L_τ as the *error locations* and to E_1, \dots, E_τ as the *error values*. The location L_j (a column vector) indicates that, for $i = 1, \dots, n$, the j th error value E_j (a row vector) occurred in row i multiplied by the coefficient L_{ij} . Of course, $L_{ij} = 0$ means that the j th error value is not present in row i .

Note that, in contrast to the classical case, the distinction between error locations and error values in the rank metric is merely a convention. If we prefer to think of errors as occurring on columns rather than rows, then the roles of L_j and E_j would be interchanged. The same observation will also apply to any concept derived from the interpretation of these quantities as error locations and error values.

It is important to mention that, in contrast with classical coding theory, the expansion (29) is not unique, since

$$\mathbf{e} = \mathbf{L}\mathbf{E} = \mathbf{L}\mathbf{T}^{-1}\mathbf{T}\mathbf{E}$$

for any nonsingular $\mathbf{T} \in \mathbb{F}_q^{\tau \times \tau}$. Thus, strictly speaking, L_1, \dots, L_τ and E_1, \dots, E_τ are just one possible set of error locations/values describing the error word \mathbf{e} .

B. Erasures and Deviations

We now reformulate the generalized rank decoding problem in a way that facilitates its understanding and solution.

First, observe that the problem (28) is equivalent to the problem of finding an error word \hat{e} , given by

$$\hat{e} = \underset{e \in \mathbf{r} - \mathcal{C}}{\operatorname{argmin}} \operatorname{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{E} \end{bmatrix}, \quad (30)$$

from which the output of the decoder can be computed as $\hat{\mathbf{x}} = \mathbf{r} - \hat{e}$.

Proposition 10: Let $e \in \mathbb{F}_q^{n \times m}$, $\hat{L} \in \mathbb{F}_q^{n \times \mu}$ and $\hat{E} \in \mathbb{F}_q^{\delta \times n}$. The following statements are equivalent:

- 1) $\tau^* = \operatorname{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{E} \end{bmatrix}$.
- 2) $\tau^* - \mu - \delta$ is the minimum value of

$$\operatorname{rank}(e - \hat{L}E^{(1)} - L^{(2)}\hat{E})$$

for all $E^{(1)} \in \mathbb{F}_q^{\mu \times m}$ and all $L^{(2)} \in \mathbb{F}_q^{n \times \delta}$.

- 3) τ^* is the minimum value of τ for which there exist $L_1, \dots, L_\tau \in \mathbb{F}_q^n$ and $E_1, \dots, E_\tau \in \mathbb{F}_q^{1 \times m}$ satisfying:

$$e = \sum_{j=1}^{\tau} L_j E_j$$

$$L_j = \hat{L}_j, \quad j = 1, \dots, \mu$$

$$E_{\mu+j} = \hat{E}_j, \quad j = 1, \dots, \delta.$$

Proof: See the Appendix. ■

With the help of Proposition 10, the influence of \hat{L} and \hat{E} in the decoding problem can be interpreted as follows. Suppose $e \in \mathbf{r} - \mathcal{C}$ is the unique solution to (30). Then e can be expanded as $e = \sum_{j=1}^{\tau} L_j E_j$, where L_1, \dots, L_μ and $E_{\mu+1}, \dots, E_{\mu+\delta}$ are *known* to the decoder. In other words, the decoding problem is facilitated, since the decoder has side information about the expansion of e .

Recall the terminology of Section V-A. Observe that, for $j \in \{1, \dots, \mu\}$, the decoder knows the *location* of the j th error term but not its value, while for $j \in \{\mu+1, \dots, \mu+\delta\}$, the decoder knows the *value* of the j th error term but not its location. Since in classical coding theory knowledge of an error location but not its value corresponds to an erasure, we will adopt a similar terminology here. However we will need to introduce a new term to handle the case where the value of an error is known, but not its location. In the expansion (29) of the error word, each term $L_j E_j$ will be called

- an *erasure*, if L_j is known;
- a *deviation*, if E_j is known; and
- a *full error* (or simply an *error*), if neither L_j nor E_j are known.

Collectively, erasures, deviations and errors will be referred to as “errata.” We say that an errata pattern is *correctable* when (28) has a unique solution equal to the original transmitted codeword.

The following theorem characterizes the errata-correction capability of rank-metric codes.

Theorem 11: A rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ of minimum distance d is able to correct every pattern of ϵ errors, μ erasures and δ deviations if and only if $2\epsilon + \mu + \delta \leq d - 1$.

Proof: Let $\mathbf{x} \in \mathcal{C}$ be a transmitted codeword and let $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_q^{\delta \times m}$ be a received tuple such that $\text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} = \mu + \delta + \epsilon$. Suppose $\mathbf{x}' \in \mathcal{C}$ is another codeword such that $\text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x}' \\ 0 & \hat{E} \end{bmatrix} = \mu + \delta + \epsilon'$, where $\epsilon' \leq \epsilon$. From Proposition 10, we can write

$$\mathbf{e} = \mathbf{r} - \mathbf{x} = \hat{L}E^{(1)} + L^{(2)}\hat{E} + L^{(3)}E^{(3)}$$

$$\mathbf{e}' = \mathbf{r} - \mathbf{x}' = \hat{L}E^{(4)} + L^{(5)}\hat{E} + L^{(6)}E^{(6)}$$

for some $E^{(1)}, L^{(2)}, \dots, E^{(6)}$ with appropriate dimensions such that $\text{rank } L^{(3)}E^{(3)} = \epsilon$ and $\text{rank } L^{(6)}E^{(6)} = \epsilon'$.

Thus,

$$\mathbf{e} - \mathbf{e}' = \hat{L}(E^{(1)} - E^{(4)}) + (L^{(2)} - L^{(5)})\hat{E} + L^{(3)}E^{(3)} + L^{(6)}E^{(6)}$$

and

$$\text{rank}(\mathbf{x}' - \mathbf{x}) = \text{rank}(\mathbf{e} - \mathbf{e}') \leq \mu + \delta + \epsilon + \epsilon' \leq d - 1$$

contradicting the minimum distance of the code.

Conversely, let $\mathbf{x}, \mathbf{x}' \in \mathcal{C}$ be two codewords such that $\text{rank}(\mathbf{x}' - \mathbf{x}) = d$. For all μ, δ and ϵ such that $\mu + \delta + 2\epsilon \geq d$, we can write

$$\mathbf{x}' - \mathbf{x} = L^{(1)}E^{(1)} + L^{(2)}E^{(2)} + L^{(3)}E^{(3)} + L^{(4)}E^{(4)}$$

where the four terms above have inner dimensions equal to μ, δ, ϵ and $\epsilon' = d - \mu - \delta - \epsilon$, respectively.

Let

$$\mathbf{e} = L^{(1)}E^{(1)} + L^{(2)}E^{(2)} + L^{(3)}E^{(3)}$$

$$\mathbf{e}' = -L^{(4)}E^{(4)}$$

and observe that $\mathbf{x}' - \mathbf{x} = \mathbf{e} - \mathbf{e}'$. Let $\mathbf{r} = \mathbf{x} + \mathbf{e} = \mathbf{x}' + \mathbf{e}'$, $\hat{L} = L^{(1)}$ and $\hat{E} = E^{(2)}$. Suppose that \mathbf{x} is transmitted and the tuple $(\mathbf{r}, \hat{L}, \hat{E})$ is received. Then

$$\begin{aligned} \text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} &= \text{rank} \begin{bmatrix} \hat{L} & \mathbf{e} \\ 0 & \hat{E} \end{bmatrix} = \mu + \delta + \epsilon \\ \text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x}' \\ 0 & \hat{E} \end{bmatrix} &= \text{rank} \begin{bmatrix} \hat{L} & \mathbf{e}' \\ 0 & \hat{E} \end{bmatrix} = \mu + \delta + \epsilon'. \end{aligned}$$

Since $\epsilon' = d - \mu - \delta - \epsilon \leq \epsilon$, it follows that \mathbf{x} cannot be the unique solution to (28) and therefore the errata pattern cannot be corrected. ■

Theorem 11 shows that, similarly to erasures in the Hamming metric, erasures and deviations cost half of an error in the rank metric.

Theorem 11 also shows that taking into account information about erasures and deviations (when they occur) can strictly increase the error correction capability of a rank-metric code. Indeed, suppose that an error word of rank $t = \mu + \delta + \epsilon$ is applied to a codeword, where μ , δ and ϵ are the number of erasures, deviations and full errors, respectively, in the errata pattern. It follows that a conventional rank decoder (which ignores the information about erasures and deviations) can only guarantee successful decoding if $2t \leq d - 1$, where d is the minimum rank distance of the code. On the other hand, a generalized rank decoder requires only $2\epsilon + \mu + \delta \leq d - 1$, or $2t \leq d - 1 + \mu + \delta$, in order to guarantee successful decoding. In this case, the error correction capability is increased by $(\mu + \delta)/2$ if a generalized rank decoder is used instead of a conventional one.

We conclude this section by comparing our generalized decoding problem with previous decoding problems proposed for rank-metric codes.

There has been a significant amount of research on the problem of correcting rank errors in the presence of “row and column erasures” [13], [15]–[18], where a row erasure means that all entries of that row are replaced by an erasure symbol, and similarly for a column erasure. The decoding problem in this setting is naturally defined as finding a codeword such that, when the erased entries in the received word are replaced by those of the codeword, the difference between this new matrix and the codeword has the smallest possible rank. We now show that this problem is a special case of (28).

First, we force the received word \mathbf{r} to be in $\mathbb{F}_q^{n \times m}$ by replacing each erasure symbol with an arbitrary symbol in \mathbb{F}_q , say 0. Suppose that the rows i_1, \dots, i_μ and the columns k_1, \dots, k_δ have been erased. Let $\hat{L} \in \mathbb{F}_q^{n \times \mu}$ be given by $\hat{L}_{i_j, j} = 1$ and $\hat{L}_{i, j} = 0$, $\forall i \neq i_j$, for $j = 1, \dots, \mu$ and let $\hat{E} \in \mathbb{F}_q^{\delta \times m}$ be given by

$\hat{E}_{j,k_j} = 1$ and $\hat{E}_{j,k} = 0, \forall k \neq k_j$, for $j = 1, \dots, \delta$. Since

$$\begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} = \begin{bmatrix} \hat{L} & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{x} \\ 0 & 0 \end{bmatrix} \quad (31)$$

it is easy to see that we can perform column operations on (31) to replace the erased rows of \mathbf{r} with the same entries as \mathbf{x} , and similarly we can perform row operations on (31) to replace the erased columns of \mathbf{r} with the same entries as \mathbf{x} . The decoding problem (28) is unchanged by these operations and reduces exactly to the decoding problem with “row and column erasures” described in the previous paragraph. An example is given below.

Example 4: Let $n = m = 3$. Suppose the third row and the second column have been erased in the received word. Then

$$\mathbf{r} = \begin{bmatrix} r_{11} & 0 & r_{13} \\ r_{21} & 0 & r_{23} \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{L} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \hat{E} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

Since

$$\begin{bmatrix} 0 & r_{11} & 0 & r_{13} \\ 0 & r_{21} & 0 & r_{23} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & r_{11} & x_{12} & r_{13} \\ 0 & r_{21} & x_{22} & r_{23} \\ 1 & x_{31} & x_{32} & x_{33} \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

are row equivalent, we obtain that

$$\begin{aligned} \text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} &= \text{rank} \begin{bmatrix} 0 & r_{11} - x_{11} & 0 & r_{13} - x_{13} \\ 0 & r_{21} - x_{21} & 0 & r_{23} - x_{23} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= 2 + \text{rank} \begin{bmatrix} r_{11} - x_{11} & 0 & r_{13} - x_{13} \\ r_{21} - x_{21} & 0 & r_{23} - x_{23} \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

which is essentially the same objective function as in the decoding problem with “row and column erasures” described above. ■

While row/column erasures are a special case of erasures/deviations, it also true that the latter can always be transformed into the former. This can be accomplished by multiplying all rank-metric codewords to the left and to the right by nonsingular matrices in such a way that the corresponding \hat{L}_j and \hat{E}_j become

unit vectors. The drawback of this approach, as pointed out in Section IV-B, is that the structure of the code is changed at each decoding instance, which may raise complexity and/or implementation issues. Thus, it is practically more advantageous to fix the structure of the code and construct a decoder that can handle the generalized notions of erasures and deviations. This is the approach we take in the next section.

VI. DECODING GABIDULIN CODES WITH ERRORS, ERASURES AND DEVIATIONS

In this section, we turn our attention to the design of an efficient rank decoder that can correct any pattern of ϵ errors, μ erasures and δ deviations satisfying $2\epsilon + \mu + \delta \leq d - 1$, where d is the minimum rank distance of the code. Our decoder is applicable to Gabidulin codes, a class of MRD codes proposed in [11].

A. Preliminaries

Rank-metric codes in $\mathbb{F}_q^{n \times m}$ are typically constructed as block codes of length n over the extension field \mathbb{F}_{q^m} . More precisely, by fixing a basis for \mathbb{F}_{q^m} as an m -dimensional vector space over \mathbb{F}_q , we can regard any element of \mathbb{F}_{q^m} as a *row* vector of length m over \mathbb{F}_q (and vice-versa). Similarly, we can regard any *column* vector of length n over \mathbb{F}_{q^m} as an $n \times m$ matrix over \mathbb{F}_q (and vice-versa). All concepts previously defined for matrices in $\mathbb{F}_q^{n \times m}$ can be naturally applied to vectors in $\mathbb{F}_{q^m}^n$; in particular, the rank of a vector $\mathbf{x} \in \mathbb{F}_{q^m}^n$ is the rank of \mathbf{x} as an $n \times m$ matrix over \mathbb{F}_q .

1) *Gabidulin Codes:* In order to simplify notation, let $[i]$ denote q^i . A Gabidulin code is a linear (n, k) code over \mathbb{F}_{q^m} defined by the parity-check matrix

$$H = \begin{bmatrix} h_1^{[0]} & h_2^{[0]} & \cdots & h_n^{[0]} \\ h_1^{[1]} & h_2^{[1]} & \cdots & h_n^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{[n-k-1]} & h_2^{[n-k-1]} & \cdots & h_n^{[n-k-1]} \end{bmatrix}$$

where the elements $h_1, \dots, h_n \in \mathbb{F}_{q^m}$ are linearly independent over \mathbb{F}_q (note that $n \leq m$ is required). The minimum rank distance of a Gabidulin code is $d = n - k + 1$, satisfying the Singleton bound in the rank metric [11].

2) *Linearized Polynomials:* A class of polynomials that play an important role in the study of rank-metric codes are the *linearized polynomials* [25, Sec. 3.4]. A linearized polynomial (or q -polynomial) over \mathbb{F}_{q^m} is a polynomial of the form

$$f(x) = \sum_{i=0}^t f_i x^{[i]}$$

where $f_i \in \mathbb{F}_{q^m}$. If $f_t \neq 0$, we call t the q -degree of $f(x)$. Linearized polynomials receive their name because of the following property: for any $a_1, a_2 \in \mathbb{F}_q$ and any $\beta_1, \beta_2 \in \mathbb{F}_{q^m}$,

$$f(a_1\beta_1 + a_2\beta_2) = a_1f(\beta_1) + a_2f(\beta_2).$$

That is, evaluation of a linearized polynomial is a map $\mathbb{F}_{q^m} \rightarrow \mathbb{F}_{q^m}$ that is linear over \mathbb{F}_q . In particular, the set of all roots in \mathbb{F}_{q^m} of a linearized polynomial is a subspace of \mathbb{F}_{q^m} .

Let $A(x)$ and $B(x)$ be linearized polynomials of q -degrees t_A and t_B , respectively. The symbolic product of $A(x)$ and $B(x)$ is defined as the polynomial $A(x) \otimes B(x) \triangleq A(B(x))$. It is easy to verify that $P(x) = A(x) \otimes B(x)$ is a linearized polynomial of q -degree $t = t_A + t_B$ whose coefficients can be computed as

$$P_\ell = \sum_{i=\max\{0, \ell-t_B\}}^{\min\{\ell, t_A\}} A_i B_{\ell-i}^{[i]} = \sum_{j=\max\{0, \ell-t_A\}}^{\min\{\ell, t_B\}} A_{\ell-j} B_j^{[\ell-j]}$$

for $\ell = 0, \dots, t$. In particular, if $t_A \leq t_B$, then

$$P_\ell = \sum_{i=0}^{t_A} A_i B_{\ell-i}^{[i]}, \quad t_A \leq \ell \leq t_B, \quad (32)$$

while if $t_B \leq t_A$, then

$$P_\ell = \sum_{j=0}^{t_B} A_{\ell-j} B_j^{[\ell-j]}, \quad t_B \leq \ell \leq t_A. \quad (33)$$

It is known that the set of linearized polynomials over \mathbb{F}_{q^m} together with the operations of polynomial addition and symbolic multiplication forms a noncommutative ring with identity having many of the properties of a Euclidean domain.

We define the q -reverse of a linearized polynomial $f(x) = \sum_{i=0}^t f_i x^{[i]}$ as the polynomial $\bar{f}(x) = \sum_{i=0}^t \bar{f}_i x^{[i]}$ given by $\bar{f}_i = f_{t-i}^{[i-t]}$ for $i = 0, \dots, t$. (When t is not specified we will assume that t is the q -degree of $f(x)$.)

For a set $\mathcal{S} \subseteq \mathbb{F}_{q^m}$, define the *minimal linearized polynomial* of \mathcal{S} (with respect to \mathbb{F}_{q^m}), denoted $M_{\mathcal{S}}(x)$ or $\text{minpoly}\{\mathcal{S}\}(x)$, as the monic linearized polynomial over \mathbb{F}_{q^m} of least degree whose root space contains \mathcal{S} . It can be shown that $M_{\mathcal{S}}(x)$ is given by

$$M_{\mathcal{S}}(x) \triangleq \prod_{\beta \in \langle \mathcal{S} \rangle} (x - \beta)$$

so the q -degree of $M_{\mathcal{S}}(x)$ is equal to $\dim \langle \mathcal{S} \rangle$. Moreover, if $f(x)$ is any linearized polynomial whose root space contains \mathcal{S} , then

$$f(x) = Q(x) \otimes M_{\mathcal{S}}(x)$$

for some linearized polynomial $Q(x)$. This implies that $M_{S \cup \{\alpha\}}(x) = M_{M_S(\alpha)}(x) \otimes M_S(x)$ for any α . Thus, $M_S(x)$ can be computed in $O(t^2)$ operations in \mathbb{F}_{q^m} by taking a basis $\{\alpha_1, \dots, \alpha_t\}$ for $\langle S \rangle$ and computing $M_{\{\alpha_1, \dots, \alpha_i\}}(x)$ recursively for $i = 1, \dots, t$.

3) *Decoding of Gabidulin Codes*: Recall that, in the conventional rank decoding problem with τ errors, where $2\tau \leq d - 1$, we are given a received word $\mathbf{r} \in \mathbb{F}_{q^m}^n$ and we want to find the unique error word $\mathbf{e} \in \mathbf{r} - \mathcal{C}$ such that $\text{rank } \mathbf{e} = \tau$. We review below the usual decoding procedure, which consists of finding error values $E_1, \dots, E_\tau \in \mathbb{F}_{q^m}$ and error locations $L_1, \dots, L_\tau \in \mathbb{F}_q^n$ such that $\mathbf{e} = \sum_{j=1}^\tau L_j E_j$.

Since $\mathbf{e} \in \mathbf{r} - \mathcal{C}$, we can form the *syndromes*

$$[S_0, \dots, S_{d-2}]^T \triangleq H\mathbf{r} = H\mathbf{e}$$

which can then be related to the error values and error locations according to

$$\begin{aligned} S_\ell &= \sum_{i=1}^n h_i^{[\ell]} e_i = \sum_{i=1}^n h_i^{[\ell]} \sum_{j=1}^\tau L_{ij} E_j \\ &= \sum_{j=1}^\tau X_j^{[\ell]} E_j, \quad \ell = 0, \dots, d-2 \end{aligned} \quad (34)$$

where

$$X_j = \sum_{i=1}^n L_{ij} h_i, \quad j = 1, \dots, \tau \quad (35)$$

are called the *error locators* associated with L_1, \dots, L_τ .

Suppose, for now, that the error values E_1, \dots, E_τ (which are essentially τ linearly independent elements satisfying $\langle \mathbf{e} \rangle = \langle E_1, \dots, E_\tau \rangle$) have already been determined. Then the error locators can be determined by solving (34) or, equivalently, by solving

$$\bar{S}_\ell = S_{d-2-\ell}^{[\ell-d+2]} = \sum_{j=1}^\tau E_j^{[\ell-d+2]} X_j, \quad \ell = 0, \dots, d-2 \quad (36)$$

which is a system of equations of the form

$$B_\ell = \sum_{j=1}^\tau A_j^{[\ell]} X_j, \quad \ell = 0, \dots, d-2 \quad (37)$$

consisting of $d - 1$ linear equations (over \mathbb{F}_{q^m}) in τ unknowns X_1, \dots, X_τ . Such a system is known to have a unique solution (whenever one exists) provided $\tau \leq d - 1$ and A_1, \dots, A_τ are linearly independent (see [25], [26]). Moreover, a solution to (37) can be found efficiently in $O(d^2)$ operations in \mathbb{F}_{q^m} by an algorithm proposed by Gabidulin [11, pp. 9–10].

After the error locators have been found, the error locations L_1, \dots, L_τ can be easily recovered by solving (35). More precisely, let $\mathbf{h} \in \mathbb{F}_q^{n \times m}$ be the matrix whose rows are h_1, \dots, h_n , and let $Q \in \mathbb{F}_q^{m \times n}$ be a right inverse of \mathbf{h} , i.e., $\mathbf{h}Q = I_{n \times n}$. Then

$$L_{ij} = \sum_{k=1}^m X_{jk} Q_{ki}, \quad i = 1, \dots, n, \quad j = 1, \dots, \tau.$$

The computation of error values can be done indirectly via an *error span polynomial* $\sigma(x)$. Let $\sigma(x)$ be a linearized polynomial of q -degree τ having as roots all linear combinations of E_1, \dots, E_τ . Then, $\sigma(x)$ can be related to the *syndrome polynomial*

$$S(x) = \sum_{j=0}^{d-2} S_j x^{[j]}$$

through the following *key equation*:

$$\sigma(x) \otimes S(x) \equiv \omega(x) \pmod{x^{[d-1]}} \quad (38)$$

where $\omega(x)$ is a linearized polynomial of q -degree $\leq \tau - 1$.

An equivalent way to express (38) is

$$\sum_{i=0}^{\tau} \sigma_i S_{\ell-i}^{[i]} = 0, \quad \ell = \tau, \dots, d-2. \quad (39)$$

This key equation can be efficiently solved in $O(d^2)$ operations in \mathbb{F}_{q^m} by the modified Berlekamp-Massey algorithm proposed in [13], provided $2\tau \leq d-1$.

After the error span polynomial is found, the error values can be obtained by computing a basis E_1, \dots, E_τ for the root space of $\sigma(x)$. This can be done either by the probabilistic algorithm in [27], in an average of $O(dm)$ operations in \mathbb{F}_{q^m} , or by the methods in [28], which take at most $O(m^3)$ operations in \mathbb{F}_q plus $O(dm)$ operations in \mathbb{F}_{q^m} .

B. A Modified Key Equation Incorporating Erasures and Deviations

In the general rank decoding problem with ϵ errors, μ erasures and δ deviations, where $2\epsilon + \mu + \delta \leq d-1$, we are given a received tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_{q^m}^\delta$ and we want to find the unique error word $\mathbf{e} \in \mathbf{r} - \mathcal{C}$ such that $\text{rank} \begin{bmatrix} \hat{L} & \mathbf{e} \\ 0 & \hat{E} \end{bmatrix} = \epsilon + \mu + \delta \triangleq \tau$ (along with the value of ϵ , which is not known a priori).

First, note that if we can find a linearized polynomial $\sigma(x)$ of q -degree at most $\tau \leq d-1$ satisfying $\sigma(e_i) = 0$, $i = 1, \dots, n$, then the error word can be determined in the same manner as in Section VI-A3.

According to Proposition 10, we can write the error word as $e = \sum_{j=1}^{\tau} L_j E_j$ for some $L_1, \dots, L_{\tau} \in \mathbb{F}_q^n$ and $E_1, \dots, E_{\tau} \in \mathbb{F}_{q^m}$ satisfying $L_j = \hat{L}_j$, $j = 1, \dots, \mu$, and $E_{\mu+j} = \hat{E}_j$, $j = 1, \dots, \delta$. Let $\sigma_D(x)$, $\sigma_F(x)$ and $\sigma_U(x)$ be linearized polynomials of smallest q -degrees satisfying

$$\begin{aligned}\sigma_D(E_j) &= 0, \quad j = \mu + 1, \dots, \mu + \delta \\ \sigma_F(\sigma_D(E_j)) &= 0, \quad j = \mu + \delta + 1, \dots, \tau \\ \sigma_U(\sigma_F(\sigma_D(E_j))) &= 0, \quad j = 1, \dots, \mu.\end{aligned}$$

Clearly, the q -degrees of $\sigma_D(x)$ and $\sigma_F(x)$ are δ and ϵ , respectively, and the q -degree of $\sigma_U(x)$ is at most μ .

Define the *error span polynomial*

$$\sigma(x) = \sigma_U(x) \otimes \sigma_F(x) \otimes \sigma_D(x).$$

Then $\sigma(x)$ is a linearized polynomial of q -degree $\leq \tau$ satisfying

$$\sigma(e_i) = \sigma\left(\sum_{j=1}^{\tau} L_{ij} E_j\right) = \sum_{j=1}^{\tau} L_{ij} \sigma(E_j) = 0, \quad i = 1, \dots, n.$$

Thus, since $\sigma_D(x)$ can be readily determined from \hat{E} , decoding reduces to the determination of $\sigma_F(x)$ and $\sigma_U(x)$.

Now, let $\lambda_U(x)$ be a linearized polynomial of q -degree μ satisfying

$$\lambda_U(X_j) = 0, \quad j = 1, \dots, \mu$$

and let $\bar{\lambda}_U(x)$ be the q -reverse of $\lambda_U(x)$. We define an *auxiliary syndrome polynomial* as

$$S_{DU}(x) = \sigma_D(x) \otimes S(x) \otimes \bar{\lambda}_U(x).$$

Observe that $S_{DU}(x)$ incorporates all the information that is known at the decoder, including erasures and deviations.

Our modified key equation is given in the following theorem.

Theorem 12:

$$\sigma_F(x) \otimes S_{DU}(x) \equiv \omega(x) \pmod{x^{[d-1]}} \quad (40)$$

where $\omega(x)$ is a linearized polynomial of q -degree $\leq \tau - 1$.

Proof: Let $\omega(x) = \sigma_F(x) \otimes S_{DU}(x) \pmod{x^{[d-1]}}$. If $\tau \geq d - 1$, we have nothing to prove, so let us assume $\tau \leq d - 2$. We will show that $\omega_{\ell} = 0$ for $\ell = \tau, \dots, d - 2$.

Let $\sigma_{FD}(x) = \sigma_F(x) \otimes \sigma_D(x)$ and $S_{FD}(x) = \sigma_{FD}(x) \otimes S(x)$. According to (32), for $\epsilon + \delta \leq \ell \leq d - 2$ we have

$$\begin{aligned} S_{FD,\ell} &= \sum_{i=0}^{\epsilon+\delta} \sigma_{FD,i} S_{\ell-i}^{[i]} = \sum_{i=0}^{\epsilon+\delta} \sigma_{FD,i} \left(\sum_{j=1}^{\tau} X_j^{[\ell-i]} E_j \right)^{[i]} \\ &= \sum_{j=1}^{\tau} X_j^{[\ell]} \sigma_{FD}(E_j) = \sum_{j=1}^{\mu} X_j^{[\ell]} \beta_j, \end{aligned} \quad (41)$$

where

$$\beta_j = \sigma_{FD}(E_j), \quad j = 1, \dots, \mu.$$

Note that $\sigma_F(x) \otimes S_{DU}(x) = S_{FD}(x) \otimes \bar{\lambda}_U(x)$. Using (33) and (41), for $\mu + \epsilon + \delta \leq \ell \leq d - 2$ we have

$$\begin{aligned} \omega_\ell &= \sum_{i=0}^{\mu} \bar{\lambda}_{U,i}^{[\ell-i]} S_{FD,\ell-i} = \sum_{i=0}^{\mu} \lambda_{U,\mu-i}^{[\ell-\mu]} \sum_{j=1}^{\mu} X_j^{[\ell-i]} \beta_j \\ &= \sum_{j=1}^{\mu} \sum_{i=0}^{\mu} \lambda_{U,i}^{[\ell-\mu]} X_j^{[\ell-\mu+i]} \beta_j = \sum_{j=1}^{\mu} \lambda_U(X_j)^{[\ell-\mu]} \beta_j = 0. \end{aligned}$$

This completes the proof of the theorem. ■

The key equation can be equivalently expressed as

$$\sum_{i=0}^{\epsilon} \sigma_{F,i} S_{DU,\ell-i}^{[i]} = 0, \quad \ell = \mu + \delta + \epsilon, \dots, d - 2. \quad (42)$$

Note that this key equation reduces to the original key equation (38) when there are no erasures or deviations. Moreover, it can be solved by the same methods as the original key equation (38), e.g., using the Euclidean algorithm for linearized polynomials [11] or using the modified Berlekamp-Massey algorithm from [13], provided $2\epsilon \leq d - 1 - \mu - \delta$ (which is true by assumption). Note that a small adjustment needs to be made so that (42) becomes indeed equivalent to (39); namely, we should choose S_ℓ in (39) as $S_\ell = S_{DU,\ell+\mu+\delta}$ and replace d with $d - \mu - \delta$.

After computing $\sigma_F(x)$, we still need to determine $\sigma_U(x)$. In the proof of Theorem 12, observe that (41) has the same form as (37); thus, $\beta_1, \dots, \beta_\mu$ can be computed using Gabidulin's algorithm [11, pp. 9–10], since $S_{FD}(x)$ and X_1, \dots, X_μ are known. Finally, $\sigma_U(x)$ can be obtained as $\sigma_U(x) = \text{minpoly}\{\beta_1, \dots, \beta_\mu\}$.

C. Summary of the Algorithm and Complexity Analysis

The complete algorithm for decoding Gabidulin codes with erasures and deviations is summarized in Fig. 1. We now estimate the complexity of this algorithm.

Input: received tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_{q^m}^\delta$.

Output: error word $\mathbf{e} \in \mathbb{F}_{q^m}^n$.

1) *Computing the auxiliary syndrome polynomial:*

Compute

- a) $S_\ell = \sum_{i=1}^n h_i^{[\ell]} r_i, \ell = 0, \dots, d-2$
- b) $\hat{X}_j = \sum_{i=1}^n \hat{L}_{ij} h_i, j = 1, \dots, \mu$
- c) $\lambda_U(x) = \text{minpoly}\{\hat{X}_1, \dots, \hat{X}_\mu\}$
- d) $\sigma_D(x) = \text{minpoly}\{\hat{E}_1, \dots, \hat{E}_\delta\}$, and
- e) $S_{DU}(x) = \sigma_D(x) \otimes S(x) \otimes \bar{\lambda}_U(x)$.

2) *Computing the error span polynomial:*

- a) Use the Berlekamp-Massey algorithm [13] to find $\sigma_F(x)$ that solve the key equation (40).
- b) Compute $S_{FD}(x) = \sigma_F(x) \otimes \sigma_D(x) \otimes S(x)$.
- c) Use Gabidulin's algorithm [11] to find $\beta_1, \dots, \beta_\mu \in \mathbb{F}_{q^m}$ that solve (41).
- d) Compute $\sigma_U(x) = \text{minpoly}\{\beta_1, \dots, \beta_\mu\}$ and
- e) $\sigma(x) = \sigma_U(x) \otimes \sigma_F(x) \otimes \sigma_D(x)$.

3) *Finding the roots of the error span polynomial:*

Use either the algorithm in [27] or the methods in [28] to find a basis $E_1, \dots, E_\tau \in \mathbb{F}_{q^m}$ for the root space of $\sigma(x)$.

4) *Finding the error locations:*

- a) Solve (36) using Gabidulin's algorithm [11] to find the error locators $X_1, \dots, X_\tau \in \mathbb{F}_{q^m}$.
 - b) Compute the error locations $L_{ij} = \sum_{k=1}^m X_{jk} Q_{ki}, i = 1, \dots, n, j = 1, \dots, \tau$.
 - c) Compute the error word $\mathbf{e} = \sum_{j=1}^\tau L_j E_j$.
-

Fig. 1. Generalized decoding algorithm for Gabidulin codes.

Steps 1e), 2b) and 2e) are symbolic multiplications of linearized polynomials and can be performed in $O(d^2)$ operations in \mathbb{F}_{q^m} . Steps 1c), 1d) and 2d) involve finding a minimal linearized polynomial, which takes $O(d^2)$ operations in \mathbb{F}_{q^m} . Steps 1b), 4b) and 4c) are matrix multiplications and take $O(dnm)$ operations in \mathbb{F}_q only. Both instances 2c) and 4a) of Gabidulin's algorithm and also the Berlekamp-Massey algorithm in step 2a) take $O(d^2)$ operations in \mathbb{F}_{q^m} .

The most computationally demanding steps are 1a) computing the syndromes and 3) finding a basis for the root space of the error span polynomial. The former can be implemented in a straightforward manner using $O(dn)$ operations in \mathbb{F}_{q^m} , while the latter can be performed using an average of $O(dm)$ operations in \mathbb{F}_{q^m} with the algorithm in [27] (although the method described in [28] will usually perform faster when m is small).

We conclude that the overall complexity of the algorithm is $O(dm)$ operations in \mathbb{F}_{q^m} .

D. An Equivalent Formulation Based on the Error Locator Polynomial

Due to the perfect duality between error values and error locators (both are elements of \mathbb{F}_{q^m}), it is also possible to derive a decoding algorithm based on an error *locator* polynomial that contains all the error locators as roots.

Let the auxiliary syndrome polynomial be defined as

$$S_{UD}(x) = \lambda_U(x) \otimes \bar{S}(x) \otimes \bar{\sigma}_D(x^{[d-2]})^{[-d+2]}$$

where $\bar{\sigma}_D(x)$ is the q -reverse of $\sigma_D(x)$ and $\bar{S}(x)$ is the q -reverse of $S(x)$.

Let $\lambda_F(x)$ be a linearized polynomial of q -degree ϵ such that $\lambda_F(\lambda_U(X_i)) = 0$, for $i = \mu + \delta + 1, \dots, \tau$.

We have the following key equation:

Theorem 13:

$$\lambda_F(x) \otimes S_{UD}(x) \equiv \psi(x) \pmod{x^{[d-1]}} \quad (43)$$

where $\psi(x)$ is a linearized polynomial of q -degree $\leq \tau - 1$.

Proof: The proof is similar to that of Theorem 12 and will be omitted. ■

The complete decoding algorithm based on the error locator polynomial is given in Fig. 2.

E. Practical Considerations

We have seen that the complexity of decoding a Gabidulin code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ with $d_r(\mathcal{C}) = d$ is given by $O(dm)$ operations in \mathbb{F}_{q^m} . In many applications, in particular for network coding, we have $m \gg n$. In such cases, the decoding complexity can be significantly reduced by using, rather than a Gabidulin code, an MRD code formed by the Cartesian product of many shorter Gabidulin codes with the same distance. More precisely, let $\ell = \lfloor \frac{m}{n} \rfloor$ and $n' = m - n(\ell - 1)$. Take $\mathcal{C} = C_1 \times C_2 \times \dots \times C_\ell$, where $C_i \subseteq \mathbb{F}_q^{n \times n}$, $i = 1, \dots, \ell - 1$, and $C_\ell \subseteq \mathbb{F}_q^{n \times n'}$ are Gabidulin codes with minimum rank distance d . Then \mathcal{C} is an MRD code with $d_r(\mathcal{C}) = d$.

Input: received tuple $(\mathbf{r}, \hat{L}, \hat{E}) \in \mathbb{F}_{q^m}^n \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_{q^m}^\delta$.

Output: error word $\mathbf{e} \in \mathbb{F}_{q^m}^n$.

1) *Computing the auxiliary syndrome polynomial:*

Compute

- a) $S_\ell = \sum_{i=1}^n h_i^{[\ell]} r_i, \ell = 0, \dots, d-2$
- b) $\hat{X}_j = \sum_{i=1}^n \hat{L}_{ij} h_i, j = 1, \dots, \mu$
- c) $\lambda_U(x) = \text{minpoly}\{\hat{X}_1, \dots, \hat{X}_\mu\}$
- d) $\sigma_D(x) = \text{minpoly}\{\hat{E}_1, \dots, \hat{E}_\delta\}$, and
- e) $S_{UD}(x) = \lambda_U(x) \otimes \bar{S}(x) \otimes \bar{\sigma}_D(x^{[d-2]})^{[-d+2]}$.

2) *Computing the error locator polynomial:*

- a) Use the Berlekamp-Massey algorithm [13] to find $\lambda_F(x)$ that solve the key equation (43).
- b) Compute $S_{FU}(x) = \lambda_F(x) \otimes \lambda_U(x) \otimes \bar{S}(x)$.
- c) Use Gabidulin's algorithm [11] to find $\gamma_1, \dots, \gamma_\delta \in \mathbb{F}_{q^m}$ that solve

$$S_{FU,\ell} = \sum_{j=1}^{\delta} E_{\mu+j}^{[\ell-d+2]} \gamma_j.$$

- d) Compute $\lambda_D(x) = \text{minpoly}\{\gamma_1, \dots, \gamma_\delta\}$ and
- e) $\lambda(x) = \lambda_D(x) \otimes \lambda_F(x) \otimes \lambda_U(x)$.

3) *Finding the roots of the error locator polynomial:*

Use either the algorithm in [27] or the methods in [28] to find a basis $X_1, \dots, X_\tau \in \mathbb{F}_{q^m}$ for the root space of $\lambda(x)$.

4) *Finding the error values:*

- a) Solve (34) using Gabidulin's algorithm [11] to find the error values $E_1, \dots, E_\tau \in \mathbb{F}_{q^m}$.
 - b) Compute the error locations $L_{ij} = \sum_{k=1}^m X_{jk} Q_{ki}, i = 1, \dots, n, j = 1, \dots, \tau$.
 - c) Compute the error word $\mathbf{e} = \sum_{j=1}^{\tau} L_j E_j$.
-

Fig. 2. Generalized decoding algorithm for Gabidulin codes, alternative formulation.

Now, decoding of \mathcal{C} can be performed by decoding each \mathcal{C}_i individually. Thus, assuming for simplicity that $m = n\ell$, the overall decoding complexity is given by $\ell O(dn) = O(dm)$ operations in \mathbb{F}_{q^n} . In other words, operations in a potentially large field \mathbb{F}_{q^m} can be replaced by operations in a much smaller field \mathbb{F}_{q^n} .

Note that, in this case, additional computational savings may be obtained, since all received words will share the same set of error locations. For instance, if all error locations are known and the decoding algorithm of Fig. 2 is used, then only steps 1a), 1b) and 4a)–4c) need to be performed.

VII. CONCLUSIONS

In this paper, we have introduced a new approach to the problem of error control in random network coding. Our approach is based, on the one hand, on Kötter and Kschischang's abstraction of the problem as a coding-theoretic problem for subspaces and, on the other hand, on the existence of optimal and efficiently-decodable codes for the rank metric. We have shown that, when *lifting* is performed at the transmitter and *reduction* at the receiver, the random network coding channel behaves essentially as a matrix channel that introduces errors in the rank metric and may also supply partial information about these errors in the form of erasures and deviations.

An important consequence of our results is that many of the tools developed for rank-metric codes can be *almost* directly applied to random network coding. However, in order to fully exploit the correction capability of a rank-metric code, erasures and deviations must be taken into account. A second contribution of this work is the generalization of the decoding algorithm for Gabidulin codes in order to fulfill this task. Our proposed algorithm requires $O(dm)$ operations in \mathbb{F}_{q^m} , achieving the same complexity as conventional decoding algorithms that only correct rank errors.

Following this work, a natural step toward practical error control in random network coding is the pursuit of efficient software (and possibly hardware) implementations of encoders and decoders for Gabidulin codes. Another avenue would be the investigation of more general network coding scenarios where error and erasure correction might be useful; for example, the case of multiple heterogeneous receivers can be addressed using a priority encoding transmission scheme based on Gabidulin codes [29]. An exciting open question, paralleling the development of Reed-Solomon codes, is whether an efficient list-decoder for Gabidulin codes exists that would allow correction of errors above the error-correction bound.

We believe that, with respect to forward error (and erasure) correction, Gabidulin codes will play the same role in random network coding that Reed-Solomon codes have played in traditional communication

systems.

REFERENCES

- [1] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun. 29–Jul. 4, 2003, p. 442.
- [2] P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 2003.
- [3] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
- [4] N. Cai and R. W. Yeung, “Network coding and error correction,” in *Proc. 2002 IEEE Inform. Theory Workshop*, Oct. 20–25, 2002, pp. 119–122.
- [5] R. W. Yeung and N. Cai, “Network error correction, part I: Basic concepts and upper bounds,” *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 19–36, 2006.
- [6] N. Cai and R. W. Yeung, “Network error correction, part II: Lower bounds,” *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 37–54, 2006.
- [7] Z. Zhang, “Linear network error correction codes in packet networks,” *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, 2008.
- [8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, “Resilient network coding in the presence of Byzantine adversaries,” in *Proc. 26th IEEE Int. Conf. on Computer Commun.*, Anchorage, AK, May 2007, pp. 616–624.
- [9] R. Kötter and F. R. Kschischang, “Coding for errors and erasures in random network coding,” *IEEE Trans. Inf. Theory*, 2008, to be published. [Online]. Available: <http://arxiv.org/abs/cs/0703061>
- [10] P. Delsarte, “Bilinear forms over a finite field, with applications to coding theory,” *J. of Comb. Theory. Series A*, vol. 25, pp. 226–241, 1978.
- [11] E. M. Gabidulin, “Theory of codes with maximum rank distance,” *Probl. Inform. Transm.*, vol. 21, no. 1, pp. 1–12, 1985.
- [12] R. M. Roth, “Maximum-rank array codes and their application to crisscross error correction,” *IEEE Trans. Inf. Theory*, vol. 37, pp. 328–336, 1991.
- [13] G. Richter and S. Plass, “Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm,” in *Proc. ITG Conf. on Source and Channel Coding*, Erlangen, Germany, Jan. 2004.
- [14] P. Loidreau, “A Welch-Berlekamp like algorithm for decoding Gabidulin codes,” in *Proc. 4th Int. Workshop on Coding and Cryptography*, 2005.
- [15] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, “Rank errors and rank erasures correction,” in *Proc. 4th Int. Colloq. Coding Theory, Dilijan, Armenia, 1991*, Yerevan, 1992, pp. 11–19.
- [16] E. M. Gabidulin and N. I. Pilipchuk, “A new method of erasure correction by rank codes,” in *Proc. IEEE Int. Symp. Information Theory*, Jun. 29–Jul. 4, 2003, p. 423.
- [17] G. Richter and S. Plass, “Fast decoding of rank-codes with rank errors and column erasures,” in *Proc. IEEE Int. Symp. Information Theory*, Jun. 27–Jul. 2, 2004, pp. 398–398.
- [18] N. I. Pilipchuk and E. M. Gabidulin, “Erasure correcting algorithms for rank codes,” 2007, to be published.
- [19] R. M. Roth and G. Seroussi, “Location-correcting codes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 554–565, Mar. 1996.
- [20] P. Loidreau, “Étude et optimisation de cryptosystèmes à clé publique fondés sur la théorie des codes correcteurs,” Ph.D. Dissertation, École Polytechnique, Paris, France, May 2001.

- [21] M. Gadouleau and Z. Yan, “Properties of codes with the rank metric,” in *Proc. IEEE Globecom 2006*, San Francisco, CA, Nov. 27–Dec. 1, 2006.
- [22] H. Wang, C. Xing, and R. Safavi-Naini, “Linear authentication codes: bounds and constructions,” *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 866–872, 2003.
- [23] S.-T. Xia and F.-W. Fu, “Johnson type bounds on constant dimension codes,” 2007, submitted for publication. [Online]. Available: <http://arxiv.org/abs/0709.1074v1>
- [24] D. Silva and F. R. Kschischang, “On metrics for error correction in network coding,” 2008, submitted for publication.
- [25] R. Lidl and H. Niederreiter, *Finite Fields*. Reading, MA: Addison-Wesley, 1983.
- [26] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.
- [27] V. Skachek and R. M. Roth, “Probabilistic algorithm for finding roots of linearized polynomials,” *Designs, Codes and Cryptography*, vol. 46, no. 1, pp. 17–23, 2008, also in *Comput. Sci. Dept., Technion, Tech. Rep. CS-2004-08*, Jun. 2004.
- [28] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
- [29] D. Silva and F. R. Kschischang, “Rank-metric codes for priority encoding transmission with network coding,” in *Proc. 10th Canadian Workshop Inform. Theory*, Edmonton, Alberta, Canada, Jun. 6–8, 2007, pp. 81–84.

APPENDIX

A. Proof of Proposition 7

Before proving Proposition 7, let us recall some properties of the matrices $I_{\mathcal{U}}$ and $I_{\mathcal{U}^c}$, where $I = I_{n \times n}$, $\mathcal{U} \subseteq \{1, \dots, n\}$ and $\mathcal{U}^c = \{1, \dots, n\} \setminus \mathcal{U}$.

For any $A \in \mathbb{F}_q^{n \times k}$ (respectively, $A \in \mathbb{F}_q^{k \times n}$), the matrix $I_{\mathcal{U}}^T A$ (resp., $A I_{\mathcal{U}}$) extracts the rows (resp., columns) of A that are indexed by \mathcal{U} . Conversely, for any $B \in \mathbb{F}_q^{|\mathcal{U}| \times k}$ (resp., $B \in \mathbb{F}_q^{k \times |\mathcal{U}|}$) the matrix $I_{\mathcal{U}} B$ (resp., $B I_{\mathcal{U}}^T$) reallocates the rows (resp., columns) of B to the positions indexed by \mathcal{U} , where all-zero rows (resp., columns) are inserted at the positions indexed by \mathcal{U}^c . Furthermore, observe that $I_{\mathcal{U}}$ and $I_{\mathcal{U}^c}$ satisfy the following properties:

$$\begin{aligned} I &= I_{\mathcal{U}} I_{\mathcal{U}}^T + I_{\mathcal{U}^c} I_{\mathcal{U}^c}^T, \\ I_{\mathcal{U}}^T I_{\mathcal{U}} &= I_{|\mathcal{U}| \times |\mathcal{U}|} \\ I_{\mathcal{U}}^T I_{\mathcal{U}^c} &= 0. \end{aligned}$$

We now give a proof of Proposition 7.

Proof of Proposition 7: Let $\text{RRE}(Y)$ denote the reduced row echelon form of Y . For $i = 1, \dots, N$, let p_i be the column position of the leading entry of row i in $\text{RRE}(Y)$. Let $\mathcal{U}^c = \{p_1, \dots, p_{n-\mu}\}$ and $\mathcal{U} = \{1, \dots, n\} \setminus \mathcal{U}^c$. Note that $|\mathcal{U}| = \mu$. From the properties of the reduced row echelon form, we can write

$$\text{RRE}(Y) = \begin{bmatrix} W & \tilde{r} \\ 0 & \hat{E} \end{bmatrix}$$

where $\tilde{\mathbf{r}} \in \mathbb{F}_q^{(n-\mu) \times m}$, $\hat{E} \in \mathbb{F}_q^{\delta \times m}$ has rank δ , and $W \in \mathbb{F}_q^{(n-\mu) \times n}$ satisfies $W I_{\mathcal{U}^c} = I_{(n-\mu) \times (n-\mu)}$.

Now, let

$$\bar{Y} = \begin{bmatrix} I_{\mathcal{U}^c} & 0 \\ 0 & I_{\delta \times \delta} \end{bmatrix} \text{RRE}(Y) = \begin{bmatrix} I_{\mathcal{U}^c} W & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix}$$

where $\mathbf{r} = I_{\mathcal{U}^c} \tilde{\mathbf{r}}$. Since $I = I_{\mathcal{U}^c} I_{\mathcal{U}^c}^T + I_{\mathcal{U}} I_{\mathcal{U}}^T$, we have

$$\begin{aligned} I_{\mathcal{U}^c} W &= I_{\mathcal{U}^c} W (I_{\mathcal{U}^c} I_{\mathcal{U}^c}^T + I_{\mathcal{U}} I_{\mathcal{U}}^T) \\ &= I_{\mathcal{U}^c} I_{\mathcal{U}^c}^T + I_{\mathcal{U}^c} W I_{\mathcal{U}} I_{\mathcal{U}}^T \\ &= I - I_{\mathcal{U}} I_{\mathcal{U}}^T + I_{\mathcal{U}^c} W I_{\mathcal{U}} I_{\mathcal{U}}^T \\ &= I + \hat{L} I_{\mathcal{U}}^T \end{aligned}$$

where $\hat{L} = -I_{\mathcal{U}} + I_{\mathcal{U}^c} W I_{\mathcal{U}}$. Also, since $I_{\mathcal{U}}^T I_{\mathcal{U}} = I_{\mu \times \mu}$ and $I_{\mathcal{U}}^T I_{\mathcal{U}^c} = 0$, we have $I_{\mathcal{U}}^T \hat{L} = -I_{\mu \times \mu}$ and $I_{\mathcal{U}}^T \mathbf{r} = 0$.

Thus,

$$\bar{Y} = \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix}$$

is a matrix with the same row space as Y . The proof is complete. \blacksquare

B. Proof of Proposition 8

Proof of Proposition 8: We want to show that

$$\left\langle \begin{bmatrix} I + \hat{L} T I_S^T & \mathbf{r} \\ 0 & R \hat{E} \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \right\rangle.$$

From (5) and the fact that R is nonsingular (since $\text{rank } R \hat{E} = \delta$), this amounts to showing that

$$\left\langle \begin{bmatrix} I + \hat{L} T I_S^T & \mathbf{r} \end{bmatrix} \right\rangle = \left\langle \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & \mathbf{r} \end{bmatrix} \right\rangle.$$

Let $W_1 = I + \hat{L} I_{\mathcal{U}}^T$ and $W_2 = I + \hat{L} T I_S^T$. Note that, since $W_1 I_{\mathcal{U}^c} = I_{\mathcal{U}^c}$ and $I_{\mathcal{U}}^T \begin{bmatrix} W_1 & \mathbf{r} \end{bmatrix} = 0$, we have that $I_{\mathcal{U}^c}^T W_1$ is full rank. Similarly, $I_S^T \begin{bmatrix} W_2 & \mathbf{r} \end{bmatrix} = 0$ and $I_{\mathcal{S}^c}^T W_2$ is full rank. Thus, it suffices to prove that

$$M \begin{bmatrix} W_2 & \mathbf{r} \end{bmatrix} = \begin{bmatrix} W_1 & \mathbf{r} \end{bmatrix} \tag{44}$$

for some $M \in \mathbb{F}_q^{n \times n}$.

Let $\mathcal{A} = \mathcal{U} \cup \mathcal{S}$ and $\mathcal{B} = \mathcal{U} \cap \mathcal{S}$. Observe that M can be partitioned into three sub-matrices, $MI_{\mathcal{A}^c}$, $MI_{\mathcal{S}}$ and $MI_{\mathcal{U} \setminus \mathcal{B}}$. Choose $MI_{\mathcal{A}^c} = I_{\mathcal{A}^c}$, and $MI_{\mathcal{S}}$ arbitrarily. We will choose $MI_{\mathcal{U} \setminus \mathcal{B}}$ so that (44) is satisfied. First, note that

$$M\mathbf{r} = M(I_{\mathcal{A}^c}I_{\mathcal{A}^c}^T + I_{\mathcal{A}}I_{\mathcal{A}}^T)\mathbf{r} = I_{\mathcal{A}^c}I_{\mathcal{A}^c}^T\mathbf{r} = \mathbf{r}$$

since $I_{\mathcal{A}}^T\mathbf{r} = 0$. Thus, we just need to consider $MW_2 = W_1$ in (44). Moreover, note that

$$\begin{aligned} MW_2 &= M(I_{\mathcal{A}^c}I_{\mathcal{A}^c}^T + I_{\mathcal{S}}I_{\mathcal{S}}^T + I_{\mathcal{U} \setminus \mathcal{B}}I_{\mathcal{U} \setminus \mathcal{B}}^T)W_2 \\ &= I_{\mathcal{A}^c}I_{\mathcal{A}^c}^TW_2 + (MI_{\mathcal{U} \setminus \mathcal{B}})(I_{\mathcal{U} \setminus \mathcal{B}}^TW_2). \end{aligned}$$

Now, consider the system $MW_2 = W_1$. From basic linear algebra, we can solve for $MI_{\mathcal{U} \setminus \mathcal{B}}$ if and only if

$$\text{rank} \begin{bmatrix} I_{\mathcal{U} \setminus \mathcal{B}}^TW_2 \\ W_1 - I_{\mathcal{A}^c}I_{\mathcal{A}^c}^TW_2 \end{bmatrix} \leq |\mathcal{U} \setminus \mathcal{B}|.$$

Since $I_{\mathcal{U} \setminus \mathcal{B}}^TW_1 = 0$ and $I_{\mathcal{S}}^TW_2 = 0$, we can rearrange rows to obtain

$$\begin{aligned} \text{rank} \begin{bmatrix} I_{\mathcal{U} \setminus \mathcal{B}}^TW_2 \\ W_1 - I_{\mathcal{A}^c}I_{\mathcal{A}^c}^TW_2 \end{bmatrix} &= \text{rank} \begin{bmatrix} I_{\mathcal{U} \setminus \mathcal{B}}^T(W_1 - W_2) \\ I_{(\mathcal{U} \setminus \mathcal{B})^c}I_{(\mathcal{U} \setminus \mathcal{B})^c}^T(W_1 - W_2) \end{bmatrix} \\ &= \text{rank}(W_1 - W_2). \end{aligned}$$

To complete the proof, we will show that $\text{rank}(W_1 - W_2) \leq |\mathcal{U} \setminus \mathcal{B}|$. We have

$$\begin{aligned} \text{rank}(W_1 - W_2) &= \text{rank}(\hat{L}I_{\mathcal{U}}^T - \hat{L}TI_{\mathcal{S}}^T) \\ &\leq \text{rank}(I_{\mathcal{U}}^T - TI_{\mathcal{S}}^T) \\ &= \text{rank}(I_{\mathcal{S}}^T\hat{L}I_{\mathcal{U}}^T + I_{\mathcal{S}}^T) \\ &= \text{rank } I_{\mathcal{S}}^TW_1 \\ &= \text{rank } I_{\mathcal{S}}I_{\mathcal{S}}^TW_1 \\ &= \text{rank}(I_{\mathcal{S} \setminus \mathcal{B}}I_{\mathcal{S} \setminus \mathcal{B}}^T + I_{\mathcal{B}}I_{\mathcal{B}}^T)W_1 \\ &= \text{rank } I_{\mathcal{S} \setminus \mathcal{B}}I_{\mathcal{S} \setminus \mathcal{B}}^TW_1 \\ &\leq |\mathcal{S} \setminus \mathcal{B}| = |\mathcal{U} \setminus \mathcal{B}|. \end{aligned} \tag{45}$$

where (45) is obtained by left multiplying by $I_{\mathcal{S}}^T\hat{L} = -T^{-1}$. ■

C. Proof of Theorem 9

Proof of Theorem 9: We have

$$\begin{aligned}
 \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} &= \text{rank} \begin{bmatrix} I & \mathbf{x} \\ I + \hat{L}I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \\
 &= \text{rank} \begin{bmatrix} -\hat{L}I_{\mathcal{U}}^T & \mathbf{x} - \mathbf{r} \\ I + \hat{L}I_{\mathcal{U}}^T & \mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \\
 &= \text{rank} \begin{bmatrix} \hat{L}I_{\mathcal{U}}^T & \mathbf{r} - \mathbf{x} \\ I_{\mathcal{U}^c}^T(I + \hat{L}I_{\mathcal{U}}^T) & I_{\mathcal{U}^c}^T\mathbf{r} \\ 0 & \hat{E} \end{bmatrix} \tag{46}
 \end{aligned}$$

$$= \text{rank} \begin{bmatrix} \hat{L}I_{\mathcal{U}}^T & \mathbf{r} - \mathbf{x} \\ I_{\mathcal{U}^c}^T & I_{\mathcal{U}^c}^T\mathbf{x} \\ 0 & \hat{E} \end{bmatrix} \tag{47}$$

$$= \text{rank} \begin{bmatrix} \hat{L}I_{\mathcal{U}}^T & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} + \text{rank} \begin{bmatrix} I_{\mathcal{U}^c}^T & I_{\mathcal{U}^c}^T\mathbf{x} \end{bmatrix} \tag{48}$$

$$= \text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} + n - \mu \tag{49}$$

where (46) follows from $I_{\mathcal{U}}^T[I + \hat{L}I_{\mathcal{U}}^T \quad \mathbf{r}] = 0$, (48) follows by subtracting $I_{\mathcal{U}^c}^T[\hat{L}I_{\mathcal{U}}^T \quad \mathbf{r} - \mathbf{x}]$ from $[I_{\mathcal{U}^c}^T(I + \hat{L}I_{\mathcal{U}}^T) \quad I_{\mathcal{U}^c}^T\mathbf{r}]$, (48) follows from $I_{\mathcal{U}^c}^T I_{\mathcal{U}}^c = I_{(n-\mu) \times (n-\mu)}$ and $\hat{L}I_{\mathcal{U}}^T I_{\mathcal{U}^c} = 0$ (i.e., the two matrices in (48) have row spaces that intersect trivially), and (49) follows by deleting the all-zero columns.

Since $\text{rank } X + \text{rank } Y = 2n - \mu + \delta$, we have

$$\begin{aligned}
 d_S(\langle X \rangle, \langle Y \rangle) &= 2 \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} - \text{rank } X - \text{rank } Y \\
 &= 2 \text{rank} \begin{bmatrix} \hat{L} & \mathbf{r} - \mathbf{x} \\ 0 & \hat{E} \end{bmatrix} - \mu - \delta.
 \end{aligned}$$

■

D. Proof of Proposition 10

Before proving Proposition 10, we need the following lemma.

Lemma 14: For $X \in \mathbb{F}_q^{n \times m}$ and $Y \in \mathbb{F}_q^{N \times m}$ we have

$$\min_{A \in \mathbb{F}_q^{N \times n}} \text{rank}(Y - AX) = \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} - \text{rank } X$$

and for $X \in \mathbb{F}_q^{n \times m}$ and $Y \in \mathbb{F}_q^{n \times M}$ we have

$$\min_{B \in \mathbb{F}_q^{m \times M}} \text{rank}(Y - XB) = \text{rank} \begin{bmatrix} X & Y \end{bmatrix} - \text{rank } X.$$

Proof: For any $A \in \mathbb{F}_q^{N \times n}$, we have

$$\text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} = \text{rank} \begin{bmatrix} X \\ Y - AX \end{bmatrix} \leq \text{rank } X + \text{rank}(Y - AX)$$

which gives a lower bound on $\text{rank}(Y - AX)$. We now prove that this lower bound is achievable.

Let $Z \in \mathbb{F}_q^{t \times m}$ be such that $\langle Y \rangle = \langle X \rangle \cap \langle Y \rangle \oplus \langle Z \rangle$, where $t = \text{rank } Y - \omega$ and $\omega = \dim \langle X \rangle \cap \langle Y \rangle$.

Let $B \in \mathbb{F}_q^{\omega \times n}$ be such that $\langle BX \rangle = \langle X \rangle \cap \langle Y \rangle$. We can write $Y = T \begin{bmatrix} BX \\ Z \end{bmatrix}$ for some full-rank

$T \in \mathbb{F}_q^{N \times (\omega+t)}$. Now, let $A = T \begin{bmatrix} B \\ 0 \end{bmatrix} \in \mathbb{F}_q^{N \times n}$. Then

$$\begin{aligned} \text{rank}(Y - AX) &= \text{rank} \left(T \begin{bmatrix} BX \\ Z \end{bmatrix} - T \begin{bmatrix} BX \\ 0 \end{bmatrix} \right) \\ &= \text{rank} \left(T \begin{bmatrix} 0 \\ Z \end{bmatrix} \right) = \text{rank } Z \\ &= \text{rank } Y - \dim(\langle X \rangle \cap \langle Y \rangle) \\ &= \text{rank} \begin{bmatrix} X \\ Y \end{bmatrix} - \text{rank } X. \end{aligned}$$

This proves the first statement. The second statement is just the transposed version of the first one. ■

Proof of Proposition 10: Let

$$\epsilon' = \min_{E^{(1)}, L^{(2)}} \text{rank}(e - \hat{L}E^{(1)} - L^{(2)}\hat{E}).$$

We first show the equivalence of 1) and 2). From Lemma 14, we have

$$\min_{L^{(2)}} \text{rank}(e - \hat{L}E^{(1)} - L^{(2)}\hat{E}) = \text{rank} \begin{bmatrix} e - \hat{L}E^{(1)} \\ \hat{E} \end{bmatrix} - \text{rank } \hat{E}.$$

Similarly, from Lemma 14 we have

$$\begin{aligned} \min_{E^{(1)}} \text{rank} \begin{bmatrix} e - \hat{L}E^{(1)} \\ \hat{E} \end{bmatrix} &= \min_{E^{(1)}} \text{rank} \left(\begin{bmatrix} e \\ \hat{E} \end{bmatrix} - \begin{bmatrix} \hat{L} \\ 0 \end{bmatrix} E^{(1)} \right) \\ &= \text{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{E} \end{bmatrix} - \text{rank } \hat{L}. \end{aligned}$$

Thus,

$$\epsilon' = \text{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{E} \end{bmatrix} - \mu - \delta$$

and the equivalence is shown.

Now, observe that the statement in 3) is equivalent to the statement that $\tau^* - \mu - \delta$ is the minimum value of ϵ for which there exist $E^{(1)} \in \mathbb{F}_q^{\mu \times m}$, $L^{(2)} \in \mathbb{F}_q^{n \times \delta}$, $L^{(3)} \in \mathbb{F}_q^{n \times \epsilon}$ and $E^{(3)} \in \mathbb{F}_q^{\epsilon \times m}$ satisfying

$$e = \hat{L}E^{(1)} + L^{(2)}\hat{E} + L^{(3)}E^{(3)}.$$

To show the equivalence of 2) and 3), we will show that $\epsilon' = \epsilon''$, where

$$\epsilon'' = \min_{\substack{\epsilon, E^{(1)}, L^{(2)}, L^{(3)}, E^{(3)}: \\ e = \hat{L}E^{(1)} + L^{(2)}\hat{E} + L^{(3)}E^{(3)}}} \epsilon.$$

We can rewrite ϵ'' as

$$\begin{aligned} \epsilon'' &= \min_{E^{(1)}, L^{(2)}} \min_{\substack{\epsilon, L^{(3)}, E^{(3)}: \\ e - \hat{L}E^{(1)} - L^{(2)}\hat{E} = L^{(3)}E^{(3)}}} \epsilon \\ &= \min_{E^{(1)}, L^{(2)}} \text{rank}(e - \hat{L}E^{(1)} - L^{(2)}\hat{E}) \\ &= \epsilon'. \end{aligned} \tag{50}$$

where (50) follows from (1). This shows the equivalence between 2) and 3). ■