

Error-Correcting Regenerating and Locally Repairable Codes via Rank-Metric Codes

Natalia Silberstein, Ankit Singh Rawat and Sriram Vishwanath

Abstract

This paper presents and analyzes a novel concatenated coding scheme for enabling error resilience in two distributed storage settings: one being storage using existing regenerating codes and the second being storage using locally repairable codes. The concatenated coding scheme brings together a maximum rank distance (MRD) code as an outer code and either a globally regenerating or a locally repairable code as an inner code. Also, error resilience for combination of locally repairable codes with regenerating codes is considered. This concatenated coding system is designed to handle two different types of adversarial errors: the first type includes an adversary that can replace the content of an affected node only once; while the second type studies an adversary that is capable of polluting data an unbounded number of times. The paper establishes an upper bound on the resilience capacity for a locally repairable code and proves that this concatenated coding scheme attains the upper bound on resilience capacity for the first type of adversary. Further, the paper presents mechanisms that combine the presented concatenated coding scheme with subspace signatures to achieve error resilience for the second type of errors.

I. INTRODUCTION

Distributed storage systems (DSS) have gained importance over recent years as dependable, easily accessible and well administrated cloud resource for both individuals and businesses. There are multiple research issues that are unique to DSS; some of which are logistical and market driven, while others relate with their underlying design. A primary concern in designing DSS is to ensure resilience to failures, as it is desirable that a user (*data collector*) can retrieve the stored data even in the presence of node failures. As studied in the pioneering work by Dimakis et al. [1], coding introduces redundancy to a storage system in the most efficient manner to enable resilience to failures. In [1], Dimakis et al. go one step further: when a single node fails, they propose reconstructing the data stored on the failed node in order to maintain the required level of redundancy in the system. This process of data reconstruction for a failed node is called the *node repair process* [1]. During a node repair process, the node which is added to the system to replace the failed node, downloads data from a set of surviving nodes to reconstruct the lost (or its equivalent) data.

Regenerating codes and *locally repairable codes (LRCs)* are two families of codes that are especially designed to allow for efficient node repairs in DSS. In particular, regenerating codes are designed to reduce *repair bandwidth*, i.e., the amount of data downloaded from surviving nodes during the node repair process. On the other hand, LRCs are designed to have a small number of nodes participating in the node repair process. The constructions for these two families of codes can be found in [1]–[22] and references therein.

Although failure resilience is of critical importance as failure of storage nodes is commonplace in storage systems, there are multiple other design considerations that merit study in conjunction with failure resilience. These include security, error resilience, update efficiency and load balancing. In this paper, we address the issue of instilling error resilience in DSS, particularly against adversarial errors. In particular, we model and present coding methodologies that allow for a data collector to correctly decode data even in presence of adversarial errors.

In [1], Dimakis et al. establish close connections between DSS with node repairs and the network coding problem. Thus, it is natural to apply the techniques used for error correction in network coding for DSS with node repairs.

N. Silberstein is with the Department of Computer Science, Technion — Israel Institute of Technology, Haifa 32000, Israel (email: natalys@cs.technion.ac.il).

A. S. Rawat and S. Vishwanath are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA (e-mail: ankitsr@utexas.edu, sriram@austin.utexas.edu).

This paper was presented in part at the 2012 50th Annual Allerton Conference on Communication, Control, and Computing.

Rank-metric codes are known to be a powerful solution to error correction problem in network coding [23]–[27]. The primary idea of this paper is to apply a similar technique, based on rank-metric codes, for error resilience in DSS.

In this paper, we propose a novel concatenated coding scheme, based on rank-metric codes, which provides resilience against adversarial errors in both regenerating codes and LRCs based DSS. Moreover, we also consider error resilience in DSS employing combination of LRCs and regenerating codes [18], [20].

The problem of reliability of DSS against adversarial errors was considered in [28]–[31]. In particular, in [30], Pawar et al. derive upper bounds on the amount of data that can be stored on the system and reliably made available to a data collector when bandwidth optimal node repair is performed, and present coding strategies that achieve the upper bound for a particular range of system parameters, namely in the bandwidth-limited regime. A related but different problem of securing stored data against passive eavesdroppers is addressed in [18], [30], [32], [33].

In this paper, we study the notion of an *omniscient adversary* who can observe all nodes and has full knowledge of the coding scheme employed by the system [30]. As in [30], we assume an upper bound on the number of nodes that can be controlled by such an omniscient adversary. We classify adversarial attacks by such an adversary into two classes:

- 1) *Static errors*: an omniscient adversary replaces the content of an affected node with nonsensical (unrelated) information *only once*. The affected node uses this *same* polluted information during all subsequent repair and data collection processes. Static errors represent a common type of data corruption due to wear out of storage devices, such as latent disk errors or other physical defects of the storage media, where the data stored on a node is permanently distorted.
- 2) *Dynamic errors*: an omniscient adversary may replace the content of an affected node *each time* the node is asked for its data during data collection or repair process. This kind of errors captures any malicious behaviour, hence is more difficult to manage in comparison with static errors.

We present a novel concatenated coding scheme for DSS which provides resilience against these two classes of attacks and allow for either optimal repair bandwidth or optimal local node repairs. In our scheme for an optimal repair bandwidth DSS, the content to be stored is first encoded using a maximum rank distance (MRD) code. The output of this outer code is further encoded using a regenerating code, which allows for bandwidth efficient node repairs. Using an MRD code, which is an optimal rank-metric code, allows us to quantify the errors introduced in the system using their rank as opposed to their Hamming weights. The dynamic nature of the DSS causes a large number of nodes to get polluted even by a single erroneous node, as false information spreads from node repairs. Thus, a single polluted node infects many others, resulting in an error vector with a large Hamming weight. Using rank-metric codes can help alleviate this problem as the error that a data collector has to handle has a known rank, and can therefore be corrected by an MRD code with a sufficient rank distance. Using an (n, k) bandwidth efficient MDS array code as inner code facilitates bandwidth efficient node repair in the event of a single node failure and allows the data collector to recover the original data from any subset of k storage nodes. In this paper, we use exact-regenerating linear bandwidth efficient codes operating at the minimum-storage regenerating (MSR) point [1]. However, our construction can be utilized for regenerating codes operating at any other point of repair bandwidth vs. per node storage trade-off.

Further, we focus on error resilience for locally repairable DSS. We present a bound on the resilience capacity for such DSS. To the best of our knowledge, this is the first work which considers the issue of error resilience for locally repairable codes. Then we show that our scheme based on MRD codes can be applied to LRC codes as well. In addition, we consider error resilience in codes with optimal bandwidth *local* regeneration, where MSR or MBR codes are utilized as local codes [18], [20], [34].

Our coding schemes are directly applicable to the *static error* model and are optimal in terms of amount of data that can be reliably stored on a DSS under static error model. The model with *dynamic errors* is more complicated, as it permits a single malicious node to change its pollution pattern, and introduce an arbitrarily large error both in Hamming weight and in rank. For dynamic error model, we present two solutions based on the concatenated coding scheme employed for static error model. (We consider only the solutions for a secure regenerating code, as for a secure locally repairable code the same ideas can be applied). One solution is to exploit the inherent redundancy in the encoded data due to outer code, i.e., an MRD code, and perform error free node repair even in the presence of adversarial nodes. This solution, a naïve method, is optimal for a specific choice of parameters. An alternative solution combines our concatenated coding scheme with subspace signature based cryptographic schemes to control

the amount (rank) of pollution (error) introduced by an adversarial node. We employ the signature scheme by Zhao et al. [35], which essentially reduces the dynamic error model to the one similar to a static error model, and helps us bound the rank of error introduced by an adversarial node. Note that hash function based solutions have previously been presented in the context of DSS to deal with errors [28], [30]. While promising, these hash functions based approaches provide only probabilistic guarantees for pollution containment.

The rest of the paper is organized as follows: In Section II, we start with a description of our system model and provide a brief overview of rank-metric, regenerating and locally repairable codes. In Section III, we consider the static error model. First, we describe the construction of our storage scheme which allows for bandwidth efficient repairs and prove its error resilience. Second, we consider error resilience in LRCs. We provide an upper bound on the resiliency capacity and present modifications of our scheme which enable optimal local repairs. In Section IV, we address the dynamic error model. We conclude with Section V, where we list a set of open problems.

II. PRELIMINARIES

A. System model

Let \mathcal{M} be the size of a file \mathbf{f} over a finite field \mathbb{F} that needs to be stored on a DSS with n nodes. Each node contains α symbols over \mathbb{F} . A data collector reconstructs the original file \mathbf{f} by downloading the data stored on any set of k out of n nodes. This property of a DSS is called ‘any k out of n ’ property, and we use (n, k) -DSS to represent a storage system which has ‘any k out of n ’ property.

B. Rank-metric codes

All the constructions of coding schemes for distributed storage systems provided in this paper are based on a family of error correcting codes in rank metric, called Gabidulin codes.

Let \mathbb{F}_{q^m} be an extension field of \mathbb{F}_q . Since \mathbb{F}_{q^m} can be also considered as an m -dimensional vector space over \mathbb{F}_q , any element $\gamma \in \mathbb{F}_{q^m}$ can be represented as the vector $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_m) \in \mathbb{F}_q^m$, such that $\gamma = \sum_{i=1}^m b_i \gamma_i$, for a fixed basis $\{b_1, \dots, b_m\}$ of the field extension. Similarly, any vector $\mathbf{v} = (v_1, \dots, v_N) \in \mathbb{F}_{q^m}^N$ can be represented by an $m \times N$ matrix $\mathbf{V} = [v_{i,j}]$ over \mathbb{F}_q , where each entry v_i of \mathbf{v} is expanded as a column vector $(v_{i,1}, \dots, v_{i,m})^T$. The *rank* of a vector $\mathbf{v} \in \mathbb{F}_{q^m}^N$, denoted by $\text{rank}(\mathbf{v})$, is defined as the rank of the $m \times N$ matrix \mathbf{V} over \mathbb{F}_q . Similarly, for two vectors $\mathbf{v}, \mathbf{u} \in \mathbb{F}_{q^m}^N$, the *rank distance* is defined by

$$d_R(\mathbf{v}, \mathbf{u}) = \text{rank}(\mathbf{V} - \mathbf{U}).$$

An $[N, K, D]_{q^m}$ rank-metric code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^N$ is a linear block code over \mathbb{F}_{q^m} of length N , dimension K and minimum rank distance D . A rank-metric code that attains the Singleton bound $D \leq N - K + 1$ for the rank metric is called a *maximum rank distance* (MRD) code [36]–[38]. For $m \geq N$, a construction of MRD codes was presented by Gabidulin [37]. Similar to Reed-Solomon codes, Gabidulin codes can be obtained by evaluation of polynomials; however, for Gabidulin codes, a special family of polynomials *linearized polynomials* is used. A linearized polynomial $f(x)$ over \mathbb{F}_{q^m} of q -degree t has the form $f(x) = \sum_{i=0}^t a_i x^{q^i}$, where $a_i \in \mathbb{F}_{q^m}$, and $a_t \neq 0$.

Remark 1. Note, that evaluation of a linearized polynomial is an \mathbb{F}_q -linear transformation from \mathbb{F}_{q^m} to itself, i.e., for any $a, b \in \mathbb{F}_q$ and $\gamma_1, \gamma_2 \in \mathbb{F}_{q^m}$, we have $f(a\gamma_1 + b\gamma_2) = af(\gamma_1) + bf(\gamma_2)$ [39].

A codeword of an $[N, K, D = N - K + 1]_{q^m}$ Gabidulin code \mathcal{C}^{Gab} is defined as $\mathbf{c} = (f(g_1), f(g_2), \dots, f(g_N)) \in \mathbb{F}_{q^m}^N$, for $m \geq N$, where $f(x)$ is a linearized polynomial over \mathbb{F}_{q^m} of q -degree at most $K - 1$ with K message symbols as its coefficients, and the evaluation points $g_1, \dots, g_N \in \mathbb{F}_{q^m}$ are linearly independent over \mathbb{F}_q [37].

1) *Rank Errors and Rank Erasures Correction:* Let $\mathcal{C}^{\text{Gab}} \subseteq \mathbb{F}_{q^m}^N$ be a Gabidulin code with the minimum distance D . Let $\mathbf{c} \in \mathcal{C}^{\text{Gab}}$ be the transmitted codeword and let $\mathbf{r} = \mathbf{c} + \mathbf{e}_{\text{total}}$ be the received word. The code \mathcal{C}^{Gab} can correct any vector error of the form

$$\mathbf{e}_{\text{total}} = \mathbf{e}_{\text{error}} + \mathbf{e}_{\text{erasure}} = (e_1 \mathbf{u}_1 + \dots + e_t \mathbf{u}_t) + (r_1 \mathbf{v}_1 + \dots + r_s \mathbf{v}_s), \quad (1)$$

as long as $2t + s \leq D - 1$. The first part $\mathbf{e}_{\text{error}}$ is called a *rank error* of rank t , where $e_i \in \mathbb{F}_{q^m}$ are linearly independent over the base field \mathbb{F}_q , unknown to the decoder, and $\mathbf{u}_i \in \mathbb{F}_q^N$ are linearly independent vectors of length N , unknown to the decoder. The second part $\mathbf{e}_{\text{erasure}}$ is called a *rank erasure*, where $r_i \in \mathbb{F}_{q^m}$ are linearly

y_1	y_2	y_3	y_4	y_5
c_1	c_5	c_9	$c_1 + c_5 + c_9$	$c_1 + 2c_7 + c_{10}$
c_2	c_6	c_{10}	$c_2 + c_6 + c_{10}$	$c_2 + 2c_8 + c_9$
c_3	c_7	c_{11}	$c_3 + c_7 + c_{11}$	$c_3 + c_5 + c_{12}$
c_4	c_8	c_{12}	$c_4 + c_8 + c_{12}$	$c_4 + c_6 + 2c_{11}$

(a)

y_1	y_2	y_3	y_4	y_5
$c_1 + e_1$	$c_5 - e_1$	c_9	$c_1 + c_5 + c_9$	$c_1 + 2c_7 + c_{10}$
$c_2 + e_2$	$c_6 - e_2$	c_{10}	$c_2 + c_6 + c_{10}$	$c_2 + 2c_8 + c_9$
$c_3 + e_3$	$c_7 - e_1/2$	c_{11}	$c_3 + c_7 + c_{11}$	$c_3 + c_5 + c_{12}$
$c_4 + e_4$	$c_8 - e_2/2$	c_{12}	$c_4 + c_8 + c_{12}$	$c_4 + c_6 + 2c_{11}$

(b)

Fig. 1: Illustration of the second node repair process in $(5, 3)$ Zigzag code: (a) for error free system, (b) for system with erroneous information at the first storage node.

independent over the base field \mathbb{F}_q , unknown to the decoder, and $\mathbf{v}_i \in \mathbb{F}_q^N$ are linearly independent vectors of length N , and known to the decoder. (In this paper we are interested only in rank *column* erasures, so the rank *row* erasures are not considered). [37], [40] present decoding algorithms for rank-metric codes.

Note that an $[N, K, D]_{q^m}$ MRD code is in particular an MDS code over \mathbb{F}_{q^m} , and then can correct erasures of any $D - 1$ symbols in a codeword. Such symbols erasures are in particular column erasures, when the codeword is considered in the matrix representation.

C. Vector codes

A linear $[n, \mathcal{M}, d_{\min}, \alpha]_{\mathbb{F}}$ vector code C of length n over a finite field \mathbb{F} is defined as a linear subspace of $\mathbb{F}^{\alpha n}$ of dimension \mathcal{M} . The symbols \mathbf{c}_i , $1 \leq i \leq n$, of a codeword $\mathbf{c} \in C$ belong to \mathbb{F}^{α} , i.e., are vectors or *blocks* of size α . The minimum distance d_{\min} of C is defined as the minimum Hamming distance over \mathbb{F}^{α} .

Vector codes are also known as *array* codes. An $[n, \mathcal{M}, d_{\min}, \alpha]_{\mathbb{F}}$ array code is called *MDS array code* if $\alpha|\mathcal{M}$ and $d_{\min} = n - \frac{\mathcal{M}}{\alpha} + 1$. Constructions for MDS array codes can be found e.g. in [41], [42].

In order to store a file $\mathbf{f} \in \mathbb{F}^{\mathcal{M}}$ on a DSS using a vector code C , \mathbf{f} is first encoded to a codeword $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n) \in C$. Each symbol (block) $\mathbf{c}_i \in \mathbb{F}^{\alpha}$ of the codeword is then stored on a distinct node.

D. Regenerating codes

Regenerating codes are a family of vector codes for an (n, k) -DSS that allow for efficient repair of failed nodes. When a node fails, its content can be reconstructed by downloading $\beta \leq \alpha$ symbols from each node in a set of d , $k \leq d \leq n - 1$, surviving nodes. We denote by $[n, \mathcal{M}, d_{\min} = n - k + 1, \alpha, \beta, d]_{\mathbb{F}}$ a linear regenerating code. Note that data can be reconstructed from any k symbols of a codeword and therefore the minimum distance is $d_{\min} = n - k + 1$. A trade-off between storage per node α and *repair bandwidth* $\gamma \triangleq d\beta$ was established in [1]. Two classes of codes that achieve two extreme points of this trade-off are known as *minimum storage regenerating (MSR)* codes and *minimum bandwidth regenerating (MBR)* codes. The parameters (α, γ) for MSR and MBR codes are given by $\left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}d}{k(d-k+1)}\right)$ and $\left(\frac{2\mathcal{M}d}{2kd-k^2+k}, \frac{2\mathcal{M}d}{2kd-k^2+k}\right)$, respectively [1].

In this paper we focus on the family of linear MSR codes. Note that these codes are also MDS array codes. We refer to these codes as $[n, \mathcal{M} = \alpha k, d_{\min} = n - k + 1, \alpha, \beta = \frac{\alpha}{d-k+1}, d]_{\mathbb{F}}$ optimal repair MDS array codes.

Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k) \in \mathbb{F}^{\alpha k}$, $k = \frac{\mathcal{M}}{\alpha}$, be an information vector (a file), $\mathbf{x}_i \in \mathbb{F}^{\alpha}$ is a block of size α , for all $1 \leq i \leq k$. These k blocks are encoded into n encoded blocks $\mathbf{y}_i \in \mathbb{F}^{\alpha}$, $1 \leq i \leq n$, stored in n nodes of size α , in the following way:

$$\mathbf{y} = \mathbf{x}\mathbf{G},$$

where $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$ and the generator matrix \mathbf{G} of an MSR code is an $k \times n$ block matrix over \mathbb{F} with blocks of size $\alpha \times \alpha$ given by:

$$\mathbf{G} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{k,1} & A_{k,2} & \dots & A_{k,n} \end{bmatrix}.$$

Note that any blocks submatrix of \mathbf{G} of size $k \times k$ is of the full rank. For a systematic code, we have $\mathbf{y}_i = \mathbf{x}_i$, for $1 \leq i \leq k$, and a parity node j , $k+1 \leq j \leq n$, stores $\mathbf{y}_j = \sum_{i=1}^k \mathbf{x}_i A_{i,j}$.

When a node i fails, each node j in a set of d surviving nodes that are contacted to repair this node sends a vector of length β given by $\mathbf{y}_j V_{j,i}$, where $V_{j,i} \in \mathbb{F}^{\alpha \times \beta}$ is a repair matrix used by node j to perform repair of node i .

1) *Example of Optimal Repair MDS Array Codes:* In the following, we present an example of optimal repair MDS array codes for DSS, which we use in Section III-A to illustrate our coding scheme.

Example 1. (*Zigzag code [6]*). This class of optimal repair MDS array codes [6] is based on generalized permutation matrices. For an $[n = 5, \mathcal{M} = 12, d_{\min} = 3, \alpha = 4, \beta = 2, d = 4]_q$ zigzag code presented in Fig. 1a, first three nodes are systematic nodes which store the data $(c_1, c_2, \dots, c_{12})$, each one $\alpha = 4$ information symbols. The block generator matrix for this code is given by

$$\mathbf{G} = \begin{bmatrix} I & 0 & 0 & I & I \\ 0 & I & 0 & I & A_2 \\ 0 & 0 & I & I & A_3 \end{bmatrix},$$

where I and 0 denote the identity matrix and all-zero matrix of size 4×4 , respectively; and

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Fig. 1a describes node repair process for this code. When the second node fails, the newcomer node downloads the symbols from the shaded locations at the surviving nodes.

E. Locally repairable codes

Locally repairable codes (LRCs) are a family of codes for DSS that allow to reduce the number of nodes participating in the node repair process. LRCs are defined as follows.

We say that an $[n, \mathcal{M}, d_{\min}, \alpha]_{\mathbb{F}}$ vector code C is a (r, δ, α) locally repairable code if for each symbol $\mathbf{c}_i \in \mathbb{F}^{\alpha}$, $1 \leq i \leq n$, of a codeword $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_n) \in C$, there exists a set of indices $\Gamma(i)$ such that

- $i \in \Gamma(i)$
- $|\Gamma(i)| \leq r + \delta - 1$
- $d_{\min}(C|_{\Gamma(i)}) \geq \delta$, where $C|_{\Gamma(i)}$ denotes the puncturing of C on the set $[n] \setminus \Gamma(i)$ of coordinates.

Note that the last two properties imply that each element $j \in \Gamma(i)$ can be written as a function of a set of at most r elements in $\Gamma(i)$ (not containing j).

It was proven in [18], [20] that the minimum distance of an (r, δ, α) LRC of length n and dimension \mathcal{M} satisfies

$$d_{\min}(C) \leq n - \left\lfloor \frac{\mathcal{M}}{\alpha} \right\rfloor + 1 - \left(\left\lfloor \frac{\mathcal{M}}{r\alpha} \right\rfloor - 1 \right) (\delta - 1). \quad (2)$$

We say that an (r, δ, α) LRC for an $(n, n - d_{\min} + 1)$ -DSS is *optimal* if its minimum distance d_{\min} attains the bound (2). In this paper we consider the construction of optimal (r, δ, α) LRCs from [17] (for $\alpha = 1$) and its generalization from [18]. This construction is based on concatenation of Gabidulin codes and MDS array codes: a file \mathbf{f} over $\mathbb{F} = \mathbb{F}_{q^N}$ of size $\mathcal{M} \geq r\alpha$ is first encoded using an $[N, \mathcal{M}, D]_{q^N}$ Gabidulin code. The codeword of the Gabidulin code is then partitioned into local disjoint groups, each of size $r\alpha$, and each local group is then encoded using an $[(r + \delta - 1), r, \delta, \alpha]_q$ MDS array code over \mathbb{F}_q . (If $r\alpha \nmid N$, there is a group of size $r'\alpha < r\alpha$, which is then encoded by using an $[(r' + \delta - 1), r', \delta, \alpha]_q$ MDS array code). A code obtained by this construction is optimal if $(r + \delta - 1)|n$, $N = \frac{nr\alpha}{r + \delta - 1}$ and $q \geq (r + \delta - 1)$. For a case when $(r + \delta - 1) \nmid n$, see the details in [18].

1) *Example of Optimal LRC:* The following example of optimal LRC will be used further for illustration of our error-correcting coding scheme.

Example 2. We consider a DSS with the following parameters:

$$(\mathcal{M}, n, r, \delta, \alpha) = (28, 15, 3, 3, 4).$$

Let \mathbf{f} be a file of size 28 over $\mathbb{F}_{q^{36}}$, $q \geq 5$. Let $N = \frac{15 \cdot 3 \cdot 4}{3 + 3 - 1} = 36$ and $(a_1, \dots, a_{12}, b_1, \dots, b_{12}, c_1, \dots, c_{12})$ be a codeword of a $[36, 28, 9]_{q^{36}}$ Gabidulin code C^{Gab} , which is obtained by encoding $\mathcal{M} = 28$ symbols over $\mathbb{F} = \mathbb{F}_{q^{36}}$

of the original file. The Gabidulin codeword is then partitioned into three groups (a_1, \dots, a_{12}) , (b_1, \dots, b_{12}) , and (c_1, \dots, c_{12}) . Encoded symbols in each group are stored on three storage nodes as shown in Fig. 2. In the second stage of encoding, a $[5, 3 \cdot 4, 3, 4]_q$ MDS array code over \mathbb{F}_q is applied on each local group to obtain $\delta - 1 = 2$ parity nodes per local group. The coding scheme is illustrated in Fig. 2.

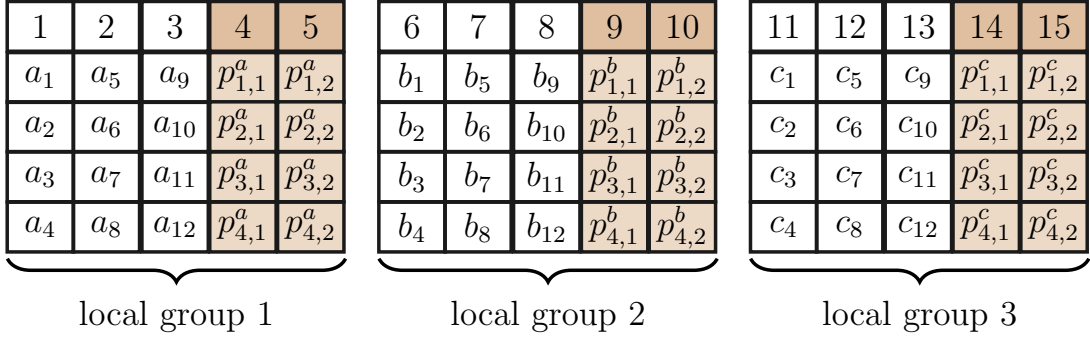


Fig. 2: Example of an $(r = 3, \delta = 3, \alpha = 4)$ LRC with $n = 15$ and $d_{\min} = 5$.

By (2) we have $d_{\min}(C^{\text{loc}}) \leq 5$. One can check that every 4 nodes failures (which is equivalent to at most 8 rank erasures [18]) can be corrected by this code, and thus it has minimum distance 5. In addition, when a single node fails, it can be repaired by using the data stored on any three other nodes from the same group.

F. Adversarial errors models

We assume the presence of an omniscient active adversary which can observe all nodes of (n, k) -DSS and has full knowledge of the coding scheme employed by the system but can modify at most t nodes, $2t < k$, similarly to [30].

Our goal is to design a coding scheme for an (n, k) -DSS that allows to deliver the original data to a data collector even in presence of t nodes ($t < k/2$) which are fully controlled by an adversary.

In this paper we consider two classes of adversarial attacks:

- 1) *Static errors*: an adversary replaces the content of each affected node with nonsensical information *only once*. The affected node uses this *same* polluted information during all subsequent repair and data collection processes.
- 2) *Dynamic errors*: an adversary may replace the content of an affected node *each time* the node is asked for the data during data collection or repair process.

As it was mentioned in Introduction, static errors represent a common type of data corruption due to wear out of storage devices, such as latent disk errors or other physical defects of the storage media, where the data stored on a node is permanently distorted. Dynamic errors represent the most general scenario and hence is more difficult to manage as compared to static errors.

The upper bound on the amount of data that can be stored reliably on an (n, k) -DSS with $t < \frac{k}{2}$ general corrupted nodes (i.e., t dynamic errors) which employs a regenerating code was presented by Pawar et al. [30]. This amount of data is called *resilience capacity* and denoted by $C_t(\alpha, \beta, d)$. The bound on the resilience capacity is given by

$$C_t(\alpha, \beta, d) \leq \sum_{i=2t+1}^k \min\{(d - i + 1)\beta, \alpha\}. \quad (3)$$

Remark 2. Although the static error model is less general than the model considered in [30], the upper bound in (3) applies to the static model as well. Pawar et al. [30] obtain the upper bound in (3) by evaluating a cut of the information flow graph corresponding to a particular node failure sequence, pattern of nodes under adversarial attack and data collector. This information flow graph is also valid in the context of the static error model. Consequently, its cut which provides an upper bound on the information flow and represents the amount of data that can be reliably stored on DSS is applicable to the static error model as well. As shown in this paper, this bound is tight for static model at the MSR point, which might not be the case for the general error model considered in [30].

III. CODING SCHEMES FOR STATIC ERROR MODEL

In this section we consider the static error model. First, we present our coding scheme for DSS which employ optimal repair MDS array codes. Next, we provide an upper bound on the resilience capacity for locally repairable codes and present a coding scheme for DSS which employ optimal locally repairable codes. We prove the error tolerance of the proposed schemes under the *static error model*. We illustrate the idea by using examples from Section II-D1 and Section II-E1 and prove that these constructions are optimal for the static error model. Finally, we consider error resilience in LRCs with local regeneration.

A. Error resilience in optimal repair MDS array codes

1) *Construction of Error-Correcting Optimal Repair MDS Array Codes*: First, we present a concatenated coding scheme which is based on a Gabidulin code as an outer code and an optimal repair MDS array code as an inner code:

Construction I. Let n, k, α, β, d be the given parameters and let \mathcal{M}, N, m be the positive integers such that $N = \alpha k$ and $\mathcal{M} \leq N \leq m$. Consider a file \mathbf{f} over $\mathbb{F} = \mathbb{F}_{q^m}$ of size \mathcal{M} . We encode the file in two steps before storing it on an (n, k) -DSS. First, the file is encoded using an $[N, \mathcal{M}, D = N - \mathcal{M} + 1]_{q^m}$ Gabidulin code \mathcal{C}^{Gab} , with N evaluation points g_1, \dots, g_N from \mathbb{F} , which are linearly independent over \mathbb{F}_q , as described in Section II-B. Second, the codeword $\mathbf{c}_f = (f(g_1), \dots, f(g_N)) \in \mathcal{C}^{\text{Gab}}$ that corresponds to the file \mathbf{f} is encoded using an $[n, N, d_{\min}, \alpha, \beta, d]_{\mathbb{F}_q}$ optimal repair MDS array (systematic) code \mathcal{C}^{MDS} over \mathbb{F}_q , where $N = \alpha k$, $d_{\min} = n - k + 1$, and $\beta = \frac{\alpha}{d - k + 1}$, as described in Section II-D: the codeword \mathbf{c}_f is partitioned into k blocks of size α , which are encoded into n blocks of size α (over \mathbb{F}) and then stored on n system nodes.

Note, that we use the optimal repair MDS array code defined \mathcal{C}^{MDS} over $\mathbb{F}_q \subseteq \mathbb{F}$, i.e., its generator matrix is over \mathbb{F}_q , and during the process of node repair, a set of surviving nodes transmits linear combinations of the stored elements with the coefficients from \mathbb{F}_q .

The following lemma is useful to show the properties of the constructed code.

Lemma 1. *Any k different nodes of the proposed scheme contain evaluation of the linearized polynomial $f(x)$ (corresponding to the given file) at αk elements of \mathbb{F} which are linearly independent over \mathbb{F}_q . In other words, the content of any k nodes corresponds to a codeword of an $[N = \alpha k, \mathcal{M}, D = N - \mathcal{M} + 1]_{\mathbb{F}}$ Gabidulin code $\widehat{\mathcal{C}}^{\text{Gab}}$.*

Proof: Let g_1, \dots, g_N be the evaluation points of the Gabidulin code \mathcal{C}^{Gab} used in the construction. Consider a set $S = \{i_1, \dots, i_k\}$ of nodes. The content of these nodes is given by $\mathbf{c}_f G_S$, where $G_S = (a_{i,j})_{i,j=1}^{\alpha k}$ is a $\alpha k \times \alpha k$ submatrix of the generator matrix \mathbf{G} of \mathcal{C}^{MDS} which corresponds to the nodes in S , i.e., which consists of columns $\{(i_j - 1)\alpha + 1, \dots, i_j \alpha\}_{j=1}^k$. The j th coordinate of the vector $\mathbf{c}_f G_S$ of length αk is given by $(\mathbf{c}_f G_S)_j = (f(g_1), \dots, f(g_{\alpha k})) (a_{1,j}, \dots, a_{\alpha k,j})^T = \sum_{i=1}^{\alpha k} a_{i,j} f(g_i) = f(\sum_{i=1}^{\alpha k} a_{i,j} g_i)$, where $(a_{1,j}, \dots, a_{\alpha k,j})^T$ is the j th column of G_S and the last equality follows from the \mathbb{F}_q -linearity of f . Now consider the vector of the new evaluation points of f : $(\widehat{g}_1, \dots, \widehat{g}_{\alpha k}) = (\sum_{i=1}^{\alpha k} a_{i,1} g_i, \dots, \sum_{i=1}^{\alpha k} a_{i,\alpha k} g_i) = (g_1, \dots, g_{\alpha k}) G_S$.

Note that since \mathcal{C}^{MDS} is an MDS array code, then G_S is a full rank matrix over \mathbb{F}_q . Therefore, $\{\widehat{g}_1, \dots, \widehat{g}_{\alpha k}\}$ are linearly independent over \mathbb{F}_q if and only if $\{g_1, \dots, g_{\alpha k}\}$ are linearly independent over \mathbb{F}_q . Therefore, the observations $\mathbf{c}_f G_S$ are essentially evaluations of linearized polynomial at $k\alpha$ linearly independent points over \mathbb{F}_q from \mathbb{F} , which correspond to a codeword of an $[N = \alpha k, \mathcal{M}, D = N - \mathcal{M} + 1]_{\mathbb{F}}$ Gabidulin code $\widehat{\mathcal{C}}^{\text{Gab}}$. Note that $\widehat{\mathcal{C}}^{\text{Gab}}$ has the same parameters as \mathcal{C}^{Gab} , however, the evaluation points of these two codes are different. ■

The following theorem shows that if $2t\alpha + 1 \leq D$, then the proposed scheme tolerates up to t erroneous nodes, i.e., from any k nodes a data collector can retrieve the original data even in presence of an adversary which controls (modifies) t nodes. Note that the node repairs are performed exactly in the same way as in an error-free DSS.

Theorem 2. *Let t be the number of erroneous nodes in the system based on concatenation of Gabidulin and optimal repair MDS array codes from Construction I. If $2t\alpha + 1 \leq D$, then the original data can be recovered from any k nodes.*

Proof: Let $\mathbf{c}_f \in \mathbb{F}$ be the codeword in \mathcal{C}^{Gab} which corresponds to the file \mathbf{f} , and let $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$, $\mathbf{x}_i \in \mathbb{F}^\alpha$, be the partition of \mathbf{c}_f into k parts of size α each. Let $(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n)$, $\mathbf{y}_i \in \mathbb{F}^\alpha$ be the encoded by \mathcal{C}^{MDS} blocks stored on n nodes. Let $S = \{i_1, i_2, \dots, i_t\}$ be the set of indices of the erroneous nodes. Hence the i_j th node, $i_j \in S$, contains $\sum_{\ell=1}^k \mathbf{x}_\ell A_{\ell, i_j} + \mathbf{e}^{i_j}$, where $\mathbf{e}^{i_j} = (e_1^{i_j}, e_2^{i_j}, \dots, e_\alpha^{i_j}) \in \mathbb{F}^\alpha$ denotes an adversarial error introduced by

the i_j th node, and $A_{\ell,i_j} \in \mathbb{F}_q^{\alpha \times \alpha}$ are the blocks of the generator matrix of C^{MDS} . When the failed nodes are being repaired, the errors from adversarial nodes propagate to the repaired nodes. In particular, ℓ th node, $1 \leq \ell \leq n$, contains $\sum_{j=1}^k \mathbf{x}_j A_{j,\ell} + \sum_{j=1}^t \mathbf{e}^{i_j} B_\ell^{i_j}$, where $B_\ell^{i_j} \in \mathbb{F}_q^{\alpha \times \alpha}$ represents the propagation of error \mathbf{e}^{i_j} and depends on the specific choice of an optimal repair MDS array code. Suppose a data collector contacts k nodes indexed by $\mathcal{D} \subset [n]$ and downloads $\sum_{j=1}^k \mathbf{x}_j A_{j,i} + \sum_{j=1}^t \mathbf{e}^{i_j} B_i^{i_j}$ from node $i \in \mathcal{D}$.

- Case 1. If these k nodes are all systematic nodes, then we obtain $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k) + \mathbf{e}\mathbf{B} = \mathbf{c}_f + \mathbf{e}\mathbf{B}$, where $\mathbf{B} \in \mathbb{F}_q^{\alpha t \times \alpha k}$ is the blocks matrix with the (j, ℓ) th block of \mathbf{B} is given by $B_\ell^{i_j}$, $1 \leq \ell \leq k$, $1 \leq j \leq t$, and $\mathbf{e} = (\mathbf{e}^{i_1}, \mathbf{e}^{i_2}, \dots, \mathbf{e}^{i_t}) \in \mathbb{F}_q^{\alpha t}$.
- Case 2. If not all the k nodes are systematic, we obtain $\widehat{\mathbf{c}}_f + \mathbf{e}\widehat{\mathbf{B}}$, where the (j, ℓ) th block of the blocks matrix $\widehat{\mathbf{B}} \in \mathbb{F}_q^{\alpha t \times \alpha k}$ is given by $B_\ell^{i_j}$, $\ell \in \mathcal{D}$, $1 \leq j \leq t$, \mathbf{e} is defined as previously, and $\widehat{\mathbf{c}}_f$ is the codeword of the Gabidulin code \widehat{C}^{Gab} with the same parameters as C^{Gab} , according to Lemma 1.

In any case, since $\text{rank}(\mathbf{e}) \leq t\alpha$ over \mathbb{F}_q , and $D \geq 2t\alpha + 1$, a Gabidulin code can correct this error; consequently, by applying erasure decoding of the MDS array code (for Case 2) used as inner code the original data can be recovered. \blacksquare

Now we illustrate the idea of the construction with the help of the example of the optimal repair MDS array code for an $(5, 3)$ -DSS, presented in Section II-D1. We consider the case where an adversary pollutes the information stored on a single storage node and demonstrate that the rank of the error introduced by the adversary does not increase due to node repair dynamics under the static error model. Hence, a data collector can recover the correct original information using decoders for an MRD code C^{Gab} and MDS array code C^{MDS} .

Example 3. Let C^{Gab} be an $[12, 4, 9]_{q^{12}}$ Gabidulin code and let C^{MDS} be the $[5, 12, 3, 4, 2, 4]_q$ zigzag code from Example 1. First we encode a file $\mathbf{f} = (f_0, f_1, f_2, f_3) \in \mathbb{F}_{q^{12}}^4$ into a codeword $\mathbf{c}_f = (c_1, \dots, c_{12}) \in C^{\text{Gab}}$ by $c_i = f(g_i)$, $1 \leq i \leq 12$, where $\{g_1, \dots, g_{12}\} \subseteq \mathbb{F}_{q^{12}}$ are linearly independent over \mathbb{F}_q and $f(x) = \sum_{j=0}^3 f_j x^{q^j}$. Then we encode \mathbf{c}_f again by using C^{MDS} . The first three systematic nodes of $(5, 3)$ -DSS store a codeword \mathbf{c}_f , i.e., the content stored in i th systematic node, $1 \leq i \leq 3$, is $\mathbf{y}_i = (c_{4(i-1)+1}, \dots, c_{4i}) \in \mathbb{F}_q^4$. Let us assume that an adversary attacks the first storage node and introduces erroneous information. The erroneous information at the first node can be modeled as $\mathbf{y}_1 + \mathbf{e} = (c_1, c_2, c_3, c_4) + (e_1, e_2, e_3, e_4)$. Now assume that the second node fails. The system is oblivious to the presence of pollution at the first node, and employs an exact regeneration strategy to reconstruct the second node. The reconstructed node downloads the symbols from the shaded locations at the surviving nodes, as described in Fig. 1b, and solves a linear system of equations to obtain $(c_5, c_6, c_7, c_8) + (-e_1, -e_2, -2^{-1}e_1, -2^{-1}e_2)$, where 2^{-1} denotes the inverse element of 2 in \mathbb{F}_q , $q \geq 3$.

- Case 1: First assume that a data collector accesses the first three (systematic) nodes to recover the original data. The data collector now has $\tilde{\mathbf{c}} = \mathbf{c}_f + \mathbf{e}[I, B_2^1, \mathbf{0}]$, where I and $\mathbf{0}$ are 4×4 identity and zeroes matrices, respectively. Note that

$$B_2^1 = \begin{bmatrix} -1 & 0 & -2^{-1} & 0 \\ 0 & -1 & 0 & -2^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

- Case 2: Assume that a data collector accesses the first, second, and the fourth nodes to recover the original data. In this case, the data collector has $\tilde{\mathbf{c}} = \widehat{\mathbf{c}}_f + \mathbf{e}[I, B_2^1, \mathbf{0}]$, where I , $\mathbf{0}$, and B_2^1 are defined as previously, and $\widehat{\mathbf{c}}_f = (c_1, c_2, \dots, c_8, c_1 + c_5 + c_9, c_2 + c_6 + c_{10}, c_3 + c_7 + c_{11}, c_4 + c_8 + c_{12})$ is a codeword of a $[12, 4, 9]_{q^{12}}$ Gabidulin code \widehat{C}^{Gab} with evaluation points $\{g_1, g_2, \dots, g_8, g_1 + g_5 + g_9, g_2 + g_6 + g_{10}, g_3 + g_7 + g_{11}, g_4 + g_8 + g_{12}\}$.

In any case, $\tilde{\mathbf{c}}$ contains an error of rank at most 4. Since the minimum rank distance of codes C^{Gab} and \widehat{C}^{Gab} is 9, the codeword $\mathbf{c} \in C^{\text{Gab}}$ (for Case 1) and the codeword $\widehat{\mathbf{c}} \in \widehat{C}^{\text{Gab}}$ (for Case 2) can be correctly decoded. Now, this allows the original information \mathbf{f} to be recovered by (1) (and by applying the erasure decoding of C^{MDS} in Case 2).

2) *Optimality of Construction I:* Next, we show that our concatenated scheme attains the upper bound (3) on resilience capacity and thus, is optimal (see Remark 2).

First, we note that for an $[n, \mathcal{M}, d_{\min}, \alpha, \beta, d]_{\mathbb{F}}$ optimal repair MDS code it holds that $\beta = \frac{\alpha}{d-k+1}$ and hence the

upper bound (3) on the resilience capacity can be rewritten as

$$C_t(\alpha, \beta, d) \leq \sum_{i=2t+1}^k \min\left\{(d-i+1)\frac{\alpha}{d-k+1}, \alpha\right\} = \alpha(k-2t). \quad (5)$$

Let the set of parameters $\{\mathcal{M}, n, k, \alpha, \beta, d, N, m, D\}$ be as described in the construction of Section III-A1. Then $N = \mathcal{M} + D - 1$ and $N = \alpha k$. Let t be an integer such that $D = 2t\alpha + 1$. Then $\alpha k = \mathcal{M} + D - 1 = \mathcal{M} + 2t\alpha$, and hence $\mathcal{M} = \alpha(k - 2t)$. Thus, our concatenated coding scheme achieves the bound in (5).

Remark 3. *The authors in [30] provided an explicit construction of codes that attain the bound in (5) for bandwidth-limited regime. However, this construction has practical limitations for large values of t since the decoding algorithm presented in [30] is exponential in t . On the other hand, the decoding of codewords in the construction presented in our paper is efficient as it is based on two efficient decoding algorithms: one for an optimal repair MDS array code, and other one for a Gabidulin code. However, our coding scheme provides resilience for a weaker model of adversarial errors.*

Remark 4. *In [31], Rashmi et al. consider a scenario, referred as ‘erasure’, where some nodes which are supposed to provide data during node repair become unavailable. It is easy to see from (1) that our construction can also correct such erasures, as long as the minimum distance of the MRD code used as an outer code is large enough. The codes obtained by our construction also attain the bound on the capacity derived in [31]. Here, we note that while our construction works with any MSR code and in particular with an MSR code with high rate, it provides a solution for a restricted error model, i.e., static errors.*

B. Error resilience in optimal LRCs

In this subsection we study DSS which employ locally repairable codes in the presence of errors. We consider only optimal LRCs, i.e., we consider an $(n, n - d_{\min} + 1)$ -DSS where d_{\min} attains the upper bound (2). First, we provide an upper bound on the resilience capacity and then we present a coding scheme which attains this bound for the static errors.

1) *Resilience Capacity for LRCs:* In the following, we derive an upper bound on the amount of data that can be reliably stored on a DSS employing an LRC which contains corrupted nodes.

We denote by $C_t(r, \delta, \alpha)$ the resilience capacity of an $(n, n - d_{\min} + 1)$ -DSS which employs an (r, δ, α) LRC in the presence of t corrupted nodes (for any type of errors). In other words, a data collector contacting any $n - d_{\min} + 1$ storage nodes can reconstruct the original data of size $C_t(r, \delta, \alpha)$ stored on DSS in the presence of at most t adversarial storage nodes.

Theorem 3. *Consider an $(n, n - d_{\min} + 1)$ -DSS which employs an (r, δ, α) LRC. If there are at most t corrupted nodes, then the upper bound on the resilience capacity is given by*

$$C_t(r, \delta, \alpha) \leq (\rho r - 2t)\alpha + \min\{h\alpha, r\alpha\}, \quad (6)$$

where $\rho = \left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor$, $h = (n - d_{\min} + 1) - \rho(r + \delta - 1)$, and $2t < \rho r + \min\{h, r\}$.

Proof: We assume that the LRC under consideration has g disjoint local groups, as the upper bound on the minimum distance for LRCs is achievable only if the local groups of the code are disjoint [20]. We use vector $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_g)$ to denote an adversarial node pattern, where τ_i denotes the number of adversarial nodes in i th local group and $\sum_{i=1}^g \tau_i = t$. Note that during a node repair a newcomer node contacts any r out of $r + \delta - 2$ surviving nodes in its local group and downloads all the data stored on these r nodes in order to regenerate the failed node. To obtain the upper bound, we evaluate the value of a cut in the information flow graph for this DSS. Similarly to [18], we consider a data collector which contacts $r + \delta - 1$ nodes from each of first $\rho = \left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor$ local groups and $h = (n - d_{\min} + 1) - \rho(r + \delta - 1)$ nodes from $(\rho + 1)$ th local group. Then we have the following cut value CUT

$$\text{CUT} = \rho r \alpha + \min\{h\alpha, r\alpha\}. \quad (7)$$

Next, we consider the adversarial node pattern $\boldsymbol{\tau} = (\tau_1 = r, \tau_2 = r, \dots, \tau_{\lfloor t/r \rfloor} = r, \tau_{\lfloor t/r \rfloor + 1} = t - r \lfloor \frac{t}{r} \rfloor, \tau_{\lfloor t/r \rfloor + 2} = 0, \dots, \tau_g = 0)$. We further assume that in each local group we have repaired $r + \delta - 1 - r = \delta - 1$ nodes by

contacting remaining r nodes and adversarial nodes always belong to non-repaired nodes. Further, we divide the independent symbols contributing to the value of CUT into three groups:

- 1) \mathbf{M}_1 : $t\alpha$ symbols corresponding to t adversarial nodes.
- 2) \mathbf{M}_2 : $t\alpha$ symbols from t intact nodes.
- 3) \mathbf{M}_3 : $\text{CUT} - 2t\alpha$ remaining symbols after excluding \mathbf{M}_1 and \mathbf{M}_2 from CUT.

Next, we use an argument similar to that used in [30]. Note that \mathbf{M}_1 and \mathbf{M}_2 can not carry any information to the data collector: Without the knowledge of the identity of adversarial nodes, the data collector can not decide which one of two sets of symbols $\{\mathbf{M}_1, \mathbf{M}_3\}$ or $\{\mathbf{M}_2, \mathbf{M}_3\}$ corresponds to legitimate information. Therefore, the data collector relies on symbols associated with \mathbf{M}_3 to reconstruct the original file. This gives us the following bound on the resilience capacity:

$$C_t(r, \delta, \alpha) \leq \text{CUT} - 2t\alpha = (\rho r - 2t)\alpha + \min\{h\alpha, r\alpha\}.$$

2) *Construction of Error-Correcting Optimal LRC*: In this subsection we present a concatenated coding scheme, also based on a Gabidulin and MDS array codes, for error-correcting locally repairable codes and prove the optimality of this construction for static errors model.

Construction II. Let $n, d_{\min}, r, \delta, \alpha$ be the given parameters. (We assume for simplicity that $(r + \delta - 1)|n$). Let $\mathcal{M}, \mathcal{M}', N, m, \rho, h$ be the positive integers such that $\mathcal{M}' \leq \mathcal{M} \leq N \leq m$, $r\alpha \leq \mathcal{M} = \rho r\alpha + \min\{h, r\}\alpha$, $N = \frac{nr\alpha}{r+\delta-1}$, $\rho = \left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor$, $h = (n - d_{\min} + 1) - \rho(r + \delta - 1)$, and d_{\min} attains the bound (2), i.e., $d_{\min} = n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right) (\delta - 1)$.

Consider a file \mathbf{f} of size \mathcal{M}' over $\mathbb{F} = \mathbb{F}_{q^m}$. The encoding is identical to the encoding for LRC, presented in Subsection II-E, the only difference is that we apply an $[N, \mathcal{M}', D = N - \mathcal{M}' + 1]_{q^m}$ Gabidulin code \mathcal{C}^{Gab} in the first step of the encoding: Let $\mathbf{c}_{\mathbf{f}} = (f(g_1), \dots, f(g_N))$ be the codeword of \mathcal{C}^{Gab} that corresponds to the file \mathbf{f} , where $g_1, \dots, g_N \in \mathbb{F}$ are evaluation points for the Gabidulin code, which are linearly independent over \mathbb{F}_q . This codeword is partitioned into local disjoint groups, each one of size $r\alpha$ and then each local group is encoded using an $[(r + \delta - 1), r, \delta, \alpha]_q$ MDS array code (over \mathbb{F}_q).

The following theorem shows the error resilience of the proposed code. Note, that the node repairs of this scheme are performed exactly in the same (local) way as in an error-free LRC.

Theorem 4. *Let t be the number of erroneous nodes in the system which employs a d_{\min} -optimal LRC based on concatenation of Gabidulin and MDS array codes, from Construction II. If the minimum distance D of the underlying Gabidulin code satisfies $D \geq 2t\alpha + \left(\frac{n}{r+\delta-1} - \left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor\right)r\alpha - \min\{h\alpha, r\alpha\} + 1$, then the original data can be recovered from any $n - d_{\min} + 1$ nodes.*

Proof: We consider the worst case data collector which contacts $n - d_{\min} + 1$ nodes which belong to ρ (or $\rho + 1$) different groups, $(r + \delta - 1)(n - d_{\min} + 1)$ (or $(r + \delta - 1) \nmid (n - d_{\min} + 1)$), which contain all the corrupted nodes. During the node repairs, an error spreads to at most all the nodes which belong to the same local group that contains an erroneous node. Based on Lemma 1 the data collector obtains $\left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor r\alpha + \min\{h\alpha, r\alpha\}$ symbols of the corresponding Gabidulin code, which can contain an error of rank at most $t\alpha$, similarly to the proof of Theorem 2. Therefore, this corresponds to $t\alpha$ rank errors and $N - \left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor r\alpha - \min\{h\alpha, r\alpha\}$ rank erasures. Thus the statement of the theorem follows from the fact that $N = \frac{n}{r+\delta-1}r\alpha$,

Next, we will show that our error-correcting LRC attains the upper bound on the resilience capacity of Theorem 3, for a static errors model.

Corollary 1. *Let \mathcal{C}^{LRC} be the LRC for $(n, n - d_{\min} + 1)$ -DSS obtained from Construction II and let t be the number of corrupted nodes, where $2t < \rho r + \min\{h, r\}$, for ρ and h defined in Theorem 3. If the corresponding Gabidulin code has the minimum distance $D = 2t\alpha + \left(\frac{n}{r+\delta-1} - \rho\right)r\alpha - \min\{h\alpha, r\alpha\} + 1$ then \mathcal{C}^{LRC} attains the bound (6) on the resilience capacity.*

Proof: We need to prove that $\mathcal{M}' = (\rho r - 2t)\alpha + \min\{h\alpha, r\alpha\}$. Since the corresponding $[N, \mathcal{M}', D]_{q^m}$

Gabidulin code is an MRD code, it holds that $\mathcal{M}' = N - D + 1$. Then

$$\begin{aligned}\mathcal{M}' &= \frac{nr\alpha}{r + \delta - 1} - (2t\alpha + \left(\frac{n}{r + \delta - 1} - \rho\right) r\alpha - \min\{h\alpha, r\alpha\} + 1) + 1 \\ &= \rho r\alpha - 2t\alpha + \min\{h\alpha, r\alpha\}.\end{aligned}$$

■

Now we illustrate the idea of the Construction II. We consider the case where an adversary pollutes the information stored on a single storage node.

Example 4. Consider the code of Example 2 with additional parameters $t = 1$ and $\mathcal{M}' = 20$. The Gabidulin code used in the first step of the encoding is the $[36, 20, 17]_{q^{36}}$ code \mathcal{C}^{Gab} and the MDS array code used in the second step of the construction is the $(5, 3)$ zigzag code from Example 1. Here we have $\rho = 2$ and $h = 1$. In other words, the system stores a file \mathbf{f} of size 20 over $\mathbb{F}_{q^{36}}$, and a data collector should reconstruct this file from any $n - d_{\min} + 1 = 15 - 5 + 1 = 11$ nodes. Let us assume that an adversary attacks the third storage node and introduces erroneous information. The erroneous information at the third node is modeled as $(a_9, a_{10}, a_{11}, a_{12}) + (e_1, e_2, e_3, e_4)$. Now assume that the second node fails. The system is oblivious to the presence of pollution at the third node, and employs the erasure decoding of the $[5, 3 \cdot 4, 3, 4]_q$ MDS code to reconstruct the second node: Assume that the reconstructed node downloads all the symbols from the first, the third and the fifth nodes and solves a linear system of equations to obtain $(a_5, a_6, a_7, a_8) + (-e_4, -2e_3, -e_2, -2^{-1}e_1)$, where 2^{-1} denotes the inverse element of 2 in \mathbb{F}_q , $q \geq 3$.

Assume that a data collector contacts 11 first nodes ($\rho = 2$ full groups, 5 nodes in each one, and $h = 1$ node in the additional group). He obtains $12 + 12 + 4$ symbols corresponding to the Gabidulin codeword \mathbf{c}_f (which is of length 36), where these 28 symbols contain an error of rank at most 4:

$$\mathbf{c}_f + (e_1, e_2, e_3, e_4)[\mathbf{0}, B_2, I_4, \mathbf{0}_{4 \times 24}] + (c_5, \dots, c_{12})[\mathbf{0}_{8 \times 28}, -I_8],$$

where I_m denotes the identity matrix of order m , $\mathbf{0}_{a \times b}$ denotes $a \times b$ matrix with all of its entries equal to zero, and B_2 is defined as follows:

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & -2^{-1} \\ 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

In other words, we have 8 erasures in the Gabidulin codeword and at most 4 rank errors. Since the distance of \mathcal{C}^{Gab} is $D = 17$, by (1) the original data is recovered by applying errors and erasures correction of the Gabidulin code. (Note that (e_1, e_2, e_3, e_4) , $[\mathbf{0}, B_2, I, \mathbf{0}_{4 \times 24}]$, (c_5, \dots, c_{12}) are unknown to the decoder but $[\mathbf{0}_{8 \times 28}, -I_8]$ is known to the decoder).

C. Error resilience in LRCs with local regeneration

In this section, we discuss the hybrid codes which for a given locality parameters minimize the repair bandwidth. These codes, as proposed in [18] and [20], are obtained by combining locally repairable codes with regenerating codes.

In a repair process for an original LRC, a newcomer node contacts r nodes in its local group and downloads *all* the data stored on these nodes. To allow a reduction in repair bandwidth, the idea of regenerating codes is used for the hybrid codes. In particular, a newcomer node contacts any $d \geq r$ nodes in its local group and downloads only $\beta \leq \alpha$ symbols stored on these nodes in order to repair the failed node, in other words, a regenerating code is applied in each local group.

That is, when an $[r + \delta - 1, r, \delta, \alpha, \beta, d]_{\mathbb{F}_q}$ MSR code is applied in each local group instead of an MDS array code in the second step of construction of LRC presented in subsection II-E, then the resulting code, denoted by MSR-LRC, has the maximal minimum distance (since an MSR code is also an MDS array code), the local minimum storage per node, and the minimized repair bandwidth. (The details of this construction and its properties can be found in [18].) Also, when an MBR code is applied in each local group instead of an MDS array code in the second step of construction of LRC presented in subsection II-E, then the resulting code, denoted by MBR-LRC, has the

maximal possible minimum distance, and the local minimum repair bandwidth. (The details of this construction and its properties can be found in [34].)

In the following, we consider error resilience for MSR-LRC and for MBR-LRC. In particular, we provide an upper bound on the resilience capacity for these codes and consider construction for error-correcting MSR-LRC and MBR-LRC. We denote by $C_t(r, \delta, \alpha, \beta, d)_{\text{MSR}}$ ($C_t(r, \delta, \alpha, \beta, d)_{\text{MBR}}$) the resilience capacity of an $(n, n - d_{\min} + 1)$ -DSS which employs an $(r, \delta, \alpha, \beta, d)$ MSR-LRC (MBR-LRC) in the presence of t corrupted nodes.

Theorem 5. Consider an $(n, n - d_{\min} + 1)$ -DSS which employs an $(r, \delta, \alpha, \beta, d)$ MSR-LRC (MBR-LRC), for $r + \delta - 1 > d > r$. If there are at most t corrupted nodes, then the upper bound on the resilience capacity of MSR-LRC is given by

$$C_t(r, \delta, \alpha, \beta, d)_{\text{MSR}} \leq \left(\rho - 2 \left\lfloor \frac{t}{d} \right\rfloor \right) r\alpha + (\min\{h, r\} - 2 \min\{\gamma, r\}) \alpha,$$

and the resilience capacity for the MBR-LRC is upper bounded by

$$C_t(r, \delta, \alpha, \beta, d)_{\text{MBR}} \leq \min\{\text{term I, term II}\},$$

where

$$\begin{aligned} \text{term I} &= \left(\rho - 2 \left\lfloor \frac{t}{d} \right\rfloor \right) B_{\text{MBR}} - 2 \sum_{i=1}^{\min\{\gamma, r\}} (d - i + 1)\beta + \sum_{i=1}^{\min\{h, r\}} (d - i + 1)\beta, \\ \text{term II} &= \tilde{\rho} \sum_{i=2s+1}^d (d - i + 1)\beta + (\rho - \tilde{\rho}) \sum_{i=2\hat{s}+1}^d (d - i + 1)\beta + \sum_{i=2\hat{s}+1}^{\min\{h, d\}} (d - i + 1)\beta, \end{aligned}$$

Here, $\rho = \left\lfloor \frac{n - d_{\min} + 1}{r + \delta - 1} \right\rfloor$, $h = (n - d_{\min} + 1) - \rho(r + \delta - 1)$, $\gamma = (t - \lfloor \frac{t}{d} \rfloor d)$, $B_{\text{MBR}} = r\alpha - \frac{r(r-1)}{2}\beta$ and $\tilde{\rho}, s, \hat{s}$, and \hat{s} are such that $\tilde{\rho}s + (\rho - \tilde{\rho})\hat{s} + \hat{s} = t$, $2 \max\{s, \hat{s}\} \leq d$, and $2\hat{s} \leq h$.

Proof: MSR-LRC: Similarly to the proof of Theorem 3, we evaluate the value of a cut in the corresponding information flow graph. We again consider a data collector that contacts $r + \delta - 1$ nodes from each of first ρ local groups and h nodes from $(\rho + 1)$ th local group. We further assume that the pattern of adversarial nodes is $\tau = (\tau_1 = d, \tau_2 = d, \dots, \tau_{\lfloor t/d \rfloor} = d, \tau_{\lfloor t/d \rfloor + 1} = t - d \lfloor \frac{t}{d} \rfloor, \tau_{\lfloor t/d \rfloor + 2} = 0, \dots, \tau_g = 0)$. In each group, we assume that all but d nodes have been repaired at least once. The remaining d nodes are used to repair all node failures. The corrupted nodes are assumed to be among these d nodes in their local groups. The cut value for the information flow graph associated with this scenario is given by

$$\text{CUT}_{\text{MSR}} = \rho r\alpha + \min\{h\alpha, r\alpha\}. \quad (8)$$

Similar to the proof of Theorem 3, we divide the independent symbols in CUT_{MSR} into three groups:

- 1) \mathbf{M}_1 : $\left\lfloor \frac{t}{d} \right\rfloor r\alpha + \min\{\gamma, r\}\alpha$ symbols corresponding to t adversarial nodes.
- 2) \mathbf{M}_2 : $\left\lfloor \frac{t}{d} \right\rfloor r\alpha + \min\{\gamma, r\}\alpha$ symbols from $\left\lfloor \frac{t}{d} \right\rfloor r + \gamma$ intact nodes.
- 3) \mathbf{M}_3 : $\text{CUT}_{\text{MSR}} - 2 \left(\left\lfloor \frac{t}{d} \right\rfloor r + \min\{\gamma, r\} \right) \alpha$ remaining symbols after excluding \mathbf{M}_1 and \mathbf{M}_2 from CUT_{MSR} .

Again following the argument similar to that for LRCs, we get the following bound on the resilience capacity for an MSR-LRC:

$$\begin{aligned} C_t(r, \delta, \alpha, \beta, d)_{\text{MSR}} &\leq \text{CUT}_{\text{MSR}} - 2 \left(\left\lfloor \frac{t}{d} \right\rfloor r + \min\{\gamma, r\} \right) \alpha \\ &= \left(\rho - 2 \left\lfloor \frac{t}{d} \right\rfloor \right) r\alpha + (\min\{h, r\} - 2 \min\{\gamma, r\}) \alpha. \end{aligned}$$

Note that if $t < r$ then this bound is the same as the bound (6) for LRCs.

MBR-LRC: If we consider the same adversarial node pattern as in case MSR-LRC, we get the following bound on the resilience capacity for an MBR-LRC:

$$C_t(r, \delta, \alpha, \beta, d)_{\text{MBR}} \leq \left(\rho - 2 \left\lfloor \frac{t}{d} \right\rfloor \right) B_{\text{MBR}} - 2 \sum_{i=1}^{\min\{\gamma, r\}} (d - i + 1)\beta + \sum_{i=1}^{\min\{h, r\}} (d - i + 1)\beta. \quad (9)$$

Alternatively, we consider another pattern of eavesdropped nodes $\tau = (\tau_1 = s, \tau_2 = s, \dots, \tau_{\tilde{\rho}} = s, \tau_{\tilde{\rho}+1} = \tilde{s}, \tau_{\tilde{\rho}+2} = \tilde{s}, \tau_{\tilde{\rho}+3} = \hat{s}, \tau_{\tilde{\rho}+4} = 0, \dots, \tau_g = 0)$. Here, $\tilde{\rho}$, s , \tilde{s} , and \hat{s} are such that $\tilde{\rho}s + (\rho - \tilde{\rho})\tilde{s} + \hat{s} = t$, $2 \max\{s, \tilde{s}\} \leq d$, and $2\hat{s} \leq \min\{h, r\}$. Note that such choice is always possible given the particular choice of data collector and the assumption that $2t < \rho r + \min\{h, r\}$. For this pattern of adversarial node, we obtain the following upper bound on the resilience capacity for an MBR-LRC:

$$C_t(r, \delta, \alpha, \beta, d)_{\text{MBR}} \leq (\rho - \tilde{\rho}) \sum_{i=2\tilde{s}+1}^d (d-i+1)\beta + \tilde{\rho} \sum_{i=2s+1}^d (d-i+1)\beta + \sum_{i=2\hat{s}+1}^{\min\{h,d\}} (d-i+1)\beta. \quad (10)$$

Now, we can pick minimum of RHS of (9) and (10) as an upper bound on the file size. \blacksquare

1) *Construction of Error-Correcting LRC with Local Regeneration:* Similarly to the construction of error-correcting LRCs, where the only difference to the construction of d_{\min} -optimal LRC (without error correction) is the larger minimum rank distance of the corresponding Gabidulin code, the construction for error-correcting LRCs with local regeneration is based on the construction of an error-free MSR-LRC (MBR-LRC), where the Gabidulin code is chosen with larger minimum distance. We note that since for the case $t < r$ the bounds for LRCs and MSR-LRCs are identical, the optimal codes can be obtained simply by replacement of MDS array codes by MSR codes in Construction II. However, constructions for optimal error-correcting MSR-LRCs for general cases and also for optimal error-correcting MBR-LRC codes still remain an open problem.

IV. DYNAMIC ERROR MODEL

In this section, we consider the problem of designing coding schemes for DSS that work under dynamic error model. Note, that in a static error model each time an attacked node is requested for the data to be sent, it sends some linear combinations of the data that has been modified on it by an adversary, which the adversary is allowed to do only once. Therefore, the rank of the error that a single node under static attack causes throughout the operation of DSS is bounded above by α . This is not the case under the dynamic error model as a single attacked node can inject an error of large rank if it is utilized in multiple node repairs, which may render the data stored on DSS useless.

Towards this model, some results are presented in [30] and [31]. The coding scheme proposed in [30] does not have an efficient decoding during the data reconstruction process and it works specifically with bandwidth efficiently repairable codes at the MBR point. The coding scheme of [31] deals with the dynamic error model at the MSR point, but there scheme works only for low rate, i.e., $2k \leq n + 1$.

Next, we present two solutions to deal with attack under the dynamic error model. The first solution aims to correct errors during the node repair process. The second approach is based on existing literature on subspace signatures. All the results presented in this section are given for optimal repair MDS array codes. Since our locally repairable codes make use of MDS array codes (or MSR codes for codes with local regeneration) in each local group, the similar ideas can be applied for this family of DSS codes as well.

A. Naïve scheme for dynamic error model

A solution for the dynamic error model is to adopt a repair scheme where a newcomer node utilizes the redundancy in the downloaded data to perform error-free exact repair even in the presence of errors in the downloaded data. Next, we analyze the maximum amount of information that can be stored on the DSS employing concatenated codes proposed in Section III-A1 under the dynamic error model, if an error free node repair is performed.

When a storage node fails, a newcomer node downloads $d\beta$ symbols from any d surviving nodes ($d \geq k$) (for MSR code $d\beta = \alpha + (k-1)\beta$). Since there can be at most t adversarial nodes present in the system, the newcomer node receives at most $t\beta$ erroneous symbols. Therefore, out of $k\alpha$ symbols of a Gabidulin codeword, by Lemma 1, the newcomer has $(k-1)\beta + \alpha$ symbols (using the fact that the inner code is an MDS code and we perform bandwidth efficient repair). All the other $k\alpha - (k-1)\beta - \alpha = (\alpha - \beta)(k-1)$ symbols of a Gabidulin codeword can be considered as the erased symbols. Let \mathcal{M}' denote the number of information symbols (over \mathbb{F}_{q^m}) that are stored on the DSS. Then the minimum distance D of the corresponding Gabidulin code satisfies $D = k\alpha - \mathcal{M}' + 1$. Therefore we can reconstruct the entire Gabidulin codeword and thus the data stored on the failed node, if we have $D = k\alpha - \mathcal{M}' + 1 \geq 2t\beta + (k-1)(\alpha - \beta) + 1$.

This gives us

$$\mathcal{M}' \leq \alpha + (k - 2t - 1)\beta. \quad (11)$$

Note that the bound in (3) is still applicable. For $k = 2t + 1$, the right hand side expression in (11) is equal to that in (3). Therefore, this naïve repair scheme is optimal in terms of the capacity of DSS even in the dynamic error model. However, the difference between these bounds is monotonically increasing with $(k - 2t - 1)$ and the solution proposed in this section is suboptimal for general values of system parameters k and t .

Remark 5. *In a similar way, it can be shown that the LRCs based construction presented in Section III-B is optimal under dynamic error model when $pr + \min\{h, r\} = 2t + 1$.*

B. Subspace signatures approach

As mentioned previously, in the dynamic error model an attacked node can inject a high rank error. Thus, it is desirable to restrict the rank of the aggregate error that a particular attacked node can cause in the entire system under dynamic error model. In this subsection, we propose to combine the existing literature on detecting subspace pollution with MRD codes to counter a dynamic attack. Next, we illustrate this with the help of subspace signatures proposed in [35].

Let us consider an n -nodes DSS that employs a Gabidulin and a bandwidth efficiently repairable code based storage scheme as explained in Section III-A1. For node i , content stored on it, i.e. $\mathbf{y}_i \in \mathbb{F}^\alpha = \mathbb{F}_q^\alpha$, can be viewed as an $m \times \alpha$ matrix over \mathbb{F}_q . These α column vectors of length m stored on i th node span a subspace (column space of \mathbf{y}_i when viewed as a matrix over \mathbb{F}_q) in \mathbb{F}_q^m of dimension at most α . Since all elements of the coding matrix and repair matrices are from \mathbb{F}_q , during node repair process node i sends β vectors that lie in the subspace spanned by \mathbf{y}_i . If we make sure that even under the dynamic error model an attacked node sends vectors from the same α -dimensional subspace of \mathbb{F}_q^m during node repairs and data reconstruction, a data collector encounters at most $t\alpha$ -rank error, which can be corrected with a Gabidulin code of large enough distance as in the static model. Subspace signatures solve this problem of enforcing the requirement that a node sends data (vectors) from the same α -dimensional subspace of \mathbb{F}_q^m .

We assume existence of a trusted verifier, who stores all n subspace signatures, one signature for each storage node, generated according to the procedure explained in [35]. Whenever a particular node sends data during a node repair or data reconstruction, the trusted verifier checks the data against the stored subspace signature corresponding to that particular storage node.

For the purpose of the *data reconstruction*, whenever a node does not pass the signature test, this node is considered as α rank erasures. If $s \leq t$ nodes fail the test during data reconstruction, the data collector deals with $s\alpha$ rank erasures and $(t - s)\alpha$ rank errors. Given that the outer Gabidulin code has minimum rank distance $2t\alpha + 1 \geq 2(t - s)\alpha + s\alpha + 1$, the original data can be reconstructed without an error.

Next, we argue how subspace signatures help restrict rank of the error introduced in the system during a *node repair process*. Assume that node i fails. Let $\mathcal{R}_i \subseteq \{1, \dots, n\} \setminus \{i\}$ denote the set of d surviving nodes that are contacted to repair node i . In order to repair node i , each node $j \in \mathcal{R}_i$ is supposed to send $\mathbf{y}_j V_{ji}$, where V_{ji} is an $\alpha \times \beta$ repair matrix of node i associated with node j . Since the data downloaded through all the surviving nodes is verified against subspace signatures, data from node j passes the test if it is of the form $\mathbf{y}_j \widehat{V}_{ji}$, where $\mathbf{y}_j \widehat{V}_{ji}$ is in the column space of \mathbf{y}_j and \widehat{V}_{ji} may be different from V_{ji} .

If any of the surviving (helper) nodes does not pass the test, the trusted verifier begins the naïve repair for the failed node and the nodes that fail the test. During this naïve repair, entire data is downloaded from a set of $k - s$ nodes out of $d - s$ nodes that provide data for node repair and pass the subspace test. Here, s is the number of nodes that fail the subspace test. Note that each node of these $k - s$ selected nodes provides additional $\alpha - \beta$ symbols as it has already sent β symbols (over \mathbb{F}_q^m). The decoding algorithm for Gabidulin codes is run on $(k - s)\alpha$ symbols downloaded from this selected set of $k - s$ nodes. There can be at most $t - s$ adversary nodes present in the selected set of $k - s$ nodes (s adversarial nodes that failed the subspace test are excluded from this process), which can contribute at most $(t - s)\alpha$ erroneous symbols. Since the distance of the Gabidulin code is greater than $2(t - s)\alpha + s\alpha + 1$, the decoding algorithm recovers the original file, which is used to get the data stored on nodes being repaired.

In case when all the adversarial nodes pass the test, the data provided by each node $j \in \mathcal{R}_i$ is of the form

$$\mathbf{y}_j \widehat{V}_{ji} = \mathbf{y}_j V_{ji} + \mathbf{y}_j (\widehat{V}_{ji} - V_{ji}).$$

After performing exact repair process for node i , node i stores $\mathbf{y}_i + \mathbf{y}_e B_i$, where B_i is an $t\alpha \times \alpha$ matrix over \mathbb{F}_q and $\mathbf{y}_e = [\mathbf{y}_{i_1}, \dots, \mathbf{y}_{i_t}] \in \mathbb{F}_q^{t\alpha}$. Here $\{i_1, \dots, i_t\}$ denotes the set of t adversarial nodes. After the node repair, the trusted verifier generates a new subspace signature corresponding to the data stored on a node i for future verification. At any point of time, the data stored on DSS can be represented as

$$\widetilde{\mathbf{y}} = \mathbf{y} + \mathbf{y}_e \mathbf{B}, \quad (12)$$

where columns with indices from $\{(i-1)\alpha + 1, \dots, t\alpha\}$ of \mathbf{B} are equal to α columns of B_i , $1 \leq i \leq n$. It is evident from (12) that the rank of the aggregate error in the system is at most $t\alpha$ and a Gabidulin code with large enough distance can ensure the reliable recovery of the original data.

V. CONCLUSION AND FUTURE RESEARCH

A novel concatenated coding scheme for DSS is presented. The scheme makes use of rank-metric codes, in particular, Gabidulin codes, as the first step of the process of encoding the data. In the second step of the encoding process, MDS optimal repair array codes (locally or globally) are used. This construction ensures resilience against static adversarial errors. A modification of the scheme based on subspace signatures enables resilience against dynamic errors. Also, upper bounds on the resilience capacity for LRCs, MSR-LRCs, and MBR-LRCs are presented.

We conclude with a list of open problems for future research.

- 1) Do there exist (explicit) high-rate error-correcting MSR codes which attain the upper bound (3) on the resilience capacity for a general (dynamic) error model?
- 2) Is it possible to improve the bounds on resilience capacity for MSR-LRCs and MBR-LRCs?
- 3) Do there exist (explicit) optimal MSR-LRCs and MBR-LRCs for a general set of parameters?

REFERENCES

- [1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [2] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. 2009 IEEE International Symposium on Information Theory (ISIT)*, 2009, pp. 2276–2280.
- [3] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. 2010 IEEE Information Theory Workshop (ITW)*, 2010, pp. 1–5.
- [4] C. Suh and K. Ramchandran, "Exact-repair MDS codes for distributed storage using interference alignment," in *Proc. 2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 161–165.
- [5] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.
- [6] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1597–1616, 2013.
- [7] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," in *Proc. of the IEEE*, 2011, pp. 476–489.
- [8] A. Datta and F. Oggier, "An overview of codes tailor-made for networked distributed data storage," *CoRR*, vol. abs/1109.2317, 2011.
- [9] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)*, 2007, pp. 79–86.
- [10] J. Han and L. Lastras-Montano, "Reliable memories with subline accesses," in *Proc. 2007 IEEE International Symposium on Information Theory (ISIT)*, 2007.
- [11] F. Oggier and A. Datta, "Self-repairing codes for distributed storage - a projective geometric construction," in *Proc. 2011 IEEE Information Theory Workshop (ITW)*, 2011, pp. 30–34.
- [12] —, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. 2011 IEEE INFOCOM*, 2011, pp. 1215–1223.
- [13] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in windows azure storage," in *Proceedings of the 2012 USENIX conference on Annual Technical Conference, ser. USENIX ATC12*. USENIX Association, 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2342821.2342823>
- [14] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 2771–2775.
- [15] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, "Optimal linear codes with a local-error-correction property," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 2776–2780.
- [16] A. S. Rawat and S. Vishwanath, "On locality in distributed storage systems," in *Proc. 2012 IEEE Information Theory Workshop (ITW)*, 2012, pp. 497–501.

- [17] N. Silberstein, A. S. Rawat, and S. Vishwanath, "Error resilience in distributed storage via rank-metric codes," in *Proc. 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1150–1157.
- [18] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *CoRR*, vol. abs/1210.6954, 2012.
- [19] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Transactions on Information Theory*, vol. 58, no. 11, pp. 6925–6934, 2012.
- [20] G. M. Kamath, N. Prakash, V. Lalitha, and P. Kumar, "Codes with local regeneration," *CoRR*, vol. abs/1211.1932, 2012.
- [21] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: novel erasure codes for big data," in *Proceedings of the 39th international conference on Very Large Data Bases*, ser. PVLDB'13. VLDB Endowment, 2013, pp. 325–336. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2488335.2488339>
- [22] H. D. L. Hollmann, "Storage codes—coding rate and repair locality," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, 2013, pp. 830–834.
- [23] D. Silva, F. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.
- [24] T. Etzion and N. Silberstein, "Error-correcting codes in projective space via rank-metric codes and ferrers diagrams," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 2909–2919, 2009.
- [25] R. W. Nóbrega and B. F. Uchoa-Filho, "Multishot codes for network coding using rank-metric codes," in *2010 IEEE Wireless Network Coding Conference (WiNC)*, pp. 1–6.
- [26] D. Silva and F. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1124–1135, 2011.
- [27] C. Ning, Y. Zhiyuan, M. Gadouleau, W. Ying, and B. W. Suter, "Rank metric decoder architectures for random linear network coding with error control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 296–309, 2012.
- [28] T. K. Dikaliotis, A. G. Dimakis, and T. Ho, "Security in distributed storage systems by communicating a logarithmic number of bits," in *Proc. 2010 IEEE International Symposium on Information Theory (ISIT)*, 2010, pp. 1948–1952.
- [29] F. Oggier and A. Datta, "Byzantine fault tolerance of regenerating codes," in *2011 IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2011, pp. 112–121.
- [30] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6734–6753, 2011.
- [31] K. V. Rashmi, N. B. Shah, K. Ramchandran, and P. V. Kumar, "Regenerating codes for errors and erasures in distributed storage," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT)*, 2012, pp. 1202–1206.
- [32] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," in *Proc. 2011 IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–5.
- [33] S. Goparaju, S. El Rouayheb, R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," *CoRR*, vol. abs/1304.3156v2, 2013.
- [34] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, V. Lalitha, O. O. Koyluoglu, P. V. Kumar, and S. Vishwanath, "Explicit MBR all-symbol locality codes," in *Proc. 2013 IEEE International Symposium on Information Theory (ISIT)*, 2013.
- [35] F. Zhao, T. Kalker, M. Medard, and K. J. Han, "Signatures for content distribution with network coding," in *Proc. 2007 IEEE International Symposium on Information Theory (ISIT)*, 2007.
- [36] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *Journal of Comb. Theory, Series A*, vol. 25, pp. 226 – 241, 1978.
- [37] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problems of Information Transmission*, vol. 21, pp. 1 – 12, July 1985.
- [38] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 328–336, 1991.
- [39] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. North-Holland, 1978.
- [40] D. Silva and F. R. Kschischang, "Fast encoding and decoding of gabidulin codes," in *Proc. 2009 IEEE International Symposium on Information Theory (ISIT)*, 2009.
- [41] M. Blaum and R. M. Roth, "On lowest density MDS codes," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 46–59, 1999.
- [42] Y. Cassuto and J. Bruck, "Cyclic lowest density MDS array codes," *IEEE Transactions on Information Theory*, vol. 55, no. 4, pp. 1721–1729, 2009.