Optimal Locally Repairable and Secure Codes for Distributed Storage Systems

Ankit Singh Rawat, O. Ozan Koyluoglu, Natalia Silberstein, and Sriram Vishwanath

***Abstract*—This paper aims to go beyond resilience into the study of security and locality for distributed storage systems. Security and locality are both important as features of an efficient storage system, and this paper aims to understand the tradeoffs between resilience, security and locality in these systems. In particular, this paper first investigates security in the presence of colluding eavesdroppers, where eavesdroppers are assumed to work together in decoding stored information. Second, the paper focuses on coding schemes that enable optimal local repairs. It further brings these two concepts together, to develop locally-repairable coding schemes for DSS that are secure against eavesdroppers.**

The main results of this paper include: a. An improved bound on the secrecy capacity for minimum storage regenerating codes, b. secure coding schemes that achieve the bound for some special cases, c. new minimum distance bound for locally repairable codes, d. code construction for locally repairable codes that achieves the minimum distance bound, and e. repair-bandwidth-efficient locally repairable codes with and without security constraints.

***Index Terms*—Coding for distributed storage systems, locally repairable codes, repair bandwidth efficient codes, security.**

I. Introduction

A. Background

Distributed storage systems (DSS) are of increasingly importance, given the vast amounts of data being generated and accessed worldwide. OceanStore [1], Google File System (GFS) [2] and TotalRecall [3] are a few examples of existing DSS. An essential component of DSS is resilience to node failures, which is why every DSS today incorporates a mechanism to protect against failures, thus preventing permanent loss of data stored using the system. Typically, this resilience is afforded by replication, and in recent years, using coding approaches.

Node failures are one of the many design challenges faced by DSS. There are two other challenges, arguably of equal importance: security and locality. Due to the decentralized nature of such systems, it is important that they be secured against a variety of possible attacks. Our focus in this paper is on passive eavesdroppers located at multiple nodes in the DSS that can collude in attempting to gain an understanding of the stored data. In addition to being decentralized, DSS systems are often widely geographically distributed, and therefore locality in storage proves very useful. In this paper, we develop a deeper understanding of locality in storage, and subsequently combine locality and security to develop codes for secure locally-repairable DSS.

The authors are with the Laboratory of Informatics, Networks and Communications, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78751 USA. E-mail: ankitsr@utexas.edu, {ozan, natalys, sriram}@austin.utexas.edu.

The security of communication or storage systems can be analyzed with their resilience to active or passive attacks [4], [5]. Active attacks in such systems include settings were the adversary modifies existing packets or injects new ones into the system, whereas the passive attack models include eavesdroppers observing the messages being stored/transmitted. For DSS, cryptographic approaches are often ineffective, as key distribution and management between all nodes in the system is extremely challenging to accomplish. A coding/information theoretic approach to security is desired, which typically offers stronger security guarantees than cryptographic schemes [6], [7] and, in this context, is logistically easier to realize than mechanisms that require key management. A secrecy-enabling coding scheme is designed based on a worst-case estimate of the information leaked to eavesdroppers, and can naturally complement other existing coding schemes being utilized in distributed storage systems. In its simplest form, security against an eavesdropper can be achieved using a one-time pad scheme [8]. For example, consider that the contents of the two nodes are given by $X_1 = R$, and $X_2 = R \oplus d$, where R is a uniformly random bit, and d is the data bit. Then, by contacting both nodes, one can clearly obtain the data by computing $X_1 \oplus X_2$. However, one can not get any information about the data bit by observing any one of the two nodes as $I(X_i; D) = 0$ for $i = 1, 2$, i.e., the mutual information between the data and the content of one of the nodes is zero. Thus, information theoretic approach has clearly a significant value in securing DSS.

Local-repairability of DSS is an additional property, which can be one of the primary design criteria for the system. The corresponding performance metric associated with a coding scheme is its *locality* r, which is defined as the number of nodes that must participate in a repair process when a particular node fails. Locality requires fewer nodes to be involved in the node repair process, which makes the entire process easier from a logistical perspective. In addition, locality is of significant interest when a cost is associated with contacting each node in the system. Locality, in its simplest form, can be accomplished by splitting the data into groups, and each group can be coded and stored separately. However, this naïve approach requires the connection to all the groups in order to retrieve the whole data, and may not be the most efficient in terms of performance. Therefore, there is a growing interest in more sophisticated mechanisms for achieving locality in DSS. Regardless, systems designed with locality in mind can also present benefits in terms of security. In other words, locality

and security against eavesdropper attack go hand in hand, and a joint design of both features can prove to be particularly useful, as we illustrate in this paper.

In DSS, encoding data before storing it provides the same level of resilience against node failures as that of the conventional approach of uncoded replication, but with much less storage space. The advantages that can be leveraged in terms of storage space may result in a degradation of other performance metrics. Being one of such metrics, *repair bandwidth* refers to the amount of data that needs to be transferred in the event of single node failure in order to regenerate the data on the failed node. This metric is highly relevant as a large fraction of network bandwidth in DSS can be occupied by the data being transferred during repair process. Thus, it is desirable to have coding schemes with small repair bandwidth. Most of the *maximal distance separable* (MDS) codes designed for DSS, which encode k data blocks to n encoded blocks, store each encoded block on different nodes. This naïve approach entails a high repair bandwidth as the entire original file needs to be reconstructed in order to regenerate the encoded data stored at a particular storage node. In [9], Dimakis et al. explore this problem and establish a trade off between the per node storage and repair bandwidth for a code that has the MDS ("any k out of n") property, i.e., entire data can be reconstructed by a data collector by contacting to any k storage nodes. This new class of codes are referred to as *regenerating codes*, and allows for trading off repair bandwidth for storage [9]. Utilizing a network coding approach, the notion of *functional repair* is considered in [9], where the original failed node may not be replicated exactly, but can be repaired as an encoded data that is *functionally* equivalent. However, it is desirable to perform *exact repair* in DSS, where the data regenerated after the repair process is an exact replica of what was stored on the failed node. This is essential due to the ease of maintanence and other practical purposes, e.g., maintaining a code in its systematic form. Exact repair is also advantageous compared to the functional repair in the presence of eavesdroppers, as the latter scheme requires updating the coding rules which may leak additional information to eavesdroppers [10]. Noting the resilience of exact repair to eavesdropping attacks and the necessity of it for practical purposes, it is of significant interest to design regenerating codes that not only enjoy an optimal trade off in repair bandwidth vs. storage, but also satisfy exact repair in addition to security and/or locality constraints.

B. Contributions and Organization

In this paper, we consider secure and locally repairable regenerating codes for DSS. As a security constraint, we adopt the passive and colluding eavesdropper model presented in [11], where, during the entire life span of the DSS, the eavesdropper can get access to data stored on an ℓ_1 number of nodes, and, in addition, it observes both the stored content and the data downloaded (for repair) on an additional ℓ_2 number of nodes. This attack model generalizes the eavesdropper model proposed in [10], which considers the case of $\ell_2 = 0$. As the amount of information downloaded when a node repair is in progress is equal to the information stored on the repaired node for minimum bandwidth regenerating codes, the two notions are different only at the minimum storage regenerating point.

With this general eavesdropper model, we extend the existing results on the design of secure minimum storage regenerating codes for DSS. First, we derive an upper bound on secrecy capacity, the amount of data that can be stored on the system without leaking information to an eavesdropper, for a DSS employing bandwidth efficient node repair. Our bound is novel in that it can take into account the additional downloaded data at the eavesdroppers, and is tighter than the available bounds in the literature. Second, we present a secure, exact repairable coding scheme that has a higher code rate compared to that of [11]. Utilizing a special case of the obtained bound, we show our both codes achieve the optimal secure file size for any (ℓ_1, ℓ_2) when $\ell_2 \leq 2$.

Third, we shift focus to locally repairable regenerating codes. We derive an upper bound on the minimum distance of the vector codes, possibly non-linear, that satisfy a given locality constraint. We develop this bound using the proof technique used in [12], [13][1]. Fourth, based on maximal rank distance (MRD) codes, we construct a coding scheme which achieves this bound on minimum distance. Here, we establish a per node storage vs. resilience trade off similar to [13], and study bandwidth efficiency in locally repairable DSS. We present a minimum distance optimal repair bandwidth efficient coding scheme. Finally, we consider the problem of providing secrecy against passive eavesdropper for locally repairable codes and present a secure locally repairable regenerating code for DSS modifying the aforementioned coding scheme.

In all the scenarios we study in this paper, the achievability results allow for exact repair, and we obtain secure file size upper bounds from mincut analyses over the secrecy graph representation of distributed storage systems. Our main secrecy achievability coding argument are obtained by utilizing a secret sharing scheme with MRD codes, similar to the classical work of [15].

The rest of the paper is organized as follows. In the next section, we provide a summary of related work to the problems studied in this paper. In Section II, we provide a general system model together with some preliminary results utilized throughout the text. In Section III, we reproduce a classical setup for the problem, and provide an enhanced upper bound on secure file size as well as a new secure coding scheme for minimum storage regenerating codes. In Section IV, we focus on locally repairable codes, providing new bounds on minimum distance of such codes. We also present a new coding scheme that achieves these bounds. In Section V, we present locally repairable codes with security constraints. Finally, we conclude the paper in Section VI. To improve the presentation of the paper, some of the results and proofs are relegated to appendices.

C. Related Work

In [9], Dimakis et al. characterize the information theoretic trade off between repair bandwidth vs. per node storage for

[1]This also shows that the proof technique used in [14] based on generalized hamming weights, which only works for systematic codes, is not essential.

DSS satisfying the MDS ("any k out of n") property. Based on network coding results, functional repair is considered, and the life span of DSS, for a given set of node failures, is mapped to a multicast problem over a dynamic network. Using this mapping, the authors show that network coding based storage schemes achieve the lower bound on repair bandwidth allowing "functional repair" [9]. [16] and [17] present coding schemes that achieve the lower bound on repair bandwidth. The work in [18]–[20] devise low rate codes, which achieve the lower bound derived in [9] when data is downloaded from all surviving nodes during exact node repair. The coding schemes in [18] and [19], [20] are tailored for $k < 3$ and $k \leq \frac{n}{2}$, respectively. In [21], Rashmi et al. design exact-repairable codes, which allow node repair to be performed by contacting $d \leq n-1$ surviving nodes. These codes are optimal for all parameters (n, k, d) at the minimum bandwidth regeneration (MBR) point. At the minimum storage regeneration (MSR) point, these codes belong in low rate regime, as their rate is upper bounded by $\frac{1}{2} + \frac{1}{2n}$. Recently, researchers have devised high rate exact repairable codes for the MSR point. [22] presents codes for DSS with two parity nodes, which accomplish exact regeneration while being optimal in repair bandwidth. In [23] and [24], permutation-matrix based codes are designed to achieve the bound on repair bandwidth for systematic node repair for all (n, k) pairs. [25] further generalizes the idea of [24] to get MDS array codes for DSS that allow optimal exact regeneration for parity nodes as well.

Towards obtaining coding schemes with "good" locality, Oggier et al. present coding schemes which facilitate local node repair in [26], [27]. In [12], Gopalan et al. establish an upper bound on the minimum distance of locally repairable linear scalar codes, which is analogous to singleton bound. They also show that pyramid codes, presented in [28], achieve this bound. Subsequently, the work by Prakash et al. extends the bound to a more general definition of locally repairable scalar linear codes [14]. In [13], Papailiopoulos et al. generalize the bound in [12] to vector codes, possibly non-linear, and establish per node storage vs. resilience trade off. They also present locally repairable coding schemes, which exhibits "k out of n" property at the cost of small amount of excess storage space per node.

The problem of designing secure DSS against eavesdropping has been addressed in [10]. In [10], Pawar et al. consider an eavesdropper, which can get access to the data stored on ℓ $(< k)$ storage nodes of DSS, operating at the MBR point with "any k out of n" property. They derive an upper bound on the amount of data that can be stored on such a system without leaking any information to the eavesdropper, and present a coding scheme in the "bandwidth limited regime" that achieve this bound. Shah et al. consider the design of secure regenerating codes at the MSR point [11] as well. Since the amount of data downloaded for node repair at the MSR point is more than what is eventually stored on the repaired node, the eavesdropper may obtain more information if it is able to access the data downloaded when a node repair is in progress. Therefore, at the MSR point, the eavesdropper is modeled as accessing the data stored on ℓ_1 nodes and data downloaded during ℓ_2 node repairs (corresponding to distinct nodes), with $\ell_1 + \ell_2 < k$. Shah et al. present a coding scheme that achieves the bound on secrecy capacity in [10] at the MBR point based on product matrix codes [21]. They further use product matrix codes based solution for MSR point as well, which matches the bound in [10] only when $\ell_2 = 0$. Thus, the secrecy capacity for MSR codes is considered to be open when the eavesdropper is allowed to observe downloaded information. Moreover, the solution at the MSR point gives only low rate schemes as product matrix codes are themselves low rate codes.

There is a closely related line of work on designing coding schemes for DSS that are resilient against active attacks, where an adversary is allowed to modify the content stored on a certain number of nodes through out the life span of the DSS. The goal of coding scheme is to allow successful decoding of the original data at a data collector even in the presence of erroneous data injected by the active adversary [10], [29], [30].

II. System Model and Preliminaries

Consider a DSS with n live nodes at a time and a file $\mathbf{f}$ of size $\mathcal{M}$ over $\mathbb{F}_q$ that needs to be stored on the DSS. In order to store the file $\mathbf{f}$, it is divided into k blocks of size $\frac{\mathcal{M}}{k}$ each. Let $(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ denotes these k blocks. Here, we have $\mathbf{f}_i \in \mathbb{F}_q^{\frac{\mathcal{M}}{k}}$. These k data blocks are encoded into n data blocks, $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, each of length α over $\mathbb{F}_q$ ($\alpha \geq \frac{\mathcal{M}}{k}$). The encoding process is summarized by the function

$$\mathbb{G} : \left(\mathbb{F}_q^{\frac{\mathcal{M}}{k}}\right)^k \rightarrow \left(\mathbb{F}_q^{\alpha}\right)^n. \tag{1}$$

Note that we don't restrict ourselves to linear class of functions. The function $\mathbb{G}$ may very well be a nonlinear function. Let $\mathcal{C}$ denote the codebook associated with the encoding function $\mathbb{G}$. Given the codewords, node i in an n-node DSS stores encoded block $\mathbf{x}_i$. In this paper, we use $\mathbf{x}_i$, to represent both block $\mathbf{x}_i$ and a storage node storing this encoded block interchangeably. Motivated by the MDS property of the codes that are traditionally developed for data storag in centralized storage systems [31]–[33], the works on regenerating codes focus on storage schemes that have "any k out of n" property are designed and analyzed.

Given this setup, as the network evolves over failures and repairs, we use the following notation to denote the contents and downloaded symbols of the nodes. The symbols stored at node i is represented by the vector $\mathbf{s}_i$, the symbols transmitted from node i to node j is denoted as $\mathbf{d}_{i,j}$, and the set $\mathbf{d}_j$ is used to denote all of the downloaded symbols to node j. DSS is initialized with the n nodes containing encoded symbols, i.e., $\mathbf{s}_i = \mathbf{x}_i$ for $i = 1, \cdots, n$. In the event of failure of i-th storage node, a new node, namely the newcomer, is introduced to the system. This node contacts to d storage nodes and downloads β symbols from each of these nodes. The newcomer nodes use these $d\beta$ number of downloaded symbols to regenerate α symbols, $\mathbf{x}_i$, and store these symbols. This exact repair process preserves the MDS property, i.e., data stored on any k nodes (potentially including the nodes that are repaired) allows the original file $\mathbf{f}$ to be reconstructed.

We note that, for linear encoding schemes, the symbols of node i can be written as $\mathbf{s}_i = \{\mathbf{f}^T\mathbf{g}_i^1, \cdots, \mathbf{f}^T\mathbf{g}_i^\alpha\}$. In such a case, we refer to $\mathcal{S}_i$ as the subspace spanned by the vectors $\{\mathbf{g}_i^1, \cdots, \mathbf{g}_i^\alpha\}$. For node repairs, using a similar notation, we consider node i to transmit symbols $\mathbf{d}_{i,j} = \{\mathbf{f}^T\mathbf{g}_{i,j}^1, \cdots, \mathbf{f}^T\mathbf{g}_{i,j}^\beta\}$ to node j, where $g_{i,j}^r \in \mathcal{S}_i$. We also refer to $\mathcal{D}_{i,j}$ as the subspace spanned by vectors $\{\mathbf{g}_{i,j}^1, \cdots, \mathbf{g}_{i,j}^\beta\}$. $\mathcal{D}_j$ then will be referred to as the subspace downloaded to node j, which will have a certain dimension in this subspace representation. For a given set of nodes $\mathcal{A}$, we use the notation $\mathbf{s}_\mathcal{A} \triangleq \{\mathbf{s}_i, i \in \mathcal{A}\}$. A similar notation is adopted for the downloaded symbols, and the subspace representation. Throughout the text, we usually stick to the notation of having vectors denoted by lower-case bold letters; and, sets and subspaces being denoted with calligraphic fonts. $[n]$ denotes the set $\{1, 2, \ldots, n\}$.

A. Information flow graph

In their seminal work [9], Dimakis et al. models the operation DSS using a multicasting problem over information flow graph (see Fig. 1). Information flow graph consists of three types of nodes:

- Source node (S): Source node contains $\mathcal{M}$ symbols long original file $\mathbf{f}$. The source node is connected to n nodes.
- Storage nodes ($(x_i^{\text{in}}, x_i^{\text{out}})$): Each storage node is represented by a pair of nodes, input node x_i^{in} and output node x_i^{out}. Here, x_i^{in} denotes the data downloaded by node i, whereas x_i^{out} denotes the α symbols actually stored on node i. An edge of capacity α is introduced between x_i^{in} and x_i^{out} to enforce the storage constraint of α symbols per node. For a newcomer node, x_i^{in} is connected to x_i^{out} node of d live nodes with links of capacity β symbols each, representing the data downloaded during node repair.
- Data collector nodes (DC_i): Each data collector contacts x^{out} node of k live nodes by the edges of capacity ∞ each.

With the aforementioned values of capacities of various edges in the information flow graph, the DSS is said to employ an (n, k, d, α, β) code. For a given graph $\mathcal{G}$ and data collectors DC_i, the file size that can be stored in such a DSS can be bounded using the max flow-min cut theorem for multicasting using network coding [34].

Lemma 1 (Max flow-min cut theorem for multicasting [9], [34])**.**

$$\mathcal{M} \leq \min_{\mathcal{G}} \min_{\text{DC}_\text{i}} \text{maxflow}(\text{S} \rightarrow \text{DC}_\text{i}, \mathcal{G}),$$

where $\text{flow}(\text{S} \rightarrow \text{DC}_\text{i}, \mathcal{G})$ *represents the flow from the source node* S *to data collector* DC_i *over the graph* $\mathcal{G}$*.*

Therefore, e.g., for the graph in Fig. 1, $\mathcal{M}$ symbol long file can be delivered to a data collector DC, only if the min cut is at least $\mathcal{M}$. In [9], Dimakis et al. consider k successive node failures and evaluate the min-cut over possible graphs, and obtain the bound given by

$$\mathcal{M} \leq \sum_{i=0}^{k-1} \min\{(d-i)\beta, \alpha\}. \tag{2}$$

This bound can be achieved by employing linear codes, linear network code in particular. The codes that attain the bound in (2) are known as regenerating codes [9]. Given a file size $\mathcal{M}$, a trade off between storage per node α and *repair bandwidth* $\gamma \triangleq d\beta$ can be established from (2). 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 former is obtained by first choosing a minimum storage per node (i.e., $\alpha = \mathcal{M}/k$), and then minimizing γ satisfying (2), whereas the latter is obtained by first finding the minimum possible γ and then finding the minimum α in (2). For MSR codes, we have:

$$(\alpha_{\text{msr}}, \beta_{\text{msr}}) = \left(\frac{\mathcal{M}}{k}, \frac{\mathcal{M}}{k(d-k+1)}\right). \tag{3}$$

On the other hand, MBR codes are characterized by

$$(\alpha_{\text{mbr}}, \beta_{\text{mbr}}) = \left(\frac{2\mathcal{M}d}{k(2d-k+1)}, \frac{2\mathcal{M}}{k(2d-k+1)}\right). \tag{4}$$

For a given DSS with $d \leq n-1$, it can be observed that having $d = n-1$ reduces the repair bandwidth at both MSR and MBR points. Though the bound in (2) is derived for *functional repair*, the bound and the achievability of MSR and MBR points are shown to be tight for *exact repair* as well.

B. MRD codes

Most of the encoding schemes presented in this paper use optimal rank-metric codes. An $[N \times m, \varrho, \varsigma]$ *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, $d_R(A, B) \geq \varsigma$, where $d_R(\cdot, \cdot)$ denotes the rank distance defined by

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

For an $[N \times m, \varrho, \varsigma]$ rank-metric code $\mathcal{C}$ we have $\varrho \leq \min\{N(m-\varsigma+1), m(N-\varsigma+1)\}$ [35]–[37]. This bound is called Singleton bound for rank metric, and the codes that achieve this bound are called *maximum rank distance* (MRD) codes. A construction of MRD codes was given by Gabidulin [36]. These codes can be seen as the analogs of Reed-Solomon codes for rank metric. A codeword in an $[N \times m, \varrho, \varsigma]$ rank-metric code $\mathcal{C}$, for $m \leq N$, can be represented by a vector $\mathbf{c} = [c_1, c_2, \ldots, c_m]$ over $\mathbb{F}_{q^N}$. In the similar way as Reed-Solomon codes, Gabidulin codes can be obtained by evaluation of polynomials, however, for Gabidulin codes the special family of polynomials, called *linearized polynomials*, is used. A linearized polynomial $f(y)$ over $\mathbb{F}_{q^N}$ of q-degree n has the form $f(y) = \sum_{i=0}^{n} a_i y^{q^i}$, where $a_i \in \mathbb{F}_{q^N}$, and $a_n \neq 0$.

A codeword in Gabidulin code $\mathcal{C}$ is defined as $\mathbf{c} = (f(y_1), f(y_2), \ldots, f(y_m))$, where $f(y)$ is the linearized polynomial of q-degree $m - \varsigma$ with coefficients given by the information message, and $y_1, \ldots, y_m \in \mathbb{F}_{q^N}$ are linearly independent points over $\mathbb{F}_q$ [36]. Note that evaluation of a linearized polynomial is an $\mathbb{F}_q$-linear transformation from $\mathbb{F}_{q^N}$ to itself, i.e., for any $a, b \in \mathbb{F}_q$ and $y_1, y_2 \in \mathbb{F}_{q^N}$, we have $f(ay_1 + by_2) = af(y_1) + bf(y_2)$ [38].

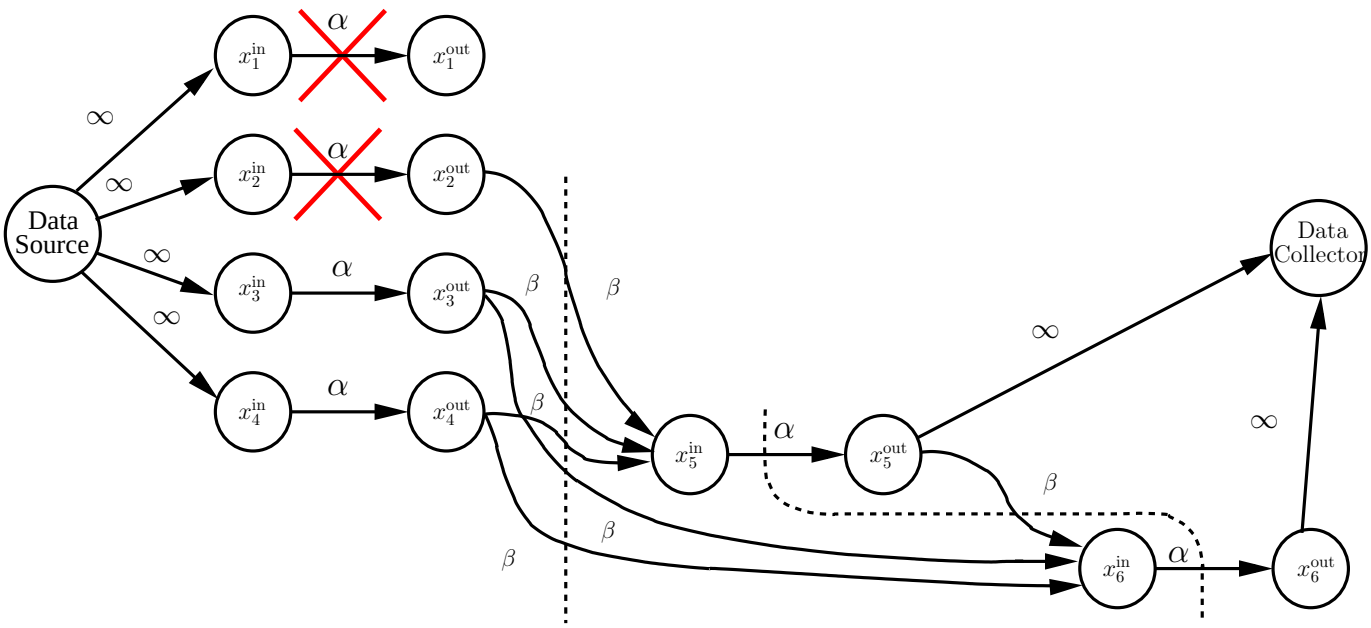

Fig. 1: Information flow graph of DSS. Assuming that x_1 fails first the newcomer, x_5 contacts $\{x_2, x_3, x_4\}$ during node repair. In the event of second node failure, x_2, data is downloaded from $\{x_3, x_4, x_5\}$ by the newcomer x_6.

C. Eavesdropper model

In this paper, we consider the eavesdropper model defined in [11], which generalizes the eavesdropper model considered in [10]. In [10], Pawar et al. consider a passive eavesdropper, who can access the data stored on ℓ $(< k)$ storage nodes. The eavesdropper is assumed to know the coding scheme employed by the DSS. At the MBR point, a newcomer downloads $\alpha_{\text{mbr}} = \gamma_{\text{mbr}} = d\beta_{\text{mbr}}$ amount of data. Thus, an eavesdropper does not gain any additional information if it is allowed to access the data downloaded during repair. However, at the MSR point repair bandwidth is strictly greater than the per node storage α_{msr}, and an eavesdropper potentially gains more information if it is has access to data downloaded during node repair as well. Motivated by this, we consider an (ℓ_1, ℓ_2) eavesdropper, which can access the stored data of nodes in the set $\mathcal{E}_1$, and additionally can access both the stored and downloaded data at the nodes in the set $\mathcal{E}_2$ with $\ell_1 = |\mathcal{E}_1|$ and $\ell_2 = |\mathcal{E}_2|$. Hence, the eavesdropper has access to $x_i^{\text{out}}, x_j^{\text{in}}, x_j^{\text{out}}$ for $i \in \mathcal{E}_1$ and $j \in \mathcal{E}_2$. We summarize the eavesdropper model together with the definition of achievablity of a secure file size in the following.

Definition 2 (Security against an (ℓ_1, ℓ_2) eavesdropper)**.** *A distributed storage system is said to achieve a secure file size of $\mathcal{M}^s$ against an (ℓ_1, ℓ_2) eavesdropper, if, for any sets $\mathcal{E}_1$ and $\mathcal{E}_2$ of size ℓ_1 and ℓ_2, respectively, $I(\mathbf{f}^s; \mathbf{e}) = 0$. Here $\mathbf{f}^s$ is the secure file of size $\mathcal{M}^s$, which is first encoded to file $\mathbf{f}$ of size $\mathcal{M}$, and $\mathbf{e}$ is the eavesdropper observation vector given by $\mathbf{e} \triangleq \{x_i^{\text{out}}, x_j^{\text{in}}, x_j^{\text{out}} : i \in \mathcal{E}_1, j \in \mathcal{E}_2\}$.*

Note that, this definition coincides with the $\{\ell, \ell'\}$ secure distributed storage system in [11], where $\ell = \ell_1 + \ell_2$ and $\ell' = \ell_2$.

In MSR coding schemes with high rate the number of parity-check nodes is negligible relatively to the number of systematic nodes. Hence in the following we consider the codes with optimal exact repair of systematic nodes, and we assume also that $\mathcal{E}_2$ is contained in the set of systematic nodes.

We remark that, as it will be clear from the following sections, when a file $\mathbf{f}$ of size $\mathcal{M}$ is stored in DSS and the secure file size achieved is $\mathcal{M}^s$, the remaining $\mathcal{M} - \mathcal{M}^s$ symbols can be utilized as public data, which does not have security constraints. Yet, noting the possibility of storing the public data, we will refer to this uniformly distributed part as the random data, which is utilized to achieve security. Throughout the text, we use the following lemma to show that the proposed codes satisfy the secrecy constraints.

Lemma 3. *Consider a system with information bits $\mathbf{u}$, random bits $\mathbf{r}$ (independent of $\mathbf{u}$), and an eavesdropper with observations given by $\mathbf{e}$. If $H(\mathbf{e}) \leq H(\mathbf{r})$ and $H(\mathbf{r}|\mathbf{u}, \mathbf{e}) = 0$, then $I(\mathbf{u};\mathbf{e})=0$.*

Proof: See Appendix A. ■

D. Locally repairable codes

First we present a general definition of the minimum distance of a code, and then we give an equivalent formulation of it, which will be used in the following sections in the sequel.

Definition 4 (Minimum distance of a code)**.** *Let $\mathcal{L}$ denotes a set of nodes that get erased. For a code associated with encoding function $\mathbb{G}$, as defined in (1), its minimum distance $d_{\min}$ is defined to be the cardinality of the smallest set $\mathcal{L}_m$, for which we have*

$$H\left(\mathbf{x}_{[n]} \backslash \mathbf{x}_{\mathcal{L}_m}\right) = H\left(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{n-|\mathcal{L}_m|}}\right) < \mathcal{M}. \tag{5}$$

Here $\{i_1, \ldots, i_{n-|\mathcal{E}|}\} = [n] \backslash \mathcal{L}_m$.

According to an alternate definition for $d_{\min}$, as given in [12] for scalar linear codes and later extended by [13] for

general codes,

$$d_{\min} = n - \max_{\mathcal{A}\subseteq[n]:H(\mathbf{x}_{\mathcal{A}})<\mathcal{M}} |\mathcal{A}| \tag{6}$$

where $\mathcal{A} = \{i_1, \ldots, i_{|\mathcal{A}|}\} \subseteq [n]$ and $\mathbf{x}_{\mathcal{A}} = (\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{|\mathcal{A}|}})$. It follows from the definition of d_{min} that a data collector can reconstruct the original data, i.e., $\mathbf{f}$, by contacting any set of $n - d_{\min} + 1$ storage nodes in the DSS. We are interested in ensuring this property of the DSS during its entire life span despite of its dynamic nature due to node repairs. Besides this in locally repairable DSS, we are interested in coding schemes, i.e., $\mathcal{C}$, that have following property:

(r, δ) **locality:** For each stored block $\mathbf{s}_i$ (of length α), there exists a set of nodes $\Gamma(i)$ of size at most $r + \delta - 1$ such that all elements of $\Gamma(i)$ have following two properties:

- Any set of r nodes in $\Gamma(i)$ are independent, i.e., for any $\{j_1, \ldots, j_r\} \subseteq \Gamma(i)$, we have
$$H(\mathbf{s}_{j_1}, \ldots, \mathbf{s}_{j_r}) = r\alpha \tag{7}$$
- Each element $j \in \Gamma(i)$ can be written as a function of any set of r elements in $\Gamma(i)$ (not containing j). In other words, minimum distance of $\mathcal{C}|_{\Gamma(i)}$, the code obtained by puncturing $\mathcal{C}$ over $\Gamma(i)$, is at least δ.

Codes that satisfy this property are called (r, δ, α) locally repairable codes.

III. Secrecy in repair bandwidth efficient DSS

Considering that the eavesdropped nodes may not carry secure information to the data collectors in the bound given by (2), [10] establishes the following upper bound on the secure file size when the eavesdropper observes the content of ℓ nodes.

$$\mathcal{M}^s \leq \sum_{i=\ell+1}^{k} \min\{(d - i + 1)\beta, \alpha\}. \tag{8}$$

Pawar et al. show that this bound is tight in the *bandwidth limited regime*, $\gamma \leq \Gamma = (n-1)\alpha$ with $d = n - 1$, by presenting a coding scheme that is secure against the passive eavesdropper observing ℓ storage nodes. This point essentially corresponds to MBR point (see (4)) when a data collector contacts all the remaining nodes. [11] proposes product matrix based secure coding schemes achieving this bound for any ℓ at the MBR point. However, the coding scheme proposed in [11] can only store a secure file size of $(k - \ell_1 - \ell_2)(\alpha - \ell_2\beta)$ at the MSR point. At the MSR point, the bound in (8) reduces to

$$\mathcal{M}^s \leq (k - \ell_1 - \ell_2)\alpha.$$

From these, it is concluded in [11] that the proposed scheme achieves secrecy capacity only when $\ell_2 = 0$. This corresponds to the scenario for which the eavesdroppers are not allowed to observe downloaded packets. This leaves the following questions open:

- Can bound (8) be further tightened for MSR point?
- Is it possible to get a secure code at the MSR point that outperforms the performance of the code proposed in [11]?

In this section, we answer both questions affirmatively. We first derive a generic upper bound on the amount of data that can be securely stored on DSS for bandwidth efficient repairable codes at the MSR point, which also applies to bandwidth efficient exact repairable code. Next, we prove a result specific to exact repairable code for $d = n - 1$, which allows us to provide an upper bound on the file size that can be securely stored on a DSS against an (ℓ_1, ℓ_2)-eavesdropper. This bound is tighter than a bound that can be obtained from the generic bound we provide. We subsequently combine the classical secret sharing scheme due to [15] with an existing class of exact repairable MSR codes to securely store data in the presence of an (ℓ_1, ℓ_2) eavesdropper. We show that this approach gives a higher rate coding scheme compared to that of [11] and achieves the secrecy capacity when $\ell_2 \leq 2$ for any ℓ_1.

A. Improved bound on secrecy capacity at the MSR point

In order to get desired bound, we rely on the standard approach of computing a cut in information flow graph associated with DSS. We consider a particular pattern of eavesdropped nodes, where eavesdropper observes content put on ℓ_1 initial nodes and data downloaded during first ℓ_2 node failures that do not involve already eavesdropped ℓ_1 nodes. Using the min cut-max flow theorem, this case translates into an upper bound on the secrecy capacity for any MDS encoding scheme that operates on MSR point (see (3)), one extreme of the repair bandwidth vs. per node storage trade off defined in (2).

Theorem 5. *For a bandwidth efficient repairable (n, k) MDS code, we have*

$$\mathcal{M}^s \leq \sum_{i=\ell_1+1}^{k-\ell_2} \left(\alpha - \dim \left(\sum_{j=1}^{\ell_2} \mathcal{D}_{i,n+j} \right) \right) \tag{9}$$

Proof: Consider Fig. 2, which describes a particular case that may arise during the lifespan of a DSS. Here, $x_1, x_2, \ldots, x_n$ represent the original n storage nodes in DSS as defined in Sec. II. Assume that nodes $x_{k-\ell_2+1}, \ldots, x_k$ fail subsequently in the order specified by their indices. These ℓ_2 failures are repaired by introducing nodes $x_{n+1}, \ldots, x_{n+\ell_2}$ in the system following a node repair process associated with the coding scheme employed by the DSS. Consider $\mathcal{E}_1 = \{x_1, \ldots, x_{\ell_1}\}$ as the set of ℓ_1 nodes, where eavesdropper observes the stored content, and $\mathcal{E}_2 = \{x_{n+1}, \ldots, x_{n+\ell_2}\}$ be the set of nodes which are exposed to the eavesdropper during their node repair, allowing eavesdropper to have access to all the data downloaded during node repair of set $\mathcal{E}_2$. Let $\mathcal{R}$ denote the set of $k - (\ell_1 + \ell_2)$ remaining original nodes $\{x_{\ell_1+1}, \ldots, x_{k-\ell_2}\}$, which are not observed by the eavesdropper directly, and information stored on these nodes may leak to eavesdroppers only when these nodes participate in node repair. Assume that a data collector contacts a set of k nodes given by $\mathcal{K} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{R}$ in order to reconstruct the original data. For a file $\mathbf{f}^s$ to be securely stored on the DSS, we have

$$H(\mathbf{f}^s) = H(\mathbf{f}^s | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) \tag{10}$$

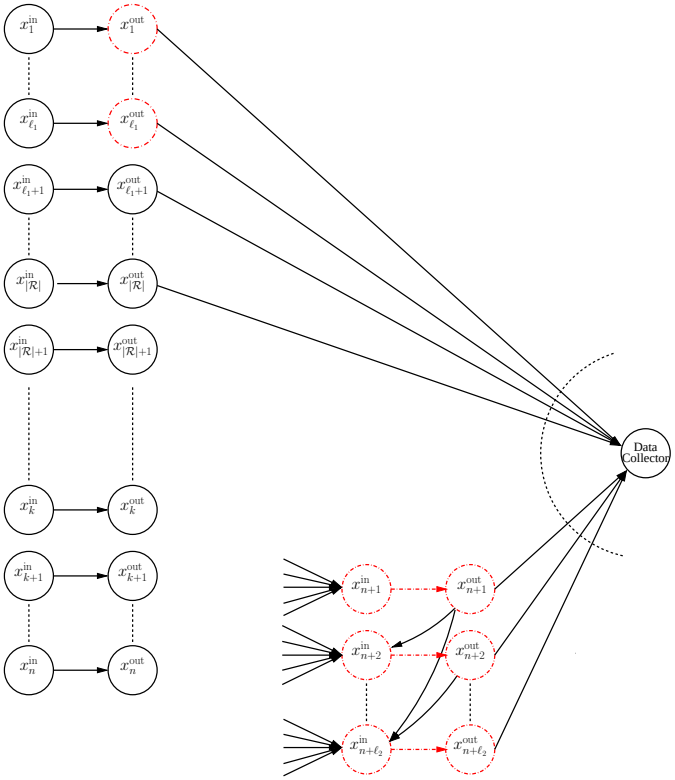

Fig. 2: Node repair in the presence of (ℓ_1, ℓ_2) passive eavesdropper.

$$
\begin{aligned}
&= H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) - H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1}, \mathbf{s}_{\mathcal{E}_2}, \mathbf{s}_{\mathcal{R}}) \qquad (11)\\
&\leq H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) - H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}, \mathbf{s}_{\mathcal{R}})\\
&= I(\mathbf{f}^s; \mathbf{s}_{\mathcal{R}}|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2})\\
&\leq H(\mathbf{s}_{\mathcal{R}}|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2})\\
&\leq H(\mathbf{s}_{\mathcal{R}}|\mathbf{d}_{\mathcal{E}_2})\\
&= \sum_{i=\ell_1+1}^{k-\ell_2} H(\mathbf{s}_i|\mathbf{s}_{\ell_1+1}, \ldots, \mathbf{s}_{i-1}, \mathbf{d}_{\mathcal{E}_2})\\
&\leq \sum_{i=\ell_1+1}^{k-\ell_2} H(\mathbf{s}_i|\mathbf{d}_{i,n+1}, \ldots, \mathbf{d}_{i,n+\ell_2})\\
&\leq \sum_{i=\ell_1+1}^{k-\ell_2} \left(\alpha - \dim\left(\sum_{j=1}^{\ell_2} \mathcal{D}_{i,n+j}\right)\right) \qquad (12)
\end{aligned}
$$

Here (10) follows from the fact that coding scheme employed in DSS is secure against an (ℓ_1, ℓ_2) eavesdropper, i.e., $I(\mathbf{f}^s; \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) = H(\mathbf{f}^s) - H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) = 0$. (11) is a consequence of MDS property of the code, i.e., the original data can be recovered from data stored on any set of k nodes. ■

In Theorem 5, $\dim\left(\sum_{j=1}^{\ell_2} D^i_{n+j}\right)$ can be trivially lower bounded by β to obtain the following corollary.

Corollary 6. *For a DSS employing an (n, k, d, α, β) MSR regenerating code, we have:*

$$\mathcal{M}^s \leq (k - \ell_1 - \ell_2)(\alpha - \beta). \qquad (13)$$

This shows that the secure code construction proposed in [11] is optimal for $\ell_2 = 1$.

The following lemma is specific to exact repairable linear codes at the MSR point that employ interference alignment for node repair with $d = n - 1$. It is shown in [39] that interference alignment is a necessary component of an exact repairable linear scalar ($\beta = 1$) MSR code. The necessity of interference alignment holds for $\beta > 1$ as well. Therefore, the following bound is fairly general and apply to all known exact repairable codes at the MSR point. Following the standard terminology in DSS literature, each node i, has $\beta \times \alpha$ repair matrices, $\{V_{i,j}\}$, associated with remaining nodes $j \neq i$. In the event of failure of node j, a newcomer downloads $V_{i,j}\mathbf{x}_i$ from every node i, $i \neq j$. In rest of the section, we use $V_{i,j}$ to denote both a matrix and row-space of the matrix.

Lemma 7. *Consider an (n, k)-DSS storing data in a systematic form with $(n - k)$ linear parity nodes. Assume that $d = n-1$, i.e., all the remaining nodes are contacted to repair a failed node. Let $V_{i,j}$ be the repair matrices associated with node i, which is used to perform interference alignment based node repair for node j. Then for each $i \in [k]$, i.e., systematic nodes, we have*

$$\dim\left(\bigcap_{j\in\mathcal{A}} V_{i,j}\right) = \operatorname{rank}\left(\bigcap_{j\in\mathcal{A}} V_{i,j}\right) \leq \frac{\alpha}{(n-k)^{|\mathcal{A}|}}, \qquad (14)$$

where $\mathcal{A} \subseteq [k]\backslash\{i\}$.

Proof: See Appendix B. ■

It follows from the well-known dimension formula for vector spaces that

$$
\begin{aligned}
&\dim\left(\mathcal{D}_{i,n+1} + \mathcal{D}_{i,n+2}\right)\\
&= \dim\left(\mathcal{D}_{i,n+1}\right) + \dim\left(\mathcal{D}_{i,n+2}\right) - \dim\left(\mathcal{D}_{i,n+1} \cap \mathcal{D}_{i,n+2}\right)\\
&= \beta + \beta - \dim\left(\mathcal{D}_{i,n+1} \cap \mathcal{D}_{i,n+2}\right)\\
&\geq 2\beta - \frac{\alpha}{(n-k)^2}, \qquad (15)
\end{aligned}
$$

where (15) follows from Lemma 7. Now combining (15) with Theorem 5, we get the following corollary:

Corollary 8. *Given a bandwidth efficient repairable (n, k) MDS code with $d = n-1$ that employs interference alignment to perform node repair, for $\ell_2 \leq 2$ we have*

$$\mathcal{M}^s \leq (k - \ell_1 - \ell_2)\left(\alpha - \kappa\left(\alpha, \beta, \ell_2\right)\right) \qquad (16)$$

where

$$\kappa\left(\alpha, \beta, \ell_2\right) = \begin{cases} \beta, & \text{if } \ell_2 = 1 \\ 2\beta - \frac{\alpha}{(n-k)^2}, & \text{if } \ell_2 = 2 \end{cases} \qquad (17)$$

B. Construction of secure MSR codes for $d = n - 1$

In this subsection we present a construction which is based on concatenation of MRD codes [35]–[37] and optimal repair MDS array codes, called zigzag codes [24], [25]. The construction of (n, k) zigzag code is given in [25]. Let $p = n-k$. Then, this construction provides a $p^k \times n$ array with a $p^k \times k$ systematic part. The repair of a systematic node (column) j is performed by accessing rows $Y_j = \{x \in [0, p^k - 1] : x \cdot e_j =$

$0\}$, where e_j is an element of the standard basis for $\mathbb{Z}_p^k$, and x is represented with an element of $\mathbb{Z}_p^k$.

We first state the following property of this repair process.

Lemma 9. *Assume that an eavesdropper gains access to the data stored in ℓ_1 nodes and the data stored as well as the data downloaded during node repair in ℓ_2 systematic nodes in a $(k+p,k)$ zigzag code. Then the eavesdropper can only observe*

$$kp^k - p^k(k-\ell_1-\ell_2)\left(1-\frac{1}{p}\right)^{\ell_2}$$

systematic symbols.

Proof: First note that

$$|Y_j| = p^{k-1}$$
$$|Y_i \cap Y_j| = p^{k-2}, \text{ for } i \neq j,$$

and in general

$$|Y_{i_1} \cap Y_{i_2} \ldots \cap Y_{i_t}| = p^{k-t}, \text{ for } i_1 \neq i_2 \neq \cdot \neq i_t.$$

Let $\mathcal{E}_2 \subseteq [k]$ be the set of size ℓ_2 of systematic nodes (columns) where an eavesdropper has access to the stored data and to the downloaded during node repair data. Then by using inclusion-exclusion principle, we have

$$\begin{aligned}
|\cup_{j\in\mathcal{E}_2} Y_j| &= \ell_2 \cdot p^{k-1} - \binom{\ell_2}{2}\cdot p^{k-2} + \binom{\ell_2}{3}\ldots p^{k-3}\ldots \\
&= \sum_{i=1}^{\ell_2}(-1)^{i-1}\binom{\ell_2}{i}p^{k-i} \\
&= -p^{k-\ell_2}\sum_{i=1}^{\ell_2}(-1)^{i}\binom{\ell_2}{i}p^{\ell_2-i} \\
&= (-p^{k-\ell_2})\left(\sum_{i=0}^{\ell_2}(-1)^{i}\binom{\ell_2}{i}p^{\ell_2-i} - p^{\ell_2}\right) \\
&= (-p^{k-\ell_2})((p-1)^{\ell_2} - p^{\ell_2}) \\
&= p^k - p^{k-\ell_2}(p-1)^{\ell_2}.
\end{aligned}$$

Then, the eavesdropper can observe

$$\begin{aligned}
&p^k(\ell_1+\ell_2) + (k-\ell_1-\ell_2)|\cup_{j\in\mathcal{E}_2} Y_j| \\
&= p^k(\ell_1+\ell_2) + (k-\ell_1-\ell_2)(p^k - p^{k-\ell_2}(p-1)^{\ell_2}) \\
&= kp^k - p^k(k-\ell_1-\ell_2)\left(1-\frac{1}{p}\right)^{\ell_2}
\end{aligned}$$

systematic symbols. ∎

We now detail the achievability scheme of this section. Let $[N\times k\alpha, Nk\alpha, 1]$ be a Gabidulin MRD code, $N \geq k\alpha$, with $\alpha = p^k$ [36]. Let $f(y) = \sum_{i=0}^{k\alpha-1} c_i y^{q^i}$, $c_i \in \mathbb{F}_{q^N}$, be the corresponding linearized polynomial, i.e., the coefficients of this polynomial are chosen as the information symbols, and a codeword of length $k\alpha$ (over $\mathbb{F}_{q^N}$) is obtained by its evaluation in $k\alpha$ linearly independent (over $\mathbb{F}_q$) elements of $\mathbb{F}_{q^N}$.

Secrecy achieving encoding of the data will be performed as follows. First, we choose $kp^k - p^k(k-\ell_1-\ell_2)(1-\frac{1}{p})^{\ell_2}$ random symbols over $\mathbb{F}_{q^N}$ and consider them as the largest coefficients of the encoding polynomial. Then, we choose the remaining $p^k(k-\ell_1-\ell_2)(1-\frac{1}{p})^{\ell_2}$ coefficients of the polynomial using the symbols of the secure file. The result of this MRD encoding will be encoded by using a $(k+p,k)$ zigzag code. Note that since the evaluation of $f(\cdot)$ is a $\mathbb{F}_q$-linear function, all the symbols in the parity-check nodes of the final code are given by the evaluation of $f(\cdot)$ in the linear combinations of the evaluation elements of the systematic nodes. This property of the constructed code will be called a *linearized property*.

This code achieves the following secure file size.

Theorem 10. *The secure code obtained by MRD secrecy precoding of a zigzag code at the MSR point with $\alpha = p^k$ achieves a secure file size given by*

$$\mathcal{M}^s = (k-\ell_1-\ell_2)p^k\left(1-\frac{1}{p}\right)^{\ell_2},$$

where $p = n-k$, for $d = n-1$. In addition, for any (ℓ_1,ℓ_2) such that $\ell_2 \leq 2$, this code attains the upper bound on the secure file size given in Corollary 8, and achieves the secrecy capacity at the MSR point with $d = n-1$.

Proof: The repair and data reconstruction properties of the proposed code follow from the construction of zigzag codes [24], [25]. The proof of security follows by Lemma 9, Lemma 3, and the linearized property of the code. (We note that a similar proof of security when utilizing polynomials for encoding is provided in the seminal paper of A. Shamir on secret sharing [15].)

Substituting $\ell_2 = 1$ (or 2), $\alpha = p^k$ and $\beta = \frac{p^k}{p} = p^{k-1}$ in (16) shows that the proposed code construction achieves the upper bound on secure file size, specified in Corollary 8, for $\ell_2 \leq 2$. ∎

IV. New Bounds and Constructions for Locally Repairable Codes

In this section, we study the notion of local repairability for DSS. As opposed to the line of work on scalar locally repairable codes [12], [14], [28], where each node stores a scalar over a field from a codeword, we consider vector locally repairable codes, which have previously been considered in [13], [27]. Furthermore, in addition to the vector construction, the (r,δ,α) codes we consider, as defined in Section II, allow for the possibility of $\alpha > \mathcal{M}/k$, and non-trivial locality, i.e., the possibility of $\delta > 2$. Thus, these codes are generalizations of vector locally repairable codes given in [13], which considered only the $\delta = 2$ case. We note that we are particularly interested in vector locally repairable code with multiple local parities. Among other advantages, codes having multiple local parities exhibits a stronger resilience to eavesdropping. In particular, as detailed in Sec. V, both scalar locally repairable codes and vector locally repairable codes with single local parity have poor secrecy rate in the presence of a passive eavesdropper.

We first derive an upper bound on the minimum distance of (r,δ,α) codes, which also applies to non-linear codes. We follow the proof technique of [12], [13], which is given for the single local parity case, and modify it for multiple local

parity nodes. The bound derived in this section gives the bound presented in [14] as a special case without the assumption of having a systematic code. As noted in [13], the bound on $d_{\min}$ establishes a resilience vs. per node storage trade off, where per node storage α can be increased over $\mathcal{M}/k$ to obtain higher $d_{\min}$. This is of particular interest in the design of codes having both locality and strong resilience to node failures.

Next, we propose a general code construction which achieves the derived bound on $d_{\min}$. We use MRD codes along with MDS array codes to obtain this construction. In this section, we further introduce the notion of *repair bandwidth* for locally repairable codes and obtain an upper bound on the amount of data that can be stored in the DSS while supporting a given repair bandwidth. We note that the idea and analysis of repair bandwidth is similar to the classical work in the area of repair bandwidth efficient code [9]. Here, the presence of multiple local parity nodes can be utilized to repair a local node efficiently by contacting more than r nodes from the same group. The notion of bandwidth efficient node repair within a local group becomes important in Sec. V, where we study the locally repairable codes under secrecy constraints.

A. Upper bound on $d_{\min}$ for an (r, δ, α) locally repairable code

We state a generic upper bound on the minimum distance $d_{\min}$ of an (r, δ, α) code $\mathcal{C}$. (The definition of $d_{\min}$ is provided in Section II.) This will establish a trade off between node failure resilience (i.e., $d_{\min}$) and per node storage (α).

Theorem 11. *Let $\mathcal{C}$ be an (r, δ, α) locally repairable code over $\mathbb{F}_q^{\alpha}$. Then, it follows that*

$$d_{\min}(\mathcal{C}) \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \quad (18)$$

Proof: In order to get the aforementioned upper bound on minimum distance of an (r, δ, α) locally repairable code, we utilize the dual definition of minimum distance of a code as given in (6). Similar to the proof in [12] and [13], we construct a set $\mathcal{A} \subseteq [n]$ such that

$$H(\mathbf{s}_{\mathcal{A}}) < \mathcal{M}. \quad (19)$$

This along with (6) give us an upper bound on $d_{\min}(\mathcal{C})$.

The construction of a set $\mathcal{A}$ is given in Fig. 3. Next, we show a lower bound on the size of the set $\mathcal{A}$, output of the algorithm described in Fig. 3. Note that at each iteration of the while loop in Fig. 3, the algorithm increases the size of the set $\mathcal{A}_{i-1}$ by at most $r+\delta-1$ to get $\mathcal{A}_i$. For each i, define

$$a_i = |\mathcal{A}_i| - |\mathcal{A}_{i-1}|. \quad (20)$$

and

$$h_i = H(\mathbf{s}_{\mathcal{A}_i}) - H(\mathbf{s}_{\mathcal{A}_{i-1}}). \quad (21)$$

Assume that the algorithm terminates at $(\ell+1)^{th}$ iteration, i.e., $\mathcal{A} = \mathcal{A}_\ell$. Then it follows from (20) and (21) that

$$|\mathcal{A}| = |\mathcal{A}_\ell| = \sum_{i=1}^{\ell} a_i, \quad (22)$$

$$H(\mathbf{s}_{\mathcal{A}}) = H(\mathbf{s}_{\mathcal{A}_\ell}) = \sum_{i=1}^{\ell} h_i. \quad (23)$$

1: Set $\mathcal{A}_0 = \emptyset$ and $i = 1$.
2: **while** $H(\mathbf{s}_{\mathcal{A}_{i-1}}) < \mathcal{M}$ **do**
3: Pick a coded block $\mathbf{s}_{j_i} \notin \mathcal{A}_{i-1}$ s.t. $|\Gamma(j_i) \backslash \mathcal{A}_{i-1}| \geq \delta - 1$.
4: **if** $H(\mathbf{s}_{\mathcal{A}_{i-1}}, \mathbf{s}_{\Gamma(j_i)}) < \mathcal{M}$ **then**
5: set $\mathcal{A}_i = \mathcal{A}_{i-1} \cup \Gamma(j_i)$
6: **else if** $H(\mathbf{s}_{\mathcal{A}_{i-1}}, \mathbf{s}_{\Gamma(j_i)}) \geq \mathcal{M}$ and $\exists \mathcal{B} \subset \Gamma(j_i)$ s.t. $H(\mathbf{s}_{\mathcal{A}_{i-1}}, \mathbf{s}_{\mathcal{B}}) < \mathcal{M}$ **then**
7: set $\mathcal{A}_i = \mathcal{A}_{i-1} \cup \mathcal{B}$
8: **else**
9: $i = i + 1$, **end while**
10: **end if**
11: $i = i + 1$
12: **end while**
13: Output: $\mathcal{A} = \mathcal{A}_{i-1}$

Fig. 3: Construction of a set $\mathcal{A}$ with $H(\mathbf{s}_{\mathcal{A}}) < \mathcal{M}$ for an (r, δ, α) code.

Consider two cases depending on the way the algorithm in Fig. 3 terminates:

Case 1: Assume that the algorithm terminates with the final set assigned at step 5, i.e., after adding $\Gamma(j_\ell)$ to $\mathcal{A}_{\ell-1}$. Now we have from (r, δ, α) property of the code that

$$\begin{aligned} h_i &= H(\mathbf{s}_{\mathcal{A}_i}) - H(\mathbf{s}_{\mathcal{A}_{i-1}}) \\ &= H(\mathbf{s}_{\mathcal{A}_{i-1} \cup (\mathcal{A}_i \backslash \mathcal{A}_{i-1})}) - H(\mathbf{s}_{\mathcal{A}_{i-1}}) \\ &= H(\mathbf{s}_{\mathcal{A}_{i-1}}) + H(\mathbf{s}_{\mathcal{A}_i \backslash \mathcal{A}_{i-1}} | \mathbf{s}_{\mathcal{A}_{i-1}}) - H(\mathbf{s}_{\mathcal{A}_{i-1}}) \\ &= H(\mathbf{s}_{\mathcal{A}_i \backslash \mathcal{A}_{i-1}} | \mathbf{s}_{\mathcal{A}_{i-1}}) \\ &\leq (a_i - \delta + 1)\alpha. \end{aligned} \quad (24)$$

The last inequality follows from the fact that any block in $\Gamma(j_i)$ can be written as a function of any set of r-blocks in $\Gamma(j_i)$ and the fact that we pick i in step 3 only if $|\Gamma(j_i) \backslash \mathcal{A}_{i-1}| \geq \delta - 1$. Since at the end of i^{th} iteration, we have all the elements of $\Gamma(j_i)$ added to $\mathcal{A}_i$, out of which a_i blocks are added at the i^{th} iteration. These newly added packets can not contribute more than $(a_i - (\delta - 1))\alpha$ to the entropy of set $\mathcal{A}_i$ as $\delta - 1$ of these packets are deterministic function of other newly added blocks of $\Gamma(j_i)$ and blocks of $\Gamma(j_i)$ that were already present in $\mathcal{A}_{i-1}$. From (24), we have that

$$a_i \geq \frac{h_i}{\alpha} + \delta - 1. \quad (25)$$

Now using (22)

$$\begin{aligned} |\mathcal{A}| = |\mathcal{A}_\ell| &= \sum_{i=1}^{\ell} a_i \\ &\geq \sum_{i=1}^{\ell} \left(\frac{h_i}{\alpha} + \delta - 1 \right) \\ &= \frac{1}{\alpha} \sum_{i=1}^{\ell} h_i + (\delta - 1)\ell. \end{aligned} \quad (26)$$

Similar to the proof of Papailiopoulos et al. [13], we have

$$\sum_{i}^{\ell} h_i = \left(\left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil \alpha - \alpha \right), \tag{27}$$

and

$$\ell = \left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1. \tag{28}$$

It follows from (26), (27), and (28) that

$$|\mathcal{A}_\ell| \geq \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - 1 + \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \tag{29}$$

Case 2: The proof of this case is exactly similar to that in [12] except a few minor modification. Consider that the algorithm terminates with the final set assigned at step 7 in ℓ^{th} iteration. Since it reaches the step 7, we have

$$H(\mathbf{s}_{\mathcal{A}_{\ell-1} \cup \Gamma(j_\ell)}) \geq \mathcal{M}. \tag{30}$$

As the increment in the entropy is at most $r\alpha$ at each iteration, we have

$$\ell \geq \left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil \tag{31}$$

For $i \leq \ell - 1$, from (25)

$$a_i \geq \frac{h_i}{\alpha} + \delta - 1. \tag{32}$$

For $i = \ell$,

$$a_\ell \geq \frac{h_\ell}{\alpha}. \tag{33}$$

Next, it follows from (22), (31), (32), and (33) that

$$\begin{aligned}
|\mathcal{A}_\ell| &= \sum_{i=1}^{\ell} a_i \\
&\geq \sum_{i=1}^{\ell-1} \left(\frac{h_i}{\alpha} + \delta - 1. \right) + \frac{h_\ell}{\alpha} \\
&= \frac{1}{\alpha} \sum_{i=1}^{\ell} h_i + (\ell - 1)(\delta - 1) \\
&\geq \frac{1}{\alpha} \left(\left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil \alpha - \alpha \right) + \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1) \quad (34) \\
&= \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - 1 + \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1) \quad (35)
\end{aligned}$$

where (34) follows from (31) and (27). Now combining (6), (29), and (35), we get

$$d_{\min}(\mathcal{C}) \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \tag{36}$$

■

Using $\alpha = (1+\epsilon)\frac{\mathcal{M}}{k}$ for the bound given in the above theorem, we obtain $d_{\min} \leq n - \left\lceil \frac{k}{1+\epsilon} \right\rceil + 1 - \left(\left\lceil \frac{k}{r(1+\epsilon)} \right\rceil - 1 \right)(\delta - 1)$. For the special case of $\delta = 1$, this bound matches with the bound in [9]. For the case of $\alpha = \mathcal{M}/k$, i.e., the minimum storage point for locally repairable codes, the bound reduces to $d_{\min} \leq n - k + 1 + (\lceil k/r \rceil - 1)(\delta - 1)$, which is coincident with the bound presented in [14].

B. Construction of $d_{\min}$-optimal locally repairable codes

In this subsection we present a construction of an (r, δ, α) locally repairable code which attains the bound given in Theorem 11. Consider a file $\mathbf{f}$, to be stored on DSS, of size $\mathcal{M} \geq r\alpha$. We encode the file in two steps before storing it on DSS. First, the file is encoded using an MRD code. The codeword (over $\mathbb{F}_{q^N}$) of the MRD code is then divided into local groups and each local group is then encoded using an MDS array code over $\mathbb{F}_q$. This construction can be viewed as a generalization of the construction proposed in [30]. In particular, let $\mathcal{C}^{\text{MRD}}$ be an $[N \times m, N\mathcal{M}, \varsigma = m - \mathcal{M} + 1]$ Gabidulin MRD code, $N \geq m$, where each codeword is considered as a vector of length m over $\mathbb{F}_{q^N}$. We take $m = gr\alpha$, where g denotes the number of local groups in the system, which is a system parameter. A codeword $\mathbf{c} \in \mathcal{C}^{\text{MRD}}$ is partitioned into g groups, each of size $r\alpha$, and each group is stored on a different set of r nodes, α symbols per node. In other words, the output of the first encoding step generate the encoded data stored on rg nodes, each one containing α symbols of a (folded) MRD codeword. In the second stage of encoding process, we generate $\delta - 1$ parity nodes per group by applying an $(r + \delta - 1, r)$ MDS array code over $\mathbb{F}_q$ on each local group of r nodes, treating these r nodes as input data blocks for the MDS array code. At the end of second round of encoding, we have $n = (r + \delta - 1)g = \frac{m}{\alpha} + \frac{m}{r\alpha}(\delta - 1)$ nodes, each storing α symbols over $\mathbb{F}_{q^N}$, partitioned into g local groups, each of size $r - \delta + 1$. We denote the concatenated code by C^{loc}. Next, we prove that the proposed locally repairable code C^{loc} indeed has the maximum minimum distance as given in (18).

Theorem 12. *The proposed (r, δ, α) locally repairable code C^{loc} attains the bound (18), i.e., its minimum distance $d_{\min}(C^{\text{loc}})$ satisfies*

$$d_{\min}(C^{\text{loc}}) = n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1). \tag{37}$$

Proof: Recall that a codeword of a Gabidulin MRD code can be considered as an evaluation of a linearized polynomial $f(y) \in \mathbb{F}_{q^N}[y]$ on m linearly independent points over $\mathbb{F}_q$, $\{y_1, \ldots, y_m\}$, where $y_i \in \mathbb{F}_{q^N}$, $1 \leq i \leq m$. The polynomial $f(\cdot)$ has original data symbols that need to be encoded as its coefficients. Note that for reconstruction of the original data it is sufficient to have evaluations of $f(\cdot)$ on $\mathcal{M}$ points in $\mathbb{F}_{q^N}$, $\{f(p_1), \ldots, f(p_{\mathcal{M}})\}$, such that $\{p_1, \ldots, p_{\mathcal{M}}\}$ are linearly independent points over $\mathbb{F}_q$. (See, e.g., [26], [35]–[37].)

Utilizing $\mathbb{F}_q$-linearity property of $f(y)$, MDS property of array code used in the second encoding stage, and the fact that $\mathcal{M} \geq r\alpha$, we have that any r nodes in any group contain evaluation of $f(y)$ at $r\alpha$ linearly independent over $\mathbb{F}_q$ points.

Let i and j be two integers such that $\mathcal{M} = m - r\alpha(i+1) + j$, $0 \leq i \leq \frac{m}{r\alpha} - 1$, and $0 \leq j \leq r\alpha - 1$. Now it follows from

(18) that

$$\begin{aligned} d_{\min}(C^{\text{loc}}) - 1 &\leq \frac{m}{\alpha} + \frac{m}{r\alpha}(\delta-1) \\ &\quad - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta-1) \\ &= \frac{m}{\alpha} - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil \\ &\quad + \left(\frac{m}{r\alpha} - \left(\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right) \right)(\delta-1) \end{aligned} \tag{38}$$

Next, we treat $j = 0$ and $j > 0$ cases separately and show that C^{loc} has optimal minimum distance in both these cases.

Case 1 **($j = 0$):** In this case $\left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil = \frac{m}{\alpha} - r(i+1)$ and $\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil = \frac{m}{r\alpha} - (i+1)$. From (38) we have,

$$\begin{aligned} d_{\min} - 1 &\leq r(i+1) + (i+1)(\delta-1) + (\delta-1) \\ &= (i+1)(r+\delta-1) + (\delta-1). \end{aligned} \tag{39}$$

Now, we show that any $(i+1)(r+\delta-1)+(\delta-1)$ node erasures can be tolerated by C^{loc}. In other words, even after $(i+1)(r+\delta-1)+(\delta-1)$ erasures, we have evaluations of $f(y)$ at $\mathcal{M}$ linearly independent points over $\mathbb{F}_q$. Here, we point out that the worst case erasure pattern is when the erasures appear in the smallest possible number of groups and the number of erasures inside a local group is maximal. Therefore, we consider the case when all the symbols in $i+1$ groups are erased, and there is a group with $\delta-1$ erased nodes. Due to application of MDS array code in each local group, less or equal to $\delta-1$ erasures in a particular group does not affect the number of evaluations of $f(y)$ on linearly independent points that particular group has to offer, i.e., $r\alpha$. So in this case, the number of the remaining symbols of an MRD codeword which correspond to linearly independent points is $m-(r\alpha(i+1)) = \mathcal{M}$.

Case 2 **($j > 0$):** In this case $\left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil = \frac{m}{\alpha} - r(i+1) + \left\lceil \frac{j}{\alpha} \right\rceil$ and $\left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil = \frac{m}{r\alpha} - (i+1) + \left\lceil \frac{j}{r\alpha} \right\rceil = \frac{m}{r\alpha} - i$. It follows from (38) that

$$\begin{aligned} d_{\min} - 1 &= r(i+1) - \left\lceil \frac{j}{\alpha} \right\rceil + (i+1)(\delta-1) \\ &= (i+1)(r+\delta-1) - \left\lceil \frac{j}{\alpha} \right\rceil. \end{aligned} \tag{40}$$

As in the previous case, we show that original data can be reconstructed even after the failure of any $(i+1)(r+\delta-1) - \left\lceil \frac{j}{\alpha} \right\rceil$ nodes. We again establish this by showing that we can find evaluations of $f(y)$ at $\mathcal{M}$ linearly independent points from the remaining nodes in the DSS. As previously, we consider the worst case erasure pattern, where the erasures appear in the smallest possible number of groups and the number of erasures inside a group is maximal. Assume that all the symbols in i local groups are erased, and there is a local group with $r + \delta - 1 - \left\lceil \frac{j}{\alpha} \right\rceil$ erased nodes. In this case the available number of evaluation of $f(x)$ at linearly independent points is

$$\begin{aligned} &m - r\alpha i + \left((r+\delta-1) - \left(r+\delta-1-\left\lceil \frac{j}{\alpha} \right\rceil \right) \right) \alpha \\ &= m - r\alpha i + \left\lceil \frac{j}{\alpha} \right\rceil \alpha \geq \mathcal{M} \end{aligned} \tag{41}$$

1	$a_1\ a_2\ a_3\ a_4$	6	$b_1\ b_2\ b_3\ b_4$	11	$c_1\ c_2\ c_3\ c_4$
2	$a_5\ a_6\ a_7\ a_8$	7	$b_5\ b_6\ b_7\ b_8$	12	$c_5\ c_6\ c_7\ c_8$
3	$a_9\ a_{10}\ a_{11}\ a_{12}$	8	$b_9\ b_{10}\ b_{11}\ b_{12}$	13	$c_9\ c_{10}\ c_{11}\ c_{12}$
4	$p_1^a\ p_2^a\ p_3^a\ p_4^a$	9	$p_1^b\ p_2^b\ p_3^b\ p_4^b$	14	$p_1^c\ p_2^c\ p_3^c\ p_4^c$
5	$p_5^a\ p_6^a\ p_7^a\ p_8^a$	10	$p_5^b\ p_6^b\ p_7^b\ p_8^b$	15	$p_5^c\ p_6^c\ p_7^c\ p_8^c$
	local group 1		local group 2		local group 3

Fig. 4: Example of an $(r=3, \delta=3, \alpha=4)$ locally repairable code with $n = 15$ and $\mathcal{M} = 26$. The code has minimum distance 5.

Therefore, the original data can be recovered even when $(i+1)(r+\delta-1) - \left\lceil \frac{j}{\alpha} \right\rceil$ nodes fail.

This establishes the optimality of C^{loc} in terms of minimum distance. ■

Next, we illustrate the construction of C^{loc} with help of an example.

Example 13. *Let us consider a DSS with $\mathcal{M} = 26$, $\delta = r = g = 3$, $\alpha = 4$, $m = rg\alpha = 36$. Then $n = 15$ and from (18), $d_{min} \leq 5$. Let $(a_1, \ldots, a_{12}, b_1, \ldots, b_{12}, c_1, \ldots, c_{12})$ be a codeword of an $[N \times 36, N \cdot 26, 11]$ MRD code, which is obtained by encoding $\mathcal{M} = 26$ symbols over $\mathbb{F}_{q^N}$ of the original file. Here we assume that $N \geq 36$. The MRD codeword is then divided into three groups $(a_1, \ldots, a_{12})$, $(b_1, \ldots, b_{12})$, and $(c_1, \ldots, c_{12})$. Encoded symbols in each group are stored on three storage nodes as shown in Fig. 4. In the second stage of encoding, an MDS array code is applied on each local group to obtain $\delta - 1 = 2$ parity nodes per local group. The coding scheme is illustrated in Fig. 4.*

Note that, any three nodes in a local group provide evaluations of the linearized polynomial $f(y)$ associated with data symbols at 12 linearly independent points over $\mathbb{F}_q$; and, the polynomial $f(y)$ can be recovered from its evaluations at 26 linearly independent points over $\mathbb{F}_q$. Here, we illustrate that any four node erasures can be tolerated by the coding scheme employed in this example. If there are at most two erasures in a group, then we can obtain evaluation of $f(y)$ at 12 linearly independent points from each local group, thus $36 > 26 = \mathcal{M}$ points from all three local groups. If there is a group with three node erasures, then this group can provide evaluations of $f(y)$ at only 8 linearly independent points. However, the other two groups can give evaluation of $f(y)$ at 24 additional linearly independent points, which makes the total numeber of desirable evaluation to be $32 > 26$. Finally we consider the worst case mentioned in the proof of Theorem 12. Suppose there is a group with four erased nodes, then this local group provides evaluation of $f(y)$ at 4 linearly independent points, which taking into account the contribution from other two local groups (additional 24 points), gives the evaluation of $f(y)$ at $28 > 26 = \mathcal{M}$ linearly independent points. Therefore, the original file can be reconstructed even after four nodes fail.

C. File size upper bound for repair bandwidth efficient locally repairable codes

In this subsection, we introduce the notion of repair bandwidth for locally repairable codes. In a naïve repair process for a locally repairable code, a newcomer contacts r nodes in its local group and download all the data stored on these nodes. The newcomer then regenerates the data stored on the failed node and stores it for future operations. Following the line of work of bandwidth efficient repair in DSS due to [9], we allow a newcomer to contact more than r nodes in its local group in order to repair the failed node. The motivation behind this is to lower the repair bandwidth of a locally repairable code. (This also improves the secrecy capacity of such codes as detailed in Section V.)

In the rest of this section, we restrict ourselves to locally repairable codes that have the maximum possible minimum distance as described in (18). Since the upper bound on minimum distance for locally repairable codes in (18) is achievable by only those codes that have disjoint local group, we focus only on such codes. Here, we assume that $(r+\delta-1)|n$. Let $\mathcal{G}_1,\ldots,\mathcal{G}_g$ denote $g=\frac{n}{r+\delta-1}$ disjoint sets of indices of storage nodes, each of size $(r+\delta-1)$. Each set represents a local group, and a failed node in a particular local group is repaired by contacting d remaining nodes within that group, where $r\leq d\leq r+\delta-2$. During the node repair process a newcomer downloads β symbols from each of these d nodes.

Next, we perform the standard min-cut max-flow based analysis for locally repairable DSS by mapping it to a multicasting problem on a dynamic information flow graph. (The information flow graph representing a locally repairable DSS is a modification of the information flow graph for classical DSS analyzed in [9] and is first introduced in [13] for naïve repair, where the newcomer contacts r nodes.) We assume a sequence of node failures and node repairs as shown in Fig. 5. We consider that each local group encounter the same sequence of node failures and the node repairs that are performed as result of these failures. Each data collector contacts $n-d_{\min}+1$ storage nodes for data reconstruction. A data collector is associated with the nodes it contacts for data reconstruction, $(\mathcal{K}_1,\mathcal{K}_2,\ldots,\mathcal{K}_g)$. Here $\mathcal{K}_i\subseteq\mathcal{G}_i$ is the set of indices of nodes that a data collector contacts in i^{th} local group and $\sum_{i=1}^{g}|\mathcal{K}_i|=n-d_{\min}+1$. Next we derive an upper bound on the amount of data that can be stored on the DSS while ensuring $n-d_{\min}+1$ property, i.e., each set of $n-d_{\min}+1$ nodes allows a data collector to recover the original file. This upper bound is used to derive a repair bandwidth vs. per node storage trade off for minimum distance optimal codes with (r,δ,α) locality. In what follows, we add two more parameters in the representation of locally repairable codes and denote them by the tuple $(r,\delta,\alpha,d,\beta)$.

Theorem 14. *For an (n,k) DSS employing an $(r,\delta,\alpha,d,\beta)$ locally repairable code, we have*

$$\mathcal{M}\leq\min\left\{r\alpha,\sum_{i=0}^{h-1}\min\{\max\{(d-i)\beta,0\},\alpha\}\right\} \tag{42}$$
$$+\sum_{j=1}^{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor}\min\left\{r\alpha,\sum_{i=0}^{r+\delta-2}\min\{\max\{(d-i)\beta,0\},\alpha\}\right\}$$

where $h=n-d_{\min}+1-(r+\delta-1)\left\lfloor\frac{n-d_{\min}+1}{r+\delta-1}\right\rfloor$

Proof: Consider a data collector with $\mathcal{K}_1=\mathcal{G}_1,\mathcal{K}_2=\mathcal{G}_2,\ldots,\mathcal{K}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor}=\mathcal{G}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor},\mathcal{K}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor+2}=\ldots=\mathcal{K}_g=\emptyset$, and $\mathcal{K}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor+1}\subset\mathcal{G}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor+1}$ s.t. $\left|\mathcal{K}_{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor+1}\right|=h$. Now, the bound in (42) follows by finding various cuts in information flow graph (Fig. 5). For each group, we consider cuts similar to the ones given in [9]. Here, the data collector connects to h nodes for the first term in (42) and $r+\delta-1$ nodes for each of the terms in the summation of the second term in (42). Now, consider the i-th node out of $\tilde{k}$ nodes that data collector connects in a particular group. (Here, $\tilde{k}=h$, or $\tilde{k}=r+\delta-1$ as described above.) A cut between x_i^{in} and x_i^{out} for each node gives a cut-value of α. On the other hand, for $i=0,\cdots,\tilde{k}-1$, if the cut is such that x_i^{in} belongs to the data collector side, we consider that $(d-i)$ live nodes are connected together with i nodes that have been previously repaired. In our setup, for such a cut, the cut-value evaluates to $\max\{(d-i)\beta,0\}$, as for $i>d$ the repair node is considered to contact only the previously repaired nodes, and hence does not contribute to the maximum flow. ■

Note that the codes that are under consideration have property that each local group has entropy of $r\alpha$ and any set of r nodes has $r\alpha$ independent symbols. (See definition of $(r,\delta)-$ locality in Section II.) Therefore, node repairs within each local group have to ensure this property. This implies that each local group and its repair can be related to an $(r+\delta-1,r,d,\alpha,\beta)$ MSR regenerating code with a file of size $r\alpha$. Hence, when a collector connects to any r nodes in a group, it can get all the information that particular group has to offer. Therefore, similar to the analysis given in [9] for the classical setup, the parameters need to satisfy

$$r\alpha=\sum_{i=0}^{r-1}\min\{(d-i)\beta,\alpha\}, \tag{43}$$

which leads to the requirement of $(d-i)\beta\geq\alpha$ for each $i=0,\cdots,r-1$. Then, minimum β is obtained as $\beta^*=\frac{\alpha}{d-r+1}$. When node repairs are performed by downloading β^* symbols from d nodes for each failed node, the bound in (42) reduces to

$$\mathcal{M}\leq\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor r\alpha+\min\{h,r\}\alpha \tag{44}$$

where h is as defined in Theorem 14. This establishes the file size bound for bandwidth efficient $d_{\min}$-optimal locally repairable codes.

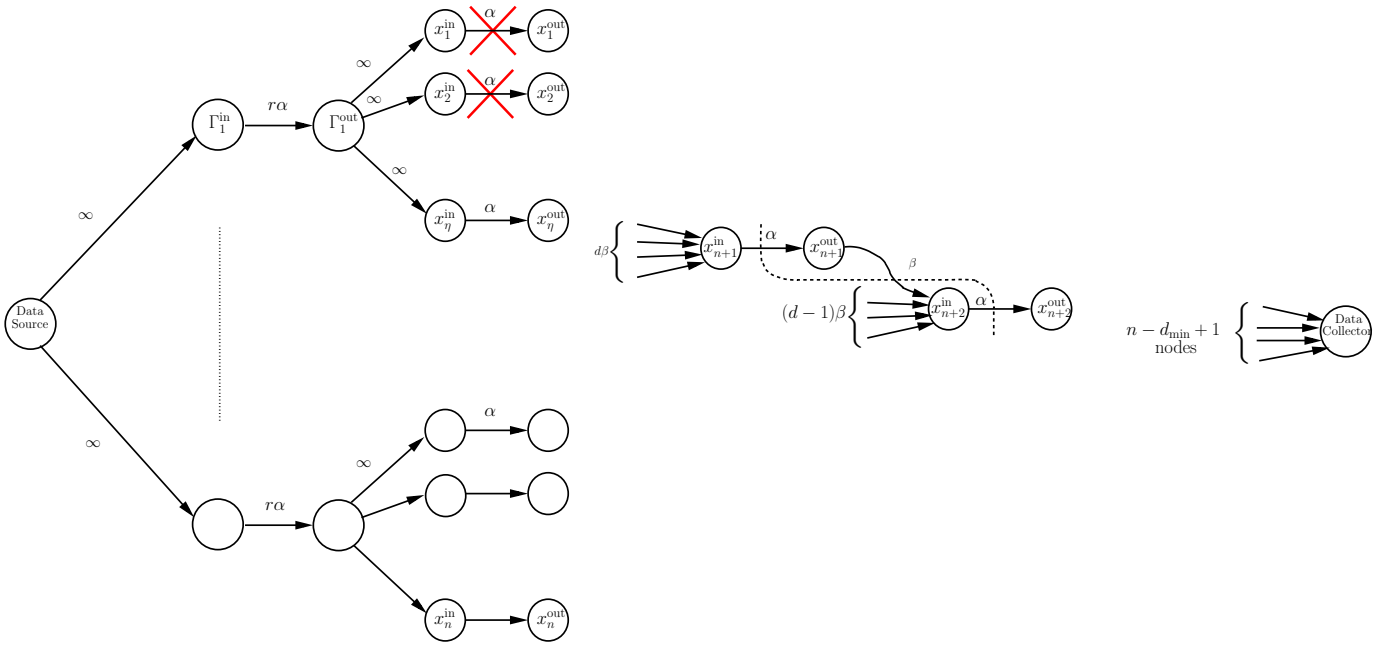

Fig. 5: Flow graph for (r, δ) locally repairable code. In this graph, node pairs $\{\Gamma_i^{\text{in}}, \Gamma_i^{\text{out}}\}_{i=1}^{g}$ with edge of capacity $r\alpha$ enforce the requirement that each local group has $r\alpha$ entropy. Here $\eta = r + \delta - 1$.

D. Construction of repair bandwidth efficient $d_{\min}$-optimal locally repairable codes

Now it is clear that node repair within a local group is performed by treating each local group as an $(r+\delta-1, r, d, \alpha, \beta^*)$ MSR regenerating code. Using a random linear network coding (RLNC) over large enough field, the bound in (44) is achievable [9], [40]. Since we don't get any reduction in repair bandwidth (β) by setting α greater than $\frac{\mathcal{M}}{k}$, we focus on the case when $\alpha = \frac{\mathcal{M}}{k}$ for the construction presented here. Remarkably, the code presented in Section IV-B, when an MSR code is employed for the second encoding stage, achieves the bound (44), when we have $\alpha|\mathcal{M}$. We establish this claim in the following theorem.

Theorem 15. *Let C^{loc} be a code obtained from the construction described in Sec. IV-B with $\alpha = \frac{\mathcal{M}}{k}$ and an MSR regenerating code employed in the second encoding stage to generate local parities. If $\alpha|\mathcal{M}$, then C^{loc} attains the bound (44), i.e., the size of a file that can be stored by using this code satisfies*

$$\mathcal{M} = \left\lfloor \frac{n - d_{min} + 1}{r + \delta - 1} \right\rfloor r\alpha + \min\{h, r\}\alpha$$

where $h = n - d_{\min} + 1 - (r + \delta - 1)\left\lfloor \frac{n-d_{\min}+1}{r+\delta-1} \right\rfloor$.

Proof: Similar to the proof of Theorem 12, we consider two cases depending on the difference between the length of MRD codeword (output of first stage of encoding) m and the file size $\mathcal{M}$. We first consider the case when $r\alpha|(m - \mathcal{M})$. (This corresponds to Case 1 in the proof of Theorem 12). Recall that in this case, we have $\mathcal{M} = m - r\alpha(i+1) = r\alpha(g - i - 1)$, $n = (r+\delta-1)g$, $d_{min} - 1 = (i+1)(r+\delta-1) + (\delta-1)$, and

$$\begin{aligned} h &= n - d_{min} + 1 - (r + \delta - 1)\left\lfloor \frac{n - d_{min} + 1}{r + \delta - 1} \right\rfloor \\ &= (r + \delta - 1)g - (i + 1)(r + \delta - 1) - (\delta - 1) \\ &\quad - (r + \delta - 1)\left(g - (i + 1) - \left\lceil \frac{\delta - 1}{r + \delta - 1} \right\rceil\right) \\ &= r. \end{aligned} \tag{45}$$

For $h = r$, the right hand side of (44) becomes $\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + r\alpha = \alpha r(g - (i+1) - 1 + 1) = \alpha r(g - (i+1)) = \mathcal{M}$, the size of file that is encoded using C^{loc}.

Now we consider the second case considered in the proof of Theorem 12 with $j = \alpha b$, where $\mathcal{M} = m - r\alpha(i+1) + b\alpha = (g - i - 1)r\alpha + b\alpha$, for some integer $0 < b \leq r - 1$. Here, we have used the fact that $m = gr\alpha$. In this case, $d_{min} - 1 = (i+1)(r+\delta-1) - b$, and $h = (r+\delta-1)g - (i+1)(r+\delta-1) + b - (r+\delta-1)(g-i-1) = b \leq r$, since $\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor = g - i - 1 + \left\lfloor \frac{b}{r+\delta-1} \right\rfloor = g - i - 1$. Therefore the upper bound on the file size in (44) becomes $\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + h\alpha = (g - i - 1)r\alpha + b\alpha = gr\alpha - (i+1)r\alpha + b\alpha = \mathcal{M}$. This establishes that C^{loc}, when MSR code used to generate its local parities, attains the bound given in (44). ■

In the following example, we illustrate the aforementioned construction for repair bandwidth efficient locally repairable codes for a particular choice for system parameters.

Example 16. *Consider the following system parameters.*

$$(\mathcal{M}, n, \alpha, r, \delta, m, N) = (24, 15, 4, 3, 3, 36, 36). \tag{46}$$

First $\mathcal{M} = 24$ symbols over $\mathbb{F}_{q^{36}}$ are encoded to a codeword represented by $(a_1, \ldots, a_{12}, b_1, \ldots, b_{12}, c_1, \ldots, c_{12})$ using the $[36 \times 36, 36 \cdot 24, 13]$ MRD code. Here 36 encoded symbols

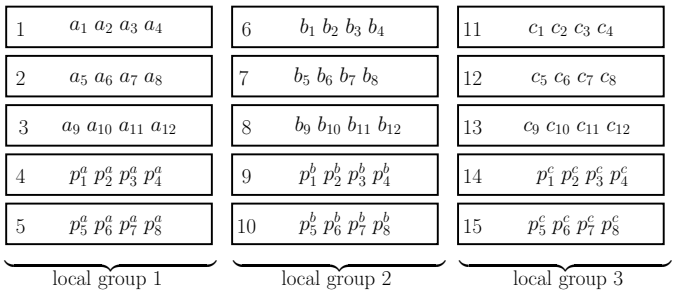

Fig. 6: Example of repair bandwidth efficient $(r = 3, \delta = 3)-$locally repairable code with $\mathcal{M} = 24$ and $n = 15$. The code has $d_{\min} = 8$.

over $\mathbb{F}_{q^{36}}$ are evaluation of a linearized polynomial on 36 linearly independent over $\mathbb{F}_q$ points. The encoded symbols are partitioned into 3 groups each of size 12 and stored on 9 nodes as shown in Fig. 6. We further add 6 nodes, 2 nodes for each local group, using a $(5, 3)$ exact repairable MSR code with $\alpha = 4$ (e.g., (5,3)-zigzag code).

From (18), the minimum distance of this code is at most 8. In fact, it is exactly 8 as we have evaluation of data polynomial over 24 lineally independent over $\mathbb{F}_q$ points even when any 7 nodes fail. Moreover, each failed node can be repaired bandwidth efficiently as an exact repairable MSR code is used within each local group.

V. Secrecy in Locally Repairable DSS

In this section, we analyze locally repairable DSS in the presence of secrecy constraints. The eavesdropping model is as defined in Section II. We first derive a generic upper bound on the secrecy capacity of an $(r, \delta, \alpha, d, \beta)$ locally repairable code, which we later specialize for specific cases of system parameters. While addressing specific cases, we also present secure coding construction that achieve the respective upper bound for certain parameters.

Consider a data collector, which contacts $n - d_{\min} + 1$ nodes. Let $\mathcal{K}_i$ denote the indices of nodes that are contacted by the data collector in i-th local group and $\mathcal{K} = \cup_{i=1}^{g} \mathcal{K}_i$ with $|\mathcal{K}| = n - d_{\min} + 1$. Similar to Section III-A, we classify eavesdropped nodes into two classes: $\mathcal{E}_1$ contains storage-eavesdropped nodes (ℓ_1 nodes in total) and $\mathcal{E}_2$ contains download-eavesdropped nodes (ℓ_2 nodes in total). Considering the local group i, we denote the set of indices of storage-eavesdropped nodes as $\mathcal{E}_1^i$ and download-eavesdropped nodes as $\mathcal{E}_2^i$. Here, we have $\mathcal{E}_1 = \cup_{i=1}^{g} \mathcal{E}_1^i$, $\mathcal{E}_2 = \cup_{i=1}^{g} \mathcal{E}_2^i$, and $\sum_{i=1}^{g} l_1^i = \ell_1$, $\sum_{i=1}^{g} l_2^i = \ell_2$, where $l_1^i = |\mathcal{E}_1^i|$ and $l_2^i = |\mathcal{E}_2^i|$. We denote $\mathcal{X}$ to represent set of tuples $\left(\{\mathcal{E}_1^i\}_{i=1}^g, \{\mathcal{E}_2^i\}_{i=1}^g, \{\mathcal{K}_i\}_{i=1}^g\right)$ satisfying these requirements. In the following, we provide our generic upper bound on the secrecy capacity of $(r, \delta, \alpha, d, \beta)$ locally repairable codes against an (ℓ_1, ℓ_2) eavesdropper.

Theorem 17. *For an (n, k) DSS employing an $(r, \delta, \alpha, d, \beta)$ locally repairable code that is secure against an $(\ell_1, \ell_2)-$eavesdropper, we have*

$$\mathcal{M}^s \leq \min_{\left(\{\mathcal{E}_1^i\}_{i=1}^g, \{\mathcal{E}_2^i\}_{i=1}^g, \{\mathcal{K}_i\}_{i=1}^g\right) \in \mathcal{X}} \sum_{i=1}^{g} H(\mathbf{s}_{\mathcal{K}_i} | \mathbf{s}_{\mathcal{E}_1^i}, \mathbf{d}_{\mathcal{E}_2^i}). \tag{47}$$

Proof: Without loss of generality we can focus on sets of indices $\{\mathcal{E}_1^i\}_{i=1}^g$ and $\{\mathcal{E}_2^i\}_{i=1}^g$ such that $|\mathcal{E}_1^i \cup \mathcal{E}_2^i| \leq r$ for the purpose of getting upper bound on secrecy capacity as eavesdropping r nodes in a group gives eavesdropper all the information that particular group has to offer. As introduced in Section II, we represent stored and downloaded content at node i (set $\mathcal{A}$) as $\mathbf{s}_i$ and $\mathbf{d}_i$ (repectively, $\mathbf{s}_\mathcal{A}$ and $\mathbf{d}_\mathcal{A}$). We assume that $(\mathcal{K}_1, \ldots, \mathcal{K}_g)$ s.t. $\mathcal{E}_1^i \cup \mathcal{E}_2^i \subseteq \mathcal{K}_i$ or $\mathcal{K}_i = \emptyset$. Note that we still need that $|\mathcal{E}_1| + |\mathcal{E}_2| = \ell_1 + \ell_2 < k$ in order to have a non-zero secure file size.

$$\begin{aligned}
H(\mathbf{f}^s) &= H(\mathbf{f}^s | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) && (48)\\
&= H(\mathbf{f}^s | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}) - H(\mathbf{f}^s | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2}, \mathbf{s}_\mathcal{K}) && (49)\\
&= I(\mathbf{f}^s; \mathbf{s}_\mathcal{K} | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2})\\
&\leq H(\mathbf{s}_\mathcal{K} | \mathbf{s}_{\mathcal{E}_1}, \mathbf{d}_{\mathcal{E}_2})\\
&= H(\mathbf{s}_{\mathcal{K}_1}, \ldots, \mathbf{s}_{\mathcal{K}_g} | \mathbf{s}_{\mathcal{E}_1^1}, \ldots, \mathbf{s}_{\mathcal{E}_1^g}, \mathbf{d}_{\mathcal{E}_2^1}, \ldots, \mathbf{d}_{\mathcal{E}_2^g})\\
&\leq \sum_{i=1}^{g} H(\mathbf{s}_{\mathcal{K}_i} | \mathbf{s}_{\mathcal{E}_1^i}, \mathbf{d}_{\mathcal{E}_2^i}), && (50)
\end{aligned}$$

where (48) follows from the secrecy constraint, and (49) follows by the data collector's ability to obtain the whole data. Since we get one such upper bound for each choice of $(\{\mathcal{E}_1^i\}_{i=1}^g, \{\mathcal{E}_2^i\}_{i=1}^g, \{\mathcal{K}_i\}_{i=1}^g)$, we have

$$H(\mathbf{f}^s) \leq \min_{\left(\{\mathcal{E}_1^i\}_{i=1}^g, \{\mathcal{E}_2^i\}_{i=1}^g, \{\mathcal{K}_i\}_{i=1}^g\right) \in \mathcal{X}} \sum_{i=1}^{g} H(\mathbf{s}_{\mathcal{K}_i} | \mathbf{s}_{\mathcal{E}_1^i}, \mathbf{d}_{\mathcal{E}_2^i}),$$

where $\mathcal{X}$ consists of all choices for $\left(\{\mathcal{E}_1^i\}_{i=1}^g, \{\mathcal{E}_2^i\}_{i=1}^g, \{\mathcal{K}_i\}_{i=1}^g\right)$ which satisfy the requirements mentioned above. ∎

Now we consider two cases depending on the number of local parities per local group. The analysis of the first case, for single parity node per local group, shows that the performance of such coding schemes degrade substantially in the presence of an eavesdropper that can observe the data downloaded during node repairs. The second case, multiple parity nodes per local group, allows the node repair to be performed with smaller repair bandwidth which results in lower leakage to such eavesdroppers observing downloaded data. In both cases, we use the vectors $\mathbf{l}_1 = (l_1^1, \ldots, l_1^g)$ and $\mathbf{l}_2 = (l_2^1, \ldots, l_2^g)$ to represent a pattern of eavesdropped nodes.

A. Case 1: $\delta = 2$

Consider locally repairable codes presented in [13], which correspond to $\delta = 2$. For such codes, during node repair a newcomer node downloads all the data stored on other nodes in the local group it belongs to. Since the data on each node in a local group is a function of data stored on any set of r nodes in a local group, all the information in that group is revealed to an eavesdropper that observe the data downloaded during a single node repair. In other words, we have $H(\mathbf{s}_{\mathcal{G}_i} | \mathbf{d}_{\mathcal{E}_2^i}) = 0 \Rightarrow$

$H(\mathbf{s}_{\mathcal{K}_i}|\mathbf{d}_{\mathcal{E}_2^i}) = 0$, when $\mathcal{E}_2^i \neq \emptyset$. Accordingly, consider the eavesdropping pattern $\mathbf{l}_2 = (1, 1, \ldots, 1, 0, \ldots, 0)$ with ones at first ℓ_2 positions and $\mathbf{l}_1 = (0, \ldots, 0, l_1^{\ell_2+1}, \ldots, l_1^g)$ with zeros in first ℓ_2 positions. Moreover, consider a data collector which accesses set of nodes as used in the proof of Theorem 14. These eavesdropping pattern and node access pattern by data collector along with (50) give us the following upper bound on the amount of information that can be stored securely on the DSS that employ an $(n, k, r, \delta = 2)$ locally repairable code:

$$H(\mathbf{f}^s) \leq \left[\left\lfloor \frac{n - d_{\min} + 1}{r+1} \right\rfloor r + h - (\ell_2 r + \ell_1) \right]^+ \alpha, \quad (51)$$

where $h = n - d_{\min} + 1 - (r+1)\lfloor \frac{n-d_{\min}+1}{r+1} \rfloor \leq r$.

In order to see that the above bound is tight, we present a coding scheme which allows file of size $\left(\left\lfloor \frac{n-d_{\min}+1}{r+1} \right\rfloor r + h - (\ell_2 r + \ell_1) \right) \alpha$ symbols to be securely stored against an (ℓ_1, ℓ_2)-eavesdropper. Take a secure file of size $\left(\left\lfloor \frac{n-d_{\min}+1}{r+1} \right\rfloor r + h - (\ell_2 r + \ell_1) \right) \alpha$ and $(\ell_2 r + \ell_1)\alpha$ random symbols $\mathbf{r} = (r_1, \ldots, r_{(\ell_2 r + \ell_1)\alpha})$. We construct a linearized polynomial $f(y)$ with the $\left(\left\lfloor \frac{n-d_{\min}+1}{r+1} \right\rfloor r + h \right) \alpha$ symbols (including both the secure file and random symbols) as its coefficients, and evaluate the polynomial at $\mathcal{M} = \left(\left\lfloor \frac{n-d_{\min}+1}{r+1} \right\rfloor r + h \right) \alpha$ linearly independent points over $\mathbb{F}_q$. These $\mathcal{M}$ symbols (evaluations of $f(y)$) are subsequently encoded with a minimum distance optimal $(r, \delta = 2, \alpha, d = r, \beta = \alpha)$ locally repairable code for an (n, k) DSS, e.g., coding scheme proposed in [13]. It follows from Lemma 3 that the file is secured against an (ℓ_1, ℓ_2)-eavesdropper if (i) $H(\mathbf{e}) \leq H(\mathbf{r})$ (which is trivially true as the eavesdropper observes at most $(\ell_2 r + \ell_1)\alpha$ linearly independent symbols) and (ii) $H(\mathbf{r}|\mathbf{u}, \mathbf{e}) = 0$. It remains to show the latter requirement also holds. We first note that as the outer code is essentially an MRD code, it can be viewed as an MDS code. Thus, given $\mathbf{u}$, original data symbols, eavesdropper can remove the contribution of monomials associated with secure data symbols from the evaluation of $f(y)$, and it can then recover the random symbols from the remaining polynomial at hand. (Note that, given $\mathbf{u}$, the eavesdropper has $(\ell_2 r + \ell_1)\alpha$ linearly independent evaluations of the reduced polynomial to solve for $(\ell_2 r + \ell_1)\alpha$ random symbols.) Thus, we obtain that $H(\mathbf{r}|\mathbf{u}, \mathbf{e}) = 0$, which establishes the secrecy claim of the proposed scheme.

Corollary 18. *For an (n, k) DSS employing an $(r, \delta = 2, \alpha, d, \beta)$ locally repairable code, the secrecy capacity against an (ℓ_1, ℓ_2) eavesdropper is given by*

$$\mathcal{M}^s = \left[\left\lfloor \frac{n - d_{\min} + 1}{r+1} \right\rfloor r + h - (\ell_2 r + \ell_1) \right]^+ \alpha. \quad (52)$$

B. Case 2: $\delta > 2$ and $\alpha = \frac{\mathcal{M}}{k}$

In this case, we assume that each node repair within a local group is performed in a bandwidth efficient manner. Therefore, in each group we can apply the result of Theorem 5 to get

$$H\left(\mathbf{s}_{\mathcal{K}_i}|\mathbf{s}_{\mathcal{E}_1^i}, \mathbf{d}_{\mathcal{E}_2^i}\right) \leq \sum_{j=1}^{\min(|\mathcal{K}_i|, r) - (l_1^i + l_2^i)} \left(\alpha - \theta\left(\alpha, \beta^*, l_2^i\right)\right) \quad (53)$$

where $\theta(\alpha, \beta^*, l_2^i)$ is the amount of information that an eavesdropper receives from one intact node (a node not eavesdropped) during the repair of $|\mathcal{E}_2^i|$ nodes in the i^{th} local group. Next, we consider data collector associated with the pattern $(\mathcal{K}_1, \ldots, \mathcal{K}_g)$ used in the proof of Theorem 14, and the following eavesdropping pattern associated with $\mathbf{l}_2$

$$\begin{aligned} l_2^1 = \ldots = l_2^\rho &= s + 1, \\ l_2^{\rho+1} = \ldots = l_2^{\lfloor \frac{n-d_{min}+1}{r+\delta-1} \rfloor} &= s, \\ l_2^{\lfloor \frac{n-d_{min}+1}{r+\delta-1} \rfloor + 2} = \ldots = l_2^g &= 0, \\ \text{and } l_2^{\lfloor \frac{n-d_{min}+1}{r+\delta-1} \rfloor + 1} &= \nu. \end{aligned} \quad (54)$$

Here we assume that $\ell_2 = s\left(\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor \right) + \rho + \nu$, for some (s, ρ, ν) satisfying $0 \leq \rho + \nu < s$ and $\nu \leq h$).

Combining (53) and (54) we get

$$\begin{aligned} H(\mathbf{f}^s) \leq & \sum_{i=1}^{\rho} (r - (l_1^i + s + 1)) (\alpha - \theta(\alpha, \beta^*, s+1)) \\ & + \sum_{i=\rho+1}^{\lfloor \frac{n-d_{min}+1}{r+\delta-1} \rfloor} (r - (l_1^i + s)) (\alpha - \theta(\alpha, \beta^*, s)) \\ & + (\min\{r, h\} - (l_1^i + \nu)) (\alpha - \theta(\alpha, \beta^*, \nu)). \end{aligned} \quad (55)$$

If we further assume that the encoding process within each local group is a linear array code (MDS by the definition of (r, δ) locality) and $d = r + \delta - 2$ within each local group for node repair (i.e., all the live local nodes are contacted for repair), then similar to Corollary 8, it follows from Lemma 7 that for $l_2^i \leq 2$,

$$\theta(\alpha, \beta^*, l_2^i) \geq \begin{cases} \beta^*, & \text{if } l_2^i = 1 \\ 2\beta^* - \frac{\alpha}{(\delta-1)^2}, & \text{if } l_2^i = 2 \end{cases} \quad (56)$$

Now (55) and (56) can be combined to obtain a bound on $H(\mathbf{f}^s)$.

Next, we present a code construction for securely storing data against an eavesdropper when $\ell_2 \leq 2\left(\left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor \right) + 2$ and $l_2^i \leq 2$. We take a file with its size $\mathcal{M}^s$, equal to the right hand side expression in (55), and a $\mathcal{M} - \mathcal{M}^s = \left\lfloor \frac{n-d_{min}+1}{r+\delta-1} \right\rfloor r\alpha + \min\{h, r\}\alpha - \mathcal{M}^s$ i.i.d. uniform random symbols. Note that $\mathcal{M}$ is equal to the upper bound in (44). Now we encode these $\mathcal{M}$ symbols, secure data symbols and random symbols, using the two step encoding scheme presented in Section IV-D. In particular, we employ $(r + \delta - 1, r)$ zigzag code within each local group in the second stage of encoding process. The secrecy and optimality claim of the proposed scheme under given assumption on ℓ_2 follows from linearized property of the MRD codes (used in the first stage of encoding) and the analysis given in Section III-B. We present this in the following.

Corollary 19. *For an (n, k) DSS employing an $(r, \delta > 2, \alpha, d, \beta^*)$ locally repairable bandwidth efficient code, the secrecy capacity against an (ℓ_1, ℓ_2) eavesdropper with $\ell_2 \leq$*

$2\left\lceil\frac{n-d_{min}+1}{r+\delta-1}\right\rceil$ *and* $\ell_1+\ell_2<k$ *is given by*

$$\begin{aligned}\mathcal{M}^s=&\sum_{i=1}^{\rho}(r-(l_1^i+l_2^i))\left(\alpha-\theta(\alpha,\beta^*,l_2^i)\right)\\&+\sum_{i=\rho+1}^{\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor}(r-(l_1^i+l_2^i))\left(\alpha-\theta(\alpha,\beta^*,l_2^i)\right)\\&+(\min(r,h)-(l_1^j+l_2^j))\left(\alpha-\theta(\alpha,\beta^*,l_2^j)\right),\end{aligned}\tag{57}$$

where $\sum_i l_1^i=\ell_1$, $l_1^i+l_2^i\le r$, $l_2^i\le 2$ *is given by* (54), $j=\left\lfloor\frac{n-d_{min}+1}{r+\delta-1}\right\rfloor+1$, *and* $\theta(\alpha,\beta^*,l_2^i)$ *is given by* (56).

VI. Conclusion

Distributed storage systems store data in multiple nodes. These systems not only require resilience against node failures, but also, due to their distributed nature, they may have to satisfy security and locality constraints. Regenerating codes proposed for DSS address the node failure resilience while efficiently trading off storage vs. repair bandwidth. In this paper, we considered security and locality aspects of coding schemes for DSS. The eavesdropper model analyzed in this paper belongs to the class of passive attack models, where the eavesdroppers observe the content of the nodes in the system. Accordingly, we considered an (ℓ_1,ℓ_2)-eavesdropper, where the content of any ℓ_1 nodes, and the downloaded information for any ℓ_2 nodes are leaked to the eavesdropper. With such an eavesdropper model, we first focused on the classical setup, which is resilient against single node failure at a time (without locality constraints). Noting that the secrecy capacity of this setting is open at the minimum storage regenerating point, we provided upper bounds on the secure file size and established the secrecy capacity for any (ℓ_1,ℓ_2) with $\ell_2\le 2$. Our coding scheme achieving this result also provides a better rate compared to the existing schemes. Then, we shifted focus on locality constraint, and studied the general scenario of having multiple parity nodes per local group. For this setting, we derived a new minimum distance bound for locally repairable codes, and present a $d_{\min}$-optimal coding scheme. Similar to the trade off analysis for the classical setup, we then studied the bandwidth efficient locally repairable codes, where we proposed a new bound and a coding scheme which is both $d_{\min}$-optimal and repair bandwidth efficient. This bandwidth efficient locally repairable setting is also analyzed under security constraints, for which we presented a secure file size upper bound and codes achieving the bound, and hence established the secrecy capacity, under special cases.

We list some avenues for further research here. 1) We first note that the novel bound that we establish for the minimum storage point allows for counting part of the data downloaded as additional leakage, and hence provide a tighter bound than the existing ones. Yet, we have not established the tightness of the bound for $\ell_2\ge 3$. Thus, new codes or improved bounds are of definite interest for secure MSR codes. 2) For locally repairable codes, we utilized MRD coding as the secrecy precoding, which requires extended field sizes. Designing codes that achieve the stated bounds with lower field sizes is an interesting problem. 3) One can also consider cooperative (or, multiple simultaneous node failure) repair [41]–[43] in a DSS. Secure code design in such a scenario is recently considered in [44]. Codes having both cooperative and locally repairable features can be studied. As distributed systems, storage problem may exhibit simultaneous node failures that need to be recovered with local connections. According to our best knowledge, this setting has not been studied (even without security constraints). Our ongoing efforts are on the design of coding schemes for DSS satisfying these properties.

Appendix A
Proof of Lemma 3

Proof: The proof follows from the classical techniques given by [7], where instead of 0-leakage, ϵ-leakage rate is considered. (The application of this technique in DSS is first considered in [11].) We have

$$\begin{aligned}I(\mathbf{u};\mathbf{e})&=H(\mathbf{e})-H(\mathbf{e}|\mathbf{u})&(58)\\&\overset{(a)}{\le}H(\mathbf{e})-H(\mathbf{e}|\mathbf{u})+H(\mathbf{e}|\mathbf{u},\mathbf{r})&(59)\\&\overset{(b)}{\le}H(\mathbf{r})-I(\mathbf{e};\mathbf{r}|\mathbf{u})&(60)\\&\overset{(c)}{=}H(\mathbf{r}|\mathbf{u},\mathbf{e})&(61)\\&\overset{(d)}{=}0&(62)\end{aligned}$$

where (a) follows by non-negativity of $H(\mathbf{e}|\mathbf{u},\mathbf{r})$, (b) is the condition $H(\mathbf{e})\le H(\mathbf{r})$, (c) is due to $H(\mathbf{r}|\mathbf{u})=H(\mathbf{r})$ as $\mathbf{r}$ and $\mathbf{u}$ are independent, (d) is the condition $H(\mathbf{r}|\mathbf{u},\mathbf{e})=0$. ■

Appendix B
Proof of Lemma 7

Proof: We prove the Lemma for $n-k=2$, i.e., $(k+2,k)-$DSS. The proof extends to higher number of parities in straightforward manner. Consider the following encoding matrix of the $(k+2,k)$ linear code employed by the DSS

$$\mathbf{G}=\begin{bmatrix}I&0&\dots&0\\0&I&\dots&0\\\vdots&\vdots&\ddots&\vdots\\0&0&\dots&I\\A_1&A_2&\dots&A_k\\B_1&B_2&\dots&B_k\end{bmatrix}.\tag{63}$$

Assume that a newcomer node downloads $S_{1,j}\mathbf{x}_{k+1}$ and $S_{2,j}\mathbf{x}_{k+2}$ from the first and the second parity nodes during the repair process of j-th systematic node. Here $S_{1,j}=V_{k+1,j}$ and $S_{2,j}=V_{k+2,j}$ are $\frac{\alpha}{2}\times\alpha$ matrices. In order to be able to perform bandwidth efficient repair using interference alignment, $\{S_{1,j}\}_{j=1}^k$ and $\{S_{2,j}\}_{j=1}^k$ satisfy

$$\text{rank}\begin{pmatrix}S_{1,j}A_i\\S_{2,j}B_i\end{pmatrix}=\frac{\alpha}{2}\quad\forall i\in[k]\backslash\{j\}\tag{64}$$

and

$$\text{rank}\begin{pmatrix}S_{1,j}A_j\\S_{2,j}B_j\end{pmatrix}=\alpha.\tag{65}$$

Note that data downloaded from i-th systematic node ($i \neq j$) for node repair is $V_{i,j}\mathbf{y}_i = V_{i,j}\mathbf{f}_i$. Since the repair matrix of node i associated to node repair of j-th node is $V_{i,j}$, we have

$$V_{i,j} = S_{1,j}A_i = S_{2,j}B_i. \tag{66}$$

Note that the above relationship is among subspaces. As pointed out earlier in the text, we use uppercase letters to represent both matrices and row spaces associated with those matrices. using the method of induction, we now show the main claim of Lemma 7. Note that this proof is modification of the proof of Lemma 10 in [45].

Base case ($|\mathcal{A}| = 1$)**:** The statement of Lemma 7 is true for this case as we perform a bandwidth efficient node repair, where each remaining node contributes $\frac{\alpha}{2}$ independent symbols for a single node repair.

Inductive step: Now we assume that the statement of Lemma 7 is true for all sets $\mathcal{A} \subseteq [k]\backslash\{i\}$ with $|\mathcal{A}| \leq m-1$ and prove it for all sets of indices of size m. With out loss of generality, we prove this for $\mathcal{A} = [m]$. We know from inductive hypothesis that

$$\dim\left(\bigcap_{j\in[m-1]} V_{i,j}\right) = \text{rank}\left(\bigcap_{j\in[m-1]} V_{i,j}\right) \leq \frac{\alpha}{2^{m-1}}, \tag{67}$$

Now assume that the result is false for $\mathcal{A} = [1:m]$, i.e.,

$$\begin{aligned}\dim\left(\bigcap_{j\in[m]} V_{i,j}\right) &= \text{rank}\left(\bigcap_{j\in[m]} V_{i,j}\right)\\ &= \text{rank}\left(\bigcap_{j\in[m]} S_{1,j}A_i\right)\\ &= \text{rank}\left(\bigcap_{j\in[m]} S_{2,j}B_i\right)\\ &> \frac{\alpha}{2^m},\end{aligned} \tag{68}$$

Since A_i and B_i are invertible, we have $\text{rank}\left(\bigcap_{j\in[m]} S_{1,j}A_i\right) = \text{rank}\left(\bigcap_{j\in[m]} S_{1,j}\right)$ and $\text{rank}\left(\bigcap_{j\in[m]} S_{2,j}B_i\right) = \text{rank}\left(\bigcap_{j\in[m]} S_{2,j}\right)$. Next, consider

$$\begin{aligned}\left(\bigcap_{j\in[m]} S_{1,j}\right)A_m &= \left(\bigcap_{j\in[m]} S_{1,j}A_m\right)\\ &\subseteq \left(\bigcap_{j\in[m-1]} S_{1,j}A_m\right)\\ &= \bigcap_{j\in[m-1]} V_{i,j}.\end{aligned} \tag{69}$$

Here, the above equation describe the relationship among row spaces of participating matrices. Similarly, we have the following.

$$\left(\bigcap_{j\in[m]} S_{2,j}\right)B_m \subseteq \bigcap_{j\in[m-1]} V_{i,j}. \tag{70}$$

Moreover, it follows from (68) and the fullrankness of A_m and B_m that

$$\begin{aligned}\dim\left(\left(\bigcap_{j\in[m]} S_{1,j}\right)A_m\right) &= \dim\left(\left(\bigcap_{j\in[m]} S_{2,j}\right)B_m\right)\\ &> \frac{\alpha}{2^m}\end{aligned} \tag{71}$$

Thus, we have two subspaces $\left(\bigcap_{j\in[m]} S_{1,j}\right)A_m$ and $\left(\bigcap_{j\in[m]} S_{2,j}\right)B_m$ of dimension strictly greater than $\frac{\alpha}{2^m}$ (see (71)), which are contained in the subspace $\bigcap_{j\in[m-1]} V_{i,j}$ of dimension at most $\frac{\alpha}{2^{m-1}}$ (see (69) and (70)). Therefore,

$$\begin{aligned}\left(\left(\bigcap_{j\in[m]} S_{1,j}\right)A_m\right)\bigcap\left(\left(\bigcap_{j\in[m]} S_{2,j}\right)B_m\right) \neq \{0\}\\ \Rightarrow S_{1,m}A_m \bigcap S_{2,m}B_m \neq \{0\}\end{aligned}$$

which is in contradiction with (65). This implies that

$$\dim\left(\bigcap_{j\in[m]} V_{i,j}\right) \leq \frac{\alpha}{2^m}. \tag{72}$$

■

REFERENCES

[1] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao and J. Kubiatowicz, "Pond:the OceanStore Prototype," in *Proc. USENIX File and Storage Technologies (FAST)*, 2003.

[2] S. Ghemawat, H. Gobioff and S. T. Leung, "The Google file system," in *Proc. of the 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, Oct. 2003.

[3] R. Bhagwan, K. Tati, Y. C. Cheng, S. Savage and G. M. Voelker, "Total Recall: System Support for Automated Availability Management," in *Proc. NSDI*, 2004.

[4] O. Goldreich, *Foundations of Cryptography: Volume II, Basic Applications*. Cambridge University Press, 2004.

[5] H. Delfs and H. Knebl, *Introduction to Cryptography: Principles and Applications*, 2nd ed. Springer, 2007.

[6] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, pp. 656715, 1949.

[7] A. Wyner, "The wire-tap channel," *The Bell System Technical Journal*, vol. 54, no. 8, pp. 1355-1387, Oct. 1975.

[8] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Amer. Inst. Elect. Eng.*, vol. 55, pp. 109-115, 1926.

[9] A. G. Dimakis, P. Godfrey, M. Wainwright and K. Ramachandran, "Network coding for distributed storage system," *IEEE Trans. on Information Theory*, vol. 56, no. 9, pp. 4539-4551, Sep. 2010.

[10] S. Pawar, S. El Rouayheb and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Trans. on Information Theory (Special Issue on Facets of Coding Theory: from Algorithms to Networks)*, vol. 57, no. 9, Sep. 2011.

[11] 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.

[12] P. Gopalan, C. Huang, H. Simitchi and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. on Information Theory*, vol. 58, no. 11, pp. 6925-6934, Nov. 2012.

[13] D. S. Papailiopoulos and A. G. Dimakis, "Locally Repairable Codes," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT 2012)*, Jul. 2012.

[14] 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)*, Jul. 2012.

[15] A. Shamir, "How to share a secret," *Communications of the ACM*, vol.22 n.11, p.612-613, Nov. 1979

[16] Y. Wu, A. G. Dimakis and K. Ramchandran, "Deterministic regenerating codes for distributed storage," in *Proc. 45th Allerton Conf. on Control, Computing and Communication*, Sep. 2007.

[17] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage," *IEEE J. on Selected Areas in Commun.*, vol. 28, pp. 277-288, Feb. 2010.

[18] 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)*, Jul. 2009.

[19] 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)*, Jan. 2010.

[20] C. Suh and K. Ramchandran, "Exact-repair MDS codes for distributed storage using interference alignment," *Proc. 2010 IEEE International Symposium on Information Theory (ISIT 2010)*, June 2010.

[21] 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.

[22] D. Papailiopoulos and A. Dimakis, "Repair optimal erasure codes through hadamard designs," in *Proc. 49th Allerton Conf. on Control, Computing and Communication*, Sep. 2011.

[23] 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. 2011 IEEE International Symposium on Information Theory (ISIT 2011)*, Aug. 2011.

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

[25] Z. Wang, I. Tamo and J. Bruck, "On Codes for Optimal Rebuilding Access," in *Proc. 49th Allerton Conf. on Control, Computing and Communication*, Sep. 2011.

[26] F. E. Oggier and A. Datta, "Homomorphic self-repairing codes for agile maintenance of distributed storage systems," *CoRR*, vol. abs/1107.3129, 2011.

[27] F. E. Oggier and A. Datta, "Self-Repairing Codes for Distributed Storage - A Projective Geometric Construction," *CoRR*, vol. abs/1105.0379, 2011.

[28] 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 International symposium on Network Computing and Applications (NCA 2007),pp.79–86, 2007.

[29] 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)*, Jul. 2012.

[30] N. Silberstein, A. S. Rawat and S. Vishwanath, "Error Resilience in Distributed Storage via Rank-Metric Codes," accepted to Allerton 2012, available in *http://arxiv.org/abs/1202.0800*, 2012.

[31] D. A. Patterson, G. A. Gibson and R. Katz, "A case for redundant arrays of inexpensive disks," in *SIGMOD: International Conference on Data Management*, pp. 109-116, Chicago, 1988.

[32] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 192202, Feb. 1995.

[33] M. Blaum, J. Bruck and E. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. on Information Theory*, vol. 42, pp. 529-542, 1996.

[34] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204-1216, 2000.

[35] 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.

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

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

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

[39] N. B. Shah, K. Rashmi, P. V. Kumar and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Trans. on Information Theory,* vol. 58, no. 4, pp. 2134-2158, Apr. 2012.

[40] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi and D. Karger, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 4413-4430, Oct. 2006.

[41] Y. Hu, Y. Xu, X. Wang, C. Zha, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE J. on Selected Areas in Commun.*, vol. 28, no. 2, pp. 268-275, Feb. 2010.

[42] A. Kermarrec, N. Le Scouarnec and G. Straub, "Repairing Multiple Failures with Coordinated and Adaptive Regenerating Codes," in *Proc. International Symposium on Network Coding (Netcod)*, Beijing, Jul. 2011.

[43] K. W. Shum and Y. Hu, "Existence of minimum-repair-bandwidth cooperative regenerating codes," in *Proc. International Symposium on Network Coding (Netcod)*, Beijing, Jul. 2011.

[44] O. O. Koyluoglu, A. S. Rawat and S. Vishwanath, "Secure cooperative regenerating codes for distributed storage systems," *CoRR*, vol. abs/1210.3664, 2012.

[45] I. Tamo, Z. Wang and J. Bruck, "Access vs. Bandwidth in Codes for Distributed Storage Systems," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT 2012)*, Jul. 2012.