Error Resilience in Distributed Storage via Rank-Metric Codes

Natalia Silberstein, Ankit Singh Rawat and Sriram Vishwanath
LINC, Dept. of ECE, UT Austin
Austin, TX 78712, USA.
Email: {natalys, ankitsr, sriram}@austin.utexas.edu

***Abstract*—This paper presents a novel coding scheme for distributed storage systems containing nodes with adversarial errors. The key challenge in such systems is the propagation of erroneous data from a single corrupted node to the rest of the system during a node repair process. This paper presents a concatenated coding scheme which is based on two types of codes: maximum rank distance (MRD) code as an outer code and optimal repair maximal distance separable (MDS) array code as an inner code. Given this, two different types of adversarial errors are considered: the first type considers an adversary that can replace the content of an affected node only once; while the second attack-type considers an adversary that can pollute data an unbounded number of times. This paper proves that the proposed coding scheme attains a suitable upper bound on resilience capacity for the first type of error. Further, the paper presents mechanisms that combine this code with subspace signatures to achieve error resilience for the second type of errors. Finally, the paper concludes by presenting a construction based on MRD codes for optimal locally repairable scalar codes that can tolerate adversarial errors.**

I. Introduction

Distributed storage systems (DSS) are increasingly important resources today for users and businesses than ever before. Centralized storage is proving to be rapidly infeasible given the growing sizes of databases being stored; as well as the need to rapidly and reliably access them. Thus, generally, data is distributed and often replicated across multiple locations to enable ease-of-access and reslience to failures. However, replication can be inefficient in terms of storage space, and therefore, coding is useful in instilling resilience to node failures while reducing storage requirements over replication. Given the prevalence of single node failures in DSS (a user exiting a P2P system, power outage in a single data center in the cloud), coding enables a single node to be repaired as soon as the node failure occurs in order to sustain the desired level of redundancy in the system.

In order to repair the failed node, data is downloaded from surviving nodes and a function of this data is stored as the 'restored' node. The amount of data downloaded in this repair process is called the *repair bandwidth* of the reconstruction process. Since large repair bandwidth translate to consumption of a vast amount of system resources (in terms of bandwidth and energy), it is desirable to have a coding schemes that have as small a repair bandwidth as possible In [1], Dimakis et al. establish an information theoretic lower bound on repair bandwidth for (n, k) codes with maximum distance separable (MDS) property (i.e., any set of k nodes can be used to reconstruct the data stored on a DSS). The work also presents a trade-off between repair bandwidth and the amount of data stored on each node.

A desirable notion of repair is *exact repair*, where the regenerated data is an exact replica of what was stored on the failed node. The work in [2], [3], [4] present low rate codes, which achieve the lower bound derived in [1] when data is downloaded from all surviving nodes during node repair. In [5], Rashmi et al. design exact-repairable codes, which allow node repair to be performed by contacting $d \leq n-1$ surviving nodes. At the minimum storage regeneration (MSR) point of operation (see [1] for details), these codes also correspond to a low rate regime as their rate is upper bounded by $\frac{1}{2} + \frac{1}{2n}$. Recently, various researchers have presented high(er) rate codes for the MSR point which enable bandwidth-efficient exact repair. Along a similar view, [6] presents codes for DSS with two parity nodes. In [7] and [8], permutation-matrix based codes are presented which are designed to achieve the bound on repair bandwidth for repair of systematic nodes for all (n, k) pairs. [9] further generalizes the idea of [8] to get MDS array codes for DSS that allow optimal exact regeneration for parity nodes as well.

While a bulk of existing literature in DSS addresses the storage versus repair bandwidth trade-off, another important issue that has recently received attention is the design of storage schemes that ensure security and reliability of the stored content against *adversarial errors* [10], [11], [12], [13]; and this issue of inducing reliability against adversarial errors is the main focus of our paper. The dynamic nature of DSS resulting from repeated node failures, triggering *node repairs*, makes the problem of dealing with erroneous nodes non-trivial. This is because a single corrupted node may subsequently pollute other nodes in the DSS system during the node repair process. In [12], Pawar et al. address the reliability issue in detail and 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. [12] also presents 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 [12], [14].

In this paper, we study the notion of an omniscient adversary[1], which can observe all nodes and has full knowledge of

[1] In [13], Rashmi et al. show that product matrix codes [5] can be used towards dealing with adversarial errors. This solution is again restricted to the regime $k \leq \frac{n+1}{2}$, at the MSR point of operation.

the coding scheme employed by the system [12]. As in [12], we assume an upper bound on the number of nodes that can be controlled by such an omniscient adversary. We classify adversarial attacks into two classes:

1) *Static errors:* an omniscient adversary replaces the content of an affected node with nonsensical 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 the data during data collection or repair process. This kind of errors captures any malicious behaviour, hence is more difficult to manage in comparison to static errors.

We present a novel concatenated coding scheme for DSS which provides resilience against these two classes of attacks. In our scheme, 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 maximum distance separable (MDS) array code, which can be repaired in a bandwidth efficient manner. 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. As stated before, 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 bandwidth efficient codes operating at the minimum-storage regenerating (MSR) point [1]. However, our construction works for any regenerating code.

The proposed coding scheme is directly applicable to the *static error* model and is 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 allows 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 propose two solutions based on the concatenated coding scheme employed for static error model. 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, namely naïve method, is optimal for a specific choice of parameters. 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. [15], which essentially reduces the dynamic error model to somewhat similar to static error model, and helps us bound the rank of error introduced by an adversarial node throughout its presence in the system. Note that hash function based solutions has previously been presented in the context of DSS to deal with errors [10], [12]. While promising, these hash functions based approaches provide only probabilistic guarantees for pollution containment.

Recently a new line of work has appeared in the context of DSS, which focuses on minimizing the number of surviving nodes that contribute data during node repair (locality). In [16], Gopalan et al. establish an upper bound on the minimum distance of scalar linear codes that have locality $r < k$. They further show the optimality of pyramid codes [18] with respect to the upper bound, with locality of the systematic nodes. [19] gives a similar bound for codes that have a more general definition of locality. In [20], Papailiopoulos et al. generalize the bound in [16] to vector codes (possibly nonlinear). In this paper, we also show that MRD codes can be utilized to construct optimal (in terms of minimum distance) locally repairable scalar codes, which can tolerate adversarial errors.

The rest of the paper is organized as follows: In Section II, we first give a brief description of rank-metric codes along with Gabidulin MRD codes and the error model in rank-metric for these codes. Subsequently, we describe MDS array codes and present two examples of bandwidth efficient MDS array codes that are later used as inner codes in our construction. In Section III, we describe the construction of our storage scheme and prove its error resilience under the static error model. Further, we present a few examples to illustrate our scheme and show that our codes attain the upper bound on resilience capability. In Section IV, we address the dynamic error model. In Section V, we describe MRD codes based locally repairable storage schemes. Finally, we conclude with Section VI.

II. Preliminaries

A. System Model

Let M be the size of a file to be stored in a distributed storage system with n nodes. All data symbols belong to a finite field $\mathbb{F}_Q$ of size Q. Each node contains α symbols. A data collector reconstructs the original file by downloading the data stored in any set of k out of n nodes. This property is called an *MDS property*. When a node fails, its content can be reconstructed by downloading β symbols from any $d \geq k$ surviving nodes and stored in a new node added to DSS instead of the failed node. The amount of data needed for a node repair, denoted by $\gamma = \beta d$, is called the *repair bandwidth*. In [1], Dimakis et al. show a trade-off between repair bandwidth γ and the amount of data α stored on each node. Two families of codes which attain two extremal points on the optimal tradeoff curve are called *minimum-storage regenerating (MSR) codes* and *minimum-bandwidth regenerating (MBR) codes*, respectively. In the sequel we will focuss on the MSR codes.

It was proved [1] that for MSR codes α and γ satisfy

$$(\alpha, \gamma) = \left(\frac{M}{k}, \, \frac{Md}{k(d-k+1)} \right). \tag{1}$$

B. Rank-Metric Codes

Rank-metric codes were introduced by Delsarte [21] and rediscovered in [22], [23]. These codes have applications in different fields, such as space-time coding [24], random network coding [25], [26], and public key cryptosystems [27]. Our goal in this paper is to show that rank-metric codes are useful for error correction in distributed storage as well.

Let $\mathbb{F}_q$ be the finite field of size q. For two $N \times m$ matrices A and B over $\mathbb{F}_q$ the *rank distance* is defined by

$$d_R(A, B) \overset{\text{def}}{=} \text{rank}(A - B) \ .$$

An $[N \times m, \varrho, \delta]$ *rank-metric code* $\mathcal{C}$ is a linear code, whose codewords are $N \times m$ matrices over $\mathbb{F}_q$; they form a linear subspace with dimension ϱ of $\mathbb{F}_q^{N \times m}$, and for each two distinct codewords A and B, $d_R(A, B) \geq \delta$. For an $[N \times m, \varrho, \delta]$ rank-metric code $\mathcal{C}$ we have $\varrho \leq \min\{N(m-\delta+1), m(N-\delta+1)\}$ [21], [22], [23]. This bound, called Singleton bound for rank metric, is achievable for all feasible parameters. The codes that achieve this bound are called *maximum rank distance* (MRD) codes.

1) Gabidulin MRD Codes: An important family of MRD linear codes is presented by Gabidulin [22]. These codes can be seen as the analogs of Reed-Solomon codes for rank metric. Let $m \leq N$. A codeword in an $[N \times m, \varrho, \delta]$ rank-metric code $\mathcal{C}$ can be represented by a vector $\mathbf{c} = [c_1, c_2, \ldots, c_m]$, where $c_i \in \mathbb{F}_{q^N}$, since $\mathbb{F}_{q^N}$ can be viewed as an N-dimensional vector space over $\mathbb{F}_q$. Let $g_i \in \mathbb{F}_{q^N}$, $1 \leq i \leq m$, be linearly independent over $\mathbb{F}_q$. The generator matrix $\mathcal{G}$ of an $[N \times m, \varrho, \delta]$ Gabidulin MRD code is given by

$$\mathcal{G} = \begin{bmatrix} g_1 & g_2 & \ldots & g_m \\ g_1^{[1]} & g_2^{[1]} & \ldots & g_m^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_1^{[K-1]} & g_2^{[K-1]} & \ldots & g_m^{[K-1]} \end{bmatrix},$$

where $K = m - \delta + 1$, $\varrho = NK$, and $[i] = q^i$.

In the similar way as Reed-Solomon codes, Gabidulin codes also have an equivalent interpretation as evaluation of polynomials, however, for Gabidulin codes the special family of polynomials, called *linearized polynomials*, is evaluated on a set of linearly independent points over the base field $\mathbb{F}_q$. A linearized polynomial $f(x)$ over $\mathbb{F}_{q^N}$ of q-degree n has the form $f(x) = \sum_{i=0}^{n} a_i x^{q^i}$, where $a_i \in \mathbb{F}_{q^N}$, and $a_n \neq 0$. It is easy to see that evaluation of a linearized polynomial is an $\mathbb{F}_q$-linear transformation from $\mathbb{F}_{q^N}$ to itself, i.e., for any $\alpha, \beta \in \mathbb{F}_q$ and $a, b \in \mathbb{F}_{q^N}$, we have $f(\alpha a + \beta b) = \alpha f(a) + \beta f(b)$ [28]. Therefore, a codeword in Gabidulin code $\mathcal{C}$ can be defined as $\mathbf{c} = [f(g_1), f(g_2), \ldots, f(g_m)]$, where $f(x)$ is the linearized polynomial of q-degree $K-1$ with coefficients given by the information message, and $g_1, \ldots, g_m \in \mathbb{F}_{q^N}$ are linearly independent over $\mathbb{F}_q$ [22].

2) Rank Errors and Rank Erasures Correction: Let $\mathcal{C} \subseteq \mathbb{F}_{q^N}^m$ be a Gabidulin MRD code with minimum distance δ. Let $\mathbf{c} \in \mathcal{C}$ be the transmitted codeword and let $\mathbf{r} = \mathbf{c} + \mathbf{e_{total}}$ be the received word. The code $\mathcal{C}$ can correct any vector error of the form $\mathbf{e_{total}} = \mathbf{e_{error}} + \mathbf{e_{erasure}} = (e_1\mathbf{u_1} + \ldots + e_t\mathbf{u_t}) + (r_1\mathbf{v_1} + \ldots + r_s\mathbf{v_s})$ as long as $2t + s \leq \delta - 1$. The first part $\mathbf{e_{error}}$ is called a *rank error* of rank t, where $e_i \in \mathbb{F}_{q^N}$ are linearly independent over the base field $\mathbb{F}_q$, unknown to the decoder, and $\mathbf{u}_i \in \mathbb{F}_q^m$ are linearly independent vectors of length m, unknown to the decoder. The second part $\mathbf{e_{erasure}}$ is called a *rank erasure*, where $r_i \in \mathbb{F}_{q^N}$ are linearly independent over the base field $\mathbb{F}_q$, unknown to the decoder, and $\mathbf{v}_i \in \mathbb{F}_q^m$ are linearly independent vectors of length m, and known to the decoder. Note, that in the matrix form we can write

$$\mathbf{e_{total}} = [e_1 \ldots e_t] \begin{bmatrix} \mathbf{u_1} \\ \hline \vdots \\ \hline \mathbf{u_t} \end{bmatrix} + [r_1 \ldots r_s] \begin{bmatrix} \mathbf{v_1} \\ \hline \vdots \\ \hline \mathbf{v}_s \end{bmatrix}. \tag{2}$$

[22], [29] present decoding algorithms for rank-metric codes.

C. MDS Array Codes for Distributed Storage

A linear *array code* C of dimensions $\alpha \times n$ over $\mathbb{F}_q$ is defined as a linear subspace of $\mathbb{F}_q^{\alpha n}$. Its minimum distance $d_{\min}$ is defined as the minimum Hamming distance over $\mathbb{F}_{q^\alpha}$, when we consider the codewords of C as vectors of length n over $\mathbb{F}_{q^\alpha}$. An array code C is called an (n, k) *maximum distance separable* (MDS) code if $|C| = q^{\alpha k}$, where $k = n - d_{\min} + 1$ [30], [31]. Note, that an MRD code is also an MDS array code.

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

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

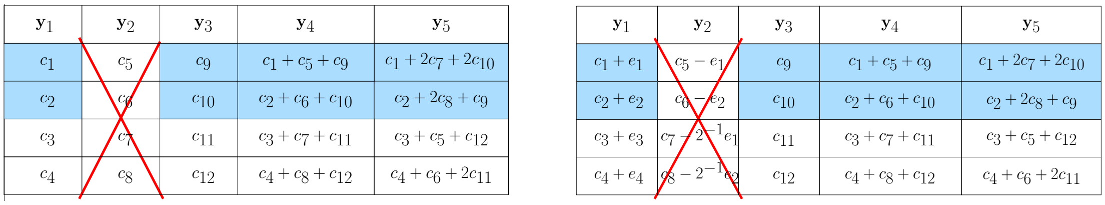
where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ and the generator matrix $\mathbf{G}$ is an $k \times n$ block matrix with blocks of size $\alpha \times \alpha$ given by:

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

An array code C *has an MDS property* if any blocks sub matrix of $\mathbf{G}$ of size $k \times k$ is of the full rank. In other words, if an (n, k) MDS code is used to store data in a system, and any set of $n - k$ storage nodes fails, the original data can be recovered from the k surviving nodes.

We say that an MDS code satisfies *optimal repair property*, if a single failed node can be repaired by downloading $\alpha d/(d-k+1)$ elements from every node of any d-subset of surviving nodes [1].

1) Examples of Optimal Repair MDS Array Codes: In the following, we present two examples of the optimal repair MDS array codes for DSS, which we will use further for illustration of our coding scheme. Due to space constraints, we only describe the MDS codes used in the examples. For general constructions, interested readers may refer to the respective papers that present these codes.



$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	$\mathbf{y}_4$	$\mathbf{y}_5$
c_1	c_5	c_9	$c_1+c_5+c_9$	$c_1+2c_7+2c_{10}$
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)

$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	$\mathbf{y}_4$	$\mathbf{y}_5$
c_1+e_1	c_5-e_1	c_9	$c_1+c_5+c_9$	$c_1+2c_7+2c_{10}$
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	$c_7-2^{-1}e_1$	c_{11}	$c_3+c_7+c_{11}$	$c_3+c_5+c_{12}$
c_4+e_4	$c_8-2^{-1}e_2$	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.

Example 1: (5,3) Zigzag code [8]. This class of MDS array codes [8] is based on generalized permutation matrices. For the $(5,3)$ Zigzag code presented in Fig. 1, the first three nodes are systematic nodes which store the data $[c_1, c_2, \dots, c_{12}]$, with $\alpha = 4$. The block generator matrix for this code is given by

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

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

where I and *0* denote the identity matrix and all-zero matrix, respectively. Fig. 1a describes node repair process for $(5,3)$ Zigzag code. When the second node fails, the newcomer node downloads the symbols from the shaded locations at the surviving nodes.

Example 2: $(5,3)$ Hadamard Design codes [6]. This class of MDS array codes employs interference alignment strategies in order to perform node repair. In the $(5,3)$ example presented in Fig. 2, the first three nodes are systematic nodes which store the data, $\mathbf{y}_i = [c_{(i-1)\alpha+1}, \dots, c_{i\alpha}]$, $1 \le i \le 3$. The block generator matrix for a $(5,3)$ Hadamard design based code is given by

$$\mathbf{G} = \begin{bmatrix} I & 0 & 0 & I & A_{1,5} \\ 0 & I & 0 & I & A_{2,5} \\ 0 & 0 & I & I & A_{3,5} \end{bmatrix},$$

where $A_{i,5}^T = a_i X_i + b_i X_4 + I$, $1 \le i \le 3$, $X_i = I_{2^{i-1}} \otimes \text{blkdiag}\left(I_{\frac{\alpha}{2^i}}, -I_{\frac{\alpha}{2^i}}\right)$, $\alpha = 2^4$, and a_i and $b_i \in \mathbb{F}_q$ satisfy $a_i^2 - b_i^2 = -1$. The process of the second node repair is illustrated in Fig. 2a. During this process, the newcomer downloads $\mathbf{y}_4 V$ and $\mathbf{y}_5 V$ from node 4 and 5, respectively (see D_1 in Fig. 2a). The newcomer uses $\widehat{V}$ and $\widetilde{V}$ as repair matrices corresponding to node 1 and 3, respectively, where $\widehat{V}$ and $\widetilde{V}$ are some basis for the column-space of $[V \;\; A_{1,5}V]$ and $[V \;\; A_{3,5}V]$, respectively. The information downloaded from node 1 and 3 is used to cancel the interference terms (contribution of $\mathbf{y}_1$ and $\mathbf{y}_3$ in D_1). After interference mitigation, a linear system of equations is solved to get $\mathbf{y}_2$.

III. Code Construction for Static Error Model

In this section we present our coding scheme and prove its error tolerance under the static error model. We illustrate the idea by using examples from Sec. II-C1 and prove that this construction is optimal for the static error model.

A. The Construction

Let $\mathbb{M} \in \mathbb{F}_q^{KN}$ denote a file of size $M = KN$, $K < N$. $\mathbb{M}$ is partitioned into K parts of size N each. We form an $N \times K$ matrix $\mathcal{M}$ over $\mathbb{F}_q$, where an ith part of $\mathbb{M}$ forms an ith column of $\mathcal{M}$, for all $1 \le i \le K$. Let $\mathcal{C}$ be an $[N \times m, \varrho = NK, \delta = m - K + 1]$ Gabidulin MRD code, with $m \le N$. Let $\mathbf{c}_{\mathcal{M}} \in \mathbb{F}_{q^N}^m$ be the codeword in $\mathcal{C}$ which corresponds to the information matrix $\mathcal{M}$. Let α, k be positive integers such that $m = \alpha k$ and let $Q = q^N$. Let C be an (n,k) optimal repair MDS array code of dimensions $\alpha \times n$. We partition the vector $\mathbf{c}_{\mathcal{M}}$ into k parts of size α and form n nodes of size α each, according to the encoding algorithm of the code C. Note, that we use an MDS array code over $\mathbb{F}_q \subseteq \mathbb{F}_{q^N} = \mathbb{F}_Q$, 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 theorem shows that if $2t\alpha + 1 \le \delta$ the system tolerates up to t erroneous nodes.

Theorem 1: Let t be the number of erroneous nodes in the system based on concatenated MRD and optimal repair MDS array codes. If $2t\alpha + 1 \le \delta$, then the original data can be recovered from any k nodes.

Proof: Let $\mathbf{c}_{\mathcal{M}} \in \mathbb{F}_Q^m$ be the codeword in $\mathcal{C}$ which corresponds to an information matrix $\mathcal{M}$, and let $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k]$, $\mathbf{x}_i \in \mathbb{F}_Q^\alpha$, be the partition of $c_{\mathcal{M}}$ into k parts of size α each. Let $[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$, $\mathbf{y}_i \in \mathbb{F}_Q^\alpha$ be the encoded blocks stored in n nodes.

Let $S = \{i_1, i_2, \dots, i_t\}$ be the set of indices of the erroneous nodes. Hence the i_jth node, $i_j \in S$, contains $\sum_{\ell=1}^k \mathbf{x}_\ell A_{\ell,i_j} + \mathbf{e}^{i_j}$, where $\mathbf{e}^{i_j} = [e_1^{i_j}, e_2^{i_j}, \dots, e_\alpha^{i_j}] \in \mathbb{F}_Q^\alpha$ denotes an adversarial error introduced by the i_jth node. When the failed nodes are being repaired, the errors from adversarial nodes propagate to the repaired nodes. In particular, ℓth node, $1 \le \ell \le n$, 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 MDS array code. Suppose a data collector contacts a subset D with k nodes 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 any ith node, $i \in D$. If these k nodes are all systematic nodes, then we obtain $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k] + \mathbf{e}\mathbf{B}$, where $\mathbf{B}$ is the blocks matrix of size $t\alpha \times k\alpha$ over $\mathbb{F}_q$ given by

$$\mathbf{B} = \begin{bmatrix} B_1^{i_1} & B_2^{i_1} & \dots & B_k^{i_1} \\ B_1^{i_2} & B_2^{i_2} & \dots & B_k^{i_2} \\ \vdots & \vdots & \ddots & \vdots \\ B_1^{i_t} & B_2^{i_t} & \dots & B_k^{i_t} \end{bmatrix}$$

and $\mathbf{e} = [\mathbf{e}^{i_1}, \mathbf{e}^{i_2}, \dots, \mathbf{e}^{i_t}]$. Otherwise, we obtain $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k] + \mathbf{e}\mathbf{B}'$, where the block matrix $\mathbf{B}' \in \mathbb{F}_q^{t\alpha \times k\alpha}$ represents the coefficients of $\mathbf{e}$ obtained by the decoding of the code C. Since the rank of $\mathbf{e}$ over $\mathbb{F}_q$ is at most $t\alpha$, and $\delta \geq 2t\alpha + 1$, the MRD code $\mathcal{C}$ can correct this error. ■

B. Examples

In this subsection we illustrate the idea of the construction for the case where an adversary pollutes the information stored on a single storage node. We demonstrate that the rank of the error introduced by an 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 a decoder for an MRD code. It is important to note that in our construction, any optimal repair MDS array code from DSS literature can be used as the inner code. In this subsection, we illustrate the idea of our construction with the help of two examples drawn from two different classes of optimal repair MDS array codes for DSS, presented in Section II-C1.

Example 3: Let C be the (5,3) Zigzag code from Example 1. Its first three systematic nodes store a codeword $\mathbf{c} = [c_1, c_2, \dots, c_{12}] \in \mathbb{F}_Q^{12}$, $Q = q^N$, from Gabidulin MRD code, which is obtained by encoding the original data. The content stored in ith systematic node, $1 \leq i \leq 3$, is $\mathbf{y}_i = [c_{(i-1)\alpha+1}, \dots, c_{i\alpha}] \in \mathbb{F}_Q^{\alpha}$. Note that $\alpha = 4$ in this example. 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 $\mathbb{F}_q$, $q \geq 3$. Now assume that a data collector accesses the first three nodes in an attempt to recover the original data. The data collector now has access to $\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{e}[I,\ B_2,\ 0]$, where

$$B_2 = \begin{bmatrix} -1 & 0 & -2^{-1} & 0 \\ 0 & -1 & 0 & -2^{-1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3}$$

Note that $\widetilde{\mathbf{c}}$ contains an error of rank at most four. Therefore, the original MRD codeword $\mathbf{c}$ and subsequently the original information can be recovered, using an MRD code with rank distance at least nine.

Example 4: Let C be a (5,3) Hadamard design based code, described in Example 2. Its first three nodes store $\mathbf{y}_i = [c_{(i-1)\alpha+1}, \dots, c_{i\alpha}] \in \mathbb{F}_Q^{\alpha}$, $1 \leq i \leq 3$, where $\mathbf{c} = [c_1, c_2, \dots, c_{3\alpha}] \in \mathbb{F}_Q^{3\alpha}$ is a codeword of a Gabidulin MRD code, which is obtained by encoding the original data. Suppose an adversary modifies the information stored at the first node to $\mathbf{y}_1 + \mathbf{e} = [c_1, \dots, c_\alpha] + [e_1, \dots, e_\alpha]$. When the second node fails, a newcomer, unaware of the presence of error at the first node, employs the interference alignment based strategy described in Example 2 and depicted in Fig. 2. After interference mitigation, a linear system of equations is solved to obtain $\mathbf{y}_2 + \mathbf{e}B_2$. Assuming that a data collector contacts the first three nodes, it receives $[\mathbf{y}_1,\ \mathbf{y}_2,\ \mathbf{y}_3] + \mathbf{e}[I,\ B_2,\ 0]$ which contains an error of rank at most α. This allows the recovery of uncorrupted information using an MRD code of sufficient minimum rank distance.

C. Code Parameters

The upper bound on the amount of data that can be stored reliably in the general system with $t < \frac{k}{2}$ corrupted nodes, called *resilience capacity* and denoted by $C(\alpha, \beta)$, was presented by Pawar et al. [12]. This bound is given by

$$C(\alpha, \beta) \leq \sum_{i=2t+1}^{k} \min\{(d-i+1)\beta, \alpha\}. \tag{4}$$

The authors in [12] provided the explicit construction of the codes that attain this bound, for bandwidth-limited regime. However, this construction has practical limitations for large values of t since the decoding algorithm presented in [12] is exponential in t.

Next, we show that our constructed codes attain bound (4) and thus, are optimal. In addition, the decoding of codewords in the construction presented in our paper is efficient since it is based on two efficient decoding algorithms: one, for an MDS array code, and two, for a Gabidulin code.

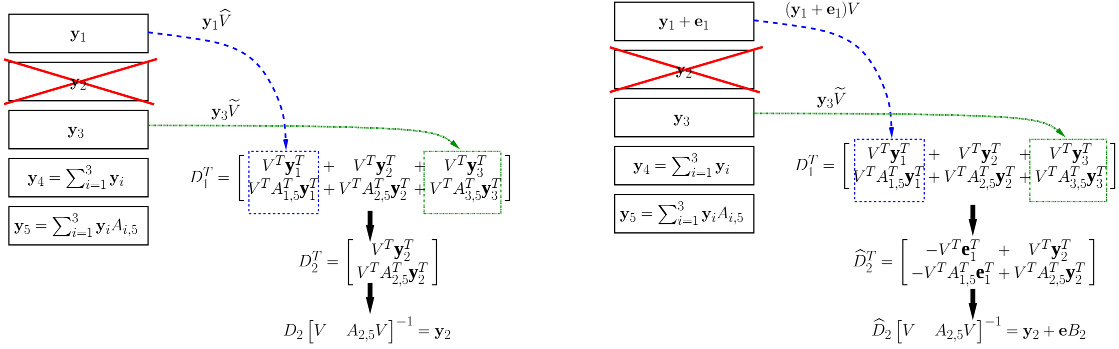
Let the parameters $K, N, m, \delta, \alpha, k, n$ be as described in the construction of Section III. Then $m = K + \delta - 1$ and $m = \alpha k$. Let t be an integer such that $\delta = 2t\alpha + 1$. Then $\alpha k = K + \delta - 1 = K + 2\alpha t$, and hence $K = \alpha(k - 2t)$.

Now we compare this result with the bound (4). Let C be an MDS array code with optimal repair property. Then $\beta = \frac{\alpha}{d-k+1}$. Therefore, we can rewrite bound (4) as follows:

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

Thus, the coding scheme proposed in this paper achieves the bound in (4).

Remark 1: Although the static error model is less general than the model considered in [12], the upper bound (4) applies to the static model as well. Pawar et al. [12] obtain this upper bound 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 correct under the static model. As shown in this paper, this bound is tight for



$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_3$
$\mathbf{y}_4 = \sum_{i=1}^3 \mathbf{y}_i$
$\mathbf{y}_5 = \sum_{i=1}^3 \mathbf{y}_i A_{i,5}$
$\mathbf{y}_1\widehat{V}$
$\mathbf{y}_3\widetilde{V}$
$D_1^T = \begin{bmatrix} V^T\mathbf{y}_1^T + V^T\mathbf{y}_2^T + V^T\mathbf{y}_3^T \\ V^T A_{1,5}^T\mathbf{y}_1^T + V^T A_{2,5}^T\mathbf{y}_2^T + V^T A_{3,5}^T\mathbf{y}_3^T \end{bmatrix}$
$D_2^T = \begin{bmatrix} V^T\mathbf{y}_2^T \\ V^T A_{2,5}^T\mathbf{y}_2^T \end{bmatrix}$
$D_2 \begin{bmatrix} V & A_{2,5}V \end{bmatrix}^{-1} = \mathbf{y}_2$
$\mathbf{y}_1 + \mathbf{e}_1$
$(\mathbf{y}_1 + \mathbf{e}_1)\widehat{V}$
$\widehat{D}_2^T = \begin{bmatrix} -V^T\mathbf{e}_1^T + V^T\mathbf{y}_2^T \\ -V^T A_{1,5}^T\mathbf{e}_1^T + V^T A_{2,5}^T\mathbf{y}_2^T \end{bmatrix}$
$\widehat{D}_2 \begin{bmatrix} V & A_{2,5}V \end{bmatrix}^{-1} = \mathbf{y}_2 + \mathbf{e}B_2$

(a) (b)

Fig. 2: Illustration of node repair in $(5,3)$ Hadamard design based codes: (a) in error free system, (b) in the presence of error at the first storage node.

static model at MSR point, which might not be the case for the general error model considered in [12].

Remark 2: Recently, Rashmi et al. [13] considered 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 (2) that our construction can also correct such erasures, as long as the minimum distance of the corresponding MRD code is large enough. The codes obtained by our construction also attain the bound on the capacity derived in [13]. However, 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.

IV. Dynamic Error Model

In this section, we consider the problem of designing coding schemes for DSS that work under dynamic error model. In this attack model, an adversary can have access to at most t storage nodes throughout the life span of DSS and can potentially send different data each time a node under its control is meant to send data during node repair as well as data reconstruction process.

Note that coding scheme proposed for the static error model hinges on the fact that 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 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 [12] and [13]. As we have discussed earlier, the coding scheme proposed in [12] does not have an efficient decoding during the data reconstruction process and it works specifically with bandwidth efficiently repairable codes at MBR point, however we focus on MSR point in this paper. The coding scheme of [13] deals with the dynamic error model at MSR point, but there scheme works only for low rate, i.e., $2k \leq n+1$. In coding scheme proposed in [13], an adversarial node does not inject errors into any other node during node repair, as this scheme allows newcomer to perform error-free exact repair.

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 by using a simple approach, which is explained in the following subsection. The second approach to cope with dynamic errors is based on existing literature on subspace signatures.

A. Naïve Scheme for Dynamic Error Model

In the dynamic error model, where there is no constraint on an adversarial nodes in terms of the data that it can provide during node repair as well as data reconstruction, one solution might be to design coding schemes that correct errors while node repair. One such scheme is proposed in [13] for low rate codes, where a newcomer node utilizes the redundancy in the downloaded data to perform correct 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 under the dynamic error model, if an error free node repair needs to be performed.

When a storage node fails, a newcomer node downloads $d\beta$ symbols from any d surviving nodes $(d \geq k)$. 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 an MRD codeword, by (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 an MRD codeword can be considered as the erased symbols. Let ℓ denote the number of information symbols (over $\mathbb{F}_{q^N}$) that are stored on the DSS. Then the minimum distance δ of the corresponding MRD code satisfies $\delta = k\alpha - \ell + 1$. Therefore we can reconstruct the entire MRD codeword and thus the data stored on the failed node, if we have

$$\delta = k\alpha - \ell + 1 \geq 2t\beta + (k-1)(\alpha - \beta) + 1.$$

This gives us

$$\ell \leq \alpha + (k - 2t - 1)\beta. \tag{5}$$

Note that the bound in (4) is still applicable. For $k = 2t + 1$, the right hand side expression in (5) is equal to that in (4). 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.

B. Subspace Signatures Approach

As mentioned previously, in 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 [15].

Let us consider an n-nodes DSS, which employs an MRD and a bandwidth efficiently repairable code based storage scheme as explained in Section III. For a node i, content stored on it, i.e. $\mathbf{y}_i$ can be viewed as an $N \times \alpha$ matrix over $\mathbb{F}_q$. These α vectors of length N stored on i^{th} node span a subspace (column space of $\mathbf{y}_i$ when viewed as a matrix over $\mathbb{F}_q$) in $\mathbb{F}_q^N$ of rank at most α. Since all elements of coding matrix and repair matrices are from $\mathbb{F}_q$, β linear combinations sent by node i for a node repair $\mathbf{y}_i V$ are nothing but, β vectors that lie in the subspace spanned by $\mathbf{y}_i$. If we make sure that even under dynamic error model an attacked node sends vectors from the same α-dimensional subspace of $\mathbb{F}_q^N$ during node repair and data construction, a data collector encounters at most $t\alpha$-rank error, which can be corrected with an MRD code of large enough rank distance as in the static model. Subspace signatures solve this problem of enforcing a node to send data (vectors) from the same α-rank subspace of $\mathbb{F}_q^N$.

At the beginning, when data is encoded to be stored on a DSS, system administrator also generates subspace signatures for data stored on each node according to the procedure explained in [15]. We assume existence of a trusted verifier, who stores all n subspace signatures, one signature for each storage node. Whenever a particular node sends data during node repair or data reconstruction, the truster verifier checks the data against the stored subspace signature corresponding to that particular storage node.

For the purpose of data reconstruction, whenever a node does not pass the signature test, that node is considered an rank erasure. 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. This bound on the rank of the error that the data collector encounters is explained in the following paragraph. Given that the outer MRD 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 by adversarial nodes under dynamic error model. Assume that node i fails. Let $\mathcal{R}_i \subseteq \{1, \ldots, n\} \backslash \{i\}$ 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 $\widehat{V}_{ji}$ is an $\alpha \times \beta$ matrix, which may be different form V_{ji}. Note that $\mathbf{y}_j \widehat{V}_{ji}$ is in the column space of $\mathbf{y}_j$.

If any of the surviving 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 in the first place 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 $\mathbb{F}_q^N$). The decoding algorithm for MRD codes is run on $(k-s)\alpha$ symbols downloaded from the 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 rank distance of the MRD 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 that is supposed to be 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 $\mathbf{y}_i + \mathbf{y}_{\mathbf{e}} B_i$, where B_i is an $t\alpha \times \alpha$ matrix over $\mathbf{F}_q$ and $\mathbf{y}_{\mathbf{e}} = [\mathbf{y}_{i_1}, \ldots, \mathbf{y}_{i_t}]$. Here $\{i_1, \ldots, i_t\}$ 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

$$\widetilde{\mathbf{y}} = \mathbf{y} + \mathbf{y}_{\mathbf{e}} \mathbf{B}, \tag{6}$$

where an ith column of $\mathbf{B}$ is equal to B_i, $1 \leq i \leq n$. It is evident from (6) that the rank of the aggregate error in the system is at most $t\alpha$ and an MRD code with large enough minimum distance can ensure the reliable recovery of the original data.

The idea of using hash functions to provide tolerance against adversarial errors in DSS has previously been presented in [10]. However, the error tolerance scheme of [10] involves periodic verification of the data stored in the system which may not be practical.

V. Error Resilience for Locally Repairable Codes

In this section we consider a special case where $\alpha = \beta = 1$ and each node can be repaired by accessing at most $r < k$ other nodes, where r is called *locality*. Note that in this scenario there *exist* r nodes for repair of a single node, and *not every* set of r nodes can be used for a node repair. Such codes were considered in [16], [17], [19], [20]. Clearly, such codes do not have an MDS property, but they have a small repair bandwidth.

It is proved [16] that the minimum distance of such a code satisfies

$$d_{\min} \le n-k+2-\left\lceil \frac{k}{r} \right\rceil . \tag{7}$$

An explicit construction of codes which attain this bound with systematic locality is presented in [16], [18]. However an explicit construction for codes that attain this bound and have all symbols locality is only known for $n = \lceil \frac{k}{r} \rceil (r+1)$ [19]. In what follows we show a new construction of codes with all symbols locality which attain bound (7) and can also correct static errors. This construction is also based on MRD codes.

A. Construction for Optimal Locally Repairable Codes

Let $\mathcal{C}$ be an $[N \times m, Nk, \delta = m-k+1]$ Gabidulin MRD code, where each codeword is considered as a vector of length m over $\mathbb{F}_{q^N}$, and let r be the given locality. Each symbol of an MRD codeword will be stored in a new node. We consider two cases: $m \equiv 0 (\text{mod } r)$ and $m \equiv k \equiv j (\text{mod } r)$, for $1 \le j < r$. In the first case we partition the set of m coordinates into $\frac{m}{r}$ disjoint groups G_i, $1 \le i \le \frac{m}{r}$, of size r, and in the second case we partition the set of m coordinates into $\left\lfloor \frac{m}{r} \right\rfloor$ disjoint groups G_i of size r and one additional group $\widehat{G}$ of size j. For each group G_i we add a new parity node p_i to be the sum of all the r symbols in the same group. For the group $\widehat{G}$ the new parity symbol $\widehat{p}$ is equal to the sum of the j elements of $\widehat{G}$. We denote by C^{loc} the constructed code.

Example 5: Let $m = N = 8$, $k = 6$, $\delta = 3$ $r = 4$, and $\mathbf{c} = [c_1, c_2, \ldots, c_8]$ be a codeword of an $[8 \times 8, 8 \cdot 6, 3]$ MRD code, $c_i \in \mathbb{F}_{q^8}$. Then the corresponding codeword of C^{loc} is equal to $\mathbf{c}^{\text{loc}} = [c_1, c_2, \ldots, c_8, p_1, p_2]$, where $p_1 = c_1 + c_2 + c_3 + c_4$, and $p_2 = c_5 + c_6 + c_7 + c_8$.

Theorem 2: A code C^{loc} attains bound (7), i.e., it is an $[n, k, d_{min}]$ code with $d_{min} = n - k + 2 - \lceil \frac{k}{r} \rceil$.

Proof: As it was mentioned in Section II, a codeword of a Gabidulin code can be considered as an evaluation of a linearized polynomial $f(x) \in \mathbb{F}_{q^N}[x]$ on the set of m linearly independent points over $\mathbb{F}_q$, $\{g_1, \ldots, g_m\}$. Note that for the reconstruction the original data we need at least k values $f(a_1), \ldots, f(a_k)$, such that $a_1, \ldots, a_k \in \mathbb{F}_{q^N}$ are linearly independent over $\mathbb{F}_q$ [17]. By the linearity of $f(x)$ we have that for any group $G \neq \widehat{G}$, the new parity p_i is equal to $f(g_{i_1} + g_{i_2} + \ldots + g_{i_r})$, for $i_s \in G$, and for $\widehat{G}$ we have $\widehat{p} = f(g_{i_1} + g_{i_2} + \ldots + g_{i_j})$, for $i_s \in \widehat{G}$. Hence, any r symbols in $G_i \cup \{p_i\}$ are the evaluation of $f(x)$ in r linearly independent points. Now we distinguish between two cases.

Case 1, $m \equiv 0 (\text{mod } r)$**:** In this case, $n = m + \frac{m}{r}$. Let i and j be two integers such that $k = m - r(i+1) + j$, $0 \le i \le \frac{m}{r} - 1$, and $0 \le j \le r - 1$. Then $\frac{m}{r} - \left\lceil \frac{k}{r} \right\rceil = i$, and according to bound (7), $d_{\min} - 1 \le m + \frac{m}{r} - r(\frac{m}{r} - (i+1)) - j - \left\lceil \frac{k}{r} \right\rceil + 1 = r(i+1) - j + i + 1 = (r+1)(i+1) - j$. We will prove that any $(r+1)(i+1) - j$ erasures can be corrected by this code. In other words, after this number of erasures we still have an evaluation of $f(x)$ in k linearly independent points. Note that the worst case is when the erasures appear in the smallest possible number of groups and when the number of erasures inside a group is maximal. So we consider the case when all the symbols in i groups are erased, and there is a group with $r+1-j$ erasures. Then the number of the remaining symbols which correspond to linearly independent points is $m - ri - (r-j) = k$.

Case 2, $m \equiv k \equiv j (\text{mod } r)$**:** for $1 \le j < r$. In this case, $n = m + \left\lceil \frac{m}{r} \right\rceil$. Let i be an integer such that $k = ir + j$, $0 \le i \le \left\lfloor \frac{m}{r} \right\rfloor$. Since $m = \left\lfloor \frac{m}{r} \right\rfloor r + j$, then by bound (7), $d_{\min} - 1 \le (\left\lfloor \frac{m}{r} \right\rfloor r + j + \left\lfloor \frac{m}{r} \right\rfloor + 1) - (ir + j) + 1 - (i+1) = (\left\lfloor \frac{m}{r} \right\rfloor - i)(r+1) + 1$. As in the previous case, we consider the worst case for erasures, when all the symbols in $(\left\lfloor \frac{m}{r} \right\rfloor - i)$ groups and one additional symbol are erased. Then the number of the remaining symbols which correspond to linearly independent points is $m - (\left\lfloor \frac{m}{r} \right\rfloor - i)r = j + ir = k$. ■

Example 6: In this example, we prove that the code from Example 5 has minimum distance 4, in other words, this code can tolerate any three erasures.

- any 2 erasures can be corrected by the MRD code, since its minimum rank distance is 3.
- if two erasures are in the same group and additional one erasure in another group, then we still have the values of the corresponding linearized polynomial $f(x)$ in the $3 + 4 = 7$ linearly independent points,
- if all three erasures are in the same group, then we have values of $f(x)$ in $2 + 4 = 6$ linearly independent points.

In all the cases, since $k = 6$, we can correct these erasures.

Note that if there is an adversarial node in such a system, then the erroneous data spreads into the whole group containing this node because of the node repair property, in other words, the Hamming weight of an error is equal to the size of a group. However, the rank of such an error is one, and it can be corrected by an MRD code with $\delta \ge 3$. We generalize this property of a code C^{loc} in the following theorem.

Theorem 3: If the related to C^{loc} MRD code $\mathcal{C}$ has minimum rank distance $\delta \ge 2t + 1$ then C^{loc} can tolerate at most t static errors.

Example 7: Consider the code from Example 5, and let $t = 1$. Its rank distance is 3, and the Hamming distance is 4. Since the group size is 5, by using only Hamming metric it is impossible to correct this error, however, by applying a decoding of MRD code we can correct it.

Remark 3: The construction proposed in this section can be seen as a particular case of the construction of self-repairing homomorphic codes [17], since it can also be described in terms of linearized polynomials. In addition, this construction is closely related to construction of pyramid codes [18], and the construction given in [19], since an MRD code can be also seen as an MDS code over an extension field.

VI. Conclusion

A novel concatenated coding scheme for distributed storage system was presented. The scheme makes use of rank-metric codes, in particular, MRD codes as the first step of encoding the data. In the second step of encoding MDS optimal repair array codes or locally repairable codes are used. This construction provides resilience against static adversarial errors. Moreover, when using MDS optimal repair array codes this scheme is optimal in terms of the resilience capacity, and when using locally repairable codes this scheme is optimal in terms of the minimum distance. A modification of the scheme based on subspace signatures provides resilience against dynamic errors. The resilience of this scheme against a passive eavesdropper is a subject of our current research [32].

References

[1] A. G. Dimakis, P. Godfrey, M. Wainwright, and K. Ramachandran, “Network coding for distributed storage system”, in *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4539-4551, September 2010.

[2] Y. Wu and A. G. Dimakis, “Reducing repair traffic for erasure coding-based storage via interference alignment”, in *Proc. of IEEE ISIT*, 2009.

[3] N. B. Shah, K. V. Rashmi, P. V. Kumar and K. Ramchandran, “Explicit codes minimizing repair bandwidth for distributed storage”, in *Proc. of IEEE ITW*, Jan 2010.

[4] C. Suh and K. Ramchandran, “Exact-repair MDS codes for distributed storage using interference alignment”, in *Proc. of IEEE ISIT*, 2010.

[5] K. V. Rashmi, N. B. Shah and P. V. Kumar, “Optimal Exact-regenerating Codes for Distributed Storage at the MSR and MBR point via a Product-Matrix Construction”, *IEEE Transactions on Information Theory*, vol. 57, no. 57, pp. 5227-5239, Aug. 2011.

[6] D. Papailiopoulos and A. Dimakis, “Repair optimal erasure codes through hadamard designs,” in *Proc. of Allerton*, 2011.

[7] V. Cadambe, C. Huang and J. Li, “Permutation code: Optimal exact-repair of a single failed node in mds code based distributed storage systems,” in *Proc. of IEEE ISIT*, 2011.

[8] I. Tamo, Z. Wang, and J. Bruck, “Zigzag codes: MDS array codes with optimal rebuilding,” *CoRR*, vol. abs/1112.0371, 2011.

[9] Z. Wang, I. Tamo and J. Bruck, “On Codes for Optimal Rebuilding Access”, in *Proc. Allerton Conf. on Control, Computing and Communication*, Sep. 2011.

[10] T. K. Dikaliotis, A. G. Dimakis, T. Ho, “Security in distributed storage systems by communicating a logarithmic number of bits”, in *Proc. of ISIT*, 2010.

[11] F. Oggier and A. Datta, “Byzantine Fault Tolerance of Regenerating Codes”, in *Proc. of IEEE Int. Conf. on Peer-to-Peer Computing*, 2011.

[12] S. Pawar, S. El Rouayheb and K. Ramchandran, “Securing Dynamic Distributed Storage Systems against Eavesdropping and Adversarial Attacks”, *IEEE Trans. Inform. Theory* , vol. 57, no. 9, Sep. 2011.

[13] K. V. Rashmi. N. B. Shah, K. Ramchandran, and P. V. Kumar, “Regenerating codes for errors and erasures in distributed storage”, *CoRR*, vol. abs/1202.1050, 2012.

[14] N. B. Shah, K. V. Rashmi and P. V. Kumar, “Information-theoretically secure Regenerating Codes for Distributed Storage”, in *Proc. of IEEE Globecom*, Dec. 2011.

[15] F. Zhao, T. Kalker, M. Medard, and K. J. Han, “Signatures for content distribution with network coding”, *in Proc. of ISIT*, June 2007.

[16] P. Gopalan, C. Huang, H. Simitchi, and S. Yekhanin, “On the locality of codeword symbols”, *CoRR*, vol. abs/1106.3625, 2011.

[17] F. Oggier and A. Datta, “Homomorphic self-repairung codes for agile maintenance of distributed storage systems”, *CoRR*, vol. abs/1107.3129, 2011.

[18] C. Huang, M. Chen, and J. Li, “Pyramid code: Flexible schemes to trade space for access efficiency in relible data storage systems”, in 6th IEEE nternational symposium on Network Computing and Applications (NCA 2007),pp.79–86, 2007.

[19] N. Prakash, G. M. Kamath, V. Lalitha, and P. V. Kumar, “Optimal linear codes with a local-error-correction property”, *CoRR*, vol. abs/1107.3129, 2011.

[20] D.S. Papailiopoulos and A.G. Dimakis, “Locally Repairable Codes”, to appear in ISIT 2012

[21] 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.

[22] E. M. Gabidulin, “Theory of codes with maximum rank distance,” *Problems of Information Transmission*, vol. 21, pp. 1-12, July 1985.

[23] R. M. Roth, “Maximum-rank array codes and their application to crisscross error correction,” *IEEE Trans. Inform. Theory*, vol. 37, pp. 328-336, March 1991.

[24] P. Lusina, E. Gabidulin, and M. Bossert,“Maximum Rank Distance codes as space-time codes,” *IEEE Trans. Inform. Theory*, vol. 49, pp. 2757–2760, Oct. 2003.

[25] D. Silva, F. R. Kschischang, and R. Koetter, “A rank-metric approach to error control in random network coding,” *IEEE Trans. Inform. Theory*, vol. 54, pp. 3951–3967, Sep. 2008.

[26] T. Etzion and N. Silberstein, “Error-correcting codes in projective space via rank-metric codes and Ferrers diagrams”, *IEEE Trans. Inform. Theory*, vol. 55, no.7, pp. 2909–2919, July 2009.

[27] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, “Ideals over a non-commutative ring and their application in cryptology,” *LNCS*, vol. 573, pp. 482-489, 1991.

[28] F. J. MacWilliams and N. J. A. Sloane, *T*he theory of error-correcting codes, North-Holland, 1978.

[29] D. Silva and F. R. Kschischang, “Fast Encoding and Decoding of Gabidulin Codes.” in *Proc. of IEEE ISIT*, 2009.

[30] M. Blaum and R. M. Roth, “On lowest density MDS codes”, *IEEE Trans. Inform. Theory*, vol. 45, pp. 46–59, 1999.

[31] Y. Cassuto and J. Bruck,“Cyclic low-density MDS array codes”, in *Proc. of IEEE ISIT*, 2006.

[32] O. O. Koyluoglu, A. S. Rawat, N. Silberstein, and S. Vishwanath, ”On secrecy in distributed storage systems”, in preparation.