

# Centralized Repair of Multiple Node Failures with Applications to Communication Efficient Secret Sharing

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

## Abstract

This paper considers a distributed storage system, where multiple storage nodes can be reconstructed simultaneously at a centralized location. This centralized multi-node repair (CMR) model is a generalization of regenerating codes that allow for bandwidth-efficient repair of a single failed node. This work focuses on the trade-off between the amount of data stored and repair bandwidth in this CMR model. In particular, repair bandwidth bounds are derived for the minimum storage multi-node repair (MSMR) and the minimum bandwidth multi-node repair (MBMR) operating points. The tightness of these bounds are analyzed via code constructions. The MSMR point is characterized through codes achieving this point under functional repair for general set of CMR parameters, as well as with codes enabling exact repair for certain CMR parameters. The MBMR point, on the other hand, is characterized with exact repair codes for all CMR parameters for systems that satisfy a certain entropy accumulation property. Finally, the model proposed here is utilized for the secret sharing problem, where the codes for the multi-node repair problem is used to construct communication efficient secret sharing schemes with the property of bandwidth efficient share repair.

## Index Terms

Codes for distributed storage, regenerating codes, cooperative regenerating codes, centralized multi-node regeneration, communication efficient secret sharing.

## I. Introduction

The ability to preserve the stored information and maintain the seamless operation in the event of permanent failures and (or) transient unavailability of the storage nodes is one of the most important issues

This paper was presented in parts at IEEE Information Theory and Applications Workshop, San Diego, CA, February 2016.

A. S. Rawat is with the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: asrawat@andrew.cmu.edu).

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).

S. Vishwanath is with the Laboratory of Informatics, Networks and Communications, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78751 USA (e-mail: sriram@austin.utexas.edu).

that need to be addressed while designing distributed storage systems. This gives rise to the so called ‘code repair’ or ‘node repair’ problem which requires a storage system to enable mechanism to regenerate (repair) the content stored on some (failed/unavailable) storage nodes with the help of the content stored on the remaining (live/available) nodes in the system. A simple replication scheme where one stores multiple copies of each data block on different nodes clearly enables the node repair as one can regenerate the data blocks stored on a node by obtaining one of their copies from the other nodes in the system. However, replication suffers from the decreasing rate as one increases the replication factor in order to enhance the resilience of the system. This motivates the use of erasure codes as they efficiently trade-off the storage space for the ability to tolerate failure/unavailability of storage nodes. However, the better utilization of the storage space should also be accompanied by a resource-efficient node repair process and efficiency of the node repair becomes a yardstick for implementing one erasure code over another.

Towards this, Dimakis et al. propose repair bandwidth, the amount of data downloaded from the contacted nodes during the repair of a single node, as a measure of the efficiency of the repair process in [1]. Considering  $n$  storage nodes where any set of  $k$  nodes are sufficient to reconstruct the entire information, Dimakis et al. further characterize an information-theoretic trade-off among the storage space vs. the repair bandwidth for such codes. The codes which attain any point on this trade-off are referred to as regenerating codes. Over the past few years, the problem of designing regenerating codes has fueled numerous research efforts which have resulted into the constructions presented in [1], [2], [3], [4], [5] and the references therein.

In this paper, we explore the problem of enabling bandwidth efficient repair of multiple nodes in a centralized manner. In particular, we consider a setting where one requires the content of any  $k$  out of  $n$  nodes in the system to be sufficient to reconstruct the entire information (as a parameter for the worst case fault-tolerance of the system). As for the centralized repair process, we consider a framework where the repair of  $t \geq 1$  node failures is performed by contacting any  $d$  out of the  $n - t$  remaining storage nodes. We also assume that  $\beta$  amount of data from each of the  $d$  contacted nodes are downloaded. We aim to characterize the storage vs. repair-bandwidth trade-off under this centralized multi-node repair (CMR) framework.

We believe that this framework is more suitable for the setting of large scale storage systems where there is a need to perform repairs at a central location. Our CMR model is perhaps useful for the following scenarios: a) Architectural and implementation related issues: Architectural constraints could make it more efficient to regenerate the content in a centralized manner. For instance, in a rack-based node placement architecture, a top-of-the-rack (TOR) switch failure would imply failure of nodes in the corresponding rack to be inaccessible, and regenerating entire content of the failed rack on a per-node basis, i.e., independently one by one, would be less efficient as compared to regenerating the content at a central location, e.g., at a leader node in that rack. b) Threshold-based data maintenance: These schemes regenerate servers after a threshold number of them fail. After regenerating the content stored on the failed nodes, the administrator can recruit  $t$  newcomers as replacements of the failed nodes and re-distribute the data to the newcomers

in order to restore the state of the system prior to the failures. c) Availability: In the event of transient unavailability of the  $t$  storage nodes, the centralized repair process allows the user to get access the content stored on the unavailable nodes in a bandwidth efficient manner.

#### A. Related work

We note that the repair of multiple nodes in a bandwidth-efficient manner has previously been considered under the cooperative repair model introduced in [6], [7]. There are two major differences between the cooperative and centralized repair frameworks: a) Under cooperative repair framework [6], [7], all  $t$  newcomer nodes are not constrained to contact the same set of  $d$  out of  $n-t$  surviving nodes. The framework allows each newcomer to contact any  $d$  surviving nodes independent of the nodes contacted by other  $t-1$  newcomers. b) Under cooperative repair framework, after downloading data from the surviving nodes, the newcomers exchange certain amount of data among themselves. On the other hand, since a centralized entity (e.g., the administrator or a master server node) has access to all the downloaded information, such information exchange is not required in the centralized repair model. Our hope is that removing the additional restriction imposed by the cooperative repair framework will enable designing codes for a broader range of system parameters.

The problem of centralized bandwidth-efficient repair of multiple node failures in a DSS employing has previously been considered by Cadambe et al. [8]. However, they restrict themselves to only MDS codes and they show existence of such codes only in the asymptotic regime where node size (amount of data stored on a node) tends to infinity.

In addition, locality, the number of nodes contacted during repair of a single node, is another measure of node repair efficiency which have been extensively studied in the literature [9]. Various minimum distance bounds and constructions achieving trade-offs are presented in [9], [10], [11], [12] and the references therein. In particular, recent works [13], [14], [15] have studied locality problem with multiple node repairs, which is a model relevant to the framework studied in this paper.

Finally, in a recent work [16], Huang et al. proposed a model for communication efficient secret sharing, where the system stores a secret over  $n$  nodes (shares) with the property that accessing to any  $z$  shares does not reveal any information about the secret, and accessing to any  $d$  shares does reveal the secret. The framework [16] is similar to that of [1] in the sense that one contacts to more than enough number of nodes (and download a partial data from each) in order to reduce the total amount of bits downloaded (to reveal secret in the former, and to repair a node in the latter). Given this setup, [16] provides a bound on required amount of communication to reconstruct the secret, constructs explicit coding schemes for certain parameter regimes achieving the stated bound, and shows an existence result for general set of parameters. More recently, [17] focuses on the same model and proposes codes that can achieve the bound provided in [16] for general set of parameters. A separate body of work [18], [19], [10], [20], [21], [22] considers secure regenerating codes, where eavesdropper accessing to a subset of nodes in the system does not get

any information about the stored data in the system. These works essentially focus on characterizing the maximum amount of secret bits that can be stored within a system that employs a given regenerating code (e.g., MSR/MBR). In this sense, these works consider a storage of data that is composed of both public (without security constraints) and private (with security constraints) information, and a data collector connects to a predefined number of nodes to recover both types of information. Whereas, in [16], only the reconstruction of the private information is the concern. In addition to this key difference, the eavesdropper models in secure regenerating code papers also include eavesdroppers that can observe the data transferred during node repairs<sup>1</sup>, whereas the framework in [16] does not consider repair problem. We note that regenerating coding schemes which are secure against such eavesdroppers are presented in [18], [19], [10], [20] and references therein. And, the problem of designing secure cooperative regenerating codes is explored in [21], [22].

## B. Contributions

The results of this work can be summarized as follows.

- We develop general repair bandwidth bounds for the CMR model at minimum per-node storage multi-node repair and minimum bandwidth multi-node repair regimes, referred to as MSMR and MBMR operating points respectively.
- We investigate tightness of the derived bounds with appropriate code constructions, and characterize the fundamental limits of the CMR model. In particular, for the MSMR scenario, the fundamental limit is characterized utilizing functional repair for all parameters. For special cases, explicit constructions that achieve the stated bound are also provided. These constructions are based on cooperative regenerating codes with minimum per-node storage (MSCR) codes as well as Zigzag codes. For the former set of codes, we show a result that any MSCR code can be utilized as MSMR code achieving the stated bounds. For the latter case, we show that multiple nodes can be repaired in Zigzag codes, and this proposed repair process is bandwidth-wise optimal, achieving the derived bound in this paper.
- For the MBMR scenario, we define minimum repair bandwidth as the property of having amount of downloaded data matching to the entropy of  $t$  nodes. For this setup, the fundamental limit is characterized for systems having a certain entropy accumulation property. In addition, we obtain a general mapping from minimum bandwidth cooperative regenerating (MBCR) codes to MBMR codes, and, utilizing MBCR with a certain entropy accumulation property, we show achievability of the stated bounds, characterizing the MBMR operating point in this special entropy accumulation case.
- Finally, we focus on the secret sharing problem, and show that the codes for the multi-node repair problem can be transformed into communication efficient secret sharing schemes that possess not only

<sup>1</sup>This eavesdropping model is important for non-MBR codes, as for MBR codes, the amount of downloaded content for a node repair is same as the data stored in the node.

the reliability (for multi-node repairs) but also the security properties. We propose a secret sharing mechanism with repairable shares that have the highest possible repair bandwidth-efficiency in the multi-node failure setup. Adversarial attack setup is considered to provide secrecy.

## II. Centralized multi-node repair model

We introduce a new model for simultaneous repair of multiple node failures in a distributed storage system (DSS), namely centralized multi-node repair (CMR) model. Consider an  $(n, k)$ -DSS, i.e., the system comprises  $n$  storage nodes and the content stored on any  $k$  nodes is sufficient to reconstruct the information stored on the system. For an  $(n, k)$ -DSS, under  $(d, t)$ -CMR model, any set of  $t$  failed nodes in the system can be repaired by downloading data from any set of  $d$  out of  $n - t$  surviving nodes. Let  $\alpha$  denote the size of each node (over a finite field  $\mathbb{F}$ ) and  $\beta$  denote the amount of data downloaded from each of the contacted  $d$  nodes under the  $(d, t)$ -CMR model. In order to denote all the relevant system parameters, we also expand the notation for the CMR model as  $(n, k, d, t, \alpha, \gamma)$ -CMR model or  $(d, t, \alpha, \gamma)$ -CMR model. After downloading  $\gamma = d\beta$  symbols from the contacted nodes, the content stored on all  $t$  failed nodes is recovered simultaneously in a centralized manner<sup>2</sup>.

## III. A file size bound for the CMR model

In this section, we initiate the study of the trade-off between the per-node storage  $\alpha$  and repair bandwidth  $\gamma$  for the CMR model. We first provide a file size bound for the CMR model.

Let the system store a uniformly distributed file  $\mathbf{f}$  of size  $|\mathbf{f}| = \mathcal{M}$  (over a finite field  $\mathbb{F}$ ). Consider the case when the nodes indexed by a set  $\mathcal{K} \subseteq [n]$  such that  $|\mathcal{K}| = k$  are used to reconstruct the file  $\mathbf{f}$ . Further, assume that this set of nodes are partitioned into  $g$  number of distinct subsets  $\mathcal{S}_i$  with  $|\mathcal{S}_i| = n_i \leq t$  such that  $\sum_{i=1}^g n_i = k$ . We have the following bound.

Lemma 1. The system parameters necessarily satisfy

$$\mathcal{M} \leq \sum_{i=1}^g \min \left\{ n_i \alpha, \left( d - \sum_{j=1}^{i-1} n_j \right) \beta \right\}. \quad (1)$$

Proof: Denoting the symbols stored on the nodes indexed by the set  $\mathcal{S}$  by  $\mathbf{x}_{\mathcal{S}}$ , we have

$$\mathcal{M} = H(\mathbf{f}) \stackrel{(a)}{=} H(\mathbf{f}) - H(\mathbf{f}|\mathbf{x}_{\mathcal{K}}) = I(\mathbf{x}_{\mathcal{K}}; \mathbf{f}) \leq H(\mathbf{x}_{\mathcal{K}}) \quad (2)$$

$$\stackrel{(b)}{=} \sum_{i=1}^g H(\mathbf{x}_{\mathcal{S}_i} | \mathbf{x}_{\mathcal{S}_1 : \mathcal{S}_{i-1}}) \quad (3)$$

$$\stackrel{(c)}{\leq} \sum_{i=1}^g \min \left\{ H(\mathbf{x}_{\mathcal{S}_i}), \left( d - \sum_{j=1}^{i-1} n_j \right) \beta \right\} \quad (4)$$

<sup>2</sup>The CMR model also allow for the distributed/parallel repair of all the  $t$  failed nodes by  $t$  newcomers independently. However, it is assumed that each of the  $t$  newcomers have an access to all the  $\gamma$  downloaded symbols.

$$\stackrel{(d)}{\leq} \sum_{i=1}^g \min \left\{ n_i \alpha, \left( d - \sum_{j=1}^{i-1} n_j \right) \beta \right\}, \quad (5)$$

where (a) is due to recoverability constraint  $H(\mathbf{f}|\mathbf{x}_{\mathcal{K}}) = 0$  as  $|\mathcal{K}| = k$ , (b) is due to  $\mathcal{K} = \cup_{i=1}^g \mathcal{S}_i$ , (c) & (d) are due to the following bounds for each term in the sum:  $H(\mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}) \leq H(\mathbf{x}_{\mathcal{S}_i}) \leq n_i \alpha$ , and

$$\begin{aligned} H(\mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}) &\stackrel{(e)}{=} H(\mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}) \\ &\quad - H(\mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}, \mathbf{d}_{\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}}) \\ &= I(\mathbf{d}_{\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}}; \mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}) \\ &\leq H(\mathbf{d}_{\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}}) \leq \left( d - \sum_{j=1}^{i-1} n_j \right) \beta \end{aligned}$$

where set of helper nodes to regenerate symbols in  $\mathcal{S}_i$  is denoted as  $\mathcal{H}_i$ , this set of  $d$  nodes is constructed by using the sets  $\mathcal{S}_1 \cdots \mathcal{S}_{i-1}$  and additional nodes not belonging to these sets (this is possible as  $\sum_{i=1}^g n_i = k \leq d$ ), downloaded symbols from these additional nodes are denoted as  $\mathbf{d}_{\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}}$  with  $|\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}| = d - \sum_{j=1}^{i-1} n_j$ , and (e) follows as  $H(\mathbf{x}_{\mathcal{S}_i}|\mathbf{x}_{\mathcal{S}_1:\mathcal{S}_{i-1}}, \mathbf{d}_{\mathcal{H}_i-\mathcal{S}_1:\mathcal{S}_{i-1}}) = 0$  as  $H(\mathbf{x}_{\mathcal{S}}|\mathbf{d}_{\mathcal{H}}) = 0$  for any  $\mathcal{S}$  such that  $|\mathcal{S}| \leq t$  and any  $\mathcal{H}$  such that  $|\mathcal{H}| = d$ . ■

Given the bound in Proposition 1, we differentiate between two operating regimes of the system: Minimum storage multi-node regeneration (MSMR) and minimum bandwidth multi-node regeneration (MBMR). The MSMR point corresponds to having an MDS code which requires that  $\alpha = \mathcal{M}/k$ . Codes that attain minimum possible repair bandwidth under this constraint, i.e.,  $\alpha = \mathcal{M}/k$ , are referred to as MSMR codes. On the other hand, the MBMR point restricts that  $H(\mathbf{x}_{\mathcal{S}}) = \gamma = d\beta$  for every  $\mathcal{S} \subseteq [n]$  such that  $|\mathcal{S}| = t$ , i.e., the amount of data downloaded during the centralized repair of  $t$  node failures is equal to the amount of information stored on the lost  $t$  nodes. MBMR codes achieve the minimum possible repair bandwidth under this restriction, i.e.,  $H(\mathbf{x}_{\mathcal{S}}) = \gamma = d\beta$ . In the following, we focus on the problem of characterizing these two operating points of the CMR model.

#### IV. MSMR Codes

We first utilize Lemma 1 to obtain a bound on the repair bandwidth at the MSMR point, and then focus on achievability.

##### A. Repair bandwidth bound

Proposition 1. Consider an  $(n, k)$ -DSS that stores a file of size  $\mathcal{M}$  and enables repair of  $t$  failed nodes under a  $(d, t, \alpha_{MSMR} = \frac{\mathcal{M}}{k}, \gamma)$ -CMR model. Then, we have

$$\gamma_{MSMR} \geq \frac{\mathcal{M}dt}{k(d-k+t)}. \quad (6)$$

Proof: Let  $a = \lfloor k/t \rfloor$  and  $b = k - at$ . We set  $n_1 = b$  and  $n_i = t$  for  $i = 2, \dots, g = a + 1$ . From the bound (1), we obtain

$$\mathcal{M} \leq \min\{b\alpha, d\beta\} + \sum_{i=1}^a \min\{t\alpha, [d - (i-1)t - b]\beta\}. \quad (7)$$

Note that we have  $\alpha = \frac{\mathcal{M}}{k}$  which implies that  $d\beta \geq b\alpha$  and

$$[d - (i-1)t - b]\beta \geq t\alpha, \forall i = 1, \dots, a,$$

From this, we obtain  $\beta \geq \frac{b\alpha}{d}$  and  $[d - (a-1)t - b]\beta \geq t\alpha$ , i.e.,  $\beta \geq \frac{t\alpha}{[d - at - b + t]} = \frac{t\alpha}{[d - k + t]}$ . This implies that

$$\gamma_{MSMR} = d\beta \geq d\alpha \max\left\{\frac{t}{d - k + t}, \frac{b}{d}\right\} \stackrel{(i)}{=} \frac{\mathcal{M}dt}{k(d - k + t)},$$

where (i) follows from the fact that we have  $b < t \leq k$  and  $\alpha = \frac{\mathcal{M}}{k}$ . ■

Remark 1. Note that the same bound is also obtained by Cadambe et al. in [8] where they consider repair of multiple failures in an MDS code.

Remark 2. A code that allows for repair of  $t$  failed nodes with the parameters  $(d, t, \alpha = \frac{\mathcal{M}}{k}, \gamma = \frac{\mathcal{M}dt}{k(d - k + t)})$ -CMR is an MSMR code.

Proposition 2. The bound above (6) does not improve when helper nodes are allowed to contribute different amounts of data for regeneration of  $t$  nodes.

Proof: The proof follows from the steps given in [16]. Assume that the  $n$  nodes in the DSS are indexed by the set  $[n]$ . Let's consider a specific failure pattern, where the  $t$  nodes indexed by the set  $[t] \subset [n]$  are under failure. Furthermore, we assume that the  $d$  nodes indexed by the set  $\{t+1, t+2, \dots, t+d\}$  are contacted to repair the  $t$  failures under the centralized repair model. For  $j \in \{t+1, t+2, \dots, t+d\}$ , let  $\mathbf{s}_j$  denote the symbols downloaded from the node indexed by  $j$  in order to repair the  $t$  failed nodes. Without loss of generality, we can assume that<sup>3</sup>

$$|\mathbf{s}_{t+1}| \geq |\mathbf{s}_{t+2}| \geq \dots \geq |\mathbf{s}_{t+d}|. \quad (8)$$

Note that an  $(n, k)$ -coding scheme with  $\alpha = \frac{\mathcal{M}}{k}$  is an MDS coding scheme. Therefore, the content of the nodes indexed by the set  $\{t+1, \dots, k\}$  does not provide any information about the content of the failed nodes, i.e., the nodes indexed by the set  $[t]$ . Therefore, in order to be able to repair the  $t$  failed nodes, we need to have

$$\sum_{j=t+1}^{t+d} |\mathbf{s}_j| \geq t\alpha, \quad (9)$$

<sup>3</sup>Note that the proof holds even when we define  $\beta_t = \frac{\gamma_t}{d} = \frac{\sum_{j=t+1}^{t+d} |\mathbf{s}_j|}{d}$ , i.e.,  $\beta_t$  represents the average number of symbols downloaded from each of the contacted nodes. In the special setting where we have each contacted node contributes the equal number of symbols during the centralized node repair process, we have  $\beta_t = |\mathbf{s}_{t+1}| = \dots = |\mathbf{s}_{t+d}|$ .

1	2	3	4	5	6
$x_{0,0}$	$x_{0,1}$	$x_{0,2}$	$x_{0,0} + x_{0,1} + x_{0,2}$	$x_{0,0} + x_{6,1} + x_{2,2}$	$x_{0,0} + x_{3,1} + x_{1,2}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$	$x_{1,0} + x_{1,1} + x_{1,2}$	$x_{1,0} + x_{7,1} + x_{0,2}$	$x_{1,0} + x_{4,1} + x_{2,2}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$	$x_{2,0} + x_{2,1} + x_{2,2}$	$x_{2,0} + x_{8,1} + x_{1,2}$	$x_{2,0} + x_{5,1} + x_{0,2}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$	$x_{3,0} + x_{3,1} + x_{3,2}$	$x_{3,0} + x_{0,1} + x_{5,2}$	$x_{3,0} + x_{6,1} + x_{4,2}$
$x_{4,0}$	$x_{4,1}$	$x_{4,2}$	$x_{4,0} + x_{4,1} + x_{4,2}$	$x_{4,0} + x_{1,1} + x_{3,2}$	$x_{4,0} + x_{7,1} + x_{5,2}$
$x_{5,0}$	$x_{5,1}$	$x_{5,2}$	$x_{5,0} + x_{5,1} + x_{5,2}$	$x_{5,0} + x_{2,1} + x_{4,2}$	$x_{5,0} + x_{8,1} + x_{3,2}$
$x_{6,0}$	$x_{6,1}$	$x_{6,2}$	$x_{6,0} + x_{6,1} + x_{6,2}$	$x_{6,0} + x_{3,1} + x_{8,2}$	$x_{6,0} + x_{0,1} + x_{7,2}$
$x_{7,0}$	$x_{7,1}$	$x_{7,2}$	$x_{7,0} + x_{7,1} + x_{7,2}$	$x_{7,0} + x_{4,1} + x_{6,2}$	$x_{7,0} + x_{1,1} + x_{8,2}$
$x_{8,0}$	$x_{8,1}$	$x_{8,2}$	$x_{8,0} + x_{8,1} + x_{8,2}$	$x_{8,0} + x_{5,1} + x_{7,2}$	$x_{8,0} + x_{2,1} + x_{6,2}$

Fig. 1: Repair of the first two systematic nodes in a  $(6,3)$ -zigzag code. (Coding coefficients of the parity symbols are not specified.) Blue (red) colored symbols contribute in the repair of only node 1 (respectively, 2) in the case of single node failure. Green colored symbols contribute in the repair of both node 1 and node 2 in the case of single node failure. Magenta colored symbols denote the additional symbols that need to be downloaded to enable the centralized repair of both the nodes.

i.e., the amount of data downloaded from the remaining  $d + t - k$  contacted nodes should be at least the amount of information lost due to node failures. Therefore, we have

$$\begin{aligned}
\gamma_t &= \sum_{j=t+d}^{t+k} |\mathbf{s}_j| \stackrel{(a)}{\geq} \frac{d}{d-k+t} \sum_{j=t+k+1}^{t+d} |\mathbf{s}_j| \\
&\stackrel{(b)}{\geq} \frac{d\alpha}{d-k+t},
\end{aligned} \tag{10}$$

where (a) and (b) follow from (8) and (9), respectively. ■

## B. Constructions and the characterization of the MSMR point

1) Constructions from existing MSCR codes: Minimum storage cooperative regenerating (MSCR) codes allow for simultaneous repair of  $t$  storage nodes with the following scheme: Each newcomer node contacts to  $d$  nodes and downloads  $\beta$  symbols from each. (Different nodes can contact to different live nodes.) Then, each newcomer node sends  $\beta'$  symbols to each other. Under this setup, the repair bandwidth per failed node is  $d\beta + (t-1)\beta'$ . MSCR codes operate at  $\alpha_{MSCR} = \mathcal{M}/k$  and  $\beta_{MSCR} = \beta'_{MSCR} = \frac{\mathcal{M}}{k(d-k+t)}$ .

**Proposition 3.** A code  $\mathcal{C}$  that operates as an MSCR code is also an MSMR code for the CMR model.



Proof: Consider that each failed node contact to the same set of  $d$  nodes in the MSCR code  $\mathcal{C}$ . Then, each failed node downloads  $\beta_{MSCR}$  symbols from these  $d$  helper nodes, resulting in a total of at most  $\gamma = td\beta_{MSCR} = \frac{\mathcal{M}dt}{k(d-k+t)}$  symbols. These symbols can recover each failed node, hence regenerates  $t$  failed nodes in the CMR model. Therefore, code  $\mathcal{C}$  is an MSMR code with  $\alpha = \frac{\mathcal{M}}{k}$  and  $\gamma = \frac{\mathcal{M}dt}{k(d-k+t)}$ . ■

We remark that random linear network coding attains MSCR point [6], hence it provides an MSMR code with functional repair. Explicit code constructions for the MSCR setup while ensuring exact-repair, on the other hand, are known for a small set of parameters. The only such constructions that we are aware of are provided in [23] for  $k = t = 2$ , in [24] for  $t = 2$  (for parameters  $(n, k, d)$  at which  $(n, k, d + 1)$  MSR codes exist), and in [6] for  $d = k$ . We believe that moving from the cooperative repair model [6], [7] to the CMR model would allow us to construct MDS codes (MSMR codes) that enable repair-bandwidth efficient repair of  $t$  nodes for an expanded set of system parameters. We exhibit this by designing a scheme to perform centralized repair of multiple nodes in a distributed storage system employing a zigzag code [3].

2) Centralized repair of multiple node failures in a zigzag code [3]: The zigzag codes, as introduced in [3], are MDS codes that allow for repair of a single node failure among systematic nodes by contacting  $d = n - 1$  (all of the) remaining nodes. The zigzag codes are associated with the MSR point [1] (or MSMR point with  $t = 1$  (cf. (6))) as each of the contacted  $d = n - 1$  nodes contributes  $\beta = \frac{\alpha}{d-k+1} = \frac{\alpha}{n-k}$  symbols during the repair of a single failed node. This amounts to the repair bandwidth of  $\gamma = d\beta = \frac{n-1}{n-k}\alpha$ . Here, we show that the framework of zigzag codes also enable repair of multiple nodes in the CMR model.

We state the achievable parameters in the following result. We then illustrate the proposed centralized repair scheme with the help an example of an  $(n = 6, k = 3)$ -zigzag code where we can simultaneously repair any 2 systematic nodes<sup>4</sup>.

**Theorem 1.** For an  $(n = k + r, k)$  zigzag code with  $r = n - k \geq 2$ , it is possible to repair any  $1 \leq t \leq 3$  systematic nodes in a centralized manner with the optimal repair-bandwidth (cf. 6) by contacting  $d = n - t$  helper nodes.

Proof: We provide the details of the repair process for  $2 \leq t \leq 3$  systematic nodes along with the necessary background on zigzag codes in Appendix A. ■

**Example 1** (Repairing  $t = 2$  systematic nodes in a  $(6, 3)$ -zigzag code). Let's consider a zigzag code with the parameters  $n = 6, k = 3$  and  $\alpha = 9$  from [3]. This code is illustrated in Table 1 where each column (indexed from 1 to 6) represents a storage node. Recall that, in the event of a single node failure, this code allows for the repair of any systematic node failure by contacting  $\hat{d} = 5$  remaining nodes and downloading  $\beta = \frac{\alpha}{n-k} = 3$  symbols from each of these nodes. We now show that we can use this same construction (with required modifications of the non-zero coefficients in coded symbols) to repair 2 systematic node failures

<sup>4</sup>In a parallel and independent work [25], the authors present a mechanism for repairing multiple failures in zigzag codes as well. They show that the zigzag codes can repair any  $t \leq n - k$  failures while achieving the lower bound in (6).

by contacting  $d = n - 2 = 4$  remaining nodes. We download  $t \frac{\alpha}{d-k+2} = 2 \frac{\alpha}{n-k} = 6$  symbols from each of the  $d = 4$  contacted nodes.

Assume that node 1 and 2 are in failure. We download the colored symbols from node 3 to node 6 in Figure 1 to repair these two nodes. Using the downloaded symbols, we get the following 18 combinations in the 18 unknown information symbols. (We suppress the coefficients of the linear combinations here.)

$$\begin{aligned}
& \textcolor{red}{x}_{0,0} + x_{6,1}, \textcolor{red}{x}_{1,0} + x_{4,1}, \textcolor{red}{x}_{2,0} + x_{2,1}, \textcolor{red}{x}_{3,0} + x_{0,1}, \\
& \textcolor{red}{x}_{4,0} + x_{7,1}, \textcolor{red}{x}_{5,0} + x_{5,1}, \textcolor{red}{x}_{6,0} + x_{0,1}, \textcolor{red}{x}_{7,0} + x_{7,1}, \\
& \textcolor{red}{x}_{8,0} + x_{5,1}, x_{2,0} + \textcolor{red}{x}_{8,1}, x_{1,0} + \textcolor{red}{x}_{7,1}, x_{6,0} + \textcolor{red}{x}_{6,1}, \\
& x_{2,0} + \textcolor{red}{x}_{5,1}, x_{7,0} + \textcolor{red}{x}_{4,1}, x_{0,0} + \textcolor{red}{x}_{3,1}, x_{8,0} + \textcolor{red}{x}_{2,1}, \\
& x_{1,0} + \textcolor{red}{x}_{1,1}, x_{0,0} + \textcolor{red}{x}_{0,1}.
\end{aligned} \tag{11}$$

Now, we need to show that it is possible to choose the coding coefficients in such a manner that these 18 equations allow us to recover the desired 18 symbols. Assuming that  $A$  denotes the  $18 \times 18$  coefficient matrix of the aforementioned 18 combinations, it is a necessary and sufficient (with large enough field size) condition for the matrix  $A$  to be full rank that the natural bipartite graph associated with the matrix  $A$  contains a perfect matching<sup>5</sup> [26], [27]. We illustrate one such perfect matching in (11), where the colored unknown symbol in a combination represents the unknown symbol matched by that combination. The similar argument can be performed for the remaining combinations of 2 failed systematic nodes.

3) MSMR point: The achievability results above together with the repair bandwidth bound reported in the previous section, see Remark 2, results in the following characterization.

Theorem 2. The MSMR point for the  $(n, k, d, t, \alpha, \gamma)$ -CMR model is given by

$$\alpha_{MSMR} = \frac{\mathcal{M}}{k}, \quad \gamma_{MSMR} = \frac{\mathcal{M}dt}{k(d-k+t)}.$$

## V. MBMR Codes

In this section, we focus on the other extremal point of the storage vs. repair-bandwidth trade-off, namely the MBMR point.

### A. Repair bandwidth bound

For the MBMR point, depending on whether  $t|k$  or  $t \nmid k$ , we state the following two results.

Proposition 4. Assume that  $t|k$ . Consider an  $(n, k)$ -DSS that stores a file of size  $\mathcal{M}$  and enables repair of  $t$  failed nodes under a  $(d, t, \alpha_{MBMR}, \gamma_{MBMR})$ -CMR model. Then, denoting the entropy of  $t$  nodes as  $H_t$ ,

<sup>5</sup>The left and the right nodes in the bipartite graph correspond to the combinations and the unknowns, respectively.

we have

$$t\alpha_{MBMR} \geq H_t = \gamma_{MBMR}, \quad (12)$$

$$\gamma_{MBMR} \geq \frac{\mathcal{M}2dt}{k(2d-k+t)}. \quad (13)$$

Proof: Note that the MBMR point has  $H(\mathbf{x}_{\mathcal{S}}) = \gamma_{MBMR}$  for every  $\mathcal{S} \subseteq [n]$  such that  $|\mathcal{S}| = t$ . Therefore, we have

$$\gamma_{MBMR} = H(\mathbf{x}_{\mathcal{S}}) \leq \sum_{i \in \mathcal{S}} H(\mathbf{x}_i) \leq t\alpha_{MBMR}.$$

In order to establish the lower bound on  $\gamma_{MBMR}$  in (13), we use  $n_i = t, \forall i \in [a]$  in the bound (1), we obtain

$$\mathcal{M} \leq \sum_{i=1}^{k/t} (d - (i-1)t)\beta = \frac{k}{t} \left( \frac{2d-k+t}{2} \right) \beta. \quad (14)$$

This implies that  $\gamma_{MBMR} = d\beta \geq \frac{\mathcal{M}2dt}{k(2d-k+t)}$ .  $\blacksquare$

Proposition 5. Consider an  $(n, k)$ -DSS that stores a file of size  $\mathcal{M}$  and enables repair of  $t$  failed nodes under a  $(d, t, \alpha_{MBMR}, \gamma_{MBMR})$ -CMR model. Then, the bounds given in (12) and (13) hold for the case of  $t \nmid k$ , if  $H_b \geq \left(\frac{\beta}{t}\right) \left[ b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right]$ , where  $b = k \pmod{t}$ , and  $H_b$  denotes entropy of  $b$  nodes in the system.

Proof: The bound in (12) follows from the similar analysis as presented in the proof of Proposition 5. In order to establish (13), we select  $g = \lfloor k/t \rfloor + 1 = a + 1$  disjoint sets of nodes indexed by the sets  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_g$  such that  $n_1 = |\mathcal{S}_1| = b$  and  $n_i = |\mathcal{S}_i| = t$  for  $i \in \{2, 3, \dots, g = a + 1\}$ . Note that we have  $\sum_i n_i = k$ . Utilizing this particular sequence of sets in (4) along with the fact that we have  $H(\mathbf{x}_{\mathcal{S}_i}) = d\beta$  for  $2 \leq i \leq g$ , we obtain

$$\begin{aligned} \mathcal{M} &\leq \min \{H(\mathbf{x}_{\mathcal{S}_1}), d\beta\} + \sum_{i=1}^a (d - (i-1)t - b)\beta \\ &= H(\mathbf{x}_{\mathcal{S}_1}) + \sum_{i=1}^a (d - (i-1)t - b)\beta. \end{aligned} \quad (15)$$

Note that the choice of the set  $\mathcal{S}_1$  is arbitrary and all the nodes in the system are equivalent in terms of their information content. Therefore,  $H_b = H(\mathcal{S}_1)$  (the amount of information stored on  $b$  nodes indexed by the set  $\mathcal{S}_1$ ) only depends on  $b$ . It follows from (15) that

$$\mathcal{M} \leq H_b + \left( \frac{2d-k+(t-b)}{2} \right) a\beta \quad (16)$$

In order to have the bound in (13) we need the RHS of (16) to be at least the RHS of (14), i.e.,

$$H_b + \left( \frac{2d-k+(t-b)}{2} \right) a\beta \geq \frac{k}{t} \left( \frac{2d-k+t}{2} \right) \beta.$$

This implies that

$$H_b \geq \left( \frac{2d-k+t}{2} \right) \frac{k}{t} \beta - \left( \frac{2d-k+(t-b)}{2} \right) a\beta$$

$$= \left(\frac{\beta}{t}\right) \left[ b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right]. \quad (17)$$

■

Remark 3. A code that allows for repair of  $t$  failed nodes with  $H_t = \gamma = \frac{\mathcal{M}2dt}{k(2d-k+t)}$  is an MBMR code for the case of  $t|k$  and  $t \nmid k$ , if for the latter case the system also operates at  $H_b \geq \left(\frac{\beta}{t}\right) \left[ b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right]$ .

## B. Constructions and the characterization of the MBMR point

1) Constructions from existing MBCR codes: MBCR codes have  $\alpha_{MBCR} = \frac{\mathcal{M}}{k} \frac{2d+t-1}{2d+t-k}$ ,  $\beta = \frac{\mathcal{M}}{k} \frac{2}{2d+t-k}$ , and  $\beta' = \frac{\mathcal{M}}{k} \frac{1}{2d+t-k}$ . A construction of MBCR codes for all parameters is provided in [28], where the entropy accumulation for MBCR codes is also characterized. In particular, entropy of  $b \leq k$  nodes is given by  $H_b = \left( b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right) \beta$ .

Proposition 6. A code  $\mathcal{C}$  that operates as an MBCR code is also an MBMR code for the CMR model that operates at  $\alpha = \frac{\mathcal{M}(2d+t-1)}{k(2d+t-k)}$  and  $H_b \geq \left(\frac{\beta}{t}\right) \left[ b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right]$ .

Proof: Consider that each failed node contact to the same set of  $d$  nodes in the MBCR code  $\mathcal{C}$ . This results in a repair bandwidth of at most  $\gamma = td\beta_{MBCR} = \frac{\mathcal{M}2dt}{k(2d+t-k)}$ . Entropy of  $t$  nodes in this code is given by  $H_t = \left( t \left( \frac{2d+t-1}{2} \right) - \binom{t}{2} \right) \frac{\mathcal{M}}{k} \frac{2}{2d+t-k} = \frac{\mathcal{M}2dt}{k(2d+t-k)} = \gamma$ . These and also the entropy of  $b$  nodes meet the conditions stated in Remark 3, establishing the claimed result. ■

Remark 4. In general, for MBMR codes, we have the condition that  $t\alpha \geq H_t = \gamma_{MBMR}$ . It is not clear if  $\alpha$  can be further reduced than that in Proposition 6, e.g., when  $b = 0$ .

2) MBMR point: The achievability results above together with the repair bandwidth bound reported in the previous section results in the following characterization.

Theorem 3. Let  $k \pmod t = b$ . Then, for the CMR models satisfying  $H_b \geq \left(\frac{\beta}{t}\right) \left[ b \left( \frac{2d+t-1}{2} \right) - \binom{b}{2} \right]$ , the MBMR point is given by

$$H_t = \gamma_{MBMR} = \frac{\mathcal{M}2dt}{k(2d+t-k)}.$$

## VI. Applications to communication and repair efficient secret sharing schemes

Recently, in [16], Huang et al. proposed a model for communication efficient secret sharing. They consider a setting where one wants to encode a secret  $\mathbf{m} \in \mathbb{F}_Q^K$  into  $N$  shares  $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N \in \mathbb{F}_Q$ . The encoding from secret to shares should satisfy two requirements: 1) Given any  $z$  shares one should not be able to learn any information about the secret  $\mathbf{m}$  and 2) given access to any  $d \geq N - r$  shares one should be able to reconstruct (or decode) the entire secret  $\mathbf{m}$ . Huang et al. refer to such secret sharing schemes as  $(N, K, r, z)_Q$  secret sharing schemes. For a naive secret reconstruction process, one downloads  $Q$  symbols over  $\mathbb{F}_Q$  from each of the  $d$  contacted shares leading to the communication bandwidth (the amount of data downloaded for secret reconstruction) of  $dQ$  symbols over  $\mathbb{F}_Q$ . In [16], Huang et al. explore the minimum

possible communication bandwidth of an  $(N, K, r, z)_Q$  secret sharing scheme as a function of the number of shares participating in the reconstruction process  $d$ . Towards this end, the authors obtain the following bound on the communication bandwidth of an  $(N, K, r, z)_Q$  secret sharing scheme when  $N - r \leq d \leq N$  shares are available during the reconstruction process<sup>6</sup>.

$$BW_d \geq \frac{d}{d-z}K, \quad (18)$$

where the communication bandwidth is counted in terms of the number of symbols over  $\mathbb{F}_Q$ . Huang et al. further present an explicit  $(N, K = N - r - z, r, z)_Q$  secret sharing scheme which attain the bound in (18) for  $d = k$  and  $d = N$ . They also show the existence of  $(N, K = N - r - z, r, z)_Q$  secret sharing schemes which attain the lower bound on the communication bandwidth for all values of  $d$  in  $\{N - r, N - r + 1, \dots, N\}$ . Note that these secret sharing schemes are designed to work for a particular value  $d$ . However, it is also an interesting question to design secret sharing schemes which simultaneously work for all the values of  $d$ . In [17], Bitar and El Rouayheb present two explicit constructions which give  $(N, K = N - r - z, r, z)_Q$  secret sharing schemes with optimal communication bandwidth. The first construction attains the bound in (18) for any fixed  $d$  and the second construction simultaneously attains the bound for  $N - r \leq d \leq N$ .

In this section, we show that the communication optimal  $(N, K, r, z)_Q$  secret sharing schemes can be designed using the codes which allow for centralized repair of multiple nodes. The added advantage of using this approach to construct secret sharing scheme is that this method also enables bandwidth efficient repair of shares in the secret sharing scheme. This can also be viewed as an attempt to unify the study of repair bandwidth efficient codes for distributed storage and communication efficient secret sharing. This allows us to employ various ideas from the work on secure distributed storage literature to the setting of communication efficient secret sharing.

Let  $\mathcal{M}^s$  be the size of the secret  $\mathbf{m}$  (over  $\mathbb{F}_q$ ) that we want secure in the secret sharing scheme. We further assume that each of the  $N$  shares in the secret sharing scheme consists of  $\alpha$  symbols over  $\mathbb{F}_q$ , i.e., we have  $\mathbb{F}_Q = \mathbb{F}_{q^\alpha}$ . Note that Huang et al. define the sizes of the secret and the shares over the same alphabet  $\mathbb{F}_Q = \mathbb{F}_{q^\alpha}$  [16]. However, we denote the size of the secret over a base field  $\mathbb{F}_q$  and assume that each share comprises a symbol from the extension field  $\mathbb{F}_Q = \mathbb{F}_{q^\alpha}$ . This representation is quite prevalent in the distributed storage literature and is consistent with the rest of the paper as well. We represent the secret sharing scheme as an  $(N, \mathcal{M}^s, r, z)_{\alpha, q}$  or  $(N, \mathcal{M}^s, r, z)_\alpha$  secret sharing scheme. First, we restate the lower bound on the communication bandwidth of an  $(N, \mathcal{M}^s, r, z)_\alpha$  secret sharing schemes (cf. (18)) in our notations as follows.

$$BW_d \geq \frac{d}{d-z}\mathcal{M}^s, \quad (19)$$

where we count the communication bandwidth  $BW_d$  in terms of number of symbols over the base field  $\mathbb{F}_q$ .

<sup>6</sup>In [16], the authors present this bound in terms of communication overhead  $CO_d$  which is the difference between the communication bandwidth  $BW_d$  and the size of the secret  $K$ .

**Definition 1.** ( $z$ -secure distributed storage system) Consider an  $(n, k)$ -DSS storing a file  $\mathbf{f}^s$  of size  $\mathcal{M}^s$  (over  $\mathbb{F}_q$ ) under the  $(d, t)$ -CMR model. We say that the DSS is  $z$ -secure if an eavesdropper who has access to the content of any set of  $z$  (out of  $n$ ) storage nodes does not gain any information about the file  $\mathbf{f}^s$ .

**Remark 5.** Recall that when there is no security requirement, we denote the file stored on the DSS and its size as  $\mathbf{f}$  and  $\mathcal{M}$  (over  $\mathbb{F}_q$ ), respectively (cf. Section III). The quantity  $\mathcal{M} - \mathcal{M}^s$  denotes the loss in the file size that the system has to bear in order to guarantee the information theoretic security of the stored file against an eavesdropper. Or, this part of the data can be considered as public information (without any secrecy constraints), as compared to the private counterpart (which has secrecy constraints).

The file size bounds for DSS which are secure against even a general eavesdropping model where an eavesdropper can observe both the content stored on a set of nodes and the content downloaded during the repair of another set of node have been previously considered in the literature. The regenerating coding schemes which are secure against such eavesdroppers are presented in [18], [19], [10], [20] and references therein. Similarly, the problem of designing secure cooperative regenerating codes is explored in [21], [22]. As discussed in Section IV and V, both regenerating codes and cooperative regenerating codes are specific sub-classes of codes for centralized repair model. Therefore, both the secure regenerating codes and secure cooperative regenerating codes which can prevent the leakage of information to an eavesdropper observing the content stored on  $z$ -storage nodes form special cases of  $z$ -secure DSS under CMR model with respective system parameters.

We can utilize  $z$ -secure coding scheme for DSS under the CMR model to obtain communication efficient secret sharing schemes. We first illustrate this approach with the help of a secure MSR code in the following subsection. We then comment on how this approach can be employed using general secure coding schemes for DSS under the CMR model.

#### A. An example

Let  $\mathcal{C}$  be a linear systematic code which operates at  $(n, \mathcal{M}, d < n - 1, \alpha = \frac{\mathcal{M}}{z+1}, \beta = \frac{\alpha}{d-(z+1)+1})_q$  MSR point. Note that this  $\mathcal{C}$  is also an MDS code where the content of any  $k = z + 1$  symbols is sufficient to recover the entire file of size  $\mathcal{M}$ . We next show how we can use  $\mathcal{C}$  to construct a communication bandwidth efficient  $(N = n - 1, \mathcal{M}^s = \alpha = \frac{\mathcal{M}}{z+1}, r = N - z - 1, z)_\alpha$  secret sharing scheme.

Let  $\mathbf{m} = (m_1, \dots, m_\alpha) \in \mathbb{F}_q^\alpha$  denote the secret of size  $\alpha$  over  $\mathbb{F}_q$ . Let  $\mathbf{r} = (r_1, r_2, \dots, r_{z\alpha}) \in \mathbb{F}_q^{z\alpha}$  be  $z\alpha$  random symbols which are distributed uniformly at random over  $\mathbb{F}_q$ . We encode the  $\mathcal{M} = (z + 1)\alpha$ -length vector  $(\mathbf{m}, \mathbf{r}) \in \mathbb{F}_q^{\mathcal{M}}$  using the MSR code  $\mathcal{C}$ . Let  $(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n) = (\mathbf{m}, \mathbf{r}, \mathbf{c}_{z+2}, \dots, \mathbf{c}_n) \in \mathbb{F}_q^n$  denote the associated MSR codeword. Note that a code symbol, say  $\mathbf{c}_i$ , can be repaired by any set of  $d$  out of the remaining  $n - 1$  code symbols by downloading at most  $d\beta = \frac{d}{d-(z+1)+1}\alpha = \frac{d}{d-z}\mathcal{M}^s$  symbols (over  $\mathbb{F}_q$ ) from the contacted  $d$  nodes. In order to obtain a secret sharing scheme we puncture the symbol  $\mathbf{c}_1$  from each of the codewords in  $\mathcal{C}$  which gives us another code  $\tilde{\mathcal{C}} \in \mathbb{F}_q^{n-1}$ . Let  $\tilde{\mathbf{c}} = (\mathbf{c}_2, \mathbf{c}_3, \dots, \mathbf{c}_n) \in \mathbb{F}_q^{n-1}$  be the codeword

in  $\tilde{\mathcal{C}}$  which is obtained by removing the first code symbol from the codeword  $\mathbf{c} \in \mathcal{C}$ . For the secret  $\mathbf{m}$  we treat  $n-1$  symbols in  $\tilde{\mathbf{c}}$  as  $N = n-1$  shares of the secret sharing scheme. In order to reconstruct the secret  $\mathbf{m}$ , we can invoke the node repair process of first node (code symbol) in the original MSR code  $\mathcal{C}$  where we contact  $d$  shares and download  $\beta = \frac{\alpha}{d-k+1}$  from each of these  $d$  shares. This leads to the communication bandwidth of

$$d\beta = \frac{d}{d-z}\mathcal{M}^s,$$

which matches the bound in (19). Using the MDS property <sup>7</sup> of  $\tilde{\mathcal{C}}$ , it is easy to argue that  $\mathcal{C}$  is a  $z$ -secure coding scheme. Note that besides reconstructing the secret  $\mathbf{m}$  in a communication efficient manner, the proposed scheme also allows the bandwidth efficient repair of any of the  $N = n-1$  shares by using  $d$  out of  $N-1 = n-2$  remaining shares. This can be performed again by invoking the repair mechanism of the original MSR code  $\mathcal{C}$ .

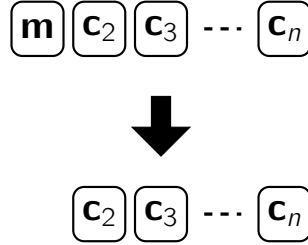


Fig. 2: Message puncturing from an MSR code gives communication efficient secret sharing with repairable shares.

#### B. Construction of communication and repair efficient secret sharing schemes using MSMR codes

Generally, we can utilize an MSMR code to obtain a communication efficient secret sharing scheme which also enables bandwidth efficient repair of the shares in the scheme. Let  $\mathcal{C}$  be a systematic linear  $(n, k = z+t, d, t, \alpha = \frac{\mathcal{M}}{k}, \gamma = \frac{t\mathcal{M}}{k} \frac{d}{d-k+t})_q$ -MSMR code. Recall that this code encodes a file  $\mathbf{f}$  of size  $\mathcal{M}$  over  $\mathbb{F}_q$  to an  $n$ -length codeword  $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n) \in \mathbb{F}_{q^\alpha}^n$  such that we have

$$\mathbf{c}_i = (\mathbf{f}_{(i-1)\alpha+1}, \mathbf{f}_{(i-1)\alpha+2}, \dots, \mathbf{f}_{i\alpha}) \in \mathbb{F}_q^\alpha \text{ for } 1 \leq i \leq k.$$

Using the code  $\mathcal{C}$ , we now construct a communication efficient  $(N = n-t, \mathcal{M}^s = t\alpha = \frac{t\mathcal{M}}{z+t}, r = N - z - t = n - k - t, z)_\alpha$  secret sharing scheme. Let  $\mathbf{m} \in \mathbb{F}_q^{\mathcal{M}^s} = \mathbb{F}_q^{t\alpha}$  denote the secret to be encoded. Let  $\mathbf{r} = (r_1, r_2, \dots, r_{z\alpha}) \in \mathbb{F}_q^{z\alpha}$  be  $z\alpha$  independent random symbols which are distributed uniformly at random over  $\mathbb{F}_q$ . We encode the  $\mathcal{M} = (z+t)\alpha$  symbols long file  $\mathbf{f} = (\mathbf{m}, \mathbf{r}) \in \mathbb{F}_q^{(z+t)\alpha}$  using the MSMR code  $\mathcal{C}$ . Given  $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n) \in \mathbb{F}_{q^\alpha}^n$ , the codeword associated with the file  $\mathbf{f}$  in the MSMR code  $\mathcal{C}$ , we puncture the codeword

<sup>7</sup>Since  $\mathcal{C}$  is an  $(n, z+1)$  MDS code, it is straightforward to observe that  $\tilde{\mathcal{C}}$  is an  $(n-1, z+1)$  MDS code.

at the first  $t$  code symbols to obtain a punctured codeword  $\tilde{\mathbf{c}} = (\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots, \tilde{\mathbf{c}}_N) = (\mathbf{c}_{t+1}, \mathbf{c}_{t+2}, \dots, \mathbf{c}_n) \in \mathbb{F}_{q^\alpha}^{n-t}$ . Assuming that  $\tilde{\mathcal{C}}$  denotes the codebook obtained by puncturing all the codewords in  $\mathcal{C}$  at the first  $t$  code symbols, we have  $\tilde{\mathbf{c}} \in \tilde{\mathcal{C}}$ . We claim that  $\tilde{\mathcal{C}}$  gives us a  $(N = n - t, \mathcal{M}^s = t\alpha = \frac{t\mathcal{M}}{z+t}, r = N - z = n - k, z)_\alpha$  secret sharing scheme.

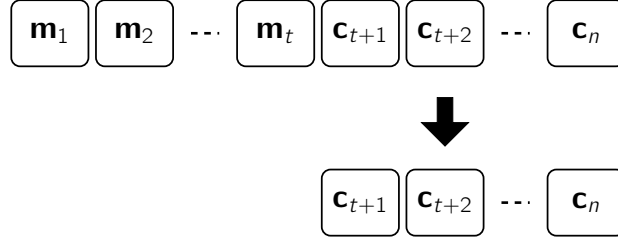


Fig. 3: Puncturing of the secret covering multiple symbols in an MSMR code gives communication efficient secret sharing with multi-share repair property.

- **Security:** Let's consider an adversary who has access to  $z$  shares  $\tilde{\mathbf{c}}_{i_1}, \tilde{\mathbf{c}}_{i_2}, \dots, \tilde{\mathbf{c}}_{i_z}$ . We make two observations: First, we have  $H(\tilde{\mathbf{c}}_{i_1}, \tilde{\mathbf{c}}_{i_2}, \dots, \tilde{\mathbf{c}}_{i_z}) \leq z\alpha = H(\mathbf{r})$ . Second, given  $\mathbf{m}$  and  $\tilde{\mathbf{c}}_{i_1}, \tilde{\mathbf{c}}_{i_2}, \dots, \tilde{\mathbf{c}}_{i_z}$  we have access to  $k = z + t$  code symbols of  $\mathbf{c} \in \mathcal{C}$ ; as a result, we can decode  $\mathbf{r}$  (by decoding  $\mathbf{f} = (\mathbf{m}, \mathbf{r})$ ) as  $\mathcal{C}$  is an  $(n, k = z + t)$  MDS code. From these two observations, it follows that the adversary does not get any information about the secret  $\mathbf{m}$  from the  $z$  shares at its disposal [19], [10].
- **Communication efficiency:** Assume that we contact a set of  $d$  shares  $\tilde{\mathbf{c}}_{i_1}, \tilde{\mathbf{c}}_{i_2}, \dots, \tilde{\mathbf{c}}_{i_d}$ . Since these  $d$  shares form  $d$  code symbols in the codeword  $\mathbf{c}$  of the MSMR code, we can use these  $d$  shares to recover the secret  $\mathbf{m}$  which constitutes the first  $t$  code symbols of the code word  $\mathbf{c}$ . Recall that we download total  $\gamma = \frac{t\mathcal{M}}{z+t} \frac{d}{d-(z+t)+t} = \frac{d}{d-z} \mathcal{M}^s$  symbols (over  $\mathbb{F}_q$ ) from the  $d$  shares we contact. Note that  $\gamma$  is exactly equal to the lower bound on (19) which establishes the communication efficiency of the obtained secret sharing scheme.

**Remark 6.** Note that if we have  $d < N - t$ , by invoking the repair process of the original MSMR code  $\mathcal{C}$ , we can repair any  $t$  shares by contacting any set of  $d$  out of  $N - t$  remaining shares and downloading  $\gamma = \frac{d}{d-z} \mathcal{M}^s$  symbols from the contacted shares. We are not necessarily required to repair the shares in the group of  $t$  failed shares at a time. If the original MSMR coding scheme also allows for bandwidth efficient repair of less than  $t$  code symbols (nodes) at a time, then we can also repair less than  $t$  failed shares at a time by using such repair mechanism. Specifically, in Section IV-B1 and IV-B2, we discuss some constructions of MSMR codes that are obtained from MSR codes. Therefore, the secret sharing schemes designed by these codes enable bandwidth efficient repair of one share at a time as well.



### C. Secret sharing schemes using MBMR codes

In this subsection, we illustrate how coding schemes at the MSMR point can be utilized to construct communication and repair efficient secret sharing scheme. Note that this allows us to increase the size of the share in an MDS code in order to lower the repair bandwidth. Furthermore, this also allows us to construct explicit secret sharing schemes for a wider set of parameters  $n, k, d$ , and  $\alpha$ . Here we note that the MBMR coding scheme that we employ in this subsection is from [28]. We define the following quantities.

$$\mathcal{M} = k(2d + t - k) = (z + t)(2d + t - (z + t)) = (z + t)(2d - z) \quad (20)$$

$$\mathcal{M}^s = t(2d + t - k - z) = t(2d + t - (z + t) - z) = 2t(d - z). \quad (21)$$

Given  $n, k = z + t$  and  $d$ , we construct an MBMR code as follows.

- Let  $\{y_1, y_2, \dots, y_{n+d+t-1}\} \subseteq \mathbb{F}_q$  be  $n + d + t - 1$  distinct elements in  $\mathbb{F}_q$ . Similarly, we select another set of  $n + d - 1$  distinct elements in  $\mathbb{F}_q$  as  $\{x_1, x_2, \dots, x_{n+d-1}\} \subseteq \mathbb{F}_q$ .
- Given an  $\mathcal{M}$ -length message vector  $\mathbf{f} = (f_1, f_2, \dots, f_{\mathcal{M}}) \in \mathbb{F}_q^{\mathcal{M}}$  construct a bi-variate polynomial such that

$$F(X, Y) = \sum_{\substack{0 \leq i < k, \\ 0 \leq j < k}} a_{i,j} X^i Y^j + \sum_{\substack{0 \leq i < k, \\ k \leq j < d+t}} b_{i,j} X^i Y^j + \sum_{\substack{k \leq i < d, \\ 0 \leq j < k}} c_{i,j} X^i Y^j. \quad (22)$$

Here,

$$(a_{0,1}, a_{0,2}, \dots, a_{k-1,k-1}, b_{0,k}, b_{1,k}, \dots, b_{k-1,d+t-1}, c_{k,0}, c_{k,1}, \dots, c_{d-1,k-1}) = \mathbf{A}\mathbf{f} \quad (23)$$

for an  $\mathcal{M} \times \mathcal{M}$  matrix  $\mathbf{A}$  with entries from  $\mathbb{F}_q$  which we specify later.

- Given the polynomial  $F(X, Y)$ , the  $i$ th code symbol  $\mathbf{c}_i$  of the codeword associated with  $\mathbf{f}$  in  $\mathcal{C}$  is obtained by evaluating  $F(X, Y)$  at

$$\{(x_i, y_i), (x_i, y_{i+1}), \dots, (x_i, y_{i+d+t-1}), (x_{i+1}, y_i), (x_{i+2}, y_i), \dots, (x_{i+d-1}, y_i)\}.$$

That is, we have

$$\mathbf{c}_i = (F(x_i, y_i), F(x_i, y_{i+1}), \dots, F(x_i, y_{i+d+t-1}), F(x_{i+1}, y_i), F(x_{i+2}, y_i), \dots, F(x_{i+d-1}, y_i)) \in \mathbb{F}_q^{2d+t-1}.$$

**Remark 7.** Note that the code symbol  $\mathbf{c}_i$  contains  $d + t$  evaluations of the degree- $(d + t)$  polynomial  $h_i(Y) = F(x_i, Y)$  at distinct points  $\{y_i, y_{i+1}, \dots, y_{i+d+t-1}\}$ . Therefore, the content of  $\mathbf{c}_i$  is sufficient to recover the polynomial  $h_i(Y) = F(x_i, Y)$ . Similarly,  $\mathbf{c}_i$  contains  $d$  evaluations of the degree- $d$  polynomial  $g_i(X) = F(X, y_i)$  at distinct points  $\{x_i, x_{i+1}, \dots, x_{i+d-1}\}$ . This implies that the content of  $\mathbf{c}_i$  is sufficient to recover the polynomial  $g_i(X) = F(X, y_i)$ .

**Remark 8.** In [28], Wang and Zhang show that this construction enables repair of any  $t$  code symbols (nodes) under a cooperative repair framework. This implies that the coding scheme can also be utilized in

the centralized repair framework. As discussed in Section V-B, these codes operate at the MBMR point with  $\alpha = 2d + t - 1$  and the repair bandwidth

$$\gamma_t = \frac{\mathcal{M}2dt}{k(2d+t-k)} = \frac{\mathcal{M}2dt}{(z+t)(2d-z)}. \quad (24)$$

The codeword associated with the information symbols  $\mathbf{f}$  in  $\mathcal{C}$  is described in Figure 4. Note that the content of evaluations highlighted in Figure 4 form an information set as the original  $\mathcal{M}$  information symbols  $\mathbf{f}$  can be reconstructed from the highlighted symbols [28]. Therefore, it is possible to precode the information symbols  $\mathbf{f}$  using an  $\mathcal{M} \times \mathcal{M}$  matrix  $\mathbf{A}$  (cf. (23)) such that the information symbols themselves appear at the highlighted positions in the codewords of  $\mathcal{C}$ . Note that this corresponds to a systematic encoding for the code  $\mathcal{C}$ .

We now describe how we can utilize the MBMR code described above to obtain a communication efficient  $(N = n - t, \mathcal{M}^s, d, r = N - z - t = n - z - 2t, z)_{\alpha, q}$ -secret sharing scheme. Let  $\mathbf{m} = (m_1, m_2, \dots, m_{\mathcal{M}^s}) \in \mathbb{F}_q^{\mathcal{M}^s}$  be a  $\mathcal{M}^s$ -length (over  $\mathbb{F}_q$ ) secret that needs to be encoded in the secret sharing scheme. Let  $\mathbf{r} = (r_1, r_2, \dots, r_{\mathcal{R}}) \in \mathbb{F}_q^{\mathcal{R}}$  be  $\mathcal{R} = \mathcal{M} - \mathcal{M}^s = 2dz + zt - z^2$  i.i.d. random variables which are uniformly distributed over  $\mathbb{F}_q$ . Given the  $\mathcal{M}^s$ -length secret  $\mathbf{m}$ , the  $(\mathcal{M} - \mathcal{M}^s)$ -length random symbols  $\mathbf{r}$  and the precoding matrix  $\mathbf{A}$ , we construct the  $\mathcal{M}$ -length information vector <sup>8</sup>  $\mathbf{f}$  such that the secret  $\mathbf{m}$  appears in the code symbols  $\mathbf{c}_{z+1}, \dots, \mathbf{c}_{z+t}$  as described in Figure 4. In other words, there exists a permutation  $\sigma : [\mathcal{M}^s] \rightarrow [\mathcal{M}^s]$  such that we have

$$\begin{aligned} F(x_{z+1}, y_{z+1}) &= m_{\sigma(1)}, F(x_{z+1}, y_{z+2}) = m_{\sigma(2)}, \dots, F(x_{z+1}, y_{z+d+t-1}) = m_{\sigma(d+t)}, \\ &\vdots \\ F(x_{z+t}, y_{z+1}) &= m_{\sigma((t-1)(d+t)+1)}, F(x_{z+t}, y_{z+2}) = m_{\sigma((t-1)(d+t)+2)}, \dots, F(x_{z+t}, y_{z+d+t-1}) = m_{\sigma(t(d+t))}, \\ F(x_{z+t+1}, y_{z+1}) &= m_{\sigma(t(d+t)+1)}, F(x_{z+t+1}, y_{z+2}) = m_{\sigma(t(d+t)+2)}, \dots, F(x_{z+t+1}, y_{z+t}) = m_{\sigma(t(d+t)+t)}, \\ &\vdots \\ F(x_d, y_{z+1}) &= m_{\sigma((t-1)(d+t)+1)}, F(x_d, y_{z+2}) = m_{\sigma((t-1)(d+t)+2)}, \dots, F(x_d, y_{z+t}) = m_{\sigma(\mathcal{M}^s)}, \end{aligned} \quad (25)$$

where we have used the fact that  $\mathcal{M}^s = 2t(d - z)$  in the last equality. Now, the encoding of the secret  $\mathbf{m}$  in the secret sharing scheme  $\tilde{\mathcal{C}}$  is obtained by puncturing the code symbols  $\mathbf{c}_{z+1}, \mathbf{c}_{z+2}, \dots, \mathbf{c}_{z+t}$  for the codeword  $\mathbf{c} \in \mathcal{C}$  (cf. Figure 5). Thus, the  $N = n - t$  shares associated with the secret  $\mathbf{m}$  in the secret sharing scheme  $\tilde{\mathcal{C}}$  are defined as

$$\tilde{\mathbf{c}} = (\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \dots, \tilde{\mathbf{c}}_N) = (\mathbf{c}_1, \dots, \mathbf{c}_z, \mathbf{c}_{z+t+1}, \dots, \mathbf{c}_n) \in \mathbb{F}_{q^\alpha}^N. \quad (26)$$

We now argue the security and the communication efficiency of the proposed secret sharing scheme.

<sup>8</sup>Each of the  $\mathcal{M}$  symbols in the vector  $\mathbf{f}$  comprises either a symbol from  $\mathbf{m}$  or  $\mathbf{r}$ . Moreover, each symbol of  $\mathbf{m}$  and  $\mathbf{r}$  appears in exactly one coordinate of the vector  $\mathbf{f}$ .

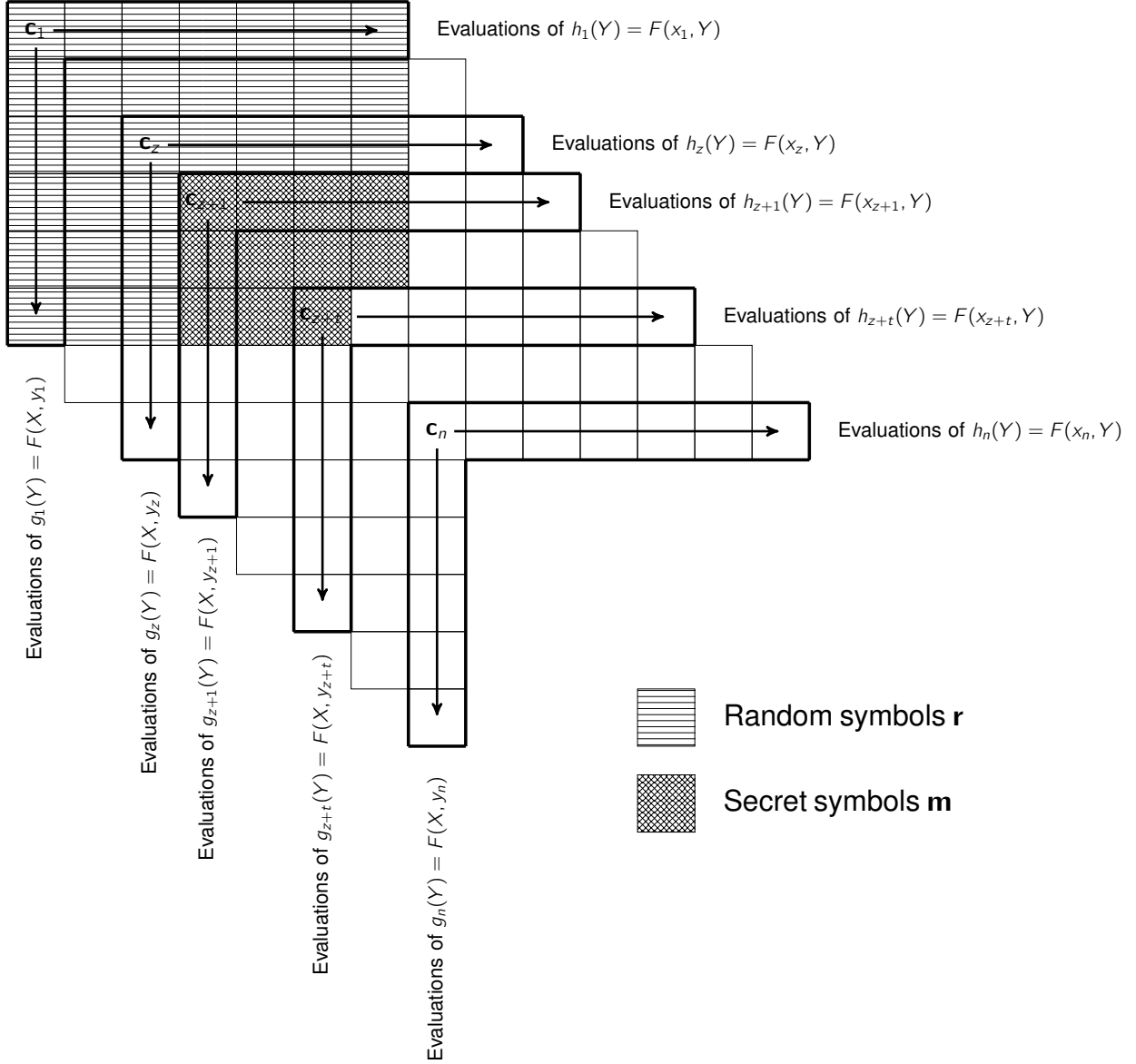


Fig. 4: Description of the MBMR coding scheme with a particular systematic encoding utilized in this paper. Each node (code symbol) is collection of  $d + t$  and  $d$  evaluations of the univariate polynomials  $\{h_i(Y)\}$  and  $\{g_i(X)\}$ , respectively. Note that the systematic encoding ensures that the random symbols  $\mathbf{r}$  and the secret symbols  $\mathbf{m}$  appear in the first  $z$  and the subsequent  $t$  code symbols, respectively.

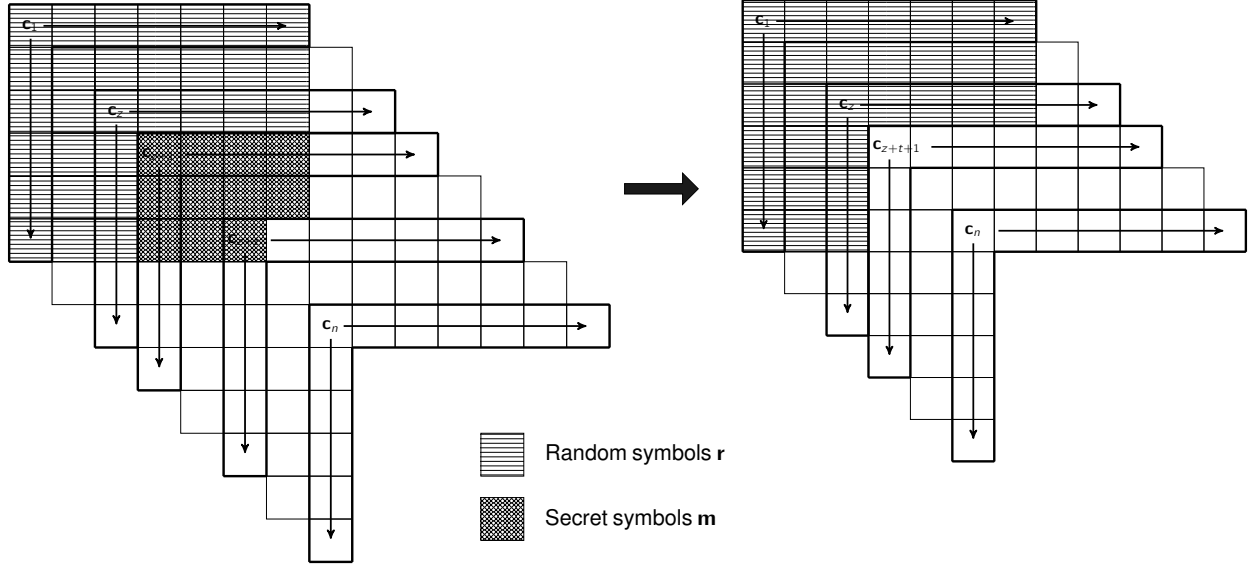


Fig. 5: Puncturing of the secret covering multiple symbols in the MBMR code gives communication efficient secret sharing with multi-share repair property.

- Security: Assume that an adversary has access to  $z$  shares  $\tilde{c}_{i_1}, \tilde{c}_{i_2}, \dots, \tilde{c}_{i_z}$ . Let these shares correspond to the code symbols  $c_{j_1}, c_{j_2}, \dots, c_{j_z}$  in the associated codeword in the MBMR code  $\mathcal{C}$  with  $\{j_1, j_2, \dots, j_z\} \subset [n] \setminus \{z+1, \dots, z+t\}$ . It follows from the Remark 7 that the adversary knows the following univariate polynomials<sup>9</sup>.

$$\{h_{j_s}(Y) = F(x_{j_s}, Y), g_{j_s}(X) = F(X, y_{j_s})\} \text{ for } s = 1, 2, \dots, z. \quad (27)$$

It is argued in [21] that if  $\mathbf{e}$  denote the symbols (in  $\mathbb{F}_q$ ) known to the adversary by observing  $z$  shares, then we have

$$H(\mathbf{e}) \leq H(\mathbf{r}) = \mathcal{M} - \mathcal{M}^s. \quad (28)$$

Next, we argue that given the observations of the adversary (cf. (27)) and the secret  $\mathbf{m}$ , one can decode the random symbols  $\mathbf{r}$ , i.e.,  $H(\mathbf{r}|\mathbf{m}, \mathbf{e}) = 0$ . Note that the secret symbols  $\mathbf{m}$  correspond to part of the code symbols  $c_{z+1}, \dots, c_{z+t}$ . As highlighted in (25), the secret symbols  $\mathbf{m}$  are evaluations of the polynomial  $F(X, Y)$  (cf. (22)). In particular, knowing the secret  $\mathbf{m}$  translates to knowing  $d + t - z$  evaluations of each of the polynomials in  $\{h_j(Y) = F(x_j, Y)\}_{j=z+1, \dots, z+t}$  and  $d - z$  evaluations of each of the polynomials in  $\{g_j(X) = F(X, y_j)\}_{j=z+1, \dots, z+t}$  (cf. (25)). Now, using the observations of the adversary (cf. (27)), we can obtain  $z$  additional observations of each of the polynomials  $\{h_j(Y) =$

<sup>9</sup>Knowing a polynomial means that the adversary knows the coefficients of the polynomials and can evaluate the polynomial at any point in  $\mathbb{F}_q$

$F(x_j, Y), g_j(X) = F(X, y_j)\}_{j=z+1, \dots, z+t}$  as follows.

$$h_j(y_{j_s}) = F(x_j, y_{j_s}) = g_{j_s}(x_j) \text{ for } j \in \{z+1, \dots, z+t\} \text{ and } s \in \{1, \dots, z\} \quad (29)$$

and

$$g_j(x_{j_s}) = F(x_{j_s}, y_j) = h_{j_s}(y_j) \text{ for } j \in \{z+1, \dots, z+t\} \text{ and } s \in \{1, \dots, z\}. \quad (30)$$

Therefore, given the observations of the adversary and the secret symbols one has access to  $k = z + t$  code symbols  $\mathbf{c}_{j_1}, \dots, \mathbf{c}_{j_s}, \mathbf{c}_{z+1}, \dots, \mathbf{c}_{z+t}$  of the associated codeword in  $\mathcal{C}$ . Now one can use a decoding algorithm of  $\mathcal{C}$  to decode  $\mathbf{A}\mathbf{f}$  and subsequently obtain  $\mathbf{f}$ . Note that the random symbols  $\mathbf{r}$  can now be obtained as these symbols constitute  $\mathcal{M} - \mathcal{M}^s$  coordinates of the vector  $\mathbf{f}$ .

From the two observations shown above that  $H(\mathbf{e}) \leq H(\mathbf{r})$  and  $H(\mathbf{r}|\mathbf{e}, \mathbf{m}) = 0$ , it follows that the adversary does not get any information about the secret  $\mathbf{m}$  from the  $z$  shares it has access to [19], [10].

- Communication efficiency: Note that the secret  $\mathbf{m}$  can be obtained by repairing  $t$  code symbols  $\mathbf{c}_{z+1}, \mathbf{c}_{z+2}, \dots, \mathbf{c}_{z+t}$  in the associated codeword  $\mathbf{c} \in \mathcal{C}$ . Since  $\mathcal{C}$  is an MBMR code, this repair process can be performed by contacting a set of  $d$  shares say  $\tilde{\mathbf{c}}_{i_1}, \tilde{\mathbf{c}}_{i_2}, \dots, \tilde{\mathbf{c}}_{i_d}$  and downloading total

$$\gamma_t = \frac{\mathcal{M}2dt}{(z+t)(2d-z)} = \frac{d}{d-z}\mathcal{M}^s$$

symbols (over  $\mathbb{F}_q$ ) from the  $d$  shares we contact (cf. 24). Comparing  $\gamma_t$  to the lower bound on (19) establishes the communication efficiency of the secret sharing scheme based on the MBMR code from [28].

Remark 9. Since the secret sharing scheme  $\tilde{\mathcal{C}}$  is obtained by puncturing  $t$  code symbols in the MBMR code  $\mathcal{C}$ . Assuming that the original MBMR code has  $d \leq N - t = n - 2t$ , we can repair any  $t$  shares in a bandwidth efficient manner by invoking the repair mechanism of the MBMR code  $\mathcal{C}$ . This repair process would involve contacting any set of  $d$  out of remaining  $N - t$  shares and downloading  $\gamma_t = \frac{\mathcal{M}2dt}{(z+t)(2d-z)}$  symbols (over  $\mathbb{F}_q$ ) from the contacted shares.

#### Acknowledgement

The authors would like to thank Salim El Rouayheb for pointing us to the work by Cadambe et al. [8], which includes a bound and an existential result on the repair of multiple failures in an MDS code.

#### References

- [1] A. G. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory*, 56(9):4539–4551, 2010.
- [2] K. Rashmi, N. Shah, and P. Kumar. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Trans. Inf. Theory*, 57:5227–5239, 2011.
- [3] I. Tamo, Z. Wang, and J. Bruck. Zigzag codes: MDS array codes with optimal rebuilding. *IEEE Trans. Inf. Theory*, 59(3):1597–1616, 2013.
- [4] D. Papailiopoulos, A. G. Dimakis, and V. Cadambe. Repair optimal erasure codes through hadamard designs. *IEEE Trans. Inf. Theory*, 59(5):3021–3037, 2013.

- [5] B. Sasidharan, G. K. Agarwal, and P. V. Kumar. A high-rate MSR code with polynomial sub-packetization level. CoRR, abs/1501.06662, 2015.
- [6] K. W. Shum and Y. Hu. Cooperative regenerating codes. *IEEE Transactions on Information Theory*, 59(11):7229–7258, 2013.
- [7] A.-M. Kermarrec, N. Le Scouarnec, and G. Straub. Repairing multiple failures with coordinated and adaptive regenerating codes. In *Proc. of 2011 NetCod*, pages 1–6, 2011.
- [8] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh. Asymptotic interference alignment for optimal repair of mds codes in distributed storage. *IEEE Transactions on Information Theory*, 59(5):2974–2987, May 2013.
- [9] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin. On the locality of codeword symbols. *IEEE Trans. Inf. Theory*, 58(11):6925–6934, 2012.
- [10] 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*, 60(1):212–236, 2014.
- [11] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar. Codes with local regeneration and erasure correction. *IEEE Trans. Inf. Theory*, 60(8):4637–4660, Aug 2014.
- [12] I. Tamo and A. Barg. A family of optimal locally recoverable codes. *IEEE Trans. Inf. Theory*, 60(8):4661–4676, Aug 2014.
- [13] A. S. Rawat, A. Mazumdar, and S. Vishwanath. Cooperative local repair in distributed storage. *EURASIP J. Adv. Signal Process.*, pages 1–17, 2015.
- [14] N. Prakash, V. Lalitha, and P. V. Kumar. Codes with locality for two erasures. In *Proc. of 2014 IEEE International Symposium on Information Theory (ISIT)*, pages 1962–1966, June 2014.
- [15] W. Song and C. Yuen. Locally repairable codes with functional repair and multiple erasure tolerance. *arXiv preprint arXiv:1507.02796*, 2015.
- [16] W. Huang, M. Langberg, J. Kliever, and J. Bruck. Communication efficient secret sharing. CoRR, abs/1505.07515, 2015.
- [17] Rawad Bitar and Salim El Rouayheb. Staircase codes for secret sharing with optimal communication and read overheads. CoRR, abs/1512.02990, 2015.
- [18] S. Pawar, S. El Rouayheb, and K. Ramchandran. Securing dynamic distributed storage systems against eavesdropping and adversarial attacks. *IEEE Transactions on Information Theory*, 57(10):6734–6753, 2011.
- [19] N. B. Shah, K. V. Rashmi, and P. V. Kumar. Information-theoretically secure regenerating codes for distributed storage. In *Proceedings of 2011 IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, 2011.
- [20] K. Huang, U. Parampalli, and M. Xian. Characterization of secrecy capacity for general MSR codes under passive eavesdropping model. CoRR, abs/1505.01986, 2015.
- [21] O. O. Koyluoglu, A. S. Rawat, and S. Vishwanath. Secure cooperative regenerating codes for distributed storage systems. *IEEE Transactions on Information Theory*, 60(9):5228–5244, Sept 2014.
- [22] K. Huang, U. Parampalli, and M. Xian. Security concerns in minimum storage cooperative regenerating codes. CoRR, abs/1509.01324, 2015.
- [23] N. Le Scouarnec. Exact scalar minimum storage coordinated regenerating codes. In *Proceedings of 2012 IEEE International Symposium on Information Theory (ISIT)*, pages 1197–1201, 2012.
- [24] J. Li and B. Li. Cooperative repair with minimum-storage regenerating codes for distributed storage. In *Proc. of 2014 IEEE INFOCOM*, pages 316–324, 2014.
- [25] Z. Wang, I. Tamo, and J. Bruck. Optimal rebuilding of multiple erasures in MDS codes. CoRR, abs/1603.01213, 2016.
- [26] L. Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computing Theory*. Akademie-Verlag, Berlin, 1979.
- [27] N. Alon. Combinatorial nullstellensatz. *Comb. Probab. Comput.*, 8(1-2):7–29, 1999.
- [28] A. Wang and Zhang. Exact cooperative regenerating codes with minimum-repair-bandwidth for distributed storage. In *Proc. of 2013 IEEE INFOCOM*, pages 400–404, 2013.

## Appendix A

### Zigzag codes: simultaneous repair of up to 3 failed systematic nodes

#### A. Description of the zigzag construction [3]

Let  $\mathbb{Z}_r$  denote the set  $\{0, 1, \dots, r-1\}$ . Let  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m \in \mathbb{Z}_r^m$  denote the  $m$  standard  $m$ -dimensional unit vectors. For  $i \in [m]$ , the vector  $\mathbf{e}_i$  has all but one of its coordinates as zero. The vector  $\mathbf{e}_i$  has value 1 at its  $i$ -th coordinate. We use  $\mathbf{e}_0 \in \mathbb{Z}_r^m$  to denote an  $m$ -dimensional all zero vector. For an integer in  $[0, r^m - 1]$  we associate a unique vector from  $\mathbb{Z}_r^m$  as its vector representation. Given  $(m+1)r^m$  information symbols, encoding of an  $(n = m+1+r, k = m+1)$  zigzag code works as follows.

- Arrange the  $(m+1)r^m$  information symbols in an  $r^m \times (m+1)$  array. For  $j \in [m]$  and  $i \in [r^m - 1]$ , let  $x_{i,j}$  denote the  $(i+1)$ -th information symbol in the  $(j+1)$ -th column of the array. The  $k = (m+1)$  columns of the information array represent the  $k$  systematic nodes in the zigzag code construction.
- In order to generate  $r$  parity nodes, for every  $l \in [0, r-1]$  and  $s \in [0, r^m - 1]$ , we define the zigzag set

$$\mathcal{Z}_s^l = \{x_{i,j} : i + l\mathbf{e}_j = s\}. \quad (31)$$

Note that we use the vector representations of  $i, s \in [0, r^m - 1]$  while defining the set  $\mathcal{Z}_s^l$  in (31). Given the zigzag set  $\mathcal{Z}_s^l$ ,  $(s+1)$ -th symbol stored on the  $(l+1)$ -th parity node is linear combination of the information symbols in the zigzag set  $\mathcal{Z}_s^l$ . The coefficients of the linear combinations belong to non-zero elements (multiplicative group) of a large enough finite field.

1) Repair of a single systematic node in zigzag codes [3]: Here, we briefly describe the repair mechanism of a single systematic node in the zigzag code construction. For  $j \in [1, m]$  and  $l \in [0, r-1]$ , we define the set

$$\mathcal{X}_j^l = \{i \in [0, r^m - 1] : i \cdot \mathbf{e}_j = r - l\}. \quad (32)$$

Again, we use the vector representation of the integer  $j \in [0, r^m - 1]$  while defining the set  $\mathcal{X}_j^l$  in (32). For  $j = 0$  and  $l \in [0, r-1]$ , we define the corresponding set  $\mathcal{X}_0^l$  as follows.

$$\mathcal{X}_0^l = \{i \in [0, r^m - 1] : i \cdot (1, 1, \dots, 1) = l\}. \quad (33)$$

For  $j \in [0, m]$ , those information symbols stored on the  $(j+1)$ -th systematic node which are indexed by the set  $\mathcal{X}_j^l$  are recovered by downloading the code symbols from the  $(l+1)$ -th parity node. From the remaining  $k-1 = m$  systematic nodes, we download those symbols which appear in the symbols downloaded from the parity nodes. Combining the definitions in (31), (32) and (33), we obtain the following.

Proposition 7. For  $l \in [0, r-1]$ , let  $\mathcal{D}_l^1$  be the set defined as follows.

$$\mathcal{D}_l^1 = \{i \in [0, r^{k-1} - 1] : i \cdot (1, 1, \dots, 1) = l\}. \quad (34)$$

Furthermore, for  $j \in [1, k-1]$  and  $l \in [0, r-1]$ , we define the set  $\mathcal{D}_l^{j+1}$  as follows.

$$\mathcal{D}_l^{j+1} = \{i \in [0, r^{k-1} - 1] : i \cdot \mathbf{e}_j = 0 \pmod{r}\}. \quad (35)$$

Then, for  $j \in [0, k-1]$  and  $l \in [0, r-1]$ , the set  $\mathcal{D}_l^{j+1}$  denotes the indices of the parity symbols downloaded from  $(l+1)$ -th parity node in order to repair the  $(j+1)$ -th systematic node in the event of a single failure.

2) Structure of symbols downloaded to repair different node in the even of a single node failure: In our approach to repair  $t$  simultaneous node failures, we contact the remaining  $d = n - t$  nodes and download symbols in two stages. In the first stage, for each of the failed  $t$  node, we download those  $\frac{\alpha}{n-k} = r^{k-2}$  symbols from the contacted node which would have been downloaded to repair this node in the event of single node failure. Since some of the symbols from a helper node contribute to the repair of many nodes during the repair of a single node failure, we end up downloading less than  $\frac{t\alpha}{n-k}$  symbols from each of the  $d = n - k$  contacted node. Using the structure of the zigzag code, in the second stage, we then download additional symbols from the helper nodes so that each helper node contributes exactly  $\frac{t\alpha}{n-k}$  symbols. In order to identify which symbols need to be downloaded in the second stage we need to understand the structure of the parity symbols downloaded in the first stage. Therefore, we first explore this.

For the ease of exposition, without loss of generality, we assume that the first  $t$  systematic nodes are in failure, i.e., the systematic nodes indexed by the set  $[t] := \{0, 1, \dots, t-1\}$  experience failure. The analysis for other  $t$  systematic nodes can be carried out in a similar manner. Recall that for  $j \in [0, t-1]$  and  $l \in [0, r-1]$ ,  $\mathcal{D}_l^{j+1}$  denotes the indices of the parity symbols downloaded from  $(l+1)$ -th parity node to repair the  $(j+1)$ -th systematic node failure. The sets  $\{\mathcal{D}_l^{j+1}\}_{j \in [0, t-1], l \in [0, r-1]}$  are defined in Proposition 7. It follows from the definition of these sets, that for any set of  $u$  out of  $t$  failed nodes, say indexed by the set  $\{j_1, j_2, \dots, j_u\} \subseteq [0, t-1]$ , and  $l \in [0, r-1]$ , we have the following.

$$\bigcap_{j \in \{j_1, j_2, \dots, j_u\}} \mathcal{D}_l^{j+1} = r^{k-1-u}. \quad (36)$$

Given this observations, we now define the following families of sets. For  $\{j_1, j_2, \dots, j_u\} \subseteq [0, t-1]$  and  $l \in [0, r-1]$ , we define the set  $\mathcal{U}_l^{\{j_1+1, j_2+1, \dots, j_u+1\}}$  to be the indices of the parity symbols downloaded from the  $(l+1)$ -th parity node which participate in the repair of exactly  $u \leq t$  systematic nodes indexed by the set  $\{j_1, \dots, j_u\}$  in the event of single node failure. In particular, given  $\mathcal{S} \subseteq [t]$  and  $l \in [0, r-1]$ , we have

$$\mathcal{U}_l^{\mathcal{S}} = \bigcap_{j \in \mathcal{S}} \mathcal{D}_l^j \setminus \bigcup_{\mathcal{S}' \subsetneq \mathcal{S}} \bigcap_{j \in \mathcal{S}'} \mathcal{D}_l^j. \quad (37)$$

Combining (37) with the definitions of the sets  $\{\mathcal{D}_l^{j+1}\}_{j \in [0, k-1], l \in [0, r-1]}$ , we obtain that

1) Case 1:  $1 \in \mathcal{S}$ ,

$$\mathcal{U}_l^{\mathcal{S}} = \{i : i \cdot (1, \dots, 1) = l; i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S} \setminus \{1\}; \text{ and } i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [t] \setminus \mathcal{S}\} \subseteq [0, r^{k-1} - 1]. \quad (38)$$

2) Case 2:  $1 \in [t] \setminus \mathcal{S}$ ,

$$\mathcal{U}_l^{\mathcal{S}} = \{i : i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S}; i \cdot (1, \dots, 1) \neq l; \text{ and } i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [2, t] \setminus \mathcal{S}\} \subseteq [0, r^{k-1} - 1]. \quad (39)$$



Moreover, we have that

$$\begin{aligned}
|\mathcal{U}_l^{\mathcal{S}}| &= \left| \bigcap_{j \in \mathcal{S}} \mathcal{D}_l^j \right| - \left| \bigcup_{\mathcal{S} \subsetneq \mathcal{S}' \subseteq [t]} \bigcap_{j \in \mathcal{S}'} \mathcal{D}_l^j \right| \\
&\stackrel{(a)}{=} r^{k-1-|\mathcal{S}|} \left( 1 - \frac{1}{r} \right)^{t-|\mathcal{S}|},
\end{aligned} \tag{40}$$

where (a) follows from (38) and (39). Note that, by construction, for a fixed  $l \in [0, r-1]$ , the family of sets  $\{\mathcal{U}_l^{\mathcal{S}}\}_{\mathcal{S} \subseteq [0, t-1]}$  comprises disjoint sets. In case of  $t = 3$ , for a fixed value of  $l \in [0 : r-1]$ , this gives us the following sequences of disjoint sets,  $\mathcal{U}_l^{\{1\}}, \mathcal{U}_l^{\{2\}}, \mathcal{U}_l^{\{3\}}, \mathcal{U}_l^{\{1,2\}}, \mathcal{U}_l^{\{2,3\}}, \mathcal{U}_l^{\{1,3\}}, \mathcal{U}_l^{\{1,2,3\}}$ . Here, for  $j \in [0, 2]$ ,  $\mathcal{U}_l^{\{j+1\}}$  denotes the indices of those symbols from the  $(l+1)$ -th parity node which participate only in the repair of  $(j+1)$ -th systematic node in the event of single node failure. The sets  $\mathcal{U}_l^{\{1,2\}}$  represent the sets of symbols from  $(l+1)$ -th parity node that participate only in the repair of 1st and 2nd systematic nodes in the event of single node failure. Similarly, the parity symbols from  $(l+1)$ th parity nodes that enable repair of each of the first 3 systematic nodes in the event of a single node failure are represented by  $\mathcal{U}_l^{\{1,2,3\}}$ .

We now characterize the sets of information symbols on the lost (failed) systematic nodes that participate in the sets  $\{\mathcal{U}_l^{\mathcal{S}}\}_{\mathcal{S} \subseteq [t], l \in [0, r-1]}$ . For a given  $j \in [0, k-1]$ ,  $l \in [0, r-1]$ , and  $\mathcal{S} \subseteq [t]$ , let  $\mathcal{U}_{j \rightarrow l}^{\mathcal{S}}$  denotes the indices of the symbols from the  $(j+1)$ -th systematic node which participate in the parity symbols in the set  $\mathcal{U}_l^{\mathcal{S}}$ . Using (31) and (38), we can explicitly characterize these sets. In particular, we consider 3 different case.

- 1) Case 1 (a) :  $j = 0$  and  $1 \in \mathcal{S}$ ,

$$\begin{aligned}
\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} &\stackrel{(a)}{=} \{i \in [0, r-1]^{k-1} : i \in \mathcal{U}_l^{\mathcal{S}}\} \\
&\stackrel{(b)}{=} \{i : i \cdot (1, \dots, 1) = l; i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S} \setminus \{1\}; \text{ and } i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [t] \setminus \mathcal{S} \subseteq [0, r^{k-1} - 1].\}
\end{aligned} \tag{41}$$

- 2) Case 1 (b) :  $j = 0$  and  $1 \in [t] \setminus \mathcal{S}$ ,

$$\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} = \{i : i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S}; i \cdot (1, \dots, 1) \neq l; \text{ and } i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [2, t] \setminus \mathcal{S} \subseteq [0, r^{k-1} - 1].\} \tag{42}$$

- 3) Case 2 (a) :  $j \neq 0$  and  $\{1, j+1\} \subseteq \mathcal{S}$ ,

$$\begin{aligned}
\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} &\stackrel{(a)}{=} \{i \in [0, r-1]^{k-1} : i + l\mathbf{e}_j \in \mathcal{U}_l^{\mathcal{S}}\} \\
&\stackrel{(b)}{=} \{i : i \cdot (1, \dots, 1) = 0; i \cdot \mathbf{e}_j + l = 0; i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S} \setminus \{1, j+1\}; \text{ and } \\
&\quad i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [t] \setminus \mathcal{S} \subseteq [0, r-1]^{k-1}.\}
\end{aligned} \tag{43}$$

- 4) Case 2 (b) :  $j \neq 0$ ,  $1 \in \mathcal{S}$  and  $j+1 \in [t] \setminus \mathcal{S}$ ,

$$\begin{aligned}
\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} &= \{i : i \cdot (1, \dots, 1) = 0; i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S} \setminus \{1\}; i \cdot \mathbf{e}_j + l \neq 0 \text{ and } \\
&\quad i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [t] \setminus \{\mathcal{S} \cup \{j+1\}\} \subseteq [0, r-1]^{k-1}.\}
\end{aligned} \tag{44}$$

5) Case 2 (c):  $j \neq 0$ ,  $1 \in [t] \setminus \mathcal{S}$  and  $j+1 \in \mathcal{S}$ ,

$$\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} = \{i : i \cdot \mathbf{e}_j + l = 0; i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S} \setminus \{j+1\}; i \cdot (1, \dots, 1) \neq 0; \text{ and} \\ i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [2, t] \setminus \mathcal{S}\} \subseteq [0, r-1]^{k-1}. \quad (45)$$

6) Case 2 (d):  $j \neq 0$  and  $\{1, j+1\} \in [t] \setminus \mathcal{S}$ ,

$$\mathcal{U}_{j \rightarrow l}^{\mathcal{S}} = \{i : i \cdot \mathbf{e}_{w-1} = 0 \ \forall w \in \mathcal{S}; i \cdot (1, \dots, 1) \neq 0; i \cdot \mathbf{e}_j + l \neq 0; \text{ and} \\ i \cdot \mathbf{e}_{v-1} \neq 0 \ \forall v \in [2, t] \setminus \{\mathcal{S} \cup \{j+1\}\}\} \subseteq [0, r-1]^{k-1}. \quad (46)$$

## B. Repairing $t = 2$ failed nodes

1) First stage of the download process: In the first stage, we download the symbols which enable the repair of 1st and 2nd systematic nodes in the event of single node failure. In particular, for  $l \in [0, r-1]$ , we download the parity symbols indexed by the set

$$\mathcal{D}_l^1 \cup \mathcal{D}_l^2 = \mathcal{U}_l^{\{1\}} \cup \mathcal{U}_l^{\{2\}} \cup \mathcal{U}_l^{\{1,2\}}$$

from the  $(l+1)$ -th parity node. From the remaining  $k-2$  systematic nodes, we download those systematic symbols which appear in these parity nodes.

Sets	$l = 0$	$l = 1$	$\dots$	$l = r-1$
$\mathcal{U}_l^{\{1\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1\}}, \mathcal{U}_{1 \rightarrow 0}^{\{1\}} = \begin{pmatrix} \mathcal{U}_{1 \rightarrow 1}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1\}}, \mathcal{U}_{1 \rightarrow 1}^{\{1\}} = \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1\}}, \mathcal{U}_{1 \rightarrow r-1}^{\{1\}} = \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 1}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-2}^{\{1,2\}} \end{pmatrix}$
$\mathcal{U}_l^{\{2\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{2\}} = \begin{pmatrix} \mathcal{U}_{0 \rightarrow 1}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 0}^{\{2\}}$	$\mathcal{U}_{0 \rightarrow 1}^{\{2\}} = \begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 1}^{\{2\}}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{2\}} = \begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 1}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-2}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow r-1}^{\{2\}}$
$\mathcal{U}_l^{\{1,2\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow 1}^{\{1,2\}}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}}$

TABLE I: Composition of parity symbols downloaded in the first stage of the repair process to repair of 1st and 2nd systematic nodes. Blue color symbols correspond to the matched symbols from the 1st systematic node. Red color symbols represent the matched symbols from the 2nd system node.

We next illustrate a strategy to match the symbols from the failed systematic nodes using the downloaded symbols illustrated in Table I. Formally, we have the following.

- Matching symbols from the 2nd systematic node using parity symbols  $\{\mathcal{U}_l^{\{1,2\}}\}$ : We use the symbols for the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1,2\}}$  to match those symbols from the 2nd systematic node that are indexed by the set

$$\mathcal{U}_{1 \rightarrow l}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 = r - l\}.$$

- Matching symbols from the 1st systematic node using parity symbols  $\{\mathcal{U}_l^{\{1\}}\}$ : We use the symbols for the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1\}}$  to match those symbols from the 1st systematic node that are indexed by the set

$$\mathcal{U}_{0 \rightarrow l}^{\{1\}} = \{i : i \cdot (1, \dots, 1) = l, i \cdot \mathbf{e}_1 \neq 0\}.$$

- Matching symbols from the 1st and the 2nd systematic node using parity symbols  $\{\mathcal{U}_l^{\{2\}}\}$ : We use the symbols for the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1\}}$  to match those symbols from the 1st systematic node that are indexed by the set

$$\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 = 0\}. \quad (47)$$

We use the remaining  $|\mathcal{U}_{1 \rightarrow l}^{\{2\}}| - |\mathcal{U}_{0 \rightarrow (l+1)}^{\{1,2\}}| = (r^{k-2} - 2r^{k-3})$  parity symbols to match the symbols from the 2nd systematic node with the following indices.

$$\{i : i \cdot (1, \dots, 1) \notin \{0, 1\}, i \cdot \mathbf{e}_1 = r - l\} \subseteq \mathcal{U}_{1 \rightarrow l}^{\{2\}} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r - l\}. \quad (48)$$

As it is clear from Table I and the matching processing described above, the following number of symbols from the 2 failed nodes are matched by the parity symbols downloaded in the first stage.

- 1) Number of symbols matched from 1st systematic node (blue colored):

$$\underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{0 \rightarrow l}^{\{1\}}|}_{\text{from } \mathcal{U}_l^{\{1\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}}|}_{\text{from } \mathcal{U}_l^{\{2\}}} = r \cdot (r^{k-2} - r^{k-3}) + r \cdot r^{k-3} = r^{k-1} = \alpha. \quad (49)$$

- 2) Number of symbols matched from 2nd systematic node (red colored):

$$\underbrace{\sum_{l=0}^{r-1} (|\mathcal{U}_{1 \rightarrow l}^{\{2\}}| - |\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}}|)}_{\text{from } \mathcal{U}_l^{\{2\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{1 \rightarrow l}^{\{1,2\}}|}_{\text{from } \mathcal{U}_l^{\{1,2\}}} = r \cdot (r^{k-2} - 2r^{k-3}) + r \cdot r^{k-3} = r^{k-1} - r^{k-2} = \alpha - r^{k-2}. \quad (50)$$

- 2) Second Stage of download process:

- Unmatched symbols from the second systematic node : It follows from (50) that it remains to match  $r^{k-2}$  symbols from the 2nd systematic node. This requires us to download additional symbols from the intact nodes. Towards this, let's consider the unmatched symbols from the second systematic node in the  $(l+1)$ -th parity node (cf. Table I). These are exactly those symbols among the symbols indexed by the set  $\mathcal{U}_{1 \rightarrow l}^{\{2\}}$  which form linear combinations with the symbols indexed by the set  $\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}}$ . It can

be easily deduced from (48) that the unmatched symbols from the second systematic node have the following indices.

$$\mathcal{R}_{1 \rightarrow l} = \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r - l\} \quad (51)$$

Another way to see this is as follows. Using (41), we know that

$$\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = l + 1, i \cdot \mathbf{e}_1 = 0\}. \quad (52)$$

Utilizing the definition of the zigzag set (cf. (31)), the symbols from the first systematic nodes which are indexed by the set  $\mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}}$  appear in those parity symbols in the  $(l + 1)$ -th parity node which are indexed by the set

$$\{i : i \in \mathcal{U}_{0 \rightarrow l+1}^{\{1,2\}}\} \subseteq [0 : r^k - 1]. \quad (53)$$

We again utilize the definition of the zigzag sets (cf. (31)) to identify the symbols from the second systematic node that appear in those parity symbols from the  $(l + 1)$ -th parity node which are indexed by the set defined in (53). These are exactly the symbols indexed by the following set.

$$\begin{aligned} \mathcal{R}_{1 \rightarrow l} &= \{i : i \cdot (1, \dots, 1) + l = l + 1, i \cdot \mathbf{e}_1 + l = 0\} \\ &= \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r - l\} \end{aligned} \quad (54)$$

- Additional symbols downloaded to match remaining symbols from the second systematic node : Now, let's consider an integer  $i^* \in [0, r^m - 1] = [0, r^{k-1} - 1]$  such that the following two conditions hold.
  - 1)  $i^* \cdot (1, 1, \dots, 1) = 1$ . Note that we are using vector representation of  $i^*$  in  $\mathbb{Z}_r^{k-1} = \mathbb{Z}_r^m$  in defining this relationship.
  - 2)  $i^* \cdot \mathbf{e}_1 = r - 1$ , i.e., the first coordinate of the vector representation of  $i^*$  takes the nonzero value  $r - 1$ .

For  $j \in [2, k - 1]$ , the set of additional symbols downloaded from the  $(j + 1)$ -th systematic node have their row indices belonging to the following set.

$$\mathcal{S}^{\{1,2\}} = \{i^* + a_1(\mathbf{e}_3 - \mathbf{e}_2) + a_2(\mathbf{e}_4 - \mathbf{e}_2) + \dots + a_{k-3}(\mathbf{e}_{k-1} - \mathbf{e}_2) : (a_1, \dots, a_{k-3}) \in [r - 1]^{k-3}\}. \quad (55)$$

Note that we have  $|\mathcal{S}^{\{1,2\}}| = r^{k-3}$ . The reason behind this particular choice for the set  $\mathcal{S}^{\{1,2\}}$  will become clear very soon. Now, let's focus on the additional code symbols that need to be downloaded from the parity nodes. For  $l \in [0 : r - 1]$ , we download those parity symbols from the  $(l + 1)$ -th parity node which involve the information symbols associated with the set  $\mathcal{S}^{\{1,2\}}$ . Recall that one can use the definitions of the zigzag sets (cf. (31)) to identify these additional parity symbols that need to be downloaded. In particular, for  $l = 0$ , the additional symbols downloaded from the 1st parity node have their row indices belonging to the set

$$\mathcal{P}_0^{\{1,2\}} = \mathcal{S}^{\{1,2\}}. \quad (56)$$

In general, for  $l \in [0, r-1]$ , the additional parity symbols downloaded from the  $(l+1)$ -th parity nodes have their row indices belonging to the following sets

$$\mathcal{P}_l^{\{1,2\}} = \mathcal{S}^{\{1,2\}} + l \cdot \mathbf{e}_2 = \mathcal{S}^{\{1,2\}} + l \cdot \mathbf{e}_3 = \dots = \mathcal{S}^{\{1,2\}} + l \cdot \mathbf{e}_{k-1}. \quad (57)$$

Now, let's make sure if these additional parity symbols indexed by the sets  $\{\mathcal{P}_l^{\{1,2\}}\}_{l \in [0:r-1]}$  indeed help us match the unmatched symbols from the second systematic node. Let's first consider the parity symbols indexed by the set  $\mathcal{P}_0^{\{1,2\}}$ . The symbols from the second systematic node which can be matched using these parity symbols are the ones indexed by the set  $\mathcal{P}_0^{\{1,2\}} = \mathcal{S}^{\{1,2\}} \subset [0:r-1]^k$  itself, i.e.,

$$\{i : i \cdot (1, 1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r-1\}. \quad (58)$$

Note that these are exactly those symbols which remained unmatched as they appear together those symbols from the first systematic nodes that are indexed by the set  $\mathcal{U}_{0 \rightarrow 2}^{\{1,2\}}$  in the 2nd parity node, i.e., the symbols from the second systematic node which are indexed by the set  $\mathcal{R}_{1 \rightarrow 1}$  (cf. 54). Similarly, one can show that the symbols from the second systematic node which can potentially be matched using the additional symbols downloaded from the  $(l+1)$ -th parity node, i.e., the parity symbols indexed by the set  $\mathcal{P}_l^{\{1,2\}}$ , are associated with the set.

$$\{i : i + l\mathbf{e}_1 \in \mathcal{P}_l^{\{1,2\}}\} = \{i : i \cdot (1, 1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r - (l+1)\}. \quad (59)$$

Note that these are exactly those symbols denoted by the set  $\mathcal{R}_{1 \rightarrow l+1}$ . That is, the symbols from second systematic node which remained unmatched as they appear together those symbols from the first systematic nodes that are indexed by the set  $\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}}$  in the  $(l+2)$ -th parity node.

### C. Repairing $t = 3$ failed nodes

1) First stage of the download process: In the first stage, we download the symbols which enable the repair of the first 3 systematic nodes in the event of single node failure. In particular, for  $l \in [0, r-1]$ , we download the parity symbols indexed by the set

$$\mathcal{D}_l^1 \cup \mathcal{D}_l^2 \cup \mathcal{D}_l^3 = \mathcal{U}_l^{\{1\}} \cup \mathcal{U}_l^{\{2\}} \cup \mathcal{U}_l^{\{3\}} \cup \mathcal{U}_l^{\{1,2\}} \cup \mathcal{U}_l^{\{2,3\}} \cup \mathcal{U}_l^{\{1,3\}} \cup \mathcal{U}_l^{\{1,2,3\}}$$

from the  $(l+1)$ -th parity node. From the remaining  $k-2$  systematic nodes, we download those systematic symbols which appear in these parity nodes.

- Matching symbols from first and second systematic node using parity symbols  $\{\mathcal{U}_l^{\{1,2,3\}}\}$ : We use the symbols for the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1,2,3\}}$  to match those symbols from the third systematic node that are indexed by the set

$$\mathcal{U}_{2 \rightarrow l}^{\{1,2,3\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = r-l\}.$$

- Matching symbols from first and second systematic node using parity symbols  $\{\mathcal{U}_l^{\{1,3\}}\}$ : We propose the following matching scheme for the symbols from the first and the second systematic node. Given

Sets	$l = 0$	$l = 1$	$\dots$	$l = r - 1$
$\mathcal{U}_l^{\{1\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 1}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 1}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{1,3\}} \end{pmatrix}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{1,3\}} \end{pmatrix}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-2}^{\{1,2\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-2}^{\{1,3\}} \end{pmatrix}$
$\mathcal{U}_l^{\{2\}}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 1}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 3}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 0}^{\{2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 1}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{2,3\}} \end{pmatrix}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 3}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 1}^{\{2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{2,3\}} \end{pmatrix}$	$\dots$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 1}^{\{1,2\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-2}^{\{1,2\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow r-1}^{\{2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{2 \rightarrow 3}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-2}^{\{2,3\}} \end{pmatrix}$
$\mathcal{U}_l^{\{3\}}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 1}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 3}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 4}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,3\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 4}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow 0}^{\{3\}}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 3}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 4}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,3\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 4}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow 1}^{\{3\}}$	$\dots$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 1}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,3\}} \\ \mathcal{U}_{0 \rightarrow 3}^{\{1,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-2}^{\{1,3\}} \end{pmatrix}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{2,3\}} \\ \mathcal{U}_{1 \rightarrow 4}^{\{2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-2}^{\{2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow r-1}^{\{3\}}$
$\mathcal{U}_l^{\{1,2\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow 0}^{\{1,2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 1}^{\{1,2,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow 1}^{\{1,2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{2 \rightarrow 2}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1,2\}}, \mathcal{U}_{1 \rightarrow r-1}^{\{1,2\}}, \begin{pmatrix} \mathcal{U}_{2 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{2 \rightarrow 1}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{2 \rightarrow r-2}^{\{1,2,3\}} \end{pmatrix}$
$\mathcal{U}_l^{\{2,3\}}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 1}^{\{1,2,3\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 0}^{\{2,3\}}, \mathcal{U}_{2 \rightarrow 0}^{\{2,3\}}$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{0 \rightarrow 2}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow 1}^{\{2,3\}}, \mathcal{U}_{2 \rightarrow 1}^{\{2,3\}}$	$\dots$	$\begin{pmatrix} \mathcal{U}_{0 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{0 \rightarrow 1}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{0 \rightarrow r-2}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{1 \rightarrow r-1}^{\{2,3\}}, \mathcal{U}_{2 \rightarrow r-1}^{\{2,3\}}$
$\mathcal{U}_l^{\{1,3\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1,3\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 1}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow 0}^{\{1,3\}}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1,3\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 3}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-1}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow 1}^{\{1,3\}}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1,3\}}, \begin{pmatrix} \mathcal{U}_{1 \rightarrow 0}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 1}^{\{1,2,3\}} \\ \mathcal{U}_{1 \rightarrow 2}^{\{1,2,3\}} \\ \vdots \\ \mathcal{U}_{1 \rightarrow r-2}^{\{1,2,3\}} \end{pmatrix}, \mathcal{U}_{2 \rightarrow r-1}^{\{1,3\}}$
$\mathcal{U}_l^{\{1,2,3\}}$	$\mathcal{U}_{0 \rightarrow 0}^{\{1,2,3\}}, \mathcal{U}_{1 \rightarrow 0}^{\{1,2,3\}}, \mathcal{U}_{2 \rightarrow 0}^{\{1,2,3\}}$	$\mathcal{U}_{0 \rightarrow 1}^{\{1,2,3\}}, \mathcal{U}_{1 \rightarrow 1}^{\{1,2,3\}}, \mathcal{U}_{2 \rightarrow 1}^{\{1,2,3\}}$	$\dots$	$\mathcal{U}_{0 \rightarrow r-1}^{\{1,2,3\}}, \mathcal{U}_{1 \rightarrow r-1}^{\{1,2,3\}}, \mathcal{U}_{2 \rightarrow r-1}^{\{1,2,3\}}$

TABLE II: Composition of parity symbols downloaded during repair of  $t = 3$  node failures. Blue colored symbols correspond to the matched symbols from the 1st systematic node. Red colored symbols represent the matched symbols from the 2nd system node. Green colored symbols are used to denote the matched symbols from the 3rd systematic node.

the parity symbols from the  $(l + 1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1,3\}}$ , we use them to match those symbols from the second systematic node which are indexed by the set

$$\mathcal{U}_{1 \rightarrow l+3}^{\{1,2,3\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 = r - (l + 3), i \cdot \mathbf{e}_2 = 0\}. \quad (60)$$

This would allow us to use the remaining  $|\mathcal{U}_l^{\{1,3\}}| - |\mathcal{U}_{1 \rightarrow l+3}^{\{1,2,3\}}| = (r - 2)r^{k-4}$  parity symbols indexed by the set  $\mathcal{U}_l^{\{1,3\}}$  to match those symbols from the second systematic node which are indexed by the

following set.

$$\{i : i \cdot (1, \dots, 1) = l, i \cdot \mathbf{e}_1 \notin \{0, r-3\}, i \cdot \mathbf{e}_2 = 0\} \subset \mathcal{U}_{0 \rightarrow l}^{\{1,3\}} = \{i : i \cdot (1, \dots, 1) = l, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = 0\}. \quad (61)$$

- Matching symbols from first and third systematic node using parity symbols  $\{\mathcal{U}_l^{\{2,3\}}\}$ : We propose the following matching scheme for the symbols from the first and the second systematic node. Given the parity symbols from the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{2,3\}}$ , we use them to match those symbols from the first systematic node which are indexed by the set

$$\mathcal{U}_{0 \rightarrow l+1}^{\{1,2,3\}} = \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = 0\}. \quad (62)$$

This would allow us to use the remaining  $|\mathcal{U}_l^{\{2,3\}}| - |\mathcal{U}_{0 \rightarrow l+1}^{\{1,2,3\}}| = (r-2)r^{k-4}$  parity symbols indexed by the set  $\mathcal{U}_l^{\{2,3\}}$  to match those symbols from the third systematic node which are indexed by the following set.

$$\{i : i \cdot (1, \dots, 1) \notin \{0, 1\}, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = r-2\} \subset \mathcal{U}_{2 \rightarrow l}^{\{2,3\}} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = 0\}. \quad (63)$$

- Matching symbols from second systematic node using parity symbols  $\{\mathcal{U}_l^{\{1,2\}}\}$ : Given the parity symbols from the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{1,2\}}$ , we use them to match those symbols from the second systematic node which are indexed by the set

$$\mathcal{U}_{1 \rightarrow l}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 = r-l, i \cdot \mathbf{e}_2 = 0\}. \quad (64)$$

- Matching symbols from first and third systematic node using parity symbols  $\{\mathcal{U}_l^{\{1\}}\}$ : We utilize the parity symbols from the  $(l+1)$ -th parity node which are indexed by the set  $\{\mathcal{U}_l^{\{1\}}\}$  to match the symbols from the third systematic node with the following indices.

$$\mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r-(l+1)\}. \quad (65)$$

The remaining  $|\mathcal{U}_l^{\{1\}}| - |\mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}}| = r^{k-4}(r-1)^2 - r^{k-4}(r-1)$  unused parity symbols indexed by the set  $\mathcal{U}_l^{\{1\}}$  are used to match the symbols from the first systematic appearing in those parity symbols.

- Matching symbols from first and third systematic node using parity symbols  $\{\mathcal{U}_l^{\{2\}}\}$ : Using the parity symbols from the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{2\}}$ , we match the symbols from the first systematic node with the following indices.

$$\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = l+2, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 \neq 0\}. \quad (66)$$

$$(67)$$

In addition, we also use these parity symbols to match the symbols from the third systematic node with the following indices.

$$\widehat{\mathcal{U}}_{2 \rightarrow l+2}^{\{2,3\}} = \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = r-(l+2)\} \subset \mathcal{U}_{2 \rightarrow l+2}^{\{2,3\}}. \quad (68)$$

This leaves us with  $|\mathcal{U}_l^{\{2\}}| - |\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}}| - |\widehat{\mathcal{U}}_{2 \rightarrow l+2}^{\{2,3\}}| = r^{k-4}(r-1)^2 - r^{k-4}(r-1) - r^{k-4}$  unused parity symbols, which we use to match the symbols from the second systematic node appearing in those parity symbols.

- Matching symbols from first and third systematic node using parity symbols  $\{\mathcal{U}_l^{\{3\}}\}$ : Using the parity symbols from the  $(l+1)$ -th parity node which are indexed by the set  $\mathcal{U}_l^{\{3\}}$ , we match the symbols from the second systematic node with the following indices.

$$\mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r - (l+2), i \cdot \mathbf{e}_2 = 0\}. \quad (69)$$

In addition, we also use these parity symbols to match the symbols from the first systematic node with the following indices.

$$\widehat{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}} \{i : i \cdot (1, \dots, 1) = l+3, i \cdot \mathbf{e}_1 = r-3, i \cdot \mathbf{e}_2 = 0\} \subset \mathcal{U}_{0 \rightarrow l+3}^{\{1,3\}}. \quad (70)$$

We utilize the remaining  $|\mathcal{U}_l^{\{3\}}| - |\mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}}| - |\widehat{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}}| = r^{k-4}(r-1)^2 - r^{k-4}(r-1) - r^{k-4}$  unused parity symbols, which we use to match the symbols from the second systematic node appearing in those parity symbols.

This concludes the first stage of the downloading process and we have utilized all the parity symbols downloaded in the first stage to match certain systematic symbols corresponding to the three failed nodes. We now move to the second stage of the download process where we download additional symbols in order to match the remaining unmatched symbols associated with the three failed systematic nodes. We illustrate our strategy to match the symbols from the failed systematic nodes using the downloaded symbols during the first stage in Table II. Let's count the number of symbols from different failed systematic nodes that are matched according to Table II.

- 1) Symbols from 1st systematic node (blue colored):

$$\begin{aligned} & \underbrace{\sum_{l=0}^{r-1} \left( |\mathcal{U}_{0 \rightarrow l}^{\{1,3\}}| - |\mathcal{U}_{1 \rightarrow l+3}^{\{1,2,3\}}| \right)}_{\text{from } \mathcal{U}_l^{\{1,3\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{0 \rightarrow l+1}^{\{1,2,3\}}|}_{\text{from } \mathcal{U}_l^{\{2,3\}}} + \underbrace{\sum_{l=0}^{r-1} |\widehat{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}}|}_{\text{from } \mathcal{U}_l^{\{3\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}}|}_{\text{from } \mathcal{U}_l^{\{2\}}} + \underbrace{\sum_{l=0}^{r-1} \left( |\mathcal{U}_{0 \rightarrow l}^{\{1\}}| - |\mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}}| \right)}_{\text{from } \mathcal{U}_l^{\{1\}}} \\ &= (r^{k-2} - r^{k-3} - r^{k-3}) + r^{k-3} + r^{k-3} + (r^{k-2} - r^{k-3}) + (r^{k-1} - 2r^{k-2} + r^{k-3} - (r^{k-2} - r^{k-3})) \\ &= r^{k-1} - (r^{k-2} - r^{k-3}). \end{aligned} \quad (71)$$

- 2) Symbols from 2nd systematic node (red colored):

$$\begin{aligned} & \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{1 \rightarrow l+3}^{\{1,2,3\}}|}_{\text{from } \mathcal{U}_l^{\{1,3\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{1 \rightarrow l}^{\{1,2\}}|}_{\text{from } \mathcal{U}_l^{\{1,2\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}}|}_{\text{from } \mathcal{U}_l^{\{3\}}} + \underbrace{\sum_{l=0}^{r-1} \left( |\mathcal{U}_{1 \rightarrow l}^{\{2\}}| - |\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}}| - |\widehat{\mathcal{U}}_{1 \rightarrow l+2}^{\{2,3\}}| \right)}_{\text{from } \mathcal{U}_l^{\{2\}}} \\ &= r^{k-3} + (r^{k-2} - r^{k-3}) + (r^{k-2} - r^{k-3}) + ((r^{k-1} - 2r^{k-2} + r^{k-3}) - (r^{k-2} - r^{k-3}) - r^{k-3}) \\ &= r^{k-1} - r^{k-2}. \end{aligned} \quad (72)$$



3) Symbols from 3rd systematic node (green colored):

$$\begin{aligned}
& \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{2 \rightarrow l}^{\{1,2,3\}}|}_{\text{from } \mathcal{U}_l^{\{1,2,3\}}} + \underbrace{\sum_{l=0}^{r-1} (|\mathcal{U}_{2 \rightarrow l}^{\{2,3\}}| - |\mathcal{U}_{0 \rightarrow l+1}^{\{1,2,3\}}|)}_{\text{from } \mathcal{U}_l^{\{2,3\}}} + \\
& \underbrace{\sum_{l=0}^{r-1} (|\mathcal{U}_{2 \rightarrow l}^{\{3\}}| - |\widehat{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}}| - |\mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}}|)}_{\text{from } \mathcal{U}_l^{\{3\}}} + \underbrace{\sum_{l=0}^{r-1} |\mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}}|}_{\text{from } \mathcal{U}_l^{\{1\}}} + \underbrace{\sum_{l=0}^{r-1} |\widehat{\mathcal{U}}_{1 \rightarrow l+2}^{\{2,3\}}|}_{\text{from } \mathcal{U}_l^{\{2\}}} \\
& = r^{k-3} + ((r^{k-2} - r^{k-3}) - r^{k-3}) + \\
& \quad ((r^{k-1} - 2r^{k-2} + r^{k-3}) - r^{k-3} - (r^{k-2} - r^{k-3})) + (r^{k-2} - r^{k-3}) + r^{k-3} \\
& = r^{k-1} - r^{k-2}.
\end{aligned} \tag{73}$$

2) Second Stage of download process: First, let's identify the unmatched systematic symbols at the end of the first stage of the downloading process.

- Unmatched symbols from the first systematic node: The symbols from the second systematic node that are matched using the parity symbols from the  $(l+1)$ -th parity node are indexed by the set

$$\mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}} = \{i : i \cdot (1, \dots, 1) = 0, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r - (l+1)\}. \tag{74}$$

Using (31), we can identify the indices of the parity symbols from the  $(l+1)$ -parity node where these symbols participate is as follows.

$$\mathcal{Z}_{l,(2 \rightarrow l+1)}^{\{1,3\}} = \mathcal{U}_{2 \rightarrow l+1}^{\{1,3\}} + l\mathbf{e}_2 = \{i : i \cdot (1, \dots, 1) = l, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r - 1\}. \tag{75}$$

The symbols from the first systematic node which remain unmatched at the end of first stage (and require downloading additional symbols) due to their participation in the parity symbols indexed by the set  $\mathcal{Z}_{l,(2 \rightarrow l+1)}^{\{1,3\}}$  in the  $(l+1)$ -th parity node are as follows.

$$\mathcal{R}_{0 \rightarrow l} = \{i : i \in \mathcal{Z}_{l,(2 \rightarrow l+1)}^{\{1,3\}}\} = \{i : i \cdot (1, \dots, 1) = l, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r - 1\}. \tag{76}$$

- Unmatched symbols from the second systematic node: The symbols from the first systematic node that are (potentially) matched using the parity symbols downloaded from the  $(l+1)$ -th parity node during the first stage are indexed by the following two sets.

$$\mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = l+2, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 \neq 0\}. \tag{77}$$

$$\tilde{\mathcal{U}}_{2 \rightarrow l+2}^{\{2,3\}} = \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = r - (l+2)\} \subset \mathcal{U}_{2 \rightarrow l+2}^{\{2,3\}}. \tag{78}$$

Using (31), we can identify the indices of the parity symbols from the  $(l+1)$ -parity node where these symbols participate as follows.

$$\mathcal{Z}_{l,(0 \rightarrow l+2)}^{\{1,2\}} = \mathcal{U}_{0 \rightarrow l+2}^{\{1,2\}} = \{i : i \cdot (1, \dots, 1) = l+2, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 \neq 0\}. \tag{79}$$

$$\tilde{\mathcal{Z}}_{l,(2 \rightarrow l+2)}^{\{2,3\}} = \tilde{\mathcal{U}}_{2 \rightarrow l+2}^{\{2,3\}} + l\mathbf{e}_2 = \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 = 0, i \cdot \mathbf{e}_2 = r - 2\}. \tag{80}$$

The symbols from the second systematic node which remain unmatched at the end of first stage (and require downloading additional symbols) due to their participation in the parity symbols indexed by the set  $\mathcal{Z}_{l,(0 \rightarrow l+2)}^{\{1,2\}} \cup \tilde{\mathcal{Z}}_{l,(2 \rightarrow l+2)}^{\{2,3\}}$  in the  $(l+1)$ -th parity node are as follows.

$$\mathcal{R}_{1 \rightarrow l} = \{i : i + l\mathbf{e}_1 \in \mathcal{Z}_{l,(0 \rightarrow l+2)}^{\{1,2\}}\} = \{i : i \cdot (1, \dots, 1) = 2, i \cdot \mathbf{e}_1 = r - l, i \cdot \mathbf{e}_2 \neq 0\}. \quad (81)$$

$$\tilde{\mathcal{R}}_{1 \rightarrow l} = \{i : i + l\mathbf{e}_1 \in \tilde{\mathcal{Z}}_{l,(2 \rightarrow l+2)}^{\{2,3\}}\} = \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r - l, i \cdot \mathbf{e}_2 = r - 2\}. \quad (82)$$

- Unmatched symbols from the third systematic node: The symbols from the second systematic node that are (potentially) matched using the parity symbols downloaded from the  $(l+1)$ -th parity node during the first stage are indexed by the following two sets.

$$\mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r - (l+2), i \cdot \mathbf{e}_2 = 0\}. \quad (83)$$

$$\tilde{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}} = \{i : i \cdot (1, \dots, 1) = l+3, i \cdot \mathbf{e}_1 = r - 3, i \cdot \mathbf{e}_2 = 0\} \subset \mathcal{U}_{0 \rightarrow l+3}^{\{1,3\}}. \quad (84)$$

$$(85)$$

Using (31), we can identify the indices of the parity symbols from the  $(l+1)$ -parity node where these symbols participate as follows.

$$\mathcal{Z}_{l,(1 \rightarrow l+2)}^{\{2,3\}} = \mathcal{U}_{1 \rightarrow l+2}^{\{2,3\}} + l\mathbf{e}_1 = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r - 2, i \cdot \mathbf{e}_2 = 0\}. \quad (86)$$

$$\tilde{\mathcal{Z}}_{l,(0 \rightarrow l+3)}^{\{1,3\}} = \tilde{\mathcal{U}}_{0 \rightarrow l+3}^{\{1,3\}} = \{i : i \cdot (1, \dots, 1) = l+3, i \cdot \mathbf{e}_1 = r - 3, i \cdot \mathbf{e}_2 = 0\}. \quad (87)$$

The symbols from the third systematic node which remain unmatched at the end of first stage (and require downloading additional symbols) due to their participation in the parity symbols indexed by the set  $\mathcal{Z}_{l,(1 \rightarrow l+2)}^{\{2,3\}} \cup \tilde{\mathcal{Z}}_{l,(0 \rightarrow l+3)}^{\{1,3\}}$  in the  $(l+1)$ -th parity node are as follows.

$$\mathcal{R}_{2 \rightarrow l} = \{i : i + l\mathbf{e}_2 \in \mathcal{Z}_{l,(1 \rightarrow l+2)}^{\{2,3\}}\} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r - 2, i \cdot \mathbf{e}_2 = r - l\}. \quad (88)$$

$$\tilde{\mathcal{R}}_{2 \rightarrow l} = \{i : i + l\mathbf{e}_2 \in \tilde{\mathcal{Z}}_{l,(0 \rightarrow l+3)}^{\{1,3\}}\} = \{i : i \cdot (1, \dots, 1) = 3, i \cdot \mathbf{e}_1 = r - 3, i \cdot \mathbf{e}_2 = r - l\} \quad (89)$$

We now describe the set of additional symbols downloaded to match the unmatched symbols from the three failed systematic nodes (cf. (76), (81) and (88)).

- Additional symbols downloaded to match remaining symbols from the first systematic node: Consider a set of  $r-1$  integers  $\mathcal{I}_0 = \{i_{0,1}, \dots, i_{0,r-1}\} \subset [0, r^{k-1} - 1]$  such that the following two conditions hold.

$$1) \ i_{0,j} \cdot (1, 1, \dots, 1) = 1 \ \forall j \in [r-1].$$

$$2) \ i_{0,j} \cdot \mathbf{e}_1 = j \ \text{for } j \in [r-1].$$

$$3) \ i_{0,j} \cdot \mathbf{e}_2 = r - 1 \ \forall j \in [r-1].$$

Note that we are using vector representation of the integers from the  $\mathcal{I}_0$  in  $\mathbb{Z}_r^{k-1}$  in order to define these three requirements. For  $j \in [3, k-1]$ , the set of additional symbols downloaded from the  $(j+1)$ -th systematic node in order to match the remaining symbols from the first systematic node have their row indices belonging to the following set.

$$\mathcal{S}_0^{\{1,2,3\}} = \mathcal{I}_0 + \{a_1(\mathbf{e}_4 - \mathbf{e}_3) + \dots + a_{k-4}(\mathbf{e}_{k-1} - \mathbf{e}_3) : (a_1, \dots, a_{k-4}) \in [r-1]^{k-4}\}. \quad (90)$$

Note that we have  $|\mathcal{S}_0^{\{1,2,3\}}| = (r-1)r^{k-4}$ . Next, we identify the set of the parity symbols in the  $(l+1)$ -th parity nodes where these symbols appear. Let  $\mathcal{P}_{0,l}^{\{1,2,3\}}$  denote the indices of these parity symbols in the  $(l+1)$ -th parity node. Then, from the definition of the zigzag sets (cf. 31), we have that

$$\begin{aligned}\mathcal{P}_{0,l}^{\{1,2,3\}} &= \mathcal{S}_0^{\{1,2,3\}} + l\mathbf{e}_3 = \mathcal{S}_0^{\{1,2,3\}} + l\mathbf{e}_4 = \dots = \mathcal{S}_0^{\{1,2,3\}} + l\mathbf{e}_{k-1} \\ &= \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r-1\}.\end{aligned}\quad (91)$$

We can again use (31) to identify the symbols from the second systematic node that appear in the parity symbols from  $l+1$ -th parity symbols that are indexed by the set  $\mathcal{P}_{0,l}^{\{1,2,3\}}$ .

$$\{i : i \in \mathcal{P}_{0,l}^{\{1,2,3\}}\} = \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 \neq 0, i \cdot \mathbf{e}_2 = r-1\}.\quad (92)$$

Note that this is exactly equal to  $\mathcal{R}_{0 \rightarrow l+1}$  which is the indices of the unmatched symbols from the first systematic as they appeared in the parity symbols downloaded from the  $(l+1)$ -th parity node during the first stage.

- Additional symbols downloaded to match remaining symbols from the second systematic node:
  - 1) Consider a set of  $r-1$  integers  $\mathcal{I}_1 = \{i_{1,1}, \dots, i_{1,r-1}\} \subset [0, r^{k-1} - 1]$  such that the following two conditions hold.
    - a)  $i_{1,j} \cdot (1, 1, \dots, 1) = 2 \ \forall j \in [r-1]$ .
    - b)  $i_{1,j} \cdot \mathbf{e}_1 = r-1 \ \forall j \in [r-1]$ .
    - c)  $i_{1,j} \cdot \mathbf{e}_2 = j$  for  $j \in [r-1]$ .

Note that we are using vector representation of the integers from the  $\mathcal{I}_1$  in  $\mathbb{Z}_r^{k-1}$  in order to define these three requirements. For  $j \in [3, k-1]$ , the set of additional symbols downloaded from the  $(j+1)$ -th systematic node in order to match the remaining symbols from the second systematic node have their row indices belonging to the following set.

$$\mathcal{S}_1^{\{1,2,3\}} = \mathcal{I}_1 + \{a_1(\mathbf{e}_4 - \mathbf{e}_3) + \dots + a_{k-4}(\mathbf{e}_{k-1} - \mathbf{e}_3) : (a_1, \dots, a_{k-4}) \in [r-1]^{k-4}\}.\quad (93)$$

Note that we have  $|\mathcal{S}_1^{\{1,2,3\}}| = (r-1)r^{k-4}$ . Next, we identify the set of the parity symbols in the  $(l+1)$ -th parity nodes where these symbols appear. Let  $\mathcal{P}_{1,l}^{\{1,2,3\}}$  denote the indices of these parity symbols in the  $(l+1)$ -th parity node. Then, from the definition of the zigzag sets (cf. 31), we have that

$$\begin{aligned}\mathcal{P}_{1,l}^{\{1,2,3\}} &= \mathcal{S}_1^{\{1,2,3\}} + l\mathbf{e}_3 = \mathcal{S}_1^{\{1,2,3\}} + l\mathbf{e}_4 = \dots = \mathcal{S}_1^{\{1,2,3\}} + l\mathbf{e}_{k-1} \\ &= \{i : i \cdot (1, \dots, 1) = l+2, i \cdot \mathbf{e}_1 = r-1, i \cdot \mathbf{e}_2 \neq 0\}.\end{aligned}\quad (94)$$

We can again use (31) to identify the symbols from the second systematic node that appear in the parity symbols from  $l+1$ -th parity symbols that are indexed by the set  $\mathcal{P}_{1,l}^{\{1,2,3\}}$ .

$$\{i : i + l\mathbf{e}_1 \in \mathcal{P}_{1,l}^{\{1,2,3\}}\} = \{i : i \cdot (1, \dots, 1) = 2, i \cdot \mathbf{e}_1 = r - (l+1), i \cdot \mathbf{e}_2 \neq 0\}.\quad (95)$$

Note that this is exactly equal to  $\mathcal{R}_{1 \rightarrow l+1}$  which is the indices of the unmatched symbols from the second systematic as they appeared in the parity symbols downloaded from the  $(l+1)$ -th parity node during the first stage.

2) Consider an integers  $\tilde{i}_1 \in [0, r^{k-1} - 1]$  such that the following two conditions hold.

- a)  $\tilde{i}_1 \cdot (1, 1, \dots, 1) = 1$ .
- b)  $\tilde{i}_1 \cdot \mathbf{e}_1 = r - 1$ .
- c)  $\tilde{i}_1 \cdot \mathbf{e}_2 = r - 2$ .

For  $j \in [3, k-1]$ , we download additional symbols from the  $(j+1)$ -th systematic node with their row indices belonging to the following set.

$$\tilde{\mathcal{S}}_1^{\{1,2,3\}} = \tilde{i}_1 + \{a_1(\mathbf{e}_4 - \mathbf{e}_3) + \dots + a_{k-4}(\mathbf{e}_{k-1} - \mathbf{e}_3) : (a_1, \dots, a_{k-4}) \in [r-1]^{k-4}\}. \quad (96)$$

Note that we have  $|\tilde{\mathcal{S}}_1^{\{1,2,3\}}| = r^{k-4}$ . Next, we identify the set of the parity symbols in the  $(l+1)$ -th parity nodes where these symbols appear. Let  $\tilde{\mathcal{P}}_{1,l}^{\{1,2,3\}}$  denote the indices of these parity symbols in the  $(l+1)$ -th parity node. Then, from the definition of the zigzag sets (cf. 31), we have that

$$\begin{aligned} \tilde{\mathcal{P}}_{1,l}^{\{1,2,3\}} &= \tilde{\mathcal{S}}_1^{\{1,2,3\}} + l\mathbf{e}_3 = \tilde{\mathcal{S}}_1^{\{1,2,3\}} + l\mathbf{e}_4 = \dots = \tilde{\mathcal{S}}_1^{\{1,2,3\}} + l\mathbf{e}_{k-1} \\ &= \{i : i \cdot (1, \dots, 1) = l+1, i \cdot \mathbf{e}_1 = r-1, i \cdot \mathbf{e}_2 = r-2\}. \end{aligned} \quad (97)$$

We can again use (31) to identify the symbols from the second systematic node that appear in the parity symbols from  $l+1$ -th parity symbols that are indexed by the set  $\tilde{\mathcal{P}}_{1,l}^{\{1,2,3\}}$ .

$$\{i : i + l\mathbf{e}_1 \in \tilde{\mathcal{P}}_{1,l}^{\{1,2,3\}}\} = \{i : i \cdot (1, \dots, 1) = 1, i \cdot \mathbf{e}_1 = r - (l+1), i \cdot \mathbf{e}_2 = r - 2\}. \quad (98)$$

Note that this is exactly equal to the unmatched symbols from the second systematic node denoted by  $\tilde{\mathcal{R}}_{1 \rightarrow l+1}$  (cf. (82)).

- Additional symbols downloaded to match remaining symbols from the third systematic node:

1) Consider a set of  $r-1$  integers  $\mathcal{I}_2 = \{i_{2,1}, \dots, i_{2,r-1}\} \subset [0, r^{k-1} - 1]$  such that the following two conditions hold.

- a)  $i_{2,j} \cdot (1, 1, \dots, 1) = j$  for  $j \in [r-1]$ .
- b)  $i_{2,j} \cdot \mathbf{e}_1 = r - 2 \ \forall j \in [r-1]$ .
- c)  $i_{2,j} \cdot \mathbf{e}_2 = r - 2 \ \forall j \in [r-1]$ .

Note that we are using vector representation of the integers from the  $\mathcal{I}_2$  in  $\mathbb{Z}_r^{k-1}$  in order to define these three requirements. For  $j \in [3, k-1]$ , the set of additional symbols downloaded from the  $(j+1)$ -th systematic node in order to match the remaining symbols from the third systematic node have their row indices belonging to the following set.

$$\mathcal{S}_2^{\{1,2,3\}} = \mathcal{I}_2 + \{a_1(\mathbf{e}_4 - \mathbf{e}_3) + \dots + a_{k-4}(\mathbf{e}_{k-1} - \mathbf{e}_3) : (a_1, \dots, a_{k-4}) \in [r-1]^{k-4}\}. \quad (99)$$

Note that we have  $|\mathcal{S}_2^{\{1,2,3\}}| = (r-1)r^{k-4}$ . Next, we identify the set of the parity symbols in the  $(l+1)$ -th parity nodes where these symbols appear. Let  $\mathcal{P}_{2,l}^{\{1,2,3\}}$  denote the indices of these parity symbols in the  $(l+1)$ -th parity node. Then, from the definition of the zigzag sets (cf. 31), we have that

$$\begin{aligned}\mathcal{P}_{2,l}^{\{1,2,3\}} &= \mathcal{S}_2^{\{1,2,3\}} + l\mathbf{e}_3 = \mathcal{S}_2^{\{1,2,3\}} + l\mathbf{e}_4 = \dots = \mathcal{S}_2^{\{1,2,3\}} + l\mathbf{e}_{k-1} \\ &= \{i : i \cdot (1, \dots, 1) \neq l, i \cdot \mathbf{e}_1 = r-2, i \cdot \mathbf{e}_2 = r-2\}.\end{aligned}\quad (100)$$

We can again use (31) to indentify the symbols from the third systematic node that appear in the parity symbols from  $(l+1)$ -th parity symbols that are indexed by the set  $\mathcal{P}_{2,l}^{\{1,2,3\}}$ .

$$\{i : i + l\mathbf{e}_2 \in \mathcal{P}_{2,l}^{\{1,2,3\}}\} = \{i : i \cdot (1, \dots, 1) \neq 0, i \cdot \mathbf{e}_1 = r-2, i \cdot \mathbf{e}_2 = r-(l+2)\}.\quad (101)$$

Note that this is exactly equal to  $\mathcal{R}_{2 \rightarrow l+2}$  which is the indices of the unmatched symbols from the third systematic node as they appeared in the parity symbols downloaded from the  $(l+1)$ -th parity node during the first stage.

2) Consider an integers  $\tilde{i}_2 \in [0, r^{k-1} - 1]$  such that the following two conditions hold.

- a)  $\tilde{i}_2 \cdot (1, 1, \dots, 1) = 3$ .
- b)  $\tilde{i}_2 \cdot \mathbf{e}_1 = r-3$ .
- c)  $\tilde{i}_2 \cdot \mathbf{e}_2 = r-1$ .

For  $j \in [3, k-1]$ , we download additional symbols from the  $(j+1)$ -th systematic node with their row indices belonging to the following set.

$$\tilde{\mathcal{S}}_2^{\{1,2,3\}} = \tilde{i}_2 + \{a_1(\mathbf{e}_4 - \mathbf{e}_3) + \dots + a_{k-4}(\mathbf{e}_{k-1} - \mathbf{e}_3) \pmod{r} : (a_1, \dots, a_{k-4}) \in [r-1]^{k-4}\}.\quad (102)$$

Note that we have  $|\tilde{\mathcal{S}}_2^{\{1,2,3\}}| = r^{k-4}$ . Next, we identify the set of the parity symbols in the  $(l+1)$ -th parity nodes where these symbols appear. Let  $\tilde{\mathcal{P}}_{2,l}^{\{1,2,3\}}$  denote the indices of these parity symbols in the  $(l+1)$ -th parity node. Then, from the definition of the zigzag sets (cf. 31), we have that

$$\begin{aligned}\tilde{\mathcal{P}}_{2,l}^{\{1,2,3\}} &= \tilde{\mathcal{S}}_2^{\{1,2,3\}} + l\mathbf{e}_3 = \tilde{\mathcal{S}}_2^{\{1,2,3\}} + l\mathbf{e}_4 = \dots = \tilde{\mathcal{S}}_2^{\{1,2,3\}} + l\mathbf{e}_{k-1} \\ &= \{i : i \cdot (1, \dots, 1) = l+3, i \cdot \mathbf{e}_1 = r-3, i \cdot \mathbf{e}_2 = r-1\}.\end{aligned}\quad (103)$$

We can again use (31) to indentify the symbols from the third systematic node that appear in the parity symbols from  $l+1$ -th parity symbols that are indexed by the set  $\tilde{\mathcal{P}}_{2,l}^{\{1,2,3\}}$ .

$$\{i : i + l\mathbf{e}_2 \in \tilde{\mathcal{P}}_{2,l}^{\{1,2,3\}}\} = \{i : i \cdot (1, \dots, 1) = 3, i \cdot \mathbf{e}_1 = r-3, i \cdot \mathbf{e}_2 = r-(l+1)\}.\quad (104)$$

Note that this is exactly equal to the unmatched symbols from the third systematic node denoted by  $\tilde{\mathcal{R}}_{2 \rightarrow l+1}$  (cf. (89)).