Secure Cooperative Regenerating Codes for Distributed Storage Systems

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

Abstract

Regenerating codes enable trading off repair bandwidth for storage in distributed storage systems (DSS). Due to their distributed nature, these systems are intrinsically susceptible to attacks, and they may also be subject to multiple simultaneous node failures. Cooperative regenerating codes allow bandwidth efficient repair of multiple simultaneous node failures. This paper analyzes storage systems that employ cooperative regenerating codes that are robust to (passive) eavesdroppers. The analysis is divided into two parts, studying both minimum bandwidth and minimum storage cooperative regenerating scenarios. First, the secrecy capacity for minimum bandwidth cooperative regenerating codes is characterized. Second, for minimum storage cooperative regenerating codes, a secure file size upper bound and achievability results are provided. These results establish the secrecy capacity for the minimum storage scenario for certain special cases. In all scenarios, the achievability results correspond to exact repair, and secure file size upper bounds are obtained using min-cut analyses over a suitable secrecy graph representation of DSS. The main achievability argument is based on an appropriate pre-coding of the data to eliminate the information leakage to the eavesdropper.

I. Introduction

A. Background

Distributed storage systems (DSS) are designed to store data over a distributed network of nodes. DSS have become increasingly important given the growing volumes of data being generated, analyzed, and archived. OceanStore [1], Google File System (GFS) [2], and TotalRecall [3] are a few examples of storage systems employed today. Data to be stored is more than doubling every two years, and efficiency in storage and data recovery is particularly critical today. The coding schemes employed by DSS are designed to provide efficient storage while ensuring resilience against node failures in order to prevent the permanent loss of the data stored on the system. In a majority of existing literature, the analysis of DSS focuses primarily on isolated node failures. In our work, we study a more general scenario of DSS that can suffer from multiple simultaneous node failures. In addition to multiple node failures, DSS systems are also inherently susceptible to adversarial attacks such as one from eavesdroppers aiming to gain access to the stored data. Therefore, it is desirable to have DSS that meet certain security requirements while performing efficient repairs even in the case of multiple simultaneous node failures.

In [4], Dimakis et al. present a class of *regenerating codes*, which efficiently trade off per node storage and repair bandwidth for single node repair. These codes are designed to possess a property of maximum distance separable (MDS) codes, referred to as *'any k out of n'* property, wherein the entire data can be reconstructed by contacting to any k storage nodes out of n nodes in the storage system. By utilizing a network coding approach, the notion of *functional repair* is developed in [4]. Here, the original failed node may not be replicated exactly after node repair, but can be repaired such that the data stored on the repaired node is *functionally* equivalent to that stored on the failed node. On the other hand, *exact repair* requires that the regeneration process results in an exact replica of the data stored on the failed node. This is essential due to the ease of maintenance and other practical requirements such as maintaining a code in its systematic form.

Exact repair may also prove to be advantageous compared to functional repair in the presence of eavesdroppers, as the latter scheme requires updating the coding coefficients, which in turn may leak additional information to eavesdroppers [5]. The design of exact regenerating codes achieving one of the two ends of the trade off between storage and repair bandwidth has recently been investigated by researchers. In particular, Rashmi et al. [6] propose codes that are optimal for all parameters at the minimum bandwidth regeneration (MBR) point. For the minimum storage regeneration (MSR) point, on the other hand, the codes presented in [6] have their rate upper bounded by $\frac{1}{2}+\frac{1}{2n}$. In high rate regime (i.e., $\frac{k}{n}>\frac{1}{2}$), the codes at the MSR point have recently been proposed under various restrictions on per node storage α (see e.g., [7]–[10] and references therein). Some of these codes at the MSR point allow for bandwidth efficient repair of only systematic nodes, e.g., [8], [9].

B. Cooperative repair

As discussed above, DSS can also exhibit multiple simultaneous node failures, and it is desirable that these be repaired simultaneously. It is not uncommon that multiple failures occur in DSS, especially for large-scale systems. In addition, some

O. O. Koyluoglu is with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA (e-mail: ozan@email.arizona.edu).

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

DSS administrators may choose to wait to initiate a repair process after a critical number of failures has occurred (say t of them) in order to render the entire process more efficient and less frequent. For example, TotalRecall [3] currently executes a node repair process only after a certain threshold on the number of failures is reached. In such multiple failure scenarios, each new node replacing a failed one can still contact d remaining (surviving) nodes to download data for the repair process. In addition, replacement nodes, after downloading data from surviving nodes, can also exchange data within themselves to complete the repair process. This repair process is referred to as *cooperative repair* in [11], which presents network coding techniques to enable such repairs. Cooperative repair is shown to be essential as it can help in lowering the total repair bandwidth compared to the $t = 1$ case. Flexibility of the choice on download nodes during the repair process is analyzed in [12]. [13], focusing on functional repair, shows that under the constraint $n = d + t$, deliberately delaying repairs (and thus increasing t) does not result in gains in terms of MBR/MSR optimality. [13] and [14] utilize a cut-set bound argument and derive cooperative counterparts of the end points of the trade off curve between repair bandwidth and per node storage for regenerating codes. These two points are named as the minimum bandwidth cooperative regenerating (MBCR) point and the minimum storage cooperative regenerating (MSCR) point (see also [15]). The work in [14] shows the existence of cooperative regenerating codes with optimal repair bandwidth. Explicit code constructions for exact repair in this setup are presented in [16], for the MBCR point, and in [17], for the MSCR point. These constructions are designed for the setting of $d = k$. (See also [18].) Interference alignment is used in [19] to construct codes to operate at the MSCR point. This construction is limited to the case $k = 2$ with $d \geq k$, and does not generalize to $k \geq 3$ with $d > k$. An explicit construction for the MBCR point, with the restriction that $n = d + t$ for any $t \geq 1$, is presented in [20]. Finally, the reference [21] presents codes for the MBCR point for all possible parameter values. Noting the significance of cooperative repair in DSS, regenerating codes that have resilience to eavesdropping attacks will have greater value if they also have efficient cooperative repair mechanisms.

C. Security in distributed storage systems

The security of systems can be understood in terms of their resilience to either (or both) active or passive attacks [22], [23]. Active attacks include settings where the attacker modifies existing packets or injects new ones into the system, whereas passive attacks include eavesdroppers observing the information being stored/transmitted. For DSS, cryptographic approaches like private-key cryptography are often logistically prohibitive as the secret key distribution between each pair of nodes and its renewal are highly challenging, especially for large-scale systems. In addition, most cryptographic approaches are typically based on certain hardness results, which, if repudiated, could leave the system vulnerable to attacks. On the other hand, information theoretic security, see, e.g., [24], [25], presents secrecy guarantees even with infinite computational power at eavesdroppers without requiring the sharing and/or distribution of keys. This approach is based on the design of secrecy-achieving coding schemes by taking into account the amount of information leaked to eavesdroppers, and can offer new solutions to security challenges in DSS. In its simplest form, the security can be achieved with the one-time pad scheme [26], which claims the security of the data in a ciphertext, which is obtained by XOR of the data and a uniform key. This approach is of significant value to DSS. For example, consider a system storing the key at a node, and the ciphertext at another node. Then, the eavesdropper can not obtain any information regarding the data by observing one of these two nodes, whereas the data collector can contact to both nodes and decipher the data.

The problem of designing secure DSS against eavesdropping attacks has recently been studied by Pawar et al. [5], where the authors consider a passive eavesdropper model by allowing an eavesdropper to observe the data stored on any ℓ $(< k)$ storage nodes in a DSS employing an MBR code. The proposed schemes are designed for the 'bandwidth limited regime', and shown to achieve an upper bound on the secure file size, establishing its optimality. Shah et al. [27] consider the design of secure MSR codes. They point out that the eavesdropper model for an MSR code should be extended compared to that of an MBR code. The underlying reason is that at the MSR point of operation, an eavesdropper may obtain additional information by observing the data downloaded during a node repair (as compared to just observing the stored content). Thus, at the MSR point, the eavesdropper is modeled with a pair (ℓ_1, ℓ_2) with $\ell_1 + \ell_2 < k$ specifying the eavesdropper that has knowledge of the content of an ℓ_1 number of nodes, and, in addition, has knowledge of the downloaded information (and hence also the stored content) of an ℓ_2 number of nodes. We note that, as the amount of the data downloaded during a node repair and per node storage for MBR codes are the same, the two notions are different only at the MSR point. Considering such an eavesdropper model, Shah et al. present coding schemes utilizing product matrix codes [6], and show that the bound on secrecy capacity in [5] at the MBR point is achievable. They further use product matrix based codes for the MSR point as well, and show that the bound in [5] is achievable only when $\ell_2 = 0$. More recently, based on appropriate maximum rank distance pre-coding of Zigzag codes [9] and their variants [10], secure codes for the MSR point are proposed in [28]. This construction is shown to achieve the secrecy capacity for a class of systems where only the downloads of the systematic nodes are observed by the eavesdropper in [29] for $d = n - 1$ for any (ℓ_1, ℓ_2). Besides this classical MBR/MSR setting, the security aspects of locally repairable codes (see, e.g., [30]–[35]) are studied in [28]; and, security of DSS against active eavesdroppers is investigated in [36]–[39].

D. Contributions and organization

In this paper, we analyze and design secure and cooperative regenerating codes for DSS. In terms of security requirements, we utilize a passive and colluding eavesdropper model as presented in [27]. In this model, during the entire life span of the DSS, the eavesdropper can gain 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. Given this eavesdropper model, we focus on the problem of designing secure cooperative regenerating codes in the context of DSS that perform multiple simultaneous node repairs. This scenario generalizes the single node repair setting considered in earlier works to multiple node failures. In this paper, we establish the secrecy capacity for the MBCR point, and propose some secure codes for the MSCR point that are optimal for some special cases. In all scenarios, the achievability results allow for exact repair. The main secrecy achievability coding argument in this paper is obtained by utilizing a secret pre-coding (randomization) scheme to obtain secure coding schemes for DSS. In some cases, this pre-coding is established simply with the one-time pad scheme, and in others *maximum rank distance* (MRD) codes are utilized similar to the classical work of [40]. We remark that utilization of such pre-coding mechanisms is critical in showing the security of the proposed schemes. In particular, our security proofs are based on an oracle-type proof, where the eavesdropper is asked to decode the random symbols given the secure data symbols in addition to its observations. We design the pre-coding mechanisms to allow for such a security analysis. For example, when we utilize MRD pre-coding, we show that the eavesdropper has enough number of evaluations of an underlying polynomial (used in MRD pre-coding) at linearly independent points to resolve for the random symbols. We summarize our contributions in the following.

- We present an upper bound on the secrecy capacity for MBCR codes. This bound follows from the information theoretic analysis of counting the *secure flow* for a particular repair instance in DSS employing an MBCR code.
- We present a secure MBCR code for $n = d+t$ by employing a maximum rank distance pre-coding of the codes proposed in [20]. By comparing the secure file size of this code with the upper bound obtained, we characterize the secrecy capacity for MBCR codes for $n = d + t$ (for any ℓ_1).
- For $n > d + t$, we present secrecy capacity achieving codes (for any ℓ_1) by utilizing the bivariate polynomial proposed in [21]. In particular, our construction is based on randomizing some appropriate coefficients of the underlying bivariate polynomial.
- For MSCR codes, we obtain an upper bound on the secrecy capacity against a passive eavesdropper that takes into account the amount of information leaked to the download-observing eavesdroppers.
- We present a secure MSCR code for $k = t = 2$ based on the code proposed in [19]. In particular, we place random symbols on some nodes in DSS in addition to utilizing an appropriate one-time pad scheme for securing the data stored on other nodes. This construction is shown to achieve optimal secure file size. (Note that, for the case where $k = t = 2$, under the restriction that $\ell_1 + \ell_2 < k$ a non-trivial eavesdropper can only be associated with $(\ell_1, \ell_2) = (1, 0)$ or $(\ell_1, \ell_2) = (0, 1)$.)
- Finally, we construct secure MSCR codes when $d = k$ and characterize achievable secure file size using such codes for any (ℓ_1, ℓ_2). This construction is based on maximum rank distance pre-coding of the codes proposed in [17]. We show that this construction achieves the secrecy capacity of MSCR codes for a restrictive eavesdropper model specified by (ℓ_1, ℓ_2) with $\ell_2 \leq 1$.

The rest of the paper is organized as follows. In Section II, we provide the general system model together with some preliminary results utilized throughout the text. Section III provides the analysis of secure MBCR codes, and Section IV is devoted to the secure MSCR codes. The paper is concluded in Section V. Some of the results and proofs are relegated to appendices to enhance the flow of the paper.

II. System Model and Preliminaries

Consider a DSS consisting of n live nodes (at a given time) and a file $\mathbf{f}$ of size $\mathcal{M}$ over a finite field $\mathbb{F}$ that needs to be stored on the DSS[1]. The file $\mathbf{f}$ is encoded into n data blocks $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, each of length α over $\mathbb{F}$ with $\alpha \geq \frac{\mathcal{M}}{k}$. Given the codeword $\mathbf{x}$, 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 proposed to store data in centralized storage systems [41]–[43], the works on regenerating codes focus on storage schemes that have 'any k out of n' property, i.e., the contents of any k nodes suffice to recover the file. We focus on codes with this property[2].

We use the following notation throughout the text. We usually stick with the notation of having vectors denoted by lower-case bold letters; sets and subspaces are denoted by calligraphic fonts. For $a < b$, $[a : b]$ represents the set of numbers $\{a, a+1, \cdots, b\}$. Similarly, $a_{i_1:i_2,j_1:j_2}$ denotes the set $\{a_{i_1,j_1}, \ldots, a_{i_2,j_1}, \ldots, a_{i_1,j_2}, \ldots, a_{i_2,j_2}\}$. In addition, $a_{i_1:i_2,j}$ and $a_{i,j_1:j_2}$ denote $\{a_{i_1,j}, \ldots, a_{i_2,j}\}$ and $\{a_{i,j_1}, \ldots, a_{i,j_2}\}$, respectively. The symbols stored at node i is represented by the vector $\mathbf{s}_i$, the

[1]The size of $\mathbb{F}$ is specified later in the context of specific coding schemes.

[2]Note that having 'any k out of n' property does not necessarily imply that the code is an MDS code. A vector code $\mathcal{C}$ defined over $\mathbb{F}_q$ with symbols in $\mathbb{F}_q^\alpha$ is said to be MDS if $\alpha | \log_q |\mathcal{C}|$ and $d_{\min} = n - \frac{\log_q |\mathcal{C}|}{\alpha} + 1$.

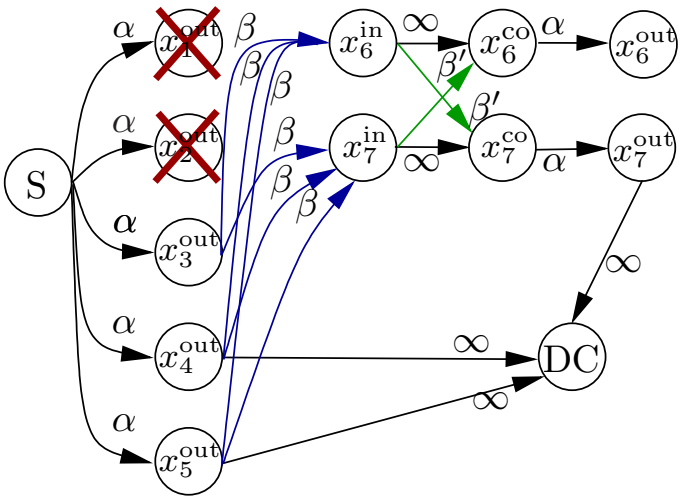

Fig. 1: Information flow graph of DSS implementing cooperative repair. In this representative example, we have $n = 5$, $d = k = 3$, and $t = 2$. Accordingly, after a failure of two nodes, namely node 1 and node 2, the system cooperatively repairs these two nodes as node 6 and node 7. Downloads from live nodes (blue) and from cooperative repair pairs (green) are shown. Due to exact repair, the network will repair the nodes to satisfy $x_6^{\text{out}} = x_1^{\text{out}}$ and $x_7^{\text{out}} = x_2^{\text{out}}$.

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 symbols downloaded at node j. DSS is initialized with the n nodes containing encoded symbols, i.e., $\mathbf{s}_i = \mathbf{x}_i$ for $i = 1, \cdots, n$.

A. Cooperative repair in DSS

In most of the studies on DSS, exact repair for regenerating codes is analyzed in the context of single node failure. However, it is not uncommon to see simultaneous multiple node failures in storage networks, especially for large ones. The basic setup involves the simultaneous repair of t (greater than one) failed nodes. After the failure of t storage nodes, the same number of newcomer nodes are introduced to the system. Each newcomer node connects to d arbitrary live storage nodes and downloads β symbols from each of these nodes. In addition, utilizing a cooperative approach, each newcomer node also contacts other newcomer nodes involved in node repair process and downloads β' symbols from each of these nodes. Hence, the total repair cost is given by

$$\gamma = d\beta + (t-1)\beta'.$$

Each newcomer node, to repair the ith node of the original network, uses these $d\beta + (t-1)\beta'$ number of downloaded symbols to regenerate α symbols, $\mathbf{x}_i$, and stores these symbols. The t nodes simultaneously repaired in a cooperative manner constitute a repair group.

This exact repair process preserves the 'k out of n 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. See Fig. 1.

We remark that, as also argued in [21], $d \geq k$ can be assumed without loss of generality. (Earlier papers on the subject assumed $d \geq k$ for simplicity. See, e.g., [16]–[20].) Remarkably, if $d < k$, a data collector can reconstruct the whole file by contacting only d nodes, as from these nodes the other nodes can be repaired in groups of size t. Thus, any (n, k, d) code with $d < k$ can be reduced to $(n, k' = d, d)$ code. Therefore, without loss of generality, we will assume $d \geq k$.

B. Information flow graph

In their seminal work [4], Dimakis et al. model the operation of DSS using a multicasting problem over an information flow graph. (See Figs. 1 and 2 for the flow graph in the cooperative setting.) An information flow graph representation of a DSS 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 ($(x_i^{\text{in}}, x_i^{\text{co}}, x_i^{\text{out}})$): In an information flow graph associated with cooperative repairs, we represent each node with a combination of three sub-nodes: x^{in}, x^{co}, and x^{out}. Here, x^{in} is the sub-node having the connections from the live nodes, x^{co} is the sub-node having the connections from the nodes under repair in the same repair group, and x^{out} is the storage sub-node which represents the content stored on the corresponding node in DSS. x^{out} is contacted by a data collector or other nodes during node repairs. x^{in} is connected to x^{co} with a link of infinite capacity, x^{co} is connected to x^{out} with a link of capacity α. We represent cuts using a notation with bars as in $(x^{\text{in}}, x^{\text{co}}|x^{\text{out}})$, meaning the cut is passing through the link between x^{co} and x^{out}. (See Fig. 2.) The nodes on the right hand side of the cuts belong to

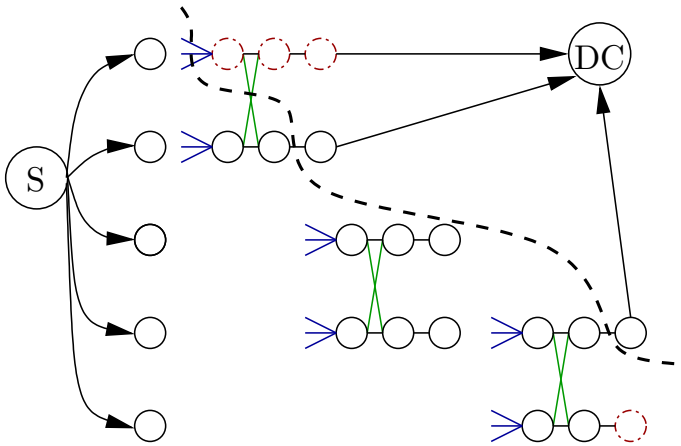

Fig. 2: Information flow graph of DSS implementing cooperative repair under security constraints. In this representative example, we have $n = 5$, $d = k = 3$, and $t = 2$. Multiple repair stages and a cut, represented by dashed line, through the nodes connected to the DC are shown. The figure has different cut types: The first repaired node has a cut of type $(|x^{\text{in}}, x^{\text{co}}, x^{\text{out}})$ and the second has a cut of type $(x^{\text{in}}, x^{\text{co}}|x^{\text{out}})$. Nodes that are being eavesdropped are indicated with dashed-dotted lines. Here, both the content and the downloads of the first repaired node is observed by the eavesdropper ($\ell_2 = 1$), and only the content of the last repaired node is observed by the eavesdropper ($\ell_1 = 1$). Accordingly, eavesdropper has observations of $d\beta + (t-1)\beta'$ downloaded symbols from the first repaired node, and has α number of symbols from the last repaired node.

data collector side, represented by the set $\mathcal{D}$, whereas the nodes belonging to the left hand side of the cuts belong to $\mathcal{D}^c$, the source side. For a newcomer node i, x_i^{in} is connected to x^{out} sub-nodes of d live nodes with links of capacity β symbols each, representing the data downloaded during node repair. x_i^{co} sub node associated with this newcomer node also connects to x^{in} sub-nodes of $(t-1)$ remaining nodes that are being repaired in the same group, and each link of these connections has a capacity of β'.

- Data collector node(s) (DC): Each data collector contacts x^{out} sub-node of k live nodes with edges of ∞-link capacity.

C. MBCR and MSCR points

With the aforementioned values of capacities of various edges in the information flow graph, the DSS is said to employ an $(n, k, d, \alpha, \beta, \beta')$ 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 utilized in network coding [44], [45].

Lemma 1 (Max-flow min-cut theorem for multicasting [4], [44], [45])**.**

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

where $\text{flow}(\text{S} \to \text{DC}_\text{i}, \mathcal{G})$ *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. 2, $\mathcal{M}$ symbol long file can be delivered to a data collector DC, only if the min cut is at least $\mathcal{M}$.

Dimakis et al. [4] obtain the following bound (for $t = 1$ case) by considering k successive node failures and evaluating the min-cut over all possible graphs.

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

We emphasize that the min-cut for this ($t = 1$) case is given by the scenario where k successively repaired nodes are connected to DC, and for each successive repair, the repaired node $i+1$ also connects to i number of previously repaired nodes. Hence, for each DC-connected node, its contribution to the value of a cut from S to DC is equal to $(d-i)\beta$ if the cut through the node is of type $(|x^{\text{in}}, x^{\text{out}})$, and is equal to α if the cut separates two sub-nodes, i.e., the cut through the node is of type $(x^{\text{in}}|x^{\text{out}})$. (Note that, x^{co} does not appear here as the model considered in [4] does not involve cooperative repair.) The codes that attain the bound in (1) are named as regenerating codes [4].

Analysis of the cut-set bounds for cooperative regenerating codes are provided in [13], [14]. (See also the arguments given in [11], [15]. Here, we follow the notations of [13], [15].) Consider a scenario where groups of nodes (each group having t nodes) are consecutively repaired in DSS. Let us enumerate the groups that are consecutively repaired as $i = 0, \cdots, \mu-1$. There are in total μt number of nodes in this repair process, and consider that DC contacts k out of these μt nodes. Let us denote

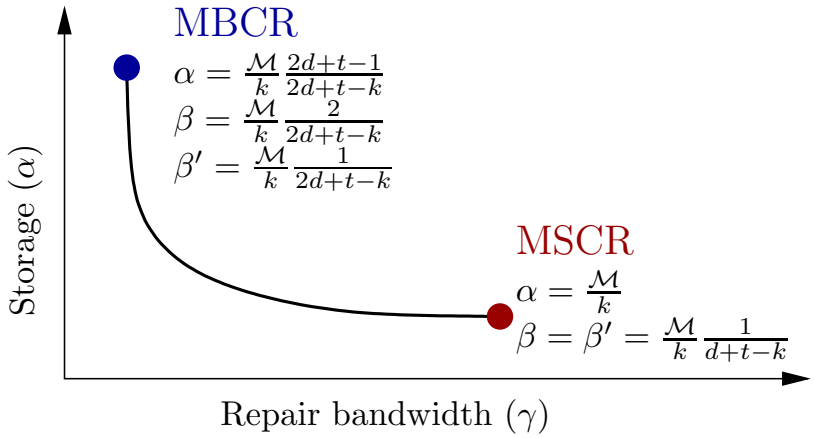

Fig. 3: Storage vs. repair bandwidth trade off for cooperative regenerating codes. The repair bandwidth is given by $\gamma = d\beta + (t-1)\beta'$.

the number of nodes in repair group i that are contacted by the DC as n_i such that $n_i \in [0:t]$, for all $i \in \{0,1,\ldots,\mu-1\}$, and $\sum_{i=0}^{\mu-1} n_i = k$.

The cut-set bound for this scenario is given by the following.

$$\mathcal{M} \leq \sum_{i=0}^{\mu-1} n_i \min\left\{\alpha, \left(d - \sum_{j=0}^{i-1} n_j\right)\beta + (t-n_i)\,\beta'\right\}. \tag{2}$$

Similar to the $t=1$ case described above, the cut of type $(x^{\text{in}}, x^{\text{co}} | x^{\text{out}})$ has a value of α. The cut of type $(|x^{\text{in}}, x^{\text{co}}, x^{\text{out}})$, on the other hand, has a value of $(t-n_i)\,\beta'$ due to the links coming from the nodes under repair that are not connected to DC and additional value of $(d - \sum_{j=0}^{i-1} n_j)\beta$ due to the connections to the previously repaired live nodes that are not contacted by DC. (Here, we again subtract the values of the flows from the nodes already belonging to the data collector side, $\mathcal{D}$.) The cut of type $(x^{\text{in}} | x^{\text{co}}, x^{\text{out}})$ has value of ∞ and hence, does not appear in the min-cut.

Note that, given a file size $\mathcal{M}$, there is an inherent trade off between storage per node α and *repair bandwidth* $\gamma \triangleq d\beta + (t-1)\beta'$. This trade off, for the cooperative setting, can be established using a similar analyses leading to the MBR/MSR points from the equation (1). Two classes of codes that achieve two extreme points of this trade off are named as *minimum bandwidth cooperative regenerating (MBCR)* codes and *minimum storage cooperative regenerating (MSCR)* codes. The former is obtained by first finding the minimum possible γ and then finding the minimum α satisfying (2). This point is given by the following.

$$\begin{aligned} \alpha_{\text{MBCR}} &= \frac{\mathcal{M}}{k}\frac{2d+t-1}{2d+t-k}, \quad \gamma_{\text{MBCR}} = \alpha_{\text{MBCR}}, \\ \beta_{\text{MBCR}} &= \frac{\mathcal{M}}{k}\frac{2}{2d+t-k}, \quad \beta'_{\text{MBCR}} = \frac{\mathcal{M}}{k}\frac{1}{2d+t-k} \end{aligned} \tag{3}$$

The MSCR point, on the other hand, is obtained by first choosing a minimum storage per node (i.e., $\alpha = \mathcal{M}/k$), and then minimizing γ (via choosing minimum possible β-β' pair) satisfying the min cut (2).

$$\begin{aligned} \alpha_{\text{MSCR}} &= \frac{\mathcal{M}}{k}, \quad \gamma_{\text{MSCR}} = \frac{\mathcal{M}}{k}\frac{d+t-1}{d+t-k}, \\ \beta_{\text{MSCR}} &= \frac{\mathcal{M}}{k}\frac{1}{d+t-k}, \quad \beta'_{\text{MSCR}} = \frac{\mathcal{M}}{k}\frac{1}{d+t-k} \end{aligned} \tag{4}$$

We depict these two trade off points, which are directly computable from (2), in Fig. 3. Note that, when $t=1$, these two points correspond to the MBR/MSR points characterized in [4]. (We refer reader to [13], [14] for a detailed derivation of these two points. See also [15] for an analysis for the simplified case of when $t|k$, i.e., the number of groups satisfies $\mu = k/t$.) We note that, in the next section, we consider secure file size upper bound using similar min cut arguments in the presence of eavesdroppers.

D. Eavesdropper model and Security in DSS

We consider an (ℓ_1, ℓ_2) eavesdropper, which has access to the stored data of any ℓ_1 number of nodes, and additionally has access to both the stored and downloaded data of any ℓ_2 number of nodes. Therefore, (ℓ_1, ℓ_2) eavesdropper has access to x_i^{out} for $i \in \mathcal{E}_1$ and $x_j^{\text{in}}, x_j^{\text{co}}, x_j^{\text{out}}$ for $j \in \mathcal{E}_2$ for some $\mathcal{E}_1, \mathcal{E}_2$ such that $\mathcal{E}_1, \mathcal{E}_2 \subset [1:n]$, $|\mathcal{E}_1| = \ell_1$, $|\mathcal{E}_2| = \ell_2$, and $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$. (See Fig. 2.) This is the eavesdropper model defined in [27] (adapted here to the cooperative repair setting), which generalizes the eavesdropper model considered in [5]. The eavesdropper is assumed to know the coding scheme employed by the DSS. At the MBCR point, a newcomer downloads $\alpha_{\text{MBCR}} = \gamma_{\text{MBCR}}$ amount of data. Thus, allowing an eavesdropper to access the data downloaded during a node repair besides the content stored on the node does not provide the eavesdropper with any additional information. However, at the MSCR point, repair bandwidth is strictly greater than the per node storage α_{MSCR} and an eavesdropper potentially gains more information if in addition to content stored on a node it also has access to the data downloaded during repair of the same node. We summarize the eavesdropper model together with the definition of achievability of a secure file size in the following.

Definition 2 (Security against an (ℓ_1, ℓ_2) eavesdropper)**.** *A DSS 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 $\mathcal{M}$ before storing it on DSS, and $\mathbf{e}$ is the eavesdropper observation vector given by $\mathbf{e} \triangleq \{x_i^{\text{out}}, x_j^{\text{in}}, x_j^{\text{co}}, x_j^{\text{out}} : i \in \mathcal{E}_1, j \in \mathcal{E}_2\}$. We use $I(\cdot;\cdot)$ to denote mutual information.*

We remark that, as it will be clear from the following sections, when a file $\mathbf{f}$ (of size $\mathcal{M}$) which contains a secure file (of size $\mathcal{M}^s$) is stored on a DSS, the remaining $\mathcal{M} - \mathcal{M}^s$ symbols of $\mathbf{f}$ can be utilized as an additional data, which does not have security constraints. Yet, noting the possibility of storing this insecure data, we will refer to this uniformly distributed part as the random data, which is utilized to achieve security. Thus, we consider files of form $\mathbf{f} = (\mathbf{f}^s, \mathbf{r})$, where $\mathbf{r}$ can represent random/insecure data. (We remark that DSS properties such as repair bandwidth and per node storage at the MBCR/MSCR point are defined over the file size $\mathcal{M}$.)

Here, we note the following lemma, which will be used in the following parts of the sequel.

Lemma 3 (Secrecy Lemma [25], [27])**.** *Consider a system with secure file symbols $\mathbf{f}^s$, random symbols $\mathbf{r}$ (independent of $\mathbf{f}^s$), and an eavesdropper with observations given by $\mathbf{e}$. If $H(\mathbf{e}) \leq H(\mathbf{r})$ and $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$, then $I(\mathbf{f}^s; \mathbf{e}) = 0$.*

Proof: See Appendix A. ∎

Finally, we provide a lemma summarizing how the cut values (from source to data collector DC) in the flow graph can be computed while some of the nodes are eavesdropped in the network.

Lemma 4. *[File size upper bound under secrecy constraints [28]] Consider a secure file $\mathbf{f}^s$ of size $\mathcal{M}^s$, i.e., $\mathcal{M}^s = H(\mathbf{f}^s)$. Consider an (ℓ_1, ℓ_2) eavesdropper observing the nodes in the sets $\mathcal{E}_1, \mathcal{E}_2$ for some $\mathcal{E}_1, \mathcal{E}_2 \subset \{1, \cdots, n\}$ as defined in Definition 2. Consider also that the data collector* DC *contacts to nodes in the set $\mathcal{K} = \mathcal{E}_1' \cup \mathcal{E}_2' \cup \mathcal{R}$ for some $\mathcal{E}_1' \subseteq \mathcal{E}_1$, $\mathcal{E}_2' \subseteq \mathcal{E}_2$, and $|\mathcal{K}| = k$. (We assume $\ell_1 + \ell_2 < k$, as otherwise the secrecy capacity of the network is zero.) Enumerating the nodes in $\mathcal{K}$ as $\{1, \cdots, k\}$, we have*

$$\mathcal{M}^s \leq \sum_{j=1}^{k} H(\mathbf{s}_j | \mathbf{s}_1, \cdots, \mathbf{s}_{j-1}, \mathbf{s}_{\mathcal{E}_1'}, \mathbf{d}_{\mathcal{E}_2'}). \tag{5}$$

Proof: The proof is summarized in Appendix B. (See also [28].) ∎

We use (5) (in some cases with a loose bound on the conditional entropy term) in the the following sections to obtain bounds on the secure file size. We remark that in addition to discounting for the previously contacted nodes $\mathbf{s}_1, \cdots, \mathbf{s}_{i-1}$, this bound also discounts for the *leakage* to the eavesdropper, by taking into account the leakage to the set $\{\mathbf{s}_{\mathcal{E}_1'}, \mathbf{d}_{\mathcal{E}_2'}\}$. Here, the minimum bound on the file size occurs when $\mathcal{E}_1' = \mathcal{E}_1$ and $\mathcal{E}_2' = \mathcal{E}_2$ in the lemma above. Hence, in order to obtain tighter upper bounds on secure file size, we consider only the scenarios for which the data collector connects to all the nodes being eavesdropped.

E. Maximum rank distance (MRD) codes via Gabidulin construction

In some of the following sections of the sequel, we consider maximum rank distance (MRD) pre-coding of the data at hand before storing it on DSS by using cooperative regenerating codes. Here, we introduce the Gabidulin construction of MRD codes [46]–[49].

First, we introduce some notation. In vector representation, the norm of a vector $\mathbf{v} \in \mathbb{F}_{q^m}^N$ is the column rank of $\mathbf{v}$ over the base field $\mathbb{F}_q$, denoted by $Rk(\mathbf{v}|\mathbb{F}_q)$. (This is the maximum number of linearly independent coordinates of $\mathbf{v}$ over the base field $\mathbb{F}_q$, for a given basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. A basis also establishes an isomorphism between N-length vectors, in $\mathbb{F}_{q^m}^N$, to $m \times N$ matrices, in $\mathbb{F}_q^{m \times N}$. Then, $Rk(\mathbf{v}|\mathbb{F}_q) = \text{rank}(\mathbf{V})$, where $\mathbf{V}$ is the corresponding matrix of $\mathbf{v}$ for the given basis.) Rank distance between two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_{q^m}^N$ is defined as $d(\mathbf{v}_1, \mathbf{v}_2) = Rk(\mathbf{v}_1 - \mathbf{v}_2|\mathbb{F}_q)$. (In matrix representation, this is equivalent to the rank of the difference of the two corresponding matrices of the vectors, i.e., $\text{rank}(\mathbf{V}_1 - \mathbf{V}_2)$.) Codes utilizing this distance metric are referred to as rank metric codes.

An $[N, K, D]$ rank metric code over the extension field $\mathbb{F}_{q^m}$ achieving the maximum rank distance $D = N - K + 1$ (for $m \geq N$) can be constructed with the following linearized polynomial. (This is referred to as the Gabidulin construction of MRD codes, or Gabidulin codes [46]–[49].)

$$f(g) = \sum_{i=0}^{K-1} u_i g^{[i]}, \tag{6}$$

where $[i] \triangleq q^i$, $u_i \in \mathbb{F}_{q^m}$, and g is an indeterminate which takes value in $\mathbb{F}_{q^m}$. Given N linearly independent (over $\mathbb{F}_q$) points, $\{g_1, g_2, \ldots, g_N\} \subset \mathbb{F}_{q^m}$, the codeword $\mathbf{c} = (c_1, c_2, \ldots, c_N)$ for a given set of K message (information) symbols $\mathbf{u} = (u_0, u_2, \ldots, u_{K-1}) \in \mathbb{F}_{q^m}^K$ is obtained as $c_j = f(g_j) = \sum_{i=0}^{K-1} u_i g_j^{[i]}$ for $j = [1 : N]$. With generator matrix representation, we have $\mathbf{c} = \mathbf{uG}$, where $\mathbf{G} = [g_1, \cdots, g_N; \cdots; g_1^{[K-1]}, \cdots, g_N^{[K-1]}]$.

We remark that a linearized polynomial $f(\cdot)$ satisfies $f(a_1 g_1 + a_2 g_2) = a_1 f(g_1) + a_2 f(g_2)$, for any $a_1, a_2 \in \mathbb{F}_q$ and $g_1, g_2 \in \mathbb{F}_{q^m}$. This property is utilized in our code constructions.

III. Secure MBCR codes

In this section, we study secure minimum bandwidth cooperative regenerating codes. We first present an upper bound on the secure file size that can be supported by an MBCR code. Then, we present exact repair coding schemes achieving the derived bound. In addition, we analyze how the cooperation affects the penalty paid in securing storage systems.

A. Upper bound on secure file size of MBCR codes

We have the following result for upper bound on secure file size for a DSS employing MBCR codes.

Proposition 5. *Cooperative regenerating codes operating at the MBCR point with a secure file size of $\mathcal{M}^s$ satisfy*

$$\begin{aligned} \mathcal{M}^s &\leq k(2d - k + t)\beta' - \ell_1(2d - \ell_1 + t)\beta' \\ &= (k - \ell_1)(2d + t - k - \ell_1)\beta', \end{aligned} \tag{7}$$

and the MBCR point is given by $\beta = 2\beta'$, $\alpha = \gamma = (2d + t - 1)\beta'$ for a file size of $\mathcal{M} = k(2d - k + t)\beta'$.

Proof: We consider the scenario where μ groups of nodes (each group having t nodes) are consecutively repaired in DSS as introduced in Section II-C. Accordingly, the data collector DC contacts $n_i \in [0 : t]$ nodes in the ith repair group such that $\sum_{i=0}^{\mu-1} n_i = k$. Without loss of generality we index these nodes as $\mathcal{K} = \{1, \ldots, k\}$. We consider two types of cuts in each group contacted by the DC: m_i number of nodes have the first cut type $(x^{\text{in}}, x^{\text{co}} | x^{\text{out}})$, and $n_i - m_i$ number of nodes have the second cut type $(|x^{\text{in}}, x^{\text{co}}, x^{\text{out}})$, $0 \leq i \leq \mu - 1$[3].

We consider ℓ_1 number of colluding eavesdroppers, each observing the contents of different nodes. Note that, for MBCR point analysis, we can consider $\ell_2 = 0$ without loss of generality, as the amount of the data a particular node stores is equal to the amount of the data it downloads during its repair. We denote the number of eavesdroppers on the nodes in the first cut type as $l_1^{i,1}$, $0 \leq i \leq \mu - 1$, and the number of eavesdroppers on the nodes in the second cut type as $l_1^{i,2}$, $0 \leq i \leq \mu - 1$, such that

$$\begin{aligned} l_1^{i,1} &\leq m_i, \\ l_1^{i,2} &\leq n_i - m_i, \end{aligned}$$

and

$$\sum_{i=0}^{\mu-1} l_1^i \leq \ell_1,$$

where $l_1^i = l_1^{i,1} + l_1^{i,2}$. We consider the repair scenario represented in Fig. 4 and utilize Lemma 4 to obtain the desired bound. Here, for repair group i, due to the eavesdroppers, the nodes that belong to the first type can only add the value of $(m_i - l_1^{i,1})\alpha$ to the cut. That is, the right hand side of (5) evaluates to 0 for the $l_1^{i,1}$ number of eavesdropped nodes, and evaluates to α for the remaining $m_i - l_1^{i,1}$ number of nodes of first type in ith repair group. The second type, on the other hand, consists of $n_i - m_i$ nodes in repair group i, out of which $l_1^{i,2}$ of them are eavesdropped. As the amount of the data downloaded during a node repair is equal to per node storage at the MBCR point, the nodes that are eavesdropped do not add a value to the cut. (For node j in ith repair group, if $j \in \mathcal{K}$ and j is an eavesdropped node of second type, then right hand side of (5) evaluates

[3]Note that the cuts of the form $(x^{\text{in}}, x^{\text{co}} | x^{\text{out}})$ give a cut value of α as opposed to $(x^{\text{in}} | x^{\text{co}}, x^{\text{out}})$, which has cut value larger than α. Since we are interested in the cuts of smaller size, we do not consider the cuts $(x^{\text{in}} | x^{\text{co}}, x^{\text{out}})$.

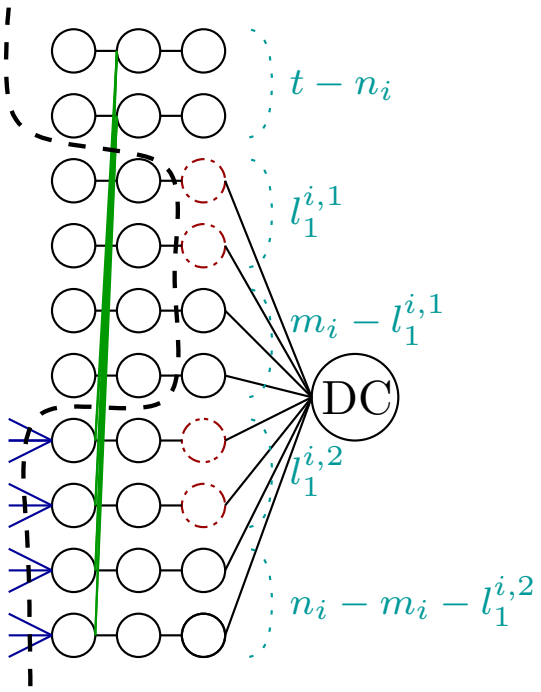

Fig. 4: The repair scenario considered to obtain an upper bound on secure file size for MBCR codes. Only a single repair group with t number of nodes is shown. First $t-n_i$ nodes are not contacted by DC. The following $l_1^{i,1}$ nodes are eavesdropped, $m_i-l_1^{i,1}$ nodes are contacted by DC with a cut of type $(x^{\text{in}},x^{\text{co}}|x^{\text{out}})$, $l_1^{i,2}$ nodes are eavesdropped, and $n_i-m_i-l_1^{i,2}$ nodes are contacted by DC with a cut of type $(|x^{\text{in}},x^{\text{co}},x^{\text{out}})$. For the last n_i-m_i nodes, only the symbols downloaded from the first $t-n_i$ nodes contribute to the flow.

as $H(\mathbf{s}_j|\mathbf{s}_1,\cdots,\mathbf{s}_{j-1},\mathbf{s}_{\mathcal{E}_1})=H(\mathbf{d}_j|\mathbf{s}_1,\cdots,\mathbf{s}_{j-1},\mathbf{s}_{\mathcal{E}_1})=0$. This follows as one can get $\mathbf{s}_j$ from $\mathbf{d}_j$ and vice versa for the MBCR point, and $\mathbf{s}_j\subseteq\mathbf{s}_{\mathcal{E}_1}$. Therefore, the contribution of these nodes to the cut through their download links evaluates to zero in value.) Consider the remaining $n_i-m_i-l_1^{i,2}$ number of nodes in the repair group i, denoted as $\mathcal{N}_i$. These nodes contact d live nodes, $\sum_{r=0}^{i-1}n_r$ number of these contacted nodes belong to the previously repaired groups. In addition, these nodes contact $t-n_i$ nodes that are previously repaired but not contacted by DC. For these nodes, the right hand side of (5) can be evaluated as

$$\begin{aligned}\sum_{j\in\mathcal{N}_i}H(\mathbf{s}_j|\mathbf{s}_1,\cdots,\mathbf{s}_{j-1},\mathbf{s}_{\mathcal{E}_1}) &\overset{(a)}{=}\sum_{j\in\mathcal{N}_i}H(\mathbf{d}_j|\mathbf{s}_1,\cdots,\mathbf{s}_{j-1},\mathbf{s}_{\mathcal{E}_1})\\ &\overset{(b)}{=}H(\mathbf{d}_{\mathcal{N}_i}|\mathbf{s}_{\mathcal{P}},\mathbf{s}_{\mathcal{E}_1})\\ &\overset{(c)}{\leq}|\mathcal{N}_i|\left((d-\sum_{r=0}^{i-1}n_r)\beta+(t-n_i)\beta'\right),\end{aligned}$$

where (a) follows as one can get $\mathbf{s}_j$ from $\mathbf{d}_j$ and vice versa for the MBCR point, (b) follows by summing the conditional entropy terms over $j\in\mathcal{N}_i$ where $\mathbf{s}_{\mathcal{P}}$ denotes the set of nodes that are repaired before repairing the ones in $\mathcal{N}_i$, and (c) follows by upper bounding $H(\mathbf{d}_{\mathcal{N}_i}|\mathbf{s}_{\mathcal{P}},\mathbf{s}_{\mathcal{E}_1})$ by assuming each node in $\mathcal{N}_i$ receives independent symbols from the previously repaired nodes and from the first $t-n_i$ nodes of this repair group. With $|\mathcal{N}_i|=n_i-m_i-l_1^{i,2}$, we have

$$\mathcal{M}^s\leq\sum_{i=0}^{\mu-1}\left(\left(m_i-l_1^{i,1}\right)\alpha+\left(n_i-m_i-l_1^{i,2}\right)C_i\right),\tag{8}$$

where

$$C_i = \left(d - \sum_{r=0}^{i-1} n_r \right) \beta + (t - n_i)\, \beta'. \tag{9}$$

We remark that the cut-value, i.e., the right hand side of (8), is minimized when we have

$$\sum_{i=0}^{\mu-1} (l_1^{i,1} + l_1^{i,2}) = \ell_1.$$

We consider two scenarios in (8), (i) $m_i = 0, l_1^{i,1} = 0, l_1^{i,2} = l_1^i$ (for which the right hand side of (8) evaluates to $(n_i - l_1^i)C_i$) and (ii) $m_i = n_i, l_1^{i,1} = l_1^i, l_1^{i,2} = 0$ (for which the right hand side of (8) evaluates to $(n_i - l_1^i)\alpha$). Hence, we obtain

$$\mathcal{M}^s \leq \sum_{i=0}^{\mu-1} \left((n_i - l_1^i) \times \min \left\{ \alpha, \left(d - \sum_{r=0}^{i-1} n_r \right) \beta + (t - n_i)\, \beta' \right\} \right), \tag{10}$$

where $\sum_{i=0}^{\mu-1} l_1^i = \ell_1$.

Note that, at the MBCR point, we have

$$\alpha = d\beta + (t-1)\beta'. \tag{11}$$

Utilizing this, we consider the following case for (10).

Case 1: $\mu = k$, $n_i = 1$, $\forall i = 0, \cdots, k-1$. This case corresponds to a repair scenario where the data collector contacts only one node from each repair group. Accordingly, we have the following bound.

$$\mathcal{M}^s \quad \leq \quad \sum_{i=0}^{k-1} (1 - l_1^i)\, ((d-i)\beta + (t-1)\beta')$$

Here, the minimum cut value corresponds to having $l_1^i = 1$ for $i = 0, 1, \cdots, \ell_1 - 1$, and $l_1^i = 0$ for $i = \ell_1, \ldots, k-1$. Hence, we get

$$\mathcal{M}^s \quad \leq \quad \sum_{i=\ell_1}^{k-1} (d-i)\beta + (t-1)\beta',$$

from which we obtain

$$\mathcal{M}^s \leq \frac{(k-\ell_1)(2d-k-\ell_1+1)}{2}\beta + (k-\ell_1)(t-1)\beta'. \tag{12}$$

The bound in (12) evaluates to the stated bound at the MBCR point. ■

Remark 6. *One can also consider the following repair scenarios in* (10)*.*

Case 2: *If $t \geq k$, $\mu = 1$, $n_0 = k$. Here, data collector contacts to nodes belonging to a single repair group. Accordingly, we have*

$$\mathcal{M}^s \quad \leq \quad (k - \ell_1)\, (d\beta + (t-k)\beta')\,. \tag{13}$$

Case 3: *If $t < k$, $\mu = \lfloor k/t \rfloor + 1$, $n_i = t$ for $i = 0, \cdots, \mu - 2$, and $n_{\mu-1} = k - \lfloor k/t \rfloor\, t$. Here, data collector contacts to every node in every repair group (except for the last repair group). Denoting $a \triangleq \lfloor k/t \rfloor$ and $b \triangleq k - at$, so that $k = at + b$, from* (10)*, we obtain*

$$\mathcal{M}^s \leq \min_{l_1^i \leq t \text{ s.t. } \sum_{i=0}^{a} l_1^i = \ell_1} \sum_{i=0}^{a-1} (t - l_1^i)(d - it)\beta + (b - l_1^a)\, \{(d-at)\beta + (t-b)\beta'\}\,. \tag{14}$$

We observe that both (13) *and* (14) *evaluate to loose bounds compared to that of* (12)*. (For some parameters, Case 2/3 provides the same bound as that in Case 1. But, in general, Case 2/3 gives loose bound compared to Case 1. Details of this analysis are omitted for brevity.)*

In the following sections, we show that the bound given by (12) is achievable, and hence it is the best bound that one can obtain for the MBCR point. That is, the repair scenario considered for Case 1 above is one of the worst cases and results in the tightest bound for the MBCR point under the secrecy constraints for all parameters. (For some parameters, Case 2/3 above represents an equivalently challenging bottleneck repair scenario for security in MBCR codes as well.)

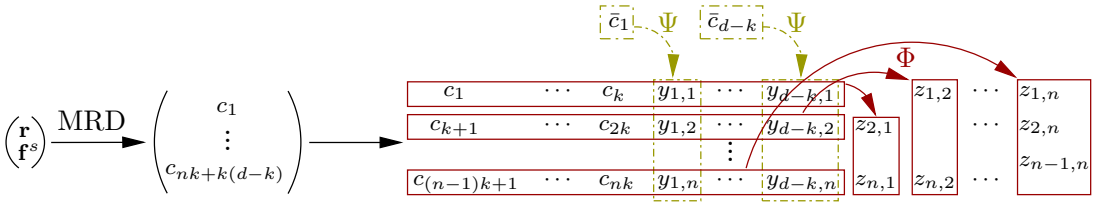

Fig. 5: The data $(\mathbf{r}, \mathbf{f}^s)$ is encoded with MRD coding into codeword symbols $c_1, \cdots, c_{nk+k(d-k)}$. Each row on the right represents a node in DSS, and the first nk symbols are placed to n nodes uniformly without additional coding. The remaining symbols are represented as $\bar{\mathbf{c}}_j = (c_{nk+(j-1)k+1}, \ldots, c_{nk+jk})$ for $j = 1, \cdots, d-k$. Then, $\bar{\mathbf{c}}_j$, of length-k, is encoded into $\mathbf{y}_j = (y_{j,1}, \ldots, y_{j,n})$ using an MDS code, generator matrix of which is denoted by Ψ. Finally, d primary symbols of node i, given by $\{c_{(i-1)k+1}, \ldots, c_{ik}, y_{1,i}, \ldots y_{d-k,i}\}$, is encoded into $(n-1)$ symbols long codeword $\mathbf{z}_i = (z_{1,i}, \ldots, z_{i-1,i}, z_{i+1,i}, \ldots, z_{n,i})$ using Φ as a generator matrix; and $z_{j,i}$ is placed at node $j \neq i$.

B. Code construction for secure MBCR when $n = d + t$

In this section, we focus on the special case $n = d + t$. (The case of $n > d + t$ will be considered in the following section.) We consider secrecy pre-coding of the data at hand before storing it to DSS using an MBCR code. We establish this pre-coding with maximum rank distance (MRD) codes introduced in Section II-E. (Please refer to Fig. 5.)

Consider the *normalized* MBCR point given by $\mathcal{M} = k(2d - k + t)$, $\beta' = 1$, $\beta = 2$, $\alpha = \gamma = 2d + t - 1$, $\mathcal{M}^s = k(2d - k + t) - \ell_1(2d - \ell_1 + t)$, and $n = d + t$. (The case of $\beta' > 1$ can be obtained by implementing independent codes parallelly in the system.) We use MRD codes with $N = K = \mathcal{M}$; hence, the rank distance bound $D \leq N - K + 1$ is saturated at $D = 1$. Accordingly, we utilize $[\mathcal{M}, \mathcal{M}, 1]$ MRD codes over $\mathbb{F}_{q^m}$, which maps length $\mathcal{M}$ vectors (each element of it being in $\mathbb{F}_{q^m}$) to length $\mathcal{M}$ codewords in $\mathbb{F}_{q^m}^{\mathcal{M}}$ (with $m \geq N = \mathcal{M}$). The coefficients of the underlying linearized polynomial $(f(g))$ are chosen by $\mathcal{M} - \mathcal{M}^s$ random symbols denoted by $\mathbf{r} \in \mathbb{F}_{q^m}^{\mathcal{M}-\mathcal{M}^s}$ and $\mathcal{M}^s$ secure data symbols denoted by $\mathbf{f}^s \in \mathbb{F}_{q^m}^{\mathcal{M}^s}$. (That is, $\mathbf{u} = (\mathbf{r}, \mathbf{f}^s)$ in (6).) The corresponding linearized polynomial $f(g)$ is evaluated at $\mathcal{M}$ points $\{g_1, \ldots, g_{\mathcal{M}}\}$, which are linearly independent over $\mathbb{F}_q$. We denote these as $c_j = f(g_j)$ for $j = 1, \cdots, \mathcal{M}$. This finalizes the secrecy pre-coding step.

The second encoding step is based on the encoding scheme for cooperative repair proposed in [20]. (Here, we will summarize file recovery and node repair processes for the case of MRD pre-coding, and provide the proof of security.) Split the $\mathcal{M}$ symbols into two parts a) c_1 to c_{nk}, and b) c_{nk+1} to $c_{nk+k(d-k)}$. (Note that $n = d + t$ and $\mathcal{M} = nk + k(d - k)$.) The first part is divided into n groups of k symbols, and stored in n nodes. Here, node i stores $c_{(i-1)k+1}$ to c_{ik}. The second part is divided into $d - k$ groups of k symbols. These symbols are encoded with an (n, k) MDS code, and stored on n nodes. In particular, $\mathbf{y}_j = (y_{j,1}, \ldots, y_{j,n})$ is generated from symbols $\bar{\mathbf{c}}_j = (c_{nk+(j-1)k+1}, \ldots, c_{nk+jk})$ by utilizing some MDS encoding matrix Ψ of size $k \times n$; then, $y_{j,i}$ is stored at node i, for $j = 1, \cdots, d - k$. Node i, having stored $\{c_{(i-1)k+1}, \ldots, c_{ik}, y_{1,i}, \ldots y_{d-k,i}\}$, which is referred to as the primary data of node i, encodes these symbols using an $(n - 1, d)$ MDS code that has a generator matrix given by a (generalized) Cauchy matrix Φ of size $d \times (n - 1)$. (This choice of Φ ensures that $[\mathbf{I}_d \ \Phi]$ is a generator matrix for an $(n + d - 1, d)$ MDS code [50].) Let $\mathbf{z}_i = (z_{1,i}, \ldots, z_{i-1,i}, z_{i+1,i}, \ldots, z_{n,i})$ denote the $(n - 1)$ symbols long codeword obtained by encoding the primary data of node i. These $n - 1$ symbols are stored in every other node one-by-one. In particular, node $j \neq i$ stores $z_{j,i}$. We call $\{z_{j,i} : i \in [1 : n], i \neq j\}$ as the secondary data. This procedure is repeated for every node, so that each node i stores $\{c_{(i-1)k+1}, \ldots, c_{ik}, y_{1,i}, \ldots, y_{d-k,i}, z_{i,1}, \ldots, z_{i,i-1}, z_{i,i+1}, \ \ldots, z_{i,n}\}$, and hence total number of symbols stored at each node is $k + (d - k) + (n - 1) = d + n - 1 = 2d + t - 1 = \alpha$.

File recovery at DC: DC connects to any k nodes, without loss of generality we assume the first k nodes. From $y_{j,1:k}$, DC can obtain $c_{nk+(j-1)k+1}, \cdots, c_{nk+jk}$, for each $j = [1 : d - k]$. It can re-encode these into $y_{j,1:n}$ using the MDS code, and obtain the other y symbols at the remaining nodes. Then, for each $i \in [k + 1 : n]$, DC can use the MDS property of $[\mathbf{I}_d \ \Phi]$, to obtain $c_{(i-1)k+1}, \cdots, c_{ik}$ symbols of node i from the k secondary data symbols of the contacted nodes, i.e., $z_{j,i}$ for $j = [1 : k]$, and additional $d - k$ symbols, $y_{j,i}$ for $j = [1 : d - k]$. Having obtained $c_1, \cdots, c_{\mathcal{M}}$, DC can perform interpolation to solve for both data and random coefficients.

Node repair: Assume that the first t nodes fail. From the secondary data stored in the remaining $d = n - t$ nodes, $z_{t+1,i}, \cdots, z_{n,i}$, one can recover $c_{(i-1)k+1}, \cdots, c_{ik}$ and $y_{1,i}, \cdots, y_{d-k,i}$ for node $i = 1, \cdots, t$. (This corresponds to sending 1 symbol from each of d nodes to each of the t nodes.) Then, to recover the secondary data stored at each node under repair, say for the node $j = 1, \cdots, t$, every other node, i.e., nodes $i \neq j$, including the nodes under repair, computes and sends its corresponding encoded primary data, i.e., $z_{j,i}$, to node j. (This corresponds to sending 1 symbol from each node to each of the t nodes.) This achieves $\beta = 2$ and $\beta' = 1$ symbols for the repair procedure.

Security: Consider that the eavesdropper is observing the first ℓ_1 nodes. (See Fig. 6.) Let $\mathcal{C} \triangleq \{c_1, \ldots, c_{\ell_1 k}\}$, $\mathcal{Y} \triangleq \{y_{1,1}, \ldots, y_{d-k,1}, \cdots, y_{1,\ell_1}, \ldots, y_{d-k,\ell_1}\}$, $\mathcal{Z} \triangleq \{z_{j,i} \text{ for } j = 1, \ldots, \ell_1, \text{ and } i = \ell_1 + 1, \cdots, n\}$.

Due to the coding scheme, the symbols $\{z_{j,i}\}$ for $j = 1, \cdots, \ell_1$; $i = 1, \cdots, \ell_1$; and $j \neq i$ are linear combinations of the symbols in $\mathcal{C} \cup \mathcal{Y}$. In addition, the symbols in the sets $\mathcal{C}, \mathcal{Y}, \mathcal{Z}$ correspond to linearly independent evaluation points. This follows due to the code construction as detailed in the following. We remark that symbols in $\mathcal{C}$ are clearly independent of the

$$
\begin{array}{ccccccccccccc}
c_1 & \cdots & c_k & y_{1,1} & \cdots & y_{d-k,1} & & z_{1,2} & \cdots & z_{1,\ell_1} & z_{1,\ell_1+1} & \cdots & z_{1,n} \\
c_{k+1} & \cdots & c_{2k} & y_{1,2} & \cdots & y_{d-k,2} & z_{2,1} & & \cdots & z_{2,\ell_1} & z_{2,\ell_1+1} & \cdots & z_{2,n} \\
 & & & & \vdots & & & & & & & & \\
c_{(\ell_1-1)k+1} & \cdots & c_{\ell_1 k} & y_{1,\ell_1} & \cdots & y_{d-k,\ell_1} & z_{\ell_1,1} & z_{\ell_1,2} & \cdots & & z_{\ell_1,\ell_1+1} & \cdots & z_{\ell_1,n} \\
 & & & & \vdots & & & & & & & & \\
c_{(n-1)k+1} & \cdots & c_{nk} & y_{1,n} & \cdots & y_{d-k,n} & & & & & & &
\end{array}
$$

Fig. 6: The eavesdropped nodes are represented by the first ℓ_1 rows, each row corresponding to a node. The symbols in the dashed-dotted (green) box are linear combinations of the symbols in $\mathcal{C} = \{c_1, \ldots, c_{\ell_1 k}\}$ and $\mathcal{Y} = \{y_{1,1}, \ldots, y_{d-k,1}, \cdots, y_{1,\ell_1}, \ldots, y_{d-k,\ell_1}\}$. The remaining symbols denoted with $z_{1:\ell_1,\ell_1+1:n}$ are linearly independent of $\mathcal{X}$ and $\mathcal{Y}$ due to encoding with the matrix Φ, where $z_{1:\ell_1,i}$ is generated from $\{c_{(i-1)k+1}, \ldots, c_{ik}, y_{1,i}, \ldots y_{d-k,i}\}$ for $i = \ell_1, \cdots, n$. For example, the last row, primary symbols of node n generates the last column, i.e., the eavesdropped symbols $z_{1:\ell_1,n}$. (See Fig. 5 for details of the encoding steps.)

ones in $\mathcal{Y} \cup \mathcal{Z}$. However, it is not clear at first sight whether the symbols in $\mathcal{Y}$ and $\mathcal{Z}$ are independent. Note that, for example, the symbols $y_{1,1:n}$ are dependent, i.e., each one can be uniquely determined by any other k of them due to MDS coding by Ψ. And, symbols $y_{1:d-k,\ell+1:n}$ generates $\mathcal{Z}$. But, the generation of $\mathcal{Z}$ is *together with* the symbols in $c_{\ell_1 k+1:nk}$. (See Fig. 6.)

In particular, we have the following,

$$[\tilde{C}\ \tilde{Y}]\tilde{\Phi} = Z,$$

where

$$\tilde{C} = \begin{bmatrix} c_{\ell_1 k+1} & \cdots & c_{(\ell_1+1)k} \\ \vdots & \ddots & \vdots \\ c_{(n-1)k+1} & \cdots & c_{nk} \end{bmatrix},$$

$$\tilde{Y} = \begin{bmatrix} y_{1,\ell_1+1} & \cdots & y_{d-k,\ell_1+1} \\ \vdots & \ddots & \vdots \\ y_{1,n} & \cdots & y_{d-k,n} \end{bmatrix},$$

$$Z = \begin{bmatrix} z_{1,\ell_1+1} & \cdots & z_{\ell_1,\ell_1+1} \\ \vdots & \ddots & \vdots \\ z_{1,n} & \cdots & z_{\ell_1,n} \end{bmatrix},$$

and $\tilde{\Phi}$ is the corresponding ℓ_1 columns of Φ. (In the example above, we consider the first ℓ_1 columns.) Consider

$$\tilde{\Phi} = \begin{bmatrix} \tilde{\Phi}^u \\ \tilde{\Phi}^l \end{bmatrix}$$

with $k \times \ell_1$ matrix $\tilde{\Phi}^u$, and, accordingly, $\tilde{C}\tilde{\Phi}^u + \tilde{Y}\tilde{\Phi}^l = Z$. Here, the symbols in $\tilde{C}\tilde{\Phi}^u$ are linearly independent. (This follows, as appending upper k rows of any $k - \ell_1$ number of remaining columns of Φ to $\tilde{\Phi}^u$ will constitute a $k \times k$ submatrix of Φ, which is non-singular. Denoting this matrix as $\Phi' = [\tilde{\Phi}^u \cdots]$, where $\cdots$ representing the added elements of Φ, we observe that all symbols in $\tilde{C}\Phi'$ are linearly independent, which implies the linear independence of symbols in $\tilde{C}\tilde{\Phi}^u$.) Then, as the symbols in the matrix $\tilde{C}$ are independent of the ones in the set $\{y_{1:d-k,1:n}\}$, it follows that symbols in the matrix $Z = \tilde{C}\tilde{\Phi}^u + \tilde{Y}\tilde{\Phi}^l$, i.e., the symbols in the set $\mathcal{Z}$, and the symbols in the set $\mathcal{Y}$ are linearly independent.

Therefore, due to the linearized property of the code, the eavesdropper, observing $\ell_1\alpha = \ell_1(2d + t - 1)$ symbols, has evaluations of the polynomial $f(\cdot)$ at $\ell_1(2d+t-\ell_1)$ linearly independent points. Using the secure data symbols, together with interpolation from these $\ell_1(2d+t-\ell_1)$ symbols, the eavesdropper can solve for $\ell_1(2d+t-\ell_1)$ random symbols. Then, denoting the eavesdroppers' observation as $\mathbf{e}$, we have $H(\mathbf{r}|\mathbf{e}, \mathbf{f}^s) = 0$. Since $H(\mathbf{e}) = H(\mathbf{r})$, from Lemma 3, we obtain $I(\mathbf{f}^s; \mathbf{e}) = 0$.

Now, using the upper bound given in Proposition 5, we obtain the following result.

Proposition 7. *The secrecy capacity at the MBCR point for a file size of $\mathcal{M} = k(2d - k + t)\beta'$ is given by $\mathcal{M}^s = k(2d - k + t)\beta' - \ell_1(2d - \ell_1 + t)\beta'$, if $n = d + t$.*

C. Does cooperation enhance/degrade security at MBCR?

The repair bandwidth for cooperative regenerating codes is defined by $\gamma = d\beta + (t-1)\beta'$. In this section, we analyze $\bar{\gamma} = \frac{\gamma}{\mathcal{M}^s}$, the ratio of repair bandwidth to the secure file size, referred to as the normalized repair bandwidth (NRBW).

Without the security constraints, for which $\ell_1 = 0$ in Proposition 5, we observe that, at the MBCR point, NRBW is given by

$$\bar{\gamma}(\ell_1 = 0) = \frac{2d + t - 1}{k(2d - k + t)},$$

which is equal to

$$\bar{\gamma}(\ell_1 = 0, n = d + t) = \frac{2n - t - 1}{k(2n - k - t)}$$

for a system with $n = d + t$. Here, the classical (i.e., non-cooperative) scenario corresponds to $t = 1$ case, which has an NRBW of

$$\bar{\gamma}(\ell_1 = 0, n = d + t, t = 1) = \frac{2n - 2}{k(2n - k - 1)}.$$

Comparing the last two equations, we see that

$$\bar{\gamma}(\ell_1 = 0, n = d + t) \geq \bar{\gamma}(\ell_1 = 0, n = d + t, t = 1),$$

with equality iff $t = 1$ (for $k > 1$). Therefore, without the security constraints, having simultaneous repairs of size greater than 1 actually increases the normalized repair bandwidth. This nature of cooperation also results in the conclusion that deliberately delaying the repairs does not bring additional savings [13]. (This observation is proposed for both MBCR and MSCR points in [13] with an analysis of derivative of γ with respect to t. Here, we provide an analysis with NRBW.)

We revisit the above conclusion under security constraints. The question is whether the cooperation (i.e., having a system with multiple failures, or deliberately delaying the repairs) results in a loss/gain in secure DSS. We have

$$\bar{\gamma}(n = d + t) = \frac{2n - t - 1}{k(2n - k - t) - \ell_1(2n - \ell_1 - t)},$$

and

$$\bar{\gamma}(n = d + t, t = 1) = \frac{2n - 2}{k(2n - k - 1) - \ell_1(2n - \ell_1 - 1)}.$$

A calculation similar to above shows that $\bar{\gamma}(n = d + t) \geq \bar{\gamma}(n = d + t, t = 1)$ with equality if and only if $t = 1$ (for $k > \ell_1 \geq 0$). This shows that NRBW for the case $t > 1$ is strictly greater than that of $t = 1$ when $n = d + t$ for $\ell_1 < k$. The MBCR points given in Proposition 5 for codes satisfying $0 \leq \ell_1 < k < n$, $d \geq k$, and $d = n - t$ are given in Table I in Appendix C. As evident from the table, we see that cooperation does not bring additional savings for secure DSS at the MBCR point when $d + t = n$. This in turn means that one may not delay the repairs to achieve a better performance than that of single failure-repair if d is chosen such that $n = d + t$ for a given t, n. However, if the downloads within the cooperative group are less costly compared to the downloads from the live nodes, then delaying repairs would be beneficial in reducing the total cost. We will revisit this analysis for codes having $n > d + t$ in the next subsection.

D. General code construction for secure MBCR

The code construction presented in Section III-B has the requirement of $d = n - t$. However, for practical systems, it may not be possible that a failed node connects to all the remaining nodes. This brings the necessity of code constructions for $d < n - t$. Remarkably, for a fixed $(n, k, d, \mathcal{M})$, increasing t can reduce the repair bandwidth in the secrecy scenario we consider here. This is reported in [16] for DSS without secrecy constraints. Hence, for a fixed d, delaying the repairs can be advantageous, e.g., when there is a limit on the number of live nodes that can be contacted for a node repair. In the following, we present a general construction which works for any parameters, in particular for $n > d + t$.

The construction is based on the MBCR code proposed in [21]. In [21], a bivariate polynomial is constructed using $\mathcal{M} = k(2d + t - k)$ message symbols (over $\mathbb{F}_q$) as the coefficients of the polynomial:

$$F(Y, Z) = \sum_{\substack{0 \leq i < k, \\ 0 \leq j < k}} a_{ij} Y^i Z^j + \sum_{\substack{0 \leq i < k, \\ k \leq j < d+t}} b_{ij} Y^i Z^j + \sum_{\substack{k \leq i < d, \\ 0 \leq j < k}} c_{ij} Y^i Z^j \tag{15}$$

Here, $\{a_{ij}\}$, $\{b_{ij}\}$, and $\{c_{ij}\}$ denote $\mathcal{M}$ message symbols. Given $q > n$, two set of n distinct points, $\{y_1, y_2, \ldots, y_n\}$ and $\{z_1, z_2, \ldots, z_n\}$, are chosen. The ith node in the DSS store the following $2d + t - 1$ evaluations of polynomial $F(Y, Z)$.

$$\begin{aligned} &F(y_i, z_i), F(y_i, z_{i\oplus 1}), \ldots, F(y_i, z_{i\oplus(d+t-1)}) \\ &F(y_{i\oplus 1}, z_i), F(y_{i\oplus 2}, z_i), \ldots, F(y_{i\oplus(d-1)}, z_i), \end{aligned} \tag{16}$$

$F(y_1,z_1)$	$F(y_1,z_2)$	$\cdots$	$F(y_1,z_{\ell_1})$	$\cdots$	$F(y_1,z_{d+t})$			
$F(y_2,z_1)$	$F(y_2,z_2)$	$\cdots$	$F(y_2,z_{\ell_1})$	$\cdots$	$F(y_2,z_{d+t})$	$F(y_2,z_{d+t+1})$		
$\cdots$								
$F(y_{\ell_1},z_1)$	$F(y_{\ell_1},z_2)$	$\cdots$	$F(y_{\ell_1},z_{\ell_1})$	$\cdots$	$F(y_{\ell_1},z_{d+t})$	$F(y_{\ell_1},z_{d+t+1})$	$\cdots$	$F(y_{\ell_1},z_{d+t-1+\ell_1})$
$\cdots$								
$F(y_d,z_1)$	$F(y_d,z_2)$	$\cdots$	$F(y_d,z_{\ell_1})$					
	$F(y_{d+1},z_2)$	$\cdots$	$F(y_{d+1},z_{\ell_1})$					
		$\cdots$	$\cdot$					
			$F(y_{d+\ell_1-1},z_{\ell_1})$					

Fig. 7: Observed symbols at the eavesdroppers for a given ℓ_1.

where $\oplus$ denotes addition modulo n. The first $d+t$ evaluations at node i can be seen as the evaluations of the univariate polynomial $f_i(Z) = F(y_i, Z)$ of degree at most $d+t-1$ at $d+t$ points. This uniquely defines the polynomial $f_i(Z)$. Similarly, the first evaluation in (16), $F(y_i, z_i)$, along with last $d-1$ evaluations, $F(y_{i\oplus 1}, z_i), \ldots, F(y_{i\oplus(d-1)}, z_i)$, uniquely define the univariate polynomial $g_i(Y) = F(Y, z_i)$ of degree at most $d-1$. This property of the proposed bivariate polynomial based coding scheme is utilized for the exact node repair and data reconstruction processes at the MBCR point. (We refer to [21] for details.)

In order to get an $(\ell_1, 0)$-secure code at the MBCR point, we rewrite the polynomial in (15) as follows:

$$\begin{aligned} F(Y,Z) = &\sum_{\substack{0\le i<\ell_1,\\ 0\le j<\ell_1}} a_{ij}Y^iZ^j + \sum_{\substack{0\le i<\ell_1,\\ \ell_1\le j<k}} a_{ij}Y^iZ^j \\ &+ \sum_{\substack{\ell_1\le i<k,\\ 0\le j<\ell_1}} a_{ij}Y^iZ^j + \sum_{\substack{\ell_1\le i<k,\\ \ell_1\le j<k}} a_{ij}Y^iZ^j \\ &+ \sum_{\substack{0\le i<\ell_1,\\ k\le j<d+t}} b_{ij}Y^iZ^j + \sum_{\substack{\ell_1\le i<k,\\ k\le j<d+t}} b_{ij}Y^iZ^j \\ &+ \sum_{\substack{k\le i<d,\\ 0\le j<\ell_1}} c_{ij}Y^iZ^j + \sum_{\substack{k\le i<d,\\ \ell_1\le j<k}} c_{ij}Y^iZ^j \end{aligned} \tag{17}$$

Next, we choose $\ell_1^2 + \ell_1(k-\ell_1) + (k-\ell_1)\ell_1 + \ell_1(d+t-k) + (d-k)\ell_1 = \ell_1(2d+t-\ell_1)$ coefficients of $F(Y,Z)$, $a_{0:\ell_1-1,0:\ell_1-1} \cup a_{0:\ell_1-1,\ell_1:k-1} \cup a_{\ell_1:k-1,0:\ell_1-1} \cup b_{0:\ell_1-1,k:d+t-1} \cup c_{k:d-1,0:\ell_1-1}$, to be symbols drawn uniformly at random from $\mathbb{F}_q$ in an i.i.d. manner. Remaining $k(2d+t-k) - \ell_1(2d+t-\ell_1) = \mathcal{M}^s$ coefficients of $F(Y,Z)$ are chosen to be the data symbols $\mathbf{f}^s$ that need to be stored on the DSS. Each node $i \in [n]$ stores the evaluation of $F(Y,Z)$ as illustrated in (16). It follows from the description of the coding scheme of [21] in the beginning of this subsection that the resulting coding scheme is an exact repairable code at the MBCR point.

Next, we show that the proposed scheme is indeed $(\ell_1, 0)$-secure. Let $\mathbf{e}$, $\mathbf{f}^s$, and $\mathbf{r}$ denote the data observed by an eavesdropper, the original data to be stored, and the randomness added to the original data before encoding, respectively. It is sufficient to show (i) $H(\mathbf{e}) \le H(\mathbf{r})$ and (ii) $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$ in order to establish the secrecy claim (see Lemma 3). To argue the first requirement, noting that number of eavesdropped symbols are $\ell_1\alpha = \ell_1(2d+t-1)$, we will show that $\ell_1^2 - \ell_1$ number of these are linearly dependent on the remaining ones. The eavesdropper, without loss of generality considering the first ℓ_1 nodes as eavesdropped nodes, observes the symbols given in Fig. 7. Due to the code construction, each row above represents evaluations of a polynomial of degree less than $d+t$ and each column represents a polynomial of degree less than d. Hence, we observe that each of the symbols denoted with a colored font in the matrix of Fig. 7 is a linear combination of the remaining ones. Therefore, $H(\mathbf{e}) \le \ell_1\alpha - \ell_1(\ell_1-1) = H(\mathbf{r})$.

In order to show that second requirement also holds, we present a method to decode randomness $\mathbf{r}$ given $\mathbf{f}^s$ and data stored on any ℓ_1 nodes. Once we know the data symbols $\mathbf{f}^s$, we can remove the monomials associated to data symbols in $F(Y,Z)$ and the contribution of these monomials from the polynomial evaluations stored on DSS. Let $\widehat{F}(Y,Z)$ denote the bivariate polynomial that we obtain by removing the data monomials:

$$\begin{aligned} \widehat{F}(Y,Z) = &\sum_{\substack{0\le i<\ell_1,\\ 0\le j<\ell_1}} a_{ij}Y^iZ^j + \sum_{\substack{0\le i<\ell_1,\\ \ell_1\le j<k}} a_{ij}Y^iZ^j \\ &+ \sum_{\substack{\ell_1\le i<k,\\ 0\le j<\ell_1}} a_{ij}Y^iZ^j + \sum_{\substack{0\le i<\ell_1,\\ k\le j<d+t}} b_{ij}Y^iZ^j \\ &+ \sum_{\substack{k\le i<d,\\ 0\le j<\ell_1}} c_{ij}Y^iZ^j \end{aligned} \tag{18}$$

$\widehat{F}(Y,Z)$ can be rewritten as:

$$\widehat{F}(Y,Z)=\sum_{\substack{0\le i<\ell_1,\\ 0\le j<\ell_1}}\hat{a}_{ij}Y^iZ^j+\sum_{\substack{0\le i<\ell_1,\\ \ell_1\le j<d+t}}\hat{b}_{ij}Y^iZ^j+\sum_{\substack{\ell_1\le i<d,\\ 0\le j<\ell_1}}\hat{c}_{ij}Y^iZ^j \tag{19}$$

where $\hat{a}_{0:\ell_1-1,0:\ell_1-1}=a_{0:\ell_1-1,0:\ell_1-1},\hat{b}_{0:\ell_1-1,\ell_1:k-1}=a_{0:\ell_1-1,\ell_1:k-1},\hat{b}_{0:\ell_1-1,k:d+t-1}=b_{0:\ell_1-1,k:d+t-1},\hat{c}_{\ell_1:k-1,0:\ell_1-1}=a_{\ell_1:k-1,0:\ell_1-1},\hat{c}_{k:d-1,0:\ell_1-1}=c_{k:d-1,0:\ell_1-1}$.

$\widehat{F}(Y,Z)$ in (19) takes the same form as $F(Y,Z)$ in (15) with k replaced with ℓ_1. Now, the randomness $\mathbf{r}$, coefficients of $\widehat{F}(Y,Z)$ in (19), can be decoded from the data observed on ℓ_1 nodes using the data reconstruction method described in [21]. Thus, we obtain the following result.

Proposition 8. *The secrecy capacity at the MBCR point for a file size of $\mathcal{M}=k(2d-k+t)$ is given by $\mathcal{M}^s=k(2d-k+t)-\ell_1(2d-\ell_1+t)$ for any $n\ge d+t$.*

We list some instances of this construction in Table II in Appendix C. As evident from the table, cooperation helps to reduce the repair bandwidth if $d<n-t$. Thus, (secure) coding schemes for the case of $n>d+t$ are of significant interest in order to reduce the repair bandwidth in cooperative repair.

IV. Secure MSCR Codes

We first consider an upper bound on the secure file size for DSS employing minimum storage cooperative regenerating (MSCR) codes. We then utilize appropriate secrecy pre-coding mechanisms to construct achievable schemes for the upper bound.

A. Upper bound on the secure file size

At the MSCR point, nodes have minimum possible storage, i.e., $\alpha=\frac{\mathcal{M}}{k}$. Using the cut-set analysis given in Section II, one can obtain that the minimum repair bandwidth can be attained with $\beta=\beta'=\frac{\alpha}{d-k+t}=\frac{\mathcal{M}}{k(d-k+t)}$. (See also [13], [15].) Therefore, the amount of data downloaded during a node repair is larger than per node storage α at the MSCR point. Keeping this in mind, we consider two kinds of eavesdropped nodes for security constraints: storage-eavesdropped nodes ($\mathcal{E}_1$) and download-eavesdropped nodes ($\mathcal{E}_2$). Using the size of these sets we denote the eavesdropper setting with (ℓ_1,ℓ_2) as introduced in Section II. Here, for a node in $\mathcal{E}_2$, an eavesdropper observes both the data downloaded from live nodes and from cooperating nodes (other nodes in the same repair group).

Similar to the analysis given in Section III-A, and utilizing Lemma 4, we obtain the following bound on the secure file size at the MSCR point.

$$\mathcal{M}^s\le\sum_{i=0}^{k-1}(1-l_1^i-l_2^i)\times\min\Big\{\alpha-I(\mathbf{s}_i;\mathbf{d}_{i,\mathcal{E}_2}),(d-i)\beta+(t-1)\beta'\Big\}, \tag{20}$$

where $\mathbf{s}_i$ and $\mathbf{d}_{i,\mathcal{E}_2}$ denote the data stored on node i and data downloaded from node i for repair of nodes in set $\mathcal{E}_2$, respectively. Here, for ith repair group, we consider $n_i=1$ number of nodes to be contacted by DC. We assume that we have l_1^i number of storage-eavesdropped nodes (from $\mathcal{E}_1$) and l_2^i number of download-eavesdroppers (from $\mathcal{E}_2$) in repair group i. Compared to the MBCR bounds, due to eavesdroppers in $\mathcal{E}_2$, nodes that are not eavesdropped may leak information during their participation in the repair of a node having an download-observing eavesdropper. Thus, the values of the cuts of type 1 include additional penalty terms $I(\mathbf{s}_i;\mathbf{d}_{i,\mathcal{E}_2})$, counting the leakage from the storage at the ith node to nodes indexed with $\mathcal{E}_2$. (That is, we consider a loose bound compared to that of (5), i.e., $H(\mathbf{s}_i|\mathbf{s}_1,\cdots,\mathbf{s}_{i-1},\mathbf{s}_{\mathcal{E}_1},\mathbf{d}_{\mathcal{E}_2})=H(\mathbf{s}_i)-I(\mathbf{s}_i;\mathbf{s}_1,\cdots,\mathbf{s}_{i-1},\mathbf{s}_{\mathcal{E}_1},\mathbf{d}_{\mathcal{E}_2})\le\alpha-I(\mathbf{s}_i;\mathbf{d}_{i,\mathcal{E}_2})$, as $\mathbf{d}_{i,\mathcal{E}_2}\subset\mathbf{d}_{\mathcal{E}_2}$.) Considering the MSCR point values of α, β, and β' given above, the second term inside each $\min\{\}$ in (20) is larger than the first term. Hence, considering that the first $k-\ell_1-\ell_2$ repairs are eavesdropper-free, (20) evaluates to the following bound.

Proposition 9. *Cooperative regenerating codes operating at the MSCR point with a secure file size of $\mathcal{M}^s$ satisfy*

$$\mathcal{M}^s\le\sum_{i=0}^{k-\ell_1-\ell_2-1}\alpha-I(\mathbf{s}_i;\mathbf{d}_{i,\mathcal{E}_2}),$$

where the MSCR point is given by $\beta=\beta'$, $\alpha=(d-k+t)\beta$, for a file size of $\mathcal{M}=k(d-k+t)\beta$. In addition, at the MSCR point, one can bound $I(\mathbf{s}_i;\mathbf{d}_{i,\mathcal{E}_2})\ge\beta'=\beta$ and obtain the bound

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

B. Code construction for secure MSCR when $k = t = 2$

We consider an interference alignment approach based on the one proposed in [19] with $k = t = 2$. For any (n, k, d, t) with $d \geq k$ and $n = d + t$, we have $\alpha = d - k + t = n - 2$ and $\mathcal{M} = k(d - k + t) = 2(d - k + t) = 2\alpha$ at the normalized MSCR point, i.e., $\beta = \beta' = 1$. (The general case of $\beta > 1$ can be obtained by a parallel utilization of independent codes with $\beta = 1$.) For $k = 2$, the bound given in Proposition 9 implies that the achievability of positive secure file size in the presence of an eavesdropper is possible only when $(\ell_1, \ell_2) = (1, 0)$ or $(\ell_1, \ell_2) = (0, 1)$. Corresponding bounds are given by $\mathcal{M}^s \leq \alpha$ and $\mathcal{M}^s \leq \alpha - 1$, respectively. (For the bound corresdpoing to $\ell_2 = |\mathcal{E}_2| = 1$, $\mathbf{d}_{i,\mathcal{E}_2}$ necessarily consists of one symbol as $\beta = \beta' = 1$, and the non-eavesdropped node participates in the repair of the eavesdropped node by sending β or β' symbols. Note that DC contacts only $k = 2$ nodes, and one of them is eavesdropped for the purpose of obtaining the upper bound here.) In the following, we construct codes achieving the stated bounds for both cases, hence establishing the secrecy capacity when $k = t = 2$. We show this with codes having $n = d + t$, i.e., all the surviving nodes participate in a node repair. The construction can be extended to cases with $n > d + t$ by following a similar approach and choosing a larger field size.

Case 1: $\mathcal{M}^s = \alpha$ when $(\ell_1, \ell_2) = (1, 0)$

Consider a large enough finite field $\mathbb{F}_q$ (conditions on the required field will be given later) which has ω as its generator, α number of random symbols $r_1, \cdots, r_\alpha$, and α number of secure information symbols $s_1, \cdots, s_\alpha$. Both information and random symbols are uniformly distributed over the field. We construct a file $\mathbf{f} = (a_1, \ldots, a_\alpha, b_1, \ldots, b_\alpha) = (r_1, \ldots, r_\alpha, r_1 + s_1, \ldots, r_\alpha + s_\alpha)$ of size $\mathcal{M}$ over $\mathbb{F}_q$. Our coding scheme is described by the following placement.

- Store $\mathbf{a} = (a_1, \cdots, a_\alpha)$ at the first node,
- Store $\mathbf{b} = (b_1, \cdots, b_\alpha)$ at the second node, and
- Store $\mathbf{r}_i = (a_1 + \omega^{(i-1) \bmod \alpha} b_1, \cdots, a_\alpha + \omega^{(i+\alpha-2) \bmod \alpha} b_\alpha)$ at ith parity node, $i \in \{1, \cdots, \alpha = n - 2\}$.

Note that we implement one-time pad scheme of Vernam [26] for the symbols represented as b_i in this construction. We represent the stored symbols of ith parity node as $\mathbf{r}_i^T = \mathbf{a}^T + \mathbf{B}_i \mathbf{b}^T$, where $\mathbf{B}_i$ is a diagonal matrix with its diagonal elements given by $\{\omega^{(i-1) \bmod \alpha}, \ldots, \omega^{(i+\alpha-2) \bmod \alpha}\}$. Data collector DC can reconstruct the file $\mathbf{f}$ by contacting to any of the $k = 2$ nodes, and solving α groups of 2 equations over 2 unknowns. (Each group gives 2 equations over 2 unknowns. For example, the first symbols for the systematic nodes are $a_1 = r_1$ and $b_1 = r_1 + s_1$, and these two equations can be solved to obtain (r_1, s_1).) From file $\mathbf{f}$, it can then obtain the secure symbols $s_1, \cdots, s_\alpha$. This establishes data reconstruction property of the code. Cooperative repair process is similar to that of the code proposed in [19], as summarized in the following.

Without loss of generality, we consider cooperative repair of two systematic nodes. Repairs of stages involving parity nodes can be performed as that of the systematic nodes after change of variables [19]. The first systematic node downloads $\mathbf{v}_{1,i} \mathbf{r}_i^T = \mathbf{v}_{1,i} \mathbf{a}^T + \mathbf{z}\mathbf{b}^T$ from ith parity node which stores $\mathbf{r}_i^T = \mathbf{a}^T + \mathbf{B}_i \mathbf{b}^T$. Here, $\mathbf{v}_{1,i} = \mathbf{z}\mathbf{B}_i^{-1}$ and $\mathbf{z} \triangleq (1, \cdots, 1)$. After this step, the first systematic node has the symbols

$$\mathbf{c}_1 = \{\mathbf{z}\mathbf{B}_1^{-1}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T, \cdots, \mathbf{z}\mathbf{B}_d^{-1}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T\}.$$

Note that the repair process is such that the interference is aligned by having terms $\mathbf{z}\mathbf{b}^T$ in each of the symbols downloaded by the first systematic node. The second systematic node obtains $d = n - 2$ symbols $\mathbf{c}_2 = \{\mathbf{v}_{2,1}\mathbf{r}_1^T, \cdots, \mathbf{v}_{2,d}\mathbf{r}_d^T\}$ from d parity nodes. Here, the repair process is such that the interference is aligned by having terms $\mathbf{z}\mathbf{a}^T$ in each of the symbol in $\mathbf{c}_2$. Accordingly, we have $\mathbf{v}_{2,i} = \mathbf{z}$, which gives $\mathbf{v}_{2,i}\mathbf{r}_i^T = \mathbf{z}\mathbf{a}^T + \mathbf{z}\mathbf{B}_i\mathbf{b}^T$ and

$$\mathbf{c}_2 = \{\mathbf{z}\mathbf{a}^T + \mathbf{z}\mathbf{B}_1\mathbf{b}^T, \cdots, \mathbf{z}\mathbf{a}^T + \mathbf{z}\mathbf{B}_d\mathbf{b}^T\}.$$

Now, the second systematic node chooses the repair vector $\mathbf{v}_{1,0} = (\omega^0 + \cdots + \omega^{\alpha-1})^{-1}\mathbf{z}$ and sends $\mathbf{v}_{1,0}\mathbf{c}_2^T = \nu\mathbf{v}_{1,0}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T$ to the first systematic node. Here ν denotes the sum of α ones over $\mathbb{F}_q$, which depends on the characteristics of the field $\mathbb{F}_q$. Then, the first systematic node solves $d+1$ equations $\{\nu\mathbf{v}_{1,0}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T, \mathbf{v}_{1,1}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T, \cdots, \mathbf{v}_{1,d}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T\}$ in $d+1$ unknowns $\{a_1, \cdots, a_\alpha, \mathbf{z}\mathbf{b}^T\}$ [19]. This follows if the matrix

$$\mathbf{M} = [\nu\mathbf{v}_{1,0}, 1; \mathbf{v}_{1,1}, 1; \cdots; \mathbf{v}_{1,d}, 1]$$

is invertible, when we represent the observed symbols at the first systematic node as $\mathbf{M}[a_1, \cdots, a_\alpha, \mathbf{z}\mathbf{b}^T]$. (Invertibility of this matrix is stated in [19]. Our analysis shows that this holds if $q > n - 1$ is a sufficiently large prime number such that the generator w of $\mathbb{F}_q$ used above satisfies $(\omega^0 + \cdots + \omega^{\alpha-1})^2 w^{-(\alpha-1)} \notin \{0, \alpha^2\}$. Details are omitted for brevity.) The second systematic node can be repaired in a similar manner. It remains to show the secrecy of the file. Here, regardless of eavesdropped node being a systematic or a parity node, given the secure symbols, $\mathbf{f}^s = \{s_1, \cdots, s_\alpha\}$, the eavesdropper can obtain α equations in α unknowns $\mathbf{r} = r_1, \cdots, r_\alpha$. This allows the eavesdropper to solve for $\mathbf{r}$, and shows that $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$, where the eavesdropper observes the content of the eavesdropped node, i.e., $\mathbf{e} = \mathbf{s}_{\mathcal{E}_1}$. We see that, at the eavesdropped node, the content of the stored data necessarily satisfies $H(\mathbf{e}) = H(\mathbf{s}_{\mathcal{E}_1}) = \alpha$. Then, as the code satisfies both $H(\mathbf{e}) \leq H(\mathbf{r})$ and $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$, we obtain from Lemma 3 that $I(\mathbf{f}^s; \mathbf{e}) = I(s_1, \cdots, s_\alpha; \mathbf{s}_{\mathcal{E}_1}) = 0$.

Case 2: $\mathcal{M}^s = \alpha - 1$ when $(\ell_1, \ell_2) = (0, 1)$

We modify the above construction by considering the file given by $\mathcal{M} = \{a_1 \triangleq r_1, \cdots, a_\alpha \triangleq r_\alpha, b_1 \triangleq r_1 + s_1, \cdots, b_{\alpha-1} \triangleq r_{\alpha-1} + s_{\alpha-1}, b_\alpha \triangleq r_{\alpha+1}\}$. The reconstruction and cooperative repair processes are the same as that of the previous case. We

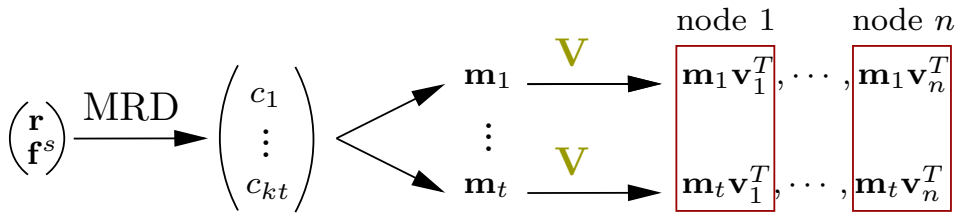

Fig. 8: MRD codeword is $\mathbf{c}_{1:kt}$. MDS input vector is $\mathbf{m}_j = (\mathbf{c}_{(j-1)k+1}, \ldots, \mathbf{c}_{jk})$ for $j = 1, \cdots, t$. Vandermonde matrix is represented with $\mathbf{V}$, columns of which are $\{\mathbf{v}_1^T, \cdots, \mathbf{v}_n^T\}$. The placement of the symbols on different nodes is shown on the right.

show that the secrecy constraint is satisfied here. The content of the eavesdropped node $\mathbf{s}_{\mathcal{E}_2}$ is generated from the downloaded data $\mathbf{d}_{\mathcal{E}_2}$. Thus, we need to show $I(\mathbf{f}^s; \mathbf{e}) = 0$ with $\mathbf{f}^s = \{s_1, \cdots, s_{\alpha-1}\}$ and $\mathbf{e} = \mathbf{d}_{\mathcal{E}_2}$. Without loss generality, we assume that the eavesdropper observes the first systematic node. Considering the repair process described above, we have $\mathbf{e} = \mathbf{d}_{\mathcal{E}_2} = \{\mathbf{v}_{1,0}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T, \mathbf{v}_{1,1}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T, \cdots, \mathbf{v}_{1,d}\mathbf{a}^T + \mathbf{z}\mathbf{b}^T\}$, from which we obtain that $H(\mathbf{e}) \leq \alpha + 1$. In addition, as the eavesdropper can solve for $(\mathbf{a}, \mathbf{z}\mathbf{b}^T)$, it can solve for $\mathbf{r} = \{r_1, \cdots, r_{\alpha+1}\}$ from the $\alpha + 1$ number of equations in $(\mathbf{a}, \mathbf{z}\mathbf{b}^T)$, after canceling out the contribution of secure symbols $\mathbf{f}^s = \{s_1, \cdots, s_{\alpha-1}\}$ from $\mathbf{z}\mathbf{b}^T$. This shows that $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$. Using this, together with $H(\mathbf{r}) = \alpha + 1$ and Lemma 3, we obtain that $I(\mathbf{f}^s; \mathbf{e}) = I(s_1, \cdots, s_{\alpha-1}; \mathbf{d}_{\mathcal{E}_2}) = 0$.

As a result of the above construction of secure MSCR codes with $k = t = 2$, we obtain the following result.

Proposition 10. *The secrecy capacity at the MSCR point for a file size of $\mathcal{M} = k(d - k + t)\beta$ is given by $\mathcal{M}^s = \alpha\beta$, if $(\ell_1, \ell_2) = (1, 0)$ and $k = t = 2$; and by $\mathcal{M}^s = (\alpha - 1)\beta$, if $(\ell_1, \ell_2) = (0, 1)$ and $k = t = 2$.*

C. Code construction for secure MSCR when $d = k$

The above construction is limited to the $k = 2$ case. Here, we provide secure MSCR code when $d = k$, and hence allowing $k > 2$. (Note that as $d \geq k > \ell_1 + \ell_2$, we necessarily have $\ell_1 + \ell_2 < d = k$ here.) Here, we apply a two-stage encoding, where we utilize an MRD code for pre-coding to achieve secrecy.

Consider $\mathcal{M} = k(d-k+t) = kt$, $\beta = \beta' = 1$, $\alpha = d-k+t = t$, $\mathcal{M}^s = (k-\ell_1-\ell_2)(t-\ell_2) = kt-(\ell_1+\ell_2)t-\ell_2(k-\ell_1-\ell_2)$, and $n \geq d+t$. (The general case of $\beta > 1$ can be obtained by a parallel utilization of independent codes with $\beta = 1$.) We encode the data using the linearized polynomial $f(g) = \sum_{i=0}^{\mathcal{M}-1} u_i g^{q^i}$. (This is the Gabidulin construction of MRD codes summarized in Section II-E.) The coefficients of $f(g)$ is chosen by $\mathcal{M} - \mathcal{M}^s$ number of random symbols denoted by $\mathbf{r}$ and $\mathcal{M}^s$ data symbols denoted by $\mathbf{f}^s$. The function $f(g)$ is evaluated at $\mathcal{M}$ points in $\mathbb{F}_{q^m}$, $\{g_1, \ldots, g_{\mathcal{M}}\}$, that are linearly independent over $\mathbb{F}_q$. (Here, the data and random symbols belong to $\mathbb{F}_{q^m}$ with $m \geq \mathcal{M}$.) We denote these evaluations as $c_i = f(g_i)$ for $i = 1, \cdots, \mathcal{M} = kt$. We consider the code provided in [17] for the secrecy setting here. We place these $\mathcal{M} = kt$ symbols into vectors $\mathbf{m}_1, \cdots, \mathbf{m}_t$, each having k symbols. We encode these vectors with a Vandermonde matrix of size $k \times n$, whose columns are represented as $\mathbf{v}_i^T$ for $i = 1, \cdots, n$. We store $\{\mathbf{m}_1\mathbf{v}_i^T, \ldots, \mathbf{m}_t\mathbf{v}_i^T\}$ at node i. (See Fig. 8.)

Data collector DC, by contacting any k nodes, can obtain k equations for each of $\mathbf{m}_j$, and solve them to obtain c_i for $i = 1, \cdots, \mathcal{M} = kt$. It can then obtain the secure data symbols by performing interpolation for the underlying linearized polynomial $f(g)$ [46]. Next, we briefly describe the cooperative node repair process for the codes under consideration. For node repair, consider that node $j_l \in \{j_1, j_2, \ldots, j_t\}$ contacts $d = k$ live nodes, referred to as $\{i_1, i_2, \ldots, i_k\}$. It downloads $\mathbf{m}_l\mathbf{v}_{i_r}^T$ from live node i_r for $r = 1, \cdots, k$. Node j_l then obtains $\mathbf{m}_l$ by solving these k equations. It stores $\mathbf{m}_l\mathbf{v}_{j_l}^T$, and sends $\mathbf{m}_l\mathbf{v}_{j_{l'}}^T$ to node $j_{l'} \in \{j_1, j_2, \ldots, j_t\}, l' \neq l$, i.e., the remaining nodes under repair. Each node $j_l \in \{j_1, j_2, \ldots, j_t\}$ repeats this procedure. As a result, node j_l recovers its $\mathbf{m}_{l'}\mathbf{v}_{j_l}^T$ for $l' \in [1:t], l' \neq l$ by downloading a symbol from each node under repair.

We show the secrecy constraint has met assuming $\ell_2 \leq t$ here. (Otherwise, this construction can not achieve a positive secure file size as an eavesdropper obtains all $\mathbf{m}_{1:t}$ symbols from download-eavesdropped nodes.) We note that an eavesdropper obtain $\ell_2 k$ equations from the $d = k$ live nodes by observing repair of ℓ_2 nodes (the observed symbols reveal ℓ_2 number of $\mathbf{m}_j$s), and an additional $\ell_2(t - \ell_2) = \ell_2(\alpha - \ell_2)$ symbols from the remaining nodes under repair. Besides this, the eavesdropper gets $\ell_1\alpha$ number of symbols from the content stored on ℓ_1 storage-eavesdropped nodes. However, $\ell_1\ell_2$ of these symbols are linearly dependent to the ones downloaded by nodes in $\mathcal{E}_2$ (as the nodes in $\mathcal{E}_2$ recover ℓ_2 number of $\mathbf{m}_j$s during node repair). (See Fig. 9.) Therefore, using the given polynomial and the secure data of length $\mathcal{M}^s$, the eavesdropper can solve for the random symbols using these $\ell_2(k+\alpha-\ell_2)+\ell_1(\alpha-\ell_2) = \ell_2(k+t-\ell_2)+\ell_1(t-\ell_2) = (k-\ell_1-\ell_2)\ell_2+(\ell_1+\ell_2)t = \mathcal{M}-\mathcal{M}^s$ linearly independent evaluations of the polynomial $f(g)$. This implies that we have $H(\mathbf{r}|\mathbf{f}^s, \mathbf{e}) = 0$, where $\mathbf{e}$ denotes the observations of the eavesdropper associated with $\mathcal{E}_1$ and $\mathcal{E}_2$. This construction also satisfies $H(\mathbf{e}) = \ell_2 k + \ell_2(\alpha - \ell_2) + \ell_1(\alpha - \ell_2) = H(\mathbf{r})$ as argued above, and it follows from Lemma 3 that we have $I(\mathbf{f}^s; \mathbf{e}) = 0$. Thus, the proposed coding scheme achieves the secure file size of $kt - (\ell_1 + \ell_2)t - \ell_2(k - \ell_1 - \ell_2)$ when $\ell_2 \leq t$; consequently, we have the following.

$$
\begin{array}{l|cccc|ccc|cccc}
\text{node } 1 & \mathbf{m}_1\mathbf{v}_1^T & \mathbf{m}_2\mathbf{v}_1^T & \cdots & \mathbf{m}_{\ell_2}\mathbf{v}_1^T & \mathbf{m}_{\ell_2+1}\mathbf{v}_1^T & \cdots & \mathbf{m}_t\mathbf{v}_1^T & \mathbf{m}_1\mathbf{v}_{1(1)}^T & \mathbf{m}_1\mathbf{v}_{1(2)}^T & \cdots & \mathbf{m}_1\mathbf{v}_{1(k)}^T \\
 & \vdots & & & & & & & & & & \\
 & \mathbf{m}_1\mathbf{v}_{\ell_2}^T & \mathbf{m}_2\mathbf{v}_{\ell_2}^T & \cdots & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2}^T & \mathbf{m}_{\ell_2+1}\mathbf{v}_{\ell_2}^T & \cdots & \mathbf{m}_t\mathbf{v}_{\ell_2}^T & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2(1)}^T & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2(2)}^T & \cdots & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2(k)}^T \\
 & \mathbf{m}_1\mathbf{v}_{\ell_2+1}^T & \mathbf{m}_2\mathbf{v}_{\ell_2+1}^T & \cdots & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2+1}^T & \mathbf{m}_{\ell_2+1}\mathbf{v}_{\ell_2+1}^T & \cdots & \mathbf{m}_t\mathbf{v}_{\ell_2+1}^T & & & & \\
 & \vdots & & & & & & & & & & \\
\text{node } \ell_2+\ell_1 & \mathbf{m}_1\mathbf{v}_{\ell_2+\ell_1}^T & \mathbf{m}_2\mathbf{v}_{\ell_2+\ell_1}^T & \cdots & \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2+\ell_1}^T & \mathbf{m}_{\ell_2+1}\mathbf{v}_{\ell_2+\ell_1}^T & \cdots & \mathbf{m}_t\mathbf{v}_{\ell_2+\ell_1}^T & & & &
\end{array}
$$

Fig. 9: The observed symbols at the eavesdropper. Without loss of generality we assume first ℓ_2 nodes belong to $\mathcal{E}_2$ and the following ℓ_1 nodes belong to $\mathcal{E}_1$. We denote the symbols downloaded at $i \in \mathcal{E}_2$ from live node $x_{i(j)}$ as $\mathbf{m}_i\mathbf{v}_{i(j)}^T$ for $j = [1:d]$, where $i(j)$ denotes the jth contacted live node for repair of node i. For the nodes in $\mathcal{E}_2$, we indicate these downloaded symbols on the right hand side of the figure, within the box with solid lines. (On the left hand side, we have the stored content of each node.) From downloaded symbols to $\mathcal{E}_2$ nodes, the eavesdropper observes $\mathbf{d}_{\mathcal{E}_2} = \{\mathbf{m}_i\mathbf{v}_{i(j)}^T : i = 1, \cdots, \ell_2; j = 1, \cdots, k\}$. (Here, $d = k$.) The symbols in the first ℓ_2 columns, i.e., $\{\mathbf{m}_1\mathbf{v}_1^T, \ldots, \mathbf{m}_1\mathbf{v}_{\ell_2+\ell_1}^T, \cdots, \mathbf{m}_{\ell_2}\mathbf{v}_1^T, \ldots, \mathbf{m}_{\ell_2}\mathbf{v}_{\ell_2+\ell_1}^T\}$, are functions of the symbols in $\mathbf{d}_{\mathcal{E}_2}$ due to the construction (i.e., MDS coding). These $\ell_2 k$ symbols in $\mathbf{d}_{\mathcal{E}_2}$ in addition to the $(\ell_1+\ell_2)(t-\ell_2)$ symbols located in the middle, i.e., $\{\mathbf{m}_{\ell_2+1}\mathbf{v}_1^T, \ldots, \mathbf{m}_{\ell_2+1}\mathbf{v}_{\ell_2+\ell_1}^T, \cdots, \mathbf{m}_t\mathbf{v}_1^T, \ldots, \mathbf{m}_t\mathbf{v}_{\ell_2+\ell_1}^T\}$, correspond to $\ell_2 k + (\ell_1+\ell_2)(t-\ell_2)$ linearly independent evaluations of the linearized polynomial $f(g)$.

Proposition 11. *The secure file size of $\mathcal{M}^s = (k-\ell_1-\ell_2)[t-\ell_2]^+\beta$ is achievable at the MSCR point for a file size of $\mathcal{M} = k(d-k+t)\beta$ when $d = k$.*

Note that this achieves the secrecy capacity when $\ell_2 \leq 1$ for any ℓ_1 as can be observed from the bound given by Proposition 9.

V. Conclusion

Distributed storage systems (DSS) store data on multiple nodes. These systems not only require resilience against node failures, but also have to satisfy security constraints and to perform multiple node repairs. Regenerating codes proposed for DSS address the node failure resilience while efficiently trading off storage vs. repair bandwidth. In this paper, we considered secure cooperative regenerating codes for DSS against passive eavesdropping attacks. 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 stored content at any ℓ_1 nodes, and the downloaded content at any ℓ_2 nodes are leaked to the eavesdropper. With such an eavesdropper model, we studied the security for the multiple repair scenario, in particular secure cooperative regenerating codes. For the minimum bandwidth cooperative regenerating (MBCR) point, we established a bound on the secrecy capacity, and by modifying the existing coding schemes in the literature, devised new codes achieving the secrecy capacity. For the minimum storage cooperative regenerating (MSCR) point, on the other hand, we proposed an upper bound and lower bounds on the secure file size, which match under special cases. The results show that it is possible to design regenerating codes that not only efficiently trades storage for repair bandwidth, but are also resilient against security attacks in a cooperative repair scenario. Finally, as evident from some of our secrecy-achieving constructions, we would like to emphasize the role that the maximum rank distance (MRD) codes can take in secrecy problems. In particular, we have utilized the Gabidulin construction [46] of MRD codes and properties of linearized polynomials in obtaining some of the results. Similar properties of such codes have been utilized to achieve secrecy in earlier works [53]–[56], and they proved their potential again here as an essential component for achieving secrecy in DSS. (See also [38] for utilization of these codes in active eavesdropper settings, and [28] for constructing locally repairable codes with and without secrecy constraints.)

We list some avenues for further research here. The secrecy capacity of MSCR codes remains as an open problem, as we have established the optimal codes under some parameter settings. To attempt solving this problem, codes for MSCR without security constraints have to be further investigated. One can also consider cooperative repair in a DSS having locally repairable structure. As other distributed systems, DSS 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.

Acknowledgement

The authors would like to thank the reviewers for their insightful comments and suggestions.

APPENDIX A
PROOF OF LEMMA 3

Proof: The proof follows from the classical techniques given by [25], where instead of 0-leakage, ϵ-leakage rate is considered. The application of this technique in DSS is considered in [27], as summarized below. We have

$$\begin{aligned} I(\mathbf{f}^s;\mathbf{e}) &= H(\mathbf{e}) - H(\mathbf{e}|\mathbf{f}^s) \\ &\overset{(a)}{\leq} H(\mathbf{e}) - H(\mathbf{e}|\mathbf{f}^s) + H(\mathbf{e}|\mathbf{f}^s,\mathbf{r}) \\ &\overset{(b)}{\leq} H(\mathbf{r}) - I(\mathbf{e};\mathbf{r}|\mathbf{f}^s) \\ &\overset{(c)}{=} H(\mathbf{r}|\mathbf{f}^s,\mathbf{e}) \\ &\overset{(d)}{=} 0, \end{aligned}$$

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

Remark 12. *If the eavesdropper has a vanishing probability of error in decoding* $\mathbf{r}$ *given* $\mathbf{e}$ *and* $\mathbf{f}^s$*, then, by Fano's inequality, one can write* $H(\mathbf{r}|\mathbf{f}^s,\mathbf{e}) \leq |\mathbf{r}|\epsilon$*, and, by following the above steps, can show the bound* $I(\mathbf{f}^s;\mathbf{e}) \leq |\mathbf{r}|\epsilon$*, where* $|\mathbf{r}|$ *is the number of random bits, and* ϵ *can be made small if the probability of error is vanishing. This shows that the leakage rate* $I(\mathbf{f}^s;\mathbf{e})/|\mathbf{e}|$ *is vanishing. (See, e.g., [25].)*

■

APPENDIX B
PROOF OR LEMMA 4

We summarize the steps given in [28].

$$\begin{aligned} \mathcal{M}^s &= H(\mathbf{f}^s) \\ &\overset{(a)}{=} H(\mathbf{f}^s) - I(\mathbf{f}^s;\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) \\ &= H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) \\ &\overset{(b)}{=} H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) - H(\mathbf{f}^s|\mathbf{s}_{\mathcal{K}}) \\ &\overset{(c)}{=} H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) - H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'},\mathbf{s}_{\mathcal{K}}) \\ &\leq I(\mathbf{f}^s;\mathbf{s}_{\mathcal{K}}|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) \\ &\leq H(\mathbf{s}_{\mathcal{K}}|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) \\ &= \sum_{j=1}^{k} H(\mathbf{s}_j|\mathbf{s}_1,\cdots,\mathbf{s}_{j-1},\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}), \end{aligned}$$

where (a) follows by the security constraint, i.e., $0 \leq I(\mathbf{f}^s;\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'}) \leq I(\mathbf{f}^s;\mathbf{s}_{\mathcal{E}_1},\mathbf{d}_{\mathcal{E}_2}) = 0$, (b) is due to the data construction property, i.e., $H(\mathbf{f}^s|\mathbf{s}_{\mathcal{K}}) = 0$, (c) is due to $0 \leq H(\mathbf{f}^s|\mathbf{s}_{\mathcal{E}_1'},\mathbf{d}_{\mathcal{E}_2'},\mathbf{s}_{\mathcal{K}}) \leq H(\mathbf{f}^s|\mathbf{s}_{\mathcal{K}}) = 0$.

APPENDIX C
NRBW VALUES FOR MBCR POINT IN DSS

The parameters of Proposition 5 are given in the following tables. $\ell_1 = 0$ case corresponds to the systems without security constraints. $t = 1$ case corresponds to non-cooperative case. Red (green) font highlights cases with greater (respectively, smaller) cooperative NRBW ($\gamma/\mathcal{M}^s$) compared to that for $t = 1$. We observed that the same trend continues for higher n values.

TABLE I: NRBW for $n = 4, 5$, $d \geq k$, $d + t = n$.

n	k	l	t	d	$\beta/\mathcal{M}^s$	$\beta'/\mathcal{M}^s$	$\gamma/\mathcal{M}^s$	$\mathcal{M}$	$\mathcal{M}^s$
4	2	0	1	3	0.2000	0.1000	0.6000	10	10
4	2	0	2	2	0.2500	0.1250	**0.6250**	8	8
4	2	1	1	3	0.5000	0.2500	1.5000	10	4
4	2	1	2	2	0.6667	0.3333	**1.6667**	8	3
4	3	0	1	3	0.1667	0.0833	0.5000	12	12
4	3	1	1	3	0.3333	0.1667	1.0000	12	6
4	3	2	1	3	1.0000	0.5000	3.0000	12	2
5	2	0	1	4	0.1429	0.0714	0.5714	14	14
5	2	0	2	3	0.1667	0.0833	**0.5833**	12	12
5	2	0	3	2	0.2000	0.1000	**0.6000**	10	10
5	2	1	1	4	0.3333	0.1667	1.3333	14	6
5	2	1	2	3	0.4000	0.2000	**1.4000**	12	5
5	2	1	3	2	0.5000	0.2500	**1.5000**	10	4
5	3	0	1	4	0.1111	0.0556	0.4444	18	18
5	3	0	2	3	0.1333	0.0667	**0.4667**	15	15
5	3	1	1	4	0.2000	0.1000	0.8000	18	10
5	3	1	2	3	0.2500	0.1250	**0.8750**	15	8
5	3	2	1	4	0.5000	0.2500	2.0000	18	4
5	3	2	2	3	0.6667	0.3333	**2.3333**	15	3
5	4	0	1	4	0.1000	0.0500	0.4000	20	20
5	4	1	1	4	0.1667	0.0833	0.6667	20	12
5	4	2	1	4	0.3333	0.1667	1.3333	20	6
5	4	3	1	4	1.0000	0.5000	4.0000	20	2

TABLE II: NRBW for $n = 4, 5$, $d \geq k$, $d + t \leq n$.

n	k	l	t	d	$\beta/\mathcal{M}^s$	$\beta'/\mathcal{M}^s$	$\gamma/\mathcal{M}^s$	$\mathcal{M}$	$\mathcal{M}^s$
4	2	0	1	3	0.2000	0.1000	0.6000	10	10
4	2	0	1	2	0.3333	0.1667	0.6667	6	6
4	2	0	2	2	0.2500	0.1250	**0.6250**	8	8
4	2	1	1	3	0.5000	0.2500	1.5000	10	4
4	2	1	1	2	1.0000	0.5000	2.0000	6	2
4	2	1	2	2	0.6667	0.3333	**1.6667**	8	3
4	3	0	1	3	0.1667	0.0833	0.5000	12	12
4	3	1	1	3	0.3333	0.1667	1.0000	12	6
4	3	2	1	3	1.0000	0.5000	3.0000	12	2
5	2	0	1	4	0.1429	0.0714	0.5714	14	14
5	2	0	1	3	0.2000	0.1000	0.6000	10	10
5	2	0	2	3	0.1667	0.0833	**0.5833**	12	12
5	2	0	1	2	0.3333	0.1667	0.6667	6	6
5	2	0	2	2	0.2500	0.1250	**0.6250**	8	8
5	2	0	3	2	0.2000	0.1000	**0.6000**	10	10
5	2	1	1	4	0.3333	0.1667	1.3333	14	6
5	2	1	1	3	0.5000	0.2500	1.5000	10	4
5	2	1	2	3	0.4000	0.2000	**1.4000**	12	5
5	2	1	1	2	1.0000	0.5000	2.0000	6	2
5	2	1	2	2	0.6667	0.3333	**1.6667**	8	3
5	2	1	3	2	0.5000	0.2500	**1.5000**	10	4
5	3	0	1	4	0.1111	0.0556	0.4444	18	18
5	3	0	1	3	0.1667	0.0833	0.5000	12	12
5	3	0	2	3	0.1333	0.0667	**0.4667**	15	15
5	3	1	1	4	0.2000	0.1000	0.8000	18	10
5	3	1	1	3	0.3333	0.1667	1.0000	12	6
5	3	1	2	3	0.2500	0.1250	**0.8750**	15	8
5	3	2	1	4	0.5000	0.2500	2.0000	18	4
5	3	2	1	3	1.0000	0.5000	3.0000	12	2
5	3	2	2	3	0.6667	0.3333	**2.3333**	15	3
5	4	0	1	4	0.1000	0.0500	0.4000	20	20
5	4	1	1	4	0.1667	0.0833	0.6667	20	12
5	4	2	1	4	0.3333	0.1667	1.3333	20	6
5	4	3	1	4	1.0000	0.5000	4.0000	20	2

REFERENCES

[1] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, "Pond: The OceanStore Prototype," in *Proc. of the 2nd USENIX Conference on File and Storage Technologies (FAST'03)*, San Francisco, CA, Mar. 2003.

[2] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proc. of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP'03)*, New York, NY, 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. of the First ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI'04)*, Berkeley, CA, Mar. 2004.

[4] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[5] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems against eavesdropping and adversarial attacks," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6734–6753, Oct. 2011.

[6] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[7] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through hadamard designs," in *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.

[8] V. R. 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)*, Saint Petersburg, Russia, Jul. 31-Aug. 5 2011.

[9] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.

[10] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Proc. 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2011)*, Monticello, IL, Sep. 2011.

[11] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 268–276, Feb. 2010.

[12] X. Wang, Y. Xu, Y. Hu, and K. Ou, "MFR: Multi-loss flexible recovery in distributed storage systems," in *Proc. 2010 IEEE International Conference on Communications (ICC 2010)*, Cape Town, South Africa, May 2010.

[13] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub, "Repairing multiple failures with coordinated and adaptive regenerating codes," in *Proc. 2011 International Symposium on Network Coding (NetCod 2001)*, Beijing, China, Jul. 2011.

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

[15] F. Oggier and A. Datta, "Coding techniques for repairability in networked distributed storage systems," *Foundations and Trends in Communications and Information Theory*, vol. 9, no. 4, pp. 383–466, Jun. 2013.

[16] K. W. Shum and Y. Hu, "Exact minimum-repair-bandwidth cooperative regenerating codes for distributed storage systems," in *Proc. 2011 IEEE International Symposium on Information Theory (ISIT 2011)*, Saint Petersburg, Russia, Jul. 31-Aug. 5 2011.

[17] K. W. Shum, "Cooperative regenerating codes for distributed storage systems," in *Proc. 2011 IEEE International Conference on Communications (ICC 2011)*, Kyoto, Japan, Jun. 2011.

[18] K. W. Shum and Y. Hu, "Cooperative regenerating codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7229–7258, Nov. 2013.

[19] N. Le Scouarnec, "Exact scalar minimum storage coordinated regenerating codes," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT 2012)*, Cambridge, MA, Jul. 2012.

[20] S. Jiekak and N. Le Scouarnec, "CROSS-MBCR: Exact minimum bandwith coordinated regenerating codes," *CoRR*, vol. abs/1207.0854, Jul. 2012.

[21] A. Wang and Z. Zhang, "Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage," in *Proc. IEEE INFOCOM 2013*, Turin, Italia, Apr. 2013.

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

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

[24] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

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

[26] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Amer. Inst. Elect. Eng.*, vol. 45, pp. 295–301, Jan. 1926.

[27] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-theoretically secure regenerating codes for distributed storage," in *Proc. 2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, Houston, TX, Dec. 2011.

[28] A. S. Rawat, O. O. Koyluoglu, N. Silberstein, and S. Vishwanath, "Optimal locally repairable and secure codes for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 212–236, Jan. 2014.

[29] S. Goparaju, S. El Rouayheb, R. Calderbank, and H. V. Poor, "Data secrecy in distributed storage systems under exact repair," in *Proc. 2013 International Symposium on Network Coding (NetCod 2013)*, Calgary, Canada, June 2013.

[30] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," in *Proc. IEEE INFOCOM 2011*, Shanghai, China, Apr. 2011.

[31] ——, "Self-repairing codes for distributed storage - A projective geometric construction," in *Proc. 2011 IEEE Information Theory Workshop (ITW 2011)*, Paraty, Brazil, May 2011.

[32] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2012.

[33] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Proc. Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007)*, Cambridge, MA, Jul. 2007.

[34] 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)*, Cambridge, MA, Jul. 2012.

[35] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. 2012 IEEE International Symposium on Information Theory (ISIT 2012)*, Cambridge, MA, Jul. 2012.

[36] F. Oggier and A. Datta, "Byzantine fault tolerance of regenerating codes," in *Proc. 2011 IEEE International Conference on Peer-to-Peer Computing (P2P)*, Kyoto, Japan, Aug. 31 - Sep. 2, 2011.

[37] 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)*, Cambridge, MA, Jul. 2012.

[38] N. Silberstein, A. S. Rawat, and S. Vishwanath, "Error resilience in distributed storage via rank-metric codes," in *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2012.

[39] Y. S. Han, H. T. Pai, R. Zheng, and P. K. Varshney, "Update-efficient regenerating codes with minimum per-node storage," in *Proc. 2013 IEEE International Symposium on Information Theory (ISIT 2013)*, Istanbul, Turkey, Jul. 2013.

[40] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[41] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. 1988 ACM SIGMOD international conference on Management of data (SIGMOD'88)* New York, NY, USA: ACM, 1988, pp. 109–116.

[42] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192–202, Feb. 1995.

[43] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 529–542, Mar. 1996.

[44] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[45] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[46] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Peredachi Inf.*, vol. 21, no. 1, pp. 3–16, Jul. 1985.

[47] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *Journal of Combinatorial Theory, Series A*, vol. 25, no. 3, pp. 226–241, Nov. 1978.

[48] R. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.

[49] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, ser. North-Holland Mathematical Library. Elsevier, 1977.

[50] R. M. Roth and G. Seroussi, "On generator matrices of MDS codes," *IEEE Trans. Inf. Theory*, vol. 31, no. 6, pp. 826–830, Nov. 1985.

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

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

[53] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," in *Advances in Cryptology - EUROCRYPT '91*, ser. Lecture Notes in Computer Science, vol. 547. Berlin, Heidelberg: Springer-Verlag, 1991, pp. 482–489.

[54] K. Gibson, "The security of the Gabidulin public key cryptosystem," in *Advances in Cryptology - EUROCRYPT '96*, ser. Lecture Notes in Computer Science, vol. 1070. Springer, 1996, pp. 212–223.

[55] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[56] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.