Polynomial Length MDS Codes With Optimal Repair in Distributed Storage

Viveck R. Cadambe, Cheng Huang, Jin Li, Sanjeev Mehrotra[†]

***Abstract*— In this paper, we study maximum distance separable (MDS) codes for distributed storage with optimal repair properties. An (n, k) MDS code can be used to store data in n storage nodes, such that the system can tolerate the failure of any $(n - k)$ storage nodes because of the MDS property. Recently, MDS codes have been constructed which satisfy an additional optimal repair property as follows: the failure of a single storage node can be repaired by downloading a fraction of $1/(n-k)$ of the data stored in every surviving storage node. In previous constructions satisfying this optimal repair property, the size of the code is polynomial in k for the high-redundancy regime of $k/n \leq 1/2$, but the codes have an exponential size (w.r.t. k) for the practically important low-redundancy regime of $k/n > 1/2$. In this paper, we construct polynomial size codes in this low redundancy regime. In particular, we construct MDS codes whose size is $O(k^2)$ with optimal repair bandwidth for the special case where $k/n \geq 2/3$. Further, we show that for any fixed *rate* k/n, we can construct repair bandwidth optimal MDS codes whose size scales as a polynomial in k.**

I. Introduction

Erasure coding is a fundamental technique to build redundancy in distributed storage systems. In classical literature in coding theory, erasure codes have been developed so that they provide the maximum tolerance to disk failure (erasures) for a given storage overhead. In particular, it is well known that *maximum distance separable* (MDS) codes (such as Reed Solomon Codes) provide the maximum failure-tolerance for a given amount of storage. Because of this favorable property, the design of practical MDS codes for distributed storage systems has been an important area of research [1]–[5]. Recently, corresponding to the rapid scaling in the volume of data in storage systems, there is increased interest in the study and design of MDS codes with a second favorable property: a small (perferably minimum) *repair bandwidth,* where the repair bandwidth is defined as the amount of data downloaded to repair one of more failed nodes. The motivation for minimizing the repair bandwidth comes from both theory and practice. From a practical perspective, minimizing the repair bandwidth reduces network congestion during recovery. It can also imply faster recovery of failed nodes in distributed storage systems. From a theoretical perspective, the repair bandwidth problem in storage has fundamental connections to interference alignment and multi-source network capacity problems. In recent literature, MDS codes with optimal (minimum) repair bandwidth in distributed storage systems have been discovered [6]–[12]. However, in general, these codes have a size that scales exponentially in the number of nodes in the distributed storage system. This is in contrast with the storage codes used in practice [1]–[5] whose sizes scale as a (usually linear) polynomial in the number of nodes. Thus, while the codes of [8]–[12] improve on the codes of [1]–[5] w.r.t. the repair bandwidth, they come at a cost of the code size being large. The exponential code size of [8]–[12] poses an obstacle in practice because of several reasons. First, a large code-size corresponds to a large latency in encoding. Second, the codes have a memory requirement that is exponential in the number of nodes. Finally, the exponential size of the code also imposes a large overhead for coding small packets/files [13]. For these reasons, the design of MDS codes of polynomial size with small (preferably minimum) repair bandwidth is an important open problem. This problem has connections to the area of interference alignment over a limited number of dimensions which is an ongoing area of research work[1] [14]–[16] (See [11] for an explanation). In this paper, we make progress in this area by designing polynomial size MDS codes with minimal repair bandwidth, for certain coding parameters. We next proceed to describe the problem, its background and a detailed description of our contributions.

A. The Problem

The problem of efficient recovery of codes for storage was formulated in [17] and studied further in [6]–[8], [10]–[12], [18]–[23]. The problem studied in these references[2] is as follows. Consider a distributed storage system with n storage nodes, using an (n, k) systematic code to store data. We will assume that the system stores a file of size $M = kL$. The file is divided into k equal parts of L each and stored in an uncoded form in the first k nodes. These first k nodes are known as the systematic nodes. The remaining $n - k$ nodes, each of which store parity data of size L, are known as parity nodes. If an (n, k) maximum distance separable (MDS) code is used to store data in the system, then, the system can tolerate a failure of any set of $(n - k)$ nodes. This is because the MDS property ensures that the original data can be recovered from *any* k surviving nodes in the system. While the MDS property protects the system from data loss in the worst case failure scenario of $(n-k)$ nodes, the most common failure scenario in storage systems is the

[1] It must be noted that [14]–[16] study the problem in the context of wireless communications. The problem in wireless communications is different from to the storage code problem in this paper, although it is related to it [11].

[2] In this paper, we focus on what are called Minimum Storage Regeneration (MSR) code generation problem [17] for exact repair [18]. Some of these references also study certain other problems related to efficient repair.

case where a single node fails. For this single-node failure scenario, the conventional repair strategy is the following: download the data stored in any k nodes in the system, recover all the original data, and then replace the failed node. Therefore, with the conventional strategy, the amount of *repair bandwidth* - the amount of data to be downloaded from the surviving nodes to repair a single failed node - is equal to k times the data stored in a single node, i.e., kL. The primary objective of references [6]–[8], [10]–[12], [17]–[23] is to reduce the repair bandwidth of MDS codes. The main contributions and the results of these references are summarized next.

B. Background: High Redundancy Regime

The design of repair strategies more efficient than the conventional approach was pioneered in [18] through an example for the special case of $n = 4, k = 2$. For this case, the reference showed that a single node failure can be repaired with a repair bandwidth of 1.5 times the data stored a single node (i.e., $1.5L$), which is more efficient than the trivial strategy of downloading 2 nodes entirely. Later, references [19], [21] showed that if $k \leq \max(n/2, 3)$ then, the amount of data to be downloaded from each surviving node can be reduced to a fraction of $\frac{1}{n-k}$ of the data stored in the (surviving) node, of a single failed node. Because there are $n-1$ surviving nodes[3], the total repair bandwidth for a single failed node is $\frac{(n-1)}{n-k}$ times the amount of data stored in a single node, i.e., $\frac{L(n-1)}{n-k}$. Since $\frac{n-1}{n-k} < k$, the strategy of [19], [21] is more efficient than conventional repair. In fact, the repair bandwidth of $\frac{L(n-1)}{n-k}$ can be shown to be optimal via cut-set lower bounds [17]. Note that the results of [18], [19], [21] hold for the special case of $k/n \leq 2$. In other words, they hold for the special case where the redundancy overhead ($(n-k)L$) is at least as large as the amount of original data, kL. Therefore, this regime is referred to as the *high redundancy regime* in this paper.

A key technique used in the improved repair bandwidth in references [18], [19], [21] is the notion of vector coding - the idea that the code elements are vectors and a code can be constructed over the vectors. The idea is best explained for the case of where $n = 4, k = 2$ originally studied in [18] and depicted in Fig. 1. In this case, note that each of the 4 storage nodes stores a 2×1 vector, where the first two nodes respectively store (A_1, A_2) and (B_1, B_2) - these two vectors together form an uncoded copy of the original data. The remaining two nodes, which are parity nodes, each store two linear combinations of the A_1, A_2, B_1, B_2. It can be verified that the code depicted is an MDS code so that the failure (erasure) of any two storage nodes can be tolerated without a loss of data. If two nodes fail, there are 4 linear combinations of A_1, A_2, B_1, B_2 surviving in the system; the original data A_1, A_2, B_1, B_2 can be recovered from these 4 linear combinations. Now, consider the case where first node fails. The goal is to repair this node using the surviving nodes, i.e., to reconstruct (A_1, A_2) using the surviving nodes. A trivial solution is to download 4 linear combinations from any 2 surviving nodes and then reconstruct the entire original data, and then store A_1, A_2 in the new node. However, it is possible to reconstruct (A_1, A_2) by downloading only 3 linear combinations, i.e., a fraction of $1/(n-k) = 1/2$ the data stored in every node. The linear combinations to be downloaded are depicted in Fig. 1.

References [19], [21] generalized the code shown in Fig. 1 to (n, k) code based storage systems, where $k \leq (n/2, 3)$. In particular, for a $(n = 2k, k)$ code of [19], [21], each code element (i.e., data stored in each node) can be viewed as $L \times 1$ vector, where $L = k$. The failure of a single node can be repaired by downloading a single scalar from each surviving node. This effectively amounts to downloading a fraction of $1/(n-k) = 1/k$ of every surviving node as required for optimality[4]. By code shortening, the class of $(2k, k)$ codes can be used to construct repair-bandwidth optimal codes for any (n, k) where $n \geq 2k$. While these references found optimal codes for the minimum repair bandwidth for the high-redundancy case of $k/n \leq 1/2$ (and $k \geq 3$, [19]), the question of the minimum repair bandwidth for the low-redundancy case of $k/n > 1/2, k > 3$ was left open.

C. Background: Low Redundancy Regime

The question of the minimum repair bandwidth for arbitrary (n, k) including the low redundancy regime of $k/n \leq 1/2$ was settled in [6], [7]. The references used the asymptotic interference alignment scheme of [24] to construct asymptotic codes which approach the repair bandwidth cutset bound of $(n-1)/(n-k)$ times the data stored in a single node. In other words, they showed that as the size of the code $L \to \infty$, the repair bandwidth approaches $L(n-1)/(n-k)$ (asymptotically with 100% accuracy). The question of existence of *finite* codes for this case was left open in these references. This question was settled in the positive in references [8], [10]–[12], [23] which constructed finite repair-bandwidth optimal codes for arbitrary (n, k) including the previously open low redundancy regime. In reference [11], a framework based on a tensor product structure (also referred to as subspace interference alignment [25]) generalizing the codes of [8], [10], [26] was described for construction of MDS codes with optimal repair of *systematic* nodes. Codes which can repair the failure of *any* single node (including the failure of a parity node) are presented recently for the case of $n-k = 2$ in [12] and for arbitrary (n, k) in [23]. While the code constructions of [19], [21] for the high redundancy case of $k/n \leq 1/2$ used vectors of size k, the code constructions of [8], [10]–[12], [23] for the low redundancy case of $k/n > 1/2$ used vectors whose size is exponential in k. In other words, for $k/n > 1/2$ the parameter $L = O((n-k)^k)$ in the codes of [8], [10]–[12], [23]. An important question left open in these references is the efficient repair of codes in the low

[3] In this reference, we assume that the repair can be performed by connecting to *all* n-1 surviving nodes. References [17], [20], [21] also conder the more general case where the new node is restricted to connect to only d of the $n-1$ surviving nodes, where $d \leq n-1$.

[4] Note that if the original data is larger than $L = k$ units, the data can be divided into several portions of $L = k$ units, and the code can be used for each of these portions separately.

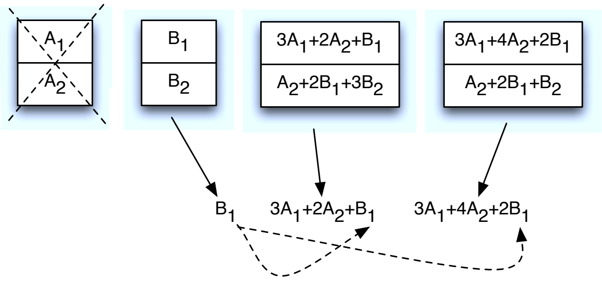


Fig. 1. A $(4,2)$ code with the repair of the first node depicted. Each node stores a $L \times 1$ vector, where $L = 2$. Note that $1/2$ of every surviving node (i.e., a scalar) is downloaded, and that on cancellation of B_1, the lost elements A_1, A_2 can be recovered from the 2 equations.

redundancy regime, when the size of the code is restricted to be of a polynomial size. In this paper, we make progress on this open problem, and provide a partial answer to this question.

The main contribution of this paper is a new polynomial size code in the low redundancy regime of $k/n > 1/2$ with optimal repair for a single *systematic* node failure. In particular, we provide a technique to combine two $(k, k/2)$ codes of [19], [21] (referred to as 'MISER' codes in the former reference) to generate codes for $k/n = 2/3$. Further, a simple code shortening technique of [21] suffices to argue that the codes are applicable, not just for $k/n = 2/3$, but for the entire regime of $k/n \geq 2/3$. Our code, which lies in this low redundancy regime of $k/n > 1/2$, has the advantage that $L = k^2/4$ and therefore has a size which is polynomial in k. For instance, if $n = 12, k = 8$, the schemes of [8], [10] encode over vectors of size $L = 4^7$. However, our scheme encodes over vectors of size $L = 16$. Since our coding scheme uses two underlying $(k, k/2)$ codes, it can be viewed as a *compound* of two MDS codes and hence termed compound codes. The compounding approach demonstrated here can also be used to form an (n, k) code with optimal repair for any arbitrary (n, k) by compounding $m = \lceil k/(n-k) \rceil$ $(2k/m, k/m)$ MISER codes (See extended paper [27] for details). The size of the compounded code for this general case is $L = (k/m)^m$. Note that this means that for a fixed *rate* k/n, the code size is polynomial in k. However in general, the code size is exponential in k.We now proceed to describe to give an overview of the main ideas behind the code construction. A more complete description can be found in the extended version of this paper [27].

II. Code Construction

Consider k sources each uniformly distributed over a field $\mathbb{F}_q^L$, where $\mathcal{F}_q$ is a field. Source $i \in \{1, 2, \ldots, k\}$ is represented by the $L \times 1$ vector $\mathbf{a}_i \in \mathbb{F}_q^L$. Note here that $M = kL$ denotes the size of the total information stored in the distributed storage system, in terms of the number of elements over the field. There are n nodes storing a code of the k source symbols in an (n, k) MDS code. Each node stores a data of size L, i.e., each coded symbol of the (n, k) code is a $L \times 1$ vector. The data stored in node i is represented by $L \times 1$ vector $\mathbf{d}_i$, where $i = 1, 2, \ldots, n$. Our code is linear and $\mathbf{d}_i$ can be represented as

$$\mathbf{d}_i = \sum_{j=1}^{k} \mathbf{C}_{i,j} \mathbf{a}_j, \tag{1}$$

where $\mathbf{C}_{i,j}$ are $L \times L$ square matrices. Our codes have a systematic structure so that, for $i \in \{1, 2, \ldots, k\}$,

$$\mathbf{C}_{i,j} = \left\{ \begin{array}{ll} \mathbf{I} & j = i \\ \mathbf{0} & j \neq i \end{array} \right\},$$

so that $\mathbf{d}_i = \mathbf{a}_i$ for $1 \leq i \leq k$. Since we restrict our attention to MDS codes, we will need the matrices $\mathbf{C}_{i,j}$ to satisfy the following property

Property 1:

$$\text{rank}\left(\begin{bmatrix} \mathbf{C}_{j_1,1} & \mathbf{C}_{j_1,2} & \ldots & \mathbf{C}_{j_1,k} \\ \mathbf{C}_{j_2,1} & \mathbf{C}_{j_2,2} & \ldots & \mathbf{C}_{j_2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{j_k,1} & \mathbf{C}_{j_k,2} & \ldots & \mathbf{C}_{j_k,k} \end{bmatrix} \right) = Lk = M \tag{2}$$

for any distinct $j_1, j_2, \ldots, j_k \in \{1, 2, \ldots, n\}$.

The MDS property ensures that the storage system can tolerate up to $(n-k)$ failures (erasures), since all the sources can be reconstructed, linearly, from any k nodes whose indices are represented by $j_1, j_2, \ldots, j_k \in \{1, 2, \ldots, n\}$. Now, consider the case where a single systematic node, node $i \in \{1, 2, \ldots, k\}$, fails. Now, to reconstruct a failure of node i, $\frac{L}{n-k}$ elements of $\mathbf{d}_i$ are downloaded from nodes $\{1, 2, \ldots, n\} - \{i\}$. The goal of this paper is to generate $\mathbf{C}_{i,j}$ such that

- The code is an MDS code, i.e., $\mathbf{C}_{i,j}$ satisfies Property 1.
- The $\mathbf{d}_i$ can be regenerated by using $L/(n-k)$ elements of each surviving node, i.e., from $\mathbf{d}_j, j \in \{1, 2, \ldots, n\} - \{i\}$. We assume that the failed node is a systematic one so that $i \in \{1, 2, \ldots, k\}$. Note that the total number of elements downloaded for reconstruction of $\mathbf{d}_i$ is equal to $\frac{(n-1)L}{n-k}$ since we download $L/(n-k)$ elements from each surviving node.

Previously, codes with the above property have been developed for $k \leq n/2$ in [19], [21]. In our construction, we will develop a technique to combine two $(2\overline{k}, \overline{k})$ codes generated in [19], [21] to develop a $(n = 3\overline{k}, k = 2\overline{k})$ code which satisfies the above properties for any $\overline{k}$. The remainder of this section is organized as follows. First, we provide an overview of the important properties of the codes described in [19], [21]. Then, we will describe the concept of a m-expansion of a code in Section II-B. Finally, in Section II-C, we will describe a technique to generate a $(3\overline{k}, 2\overline{k})$ code by combining two expanded $(2\overline{k}, \overline{k})$ codes.

Before we proceed to describe our code construction, we will introduce a notation that is used through out the paper. For any $l \times 1$ vector $\mathbf{a}$, its l elements are denoted by $a(1), a(2), \ldots, a(l)$. For instance $\mathbf{d}_i = (d_i(1) \quad d_i(2) \quad \ldots \quad d_i(L))^T$, where the ^{T} denotes the transpose of a vector. Let $\mathcal{A}$ and $\mathcal{B}$ be two subsets of $\{d_i(j) : i = 1, 2, \ldots, k, j = 1, 2, \ldots, L\}$. Then we use the notation $\mathcal{A} \rightarrow \mathcal{B}$ if the elements of $\mathcal{B}$ can be generated linearly from the elements of set $\mathcal{A}$. In other words,

$$\mathcal{A} \rightarrow \mathcal{B} \Leftrightarrow, \forall y \in \mathcal{B} \exists \alpha_x \in \mathcal{F}_q, y = \sum_{x \in \mathcal{A}} \alpha_x x, \forall \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k,$$

where the constants α_x do not depend on (the realization of) $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$.

On a similar note, we also sometimes use the notation $\mathcal{A} \rightarrow \mathbf{d}_i$ if the vector $\mathbf{d}_i$ can be generated linearly from the elements in $\mathcal{A}$. For example, because any parity node can be generated from the systematic node by (1) we have $\{d_i(j) : i = 1, 2, \ldots, k, j = 1, 2, \ldots, L\} \rightarrow \mathbf{d}_m$ for $k < m \leq n$.

A. Background : MISER codes

References [19], [21] developed a class of codes for optimal repair of a single failed node. The codes were termed MISER codes in the latter reference, whose nomenclature we borrow to denote the codes. We shall illustrate the main properties of the $(n = 2\overline{k}, \overline{k})$ MISER codes here. We do not present the MISER codes constructions here because of space constraints; the properties listed below suffice for our purposes.

1) Each element of the code is a $\overline{k} \times 1$ vector, i.e., $L = \overline{k}$.
2) To repair a failure of the jth node for $1 \leq j \leq \overline{k}$, the jth element of every surviving node is downloaded. In other words, $\mathbf{d}_j$ can be recovered using linear combinations of the elements of the set $\{d_i(j) : i \in \{1, 2, \ldots, n\} - \{j\}\}$.

The $2\overline{k}$ symbols of the codes satisfying the above properties are denoted by $\overline{\mathbf{d}}_i, i = 1, 2, \ldots, 2\overline{k}$, where $\overline{\mathbf{d}}_i$ is a $\overline{k} \times 1$ vector (because of the first property listed above). The coding matrices satisfying the above properties are denoted by $\mathbf{H}_{i,j}$, where $i \in \{\overline{k}+1, \overline{k}+2, \ldots, 2\overline{k}\}, j \in \{1, 2, \ldots, \overline{k}\}$. If we denote

$$\overline{\mathbf{d}}_i = \sum_{j=1}^{k} \mathbf{H}_{i,j} \mathbf{a}_i$$

for $i = \overline{k}+1, \overline{k}+2, \ldots, 2\overline{k}$, where the construction of $\mathbf{H}_{i,j}$, can be found in [19], [21]. The second property of these codes listed above is re-stated formally below.

Property 2: *Optimal Repair of MISER Codes [21]* For the $(2\overline{k}, \overline{k})$ MISER code defined as in reference [19], [21], with its code symbols denoted by $\overline{\mathbf{d}}_1, \overline{\mathbf{d}}_2, \ldots, \overline{\mathbf{d}}_{2\overline{k}}$, the following property holds.

$$\{\overline{d}_i(l) : i \in \{1, 2, \ldots, 2\overline{k}\} - \{l\}\} \rightarrow \overline{\mathbf{d}}_l$$

for $l = 1, 2, \ldots, \overline{k}$.

We will effectively form a compound of two $(2\overline{k}, \overline{k})$ MISER codes into one $(3\overline{k}, 2\overline{k})$ code. The concatenation is preceded by an of the $(2\overline{k}, \overline{k})$ MISER code which is described next.

B. $\overline{k}$-Expansion of the MISER code

For a given (n, k) MDS code where each node stores a $L \times 1$ vector, an s-expansion of the code is also an (n, k) MDS code where each node stores an $p_0 L \times 1$ vector. The $sL \times 1$ vector stored at a node is essentially formed by a repeated use of the original coding matrix to form the vector. In other words, the $p_0 L \times 1$ vector of each symbol is viewed as p_0 blocks of $L \times 1$ vectors. The pth block of a parity node, which is a $L \times 1$ vector, is generated by using the pth block of all the systematic nodes. Here we consider a $\overline{k}$-expansion of the $(2\overline{k}, \overline{k})$ MISER code described above. Denoting the elements of the code by $\overline{\mathbf{d}}_i^{[\overline{k}]}$, the $\overline{k}$-expansion of the code is defined as

$$\overline{\mathbf{d}}_i^{[\overline{k}]} = \sum_{i,j} \mathbf{I}_{\overline{k}} \otimes \mathbf{H}_{i,j} \mathbf{a}_i \tag{3}$$

where the data elements, $\mathbf{a}_i$ are $\overline{k}^2 \times 1$ vectors, and $\mathbf{I}_{\overline{k}}$ denotes the $\overline{k}$-dimensional identity matrix and $\otimes$ represents the Kronecker product operation. We will use the notation $\mathbf{H}_{i,j}^{[\overline{k}]} \triangleq \mathbf{I}_{\overline{k}} \otimes \mathbf{H}_{i,j}$. Note that $\mathbf{H}_{i,j}^{[\overline{k}]}$ is a $\overline{k}^2 \times \overline{k}^2$ matrix with a block diagonal structure. Note that in this expansion, the $(p-1)\overline{k} + j$th element of a parity node, where $p = 1, 2 \ldots, \overline{k}, j = 1, 2, \ldots, \overline{k}$ is essentially a member of the pth block. Therefore, it is formed by using the MISER code over the pth block of all the systematic elements, and hence, only depends on the elements $\overline{d}_m^{[\overline{k}]}((p-1)\overline{k}+1), \overline{d}_m^{[\overline{k}]}((p-1)\overline{k}+2), \ldots, \overline{d}_m^{[\overline{k}]}((p-1)\overline{k}+\overline{k})$ for $m = 1, 2, \ldots, \overline{k}$. Therefore we can make the following observation (also depicted in Fig. 2).

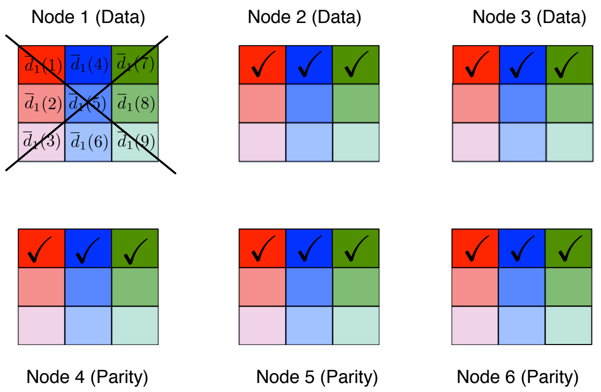


Fig. 2. A 3-expansion of the $(6, 3)$ MISER Code. The ✓denotes the elements downloaded on the failure of node 1, thus depicting Property 4. Property 3 is denoted using colors - the red (resp. blue, green) colored elements of a parity node are derived from the red (resp. blue, green) colored elements of systematic nodes.

Property 3: Consider the $\overline{k}$-expanded MISER code. Let

$$\mathcal{A}_p = \bigcup_{i=1}^{k} \{\overline{d}_i^{[\overline{k}]}((p-1)\overline{k}+j) : j = 1, 2, \ldots, \overline{k}\}$$

$$\mathcal{B}_{p,m} = \left\{ \overline{d}_m^{[\overline{k}]}((p-1)\overline{k}+j) : j = 1, 2, \ldots, \overline{k}, \right\}$$

for $p = 1, 2, \ldots, \overline{k}$ for $m = \overline{k}+1, \overline{k}+2, \ldots, 2\overline{k}$. Then, $\mathcal{A}_p \rightarrow \mathcal{B}_{p,m}, \forall m = \overline{k}+1, \overline{k}+2, \ldots, 2\overline{k}$. Note that $|\mathcal{A}_p| = \overline{k}^2$ and $|\mathcal{B}_{p,m}| = \overline{k}$.

Finally, because of Property 2 and the definition of the $\overline{k}$-expansion, the optimal repair property of the MISER code

$$\mathbf{P} = \Big(\mathbf{e}(1)\ \ \mathbf{e}(\bar{k}+1)\ \ \mathbf{e}(2\bar{k}+1)\ \ \ldots\ \ \mathbf{e}(\bar{k}(\bar{k}-1)+1)\ \ \mathbf{e}(2)\ \ \mathbf{e}(\bar{k}+1)\ \ \ldots\ \ \mathbf{e}(\bar{k}(\bar{k}-1)+\bar{k}\ \ \mathbf{e}(\bar{k}^2))\Big)$$

simply carries over to the $\bar{k}$-expansion.

Property 4: For the expanded MISER code as in equation (3), the following property holds.

$$\bigcup_{i \in \{1,2,\ldots,\bar{k}\}-\{l\}} \{\bar{d}_i^{[\bar{k}]}((p-1)\bar{k}+l) : p = 1,2,\ldots,\bar{k}\} \to \bar{\mathbf{d}}_l^{[\bar{k}]}$$

for $l = 1,2,\ldots,\bar{k}$.
Properties 3 and 4 are represented pictorially in Figure 2.

C. Main Contribution : The $(3\bar{k}, 2\bar{k})$ Compound Code

In this section, we will describe a $(n = 3\bar{k}, k = 2\bar{k})$ code which can optimally repair a single failed systematic node. The coding matrices for our construction are described as

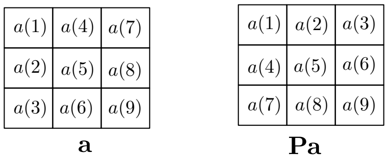
$$\mathbf{C}_{i,j} = \left\{ \begin{array}{cc} \mathbf{H}^{[\bar{k}]}_{i-\bar{k},j} & j \in \{1,2,\ldots,\bar{k}\} \\ \lambda_i \mathbf{P}\mathbf{H}^{[\bar{k}]}_{i-\bar{k},j-\bar{k}} & j \in \{\bar{k}+1, \bar{k}+2, \ldots, 2\bar{k}\} \end{array} \right\}$$

where λ_i is a scalar randomly chosen from the field $\mathbb{F}_q$. Matrix $\mathbf{P}$ is a permutation matrix shown at the top of this page, where $\mathbf{e}(i)$ represents the ith column of the $\bar{k}^2 \times \bar{k}^2$ identity matrix. Therefore, the $(p-1)\bar{k}+b$th column of $\mathbf{P}$ is $\mathbf{e}((b-1)\bar{k}+p)$, where $p, b \in \{1,2,\ldots,\bar{k}\}$. To understand the code, it is instructive to understand the structure of the above $\bar{k}^2 \times \bar{k}^2$ permutation matrix. Consider an arbitrary $\bar{k}^2 \times 1$ column vector $\mathbf{a}$. Then, the vector, $\mathbf{Pa}$, is a permutation of the elements of $\mathbf{a}$. The permutation has the following structure (also depicted pictorially in Fig. 3). If we arrange the $\bar{k}^2$ elements of $\mathbf{a}$ in a $\bar{k} \times \bar{k}$ square, where the pth column of the square contains elements $a((p-1)\bar{k}+1), a((p-1)\bar{k}+2), \ldots a(p\bar{k})$. Then, the permutation $\mathbf{Pa}$ takes a "transpose" of this square. In other words, the $(p-1)\bar{k}+l$th element of $\mathbf{Pa}$) is $a((l-1)\bar{k}+p))$ for $l, p \in \{1,2,\ldots,\bar{k}\}$. With the above permutation, the code can be interpreted as the concatenation of two MISER codes via a permutation operation because, a parity node $\mathbf{d}_i$ can be represented as follows (Also see Fig 4 and Fig. 5).

$$\mathbf{d}_i = \underbrace{\sum_{j=1}^{\bar{k}} \mathbf{H}^{[\bar{k}]}_{i-\bar{k},j} \mathbf{a}_j}_{\text{First } (2\bar{k},\bar{k}) \text{ MISER code}} + \lambda_i \mathbf{P} \underbrace{\sum_{j=1}^{\bar{k}} \mathbf{H}^{[\bar{k}]}_{i-\bar{k},j} \mathbf{a}_{j+\bar{k}}}_{\text{Second } (2\bar{k},\bar{k}) \text{ MISER code}} \qquad (4)$$

for $i = 2\bar{k}+1, 2\bar{k}+2, \ldots, 3\bar{k}$. The first MISER code is formed over $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{\bar{k}}$ and the second, over $\mathbf{a}_{\bar{k}+1}, \mathbf{a}_{\bar{k}+2}, \ldots, \mathbf{a}_{2\bar{k}}$.

Further, because of Property 3 and the nature of the permutation, note that the $(p-1)\bar{k}+l$th element of a parity node is a linear combination of the $\{d_m((p-1)\bar{k}+j) : j = 1,2,\ldots,\bar{k}\}$ for $m = 1,2,\ldots,\bar{k}$ and $\{d_r((j-1)\bar{k}+p) : j = 1,2,\ldots,\bar{k}\}$ for $r = \bar{k}+1, \bar{k}+2, \ldots, 2\bar{k}$. This observation is summarized in the following property.



$a(1)$	$a(4)$	$a(7)$
$a(2)$	$a(5)$	$a(8)$
$a(3)$	$a(6)$	$a(9)$

$\mathbf{a}$

$a(1)$	$a(2)$	$a(3)$
$a(4)$	$a(5)$	$a(6)$
$a(7)$	$a(8)$	$a(9)$

$\mathbf{Pa}$

Fig. 3. A depiction of the permutation matrix $\mathbf{P}$. If $\mathbf{a} = (a(1)\ \ a(2)\ \ \ldots\ \ a(9))^T$, then the column vector $\mathbf{Pa}$ is equal to $(a(1)\ \ a(4)\ \ a(7)\ \ a(2)\ \ a(5)\ \ a(8)\ \ a(3)\ \ a(6)\ \ a(9))^T$

Property 5: Consider the compound code defined in (4). Let

$$\mathcal{A}_p = \bigcup_{m=1}^{\bar{k}} \{d_m((p-1)\bar{k}+j) : j = 1,2,\ldots,\bar{k}\}$$

$$\mathcal{B}_p = \bigcup_{r=\bar{k}+1}^{2\bar{k}} \{d_r((j-1)\bar{k}+p) : j = 1,2,\ldots,\bar{k}\}$$

for $p \in \{1,2,\ldots,\bar{k}\}$. Then

$$\mathcal{A}_p \cup \mathcal{B}_p \to d_l((p-1)\bar{k}+j)$$

for all $j \in \{1,2,\ldots,\bar{k}\}$ for $l \in \{2\bar{k}+1, 2\bar{k}+2, \ldots, 3\bar{k}\}$.

We will use the above property, in combination with Property 4 to show that a failed systematic node can be optimally repaired, by downloading a fraction of $\frac{1}{\bar{k}}$ of every surviving node.

Repair of a failed systematic node: Suppose node 1 fails. Loosely speaking, the key idea behind repair, of say node 1, can be described as follows: First, we download enough data symbols from nodes $\bar{k}+1, \bar{k}+2, \ldots, 2\bar{k}$ so that the effect of the second MISER code is cancelled from the (appropriate) parity elements. Then, what remains among nodes $2, 3, \ldots, \bar{k}$ and the $\bar{k}$ parity nodes is a $(2\bar{k}, \bar{k})$ MISER code on which optimal repair can be performed. The repair strategy is described more specifically next. The repair of a failed systematic node is indicated in Figure 5 for $\bar{k} = 3, n = 9, k = 6$. We consider the case where node $l \in \{1,2,\ldots,\bar{k}\}$ fails. For this case, we download the following.

1) $\mathcal{C}_i = \{d_i((p-1)\bar{k}+l) : p = 1,2,\ldots,\bar{k}\}$ from node $i \in \{1,2,\ldots,\bar{k}\} - \{l\}$
2) $\mathcal{C}_i = \{d_i(l\bar{k}+(p-1)) : p = 1,2,\ldots,\bar{k}\}$ from node $i \in \{\bar{k}+1, \bar{k}+2, \ldots, 2\bar{k}\}$
3) $\mathcal{C}_i = \{d_i((p-1)\bar{k}+l) : p = 1,2,\ldots,\bar{k}\}$ from node $i \in \{2\bar{k}+1, 2\bar{k}+2, \ldots, 3\bar{k}\}$

Note that this strategy downloads $\bar{k}$ elements of every surviving node, and therefore downloads a fraction of $\frac{1}{\bar{k}}$ of every surviving node, as required. Now, because of Property 5 the second set of elements listed above can be used to cancel the effect of $\mathbf{a}_{\bar{k}+1}, \mathbf{a}_{\bar{k}+2}, \ldots, \mathbf{a}_{2\bar{k}}$ from the third set of (parity) elements downloaded (See Fig. 5). After this cancellation, the effect of the second MISER code is cancelled and the

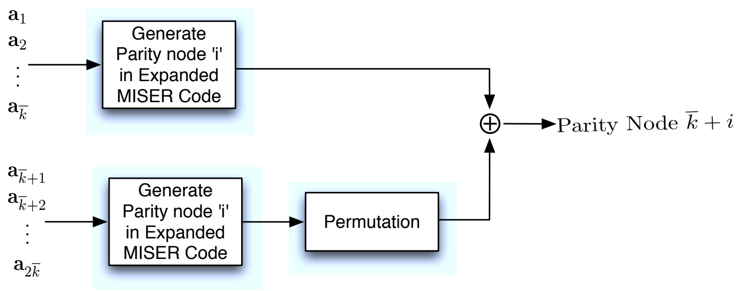

$\mathbf{a}_1$
$\mathbf{a}_2$
$\vdots$
$\mathbf{a}_{\overline{k}}$
Generate Parity node 'i' in Expanded MISER Code
$\mathbf{a}_{\overline{k}+1}$
$\mathbf{a}_{\overline{k}+2}$
$\vdots$
$\mathbf{a}_{2\overline{k}}$
Generate Parity node 'i' in Expanded MISER Code
Permutation
$\oplus$
Parity Node $\overline{k}+i$

Fig. 4. The interpretation of our code as a compound of two expanded $(2\overline{k}, \overline{k})$ MISER codes

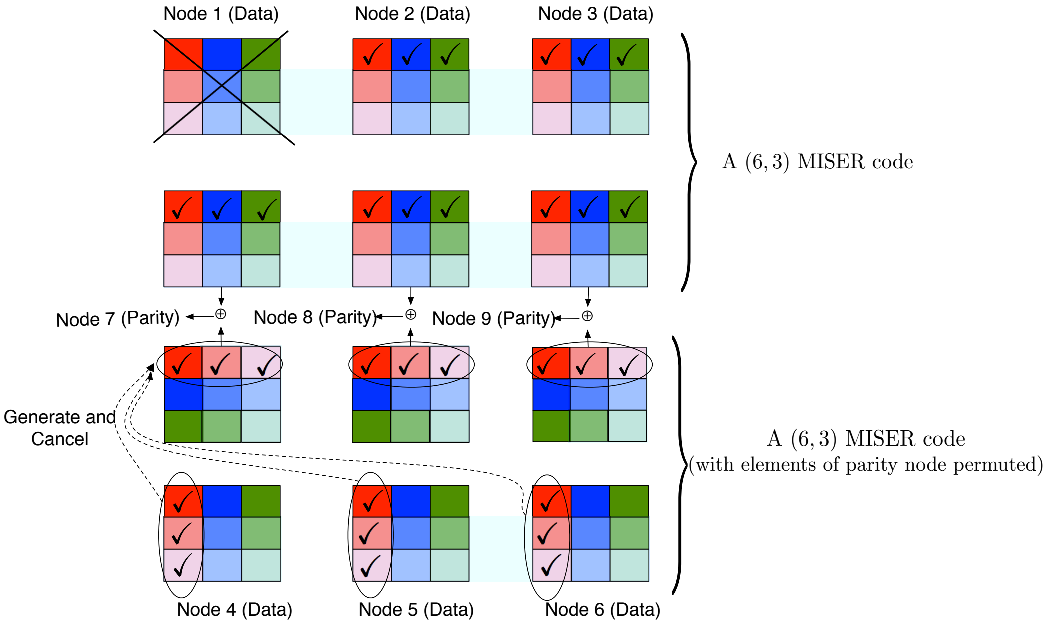


Fig. 5. Repair for a $(9,6)$ code. The ✓ denotes the elements downloaded for the repair of node 1. The elements downloaded from nodes $4,5,6$ can be used to generate and hence cancel the effect of the second (bottom) $(6,3)$ code participating in the combination, i.e., cancel the effect of $\mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6$. After this cancellation, the remaining elements form a picture similar to Fig. 2

elements that remain are exactly those elements that are needed to repair node l from the first $\overline{k}$-expanded MISER code, as described in Property 4 (See Fig. 5). Thus, using Properties 4 and 5 along with recognizing the code structure (4), the following optimal repair property can be shown

$$\bigcup_{i\in\{1,2,\ldots,3\overline{k}\}-\{l\}} \mathcal{C}_i \rightarrow \mathbf{d}_l.$$

The repair of a failed node $l \in \{\overline{k}+1, \ldots, 2\overline{k}\}$ is also similar and omitted here for the sake of brevity.

MDS Property: It can be shown (see [27]) using the Schwartz-Zippel Lemma that if the field size is chosen large enough, there exist scalars λ_i so that the MDS property can be satisfied.

III. Discussion

A. Repair of Parity Nodes

Above, we have described the repair of systematic nodes. By exploiting the fact that the underlying MISER codes have optimal repair property for parity nodes, the code developed here can be used for non-trivially efficient repair of parity nodes. For example, if a parity node fails, then we can download $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{\overline{k}}$ completely - this involves a cost (of repair bandwidth) of $\overline{k}$ nodes. Now, what remains is a $(2\overline{k}, \overline{k})$ MISER code (with one failed parity node). By applying the optimal repair strategy over these remaining nodes, the total repair bandwidth can be reduced to $\overline{k} + \frac{1}{\overline{k}}(\overline{k}-1)$ times the amount of data stored in each node. In other words, the total repair bandwidth can be reduced to be equivalent to downloading $k/2 + \frac{k/2-1}{k/2} < k$ nodes completely.

B. Compound of More than two $(2\overline{k}, \overline{k})$ MISER codes

The principle of combining $(2\overline{k}, \overline{k})$ MISER codes illustrated here can be used to combine more than two MISER codes. In general, by using a $\overline{k}^{m-1}$-expanded MISER codes, m $(2\overline{k}, \overline{k})$ codes can be combined to obtain a $(m\overline{k}, (m-1)\overline{k})$ code with optimal repair properties. However, the size of the code vectors, L, grows exponentially in $(n-k)$ with such an expansion. Details of combination of more than two MISER codes will be provided in the extended version of this paper.

References

[1] M. Blaum, J. Brady, J. Bruck, and J. Menon, "Evenodd: an optimal scheme for tolerating double disk failures in raid architectures," in *Computer Architecture, 1994., Proceedings the 21st Annual International Symposium on*, pp. 245 –254, Apr. 1994.

[2] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-diagonal parity for double disk failure correction," in *In Proceedings of the 3rd USENIX Symposium on File and Storage Technologies (FAST)*, pp. 1–14, 2004.

[3] J. S. Plank, "The raid-6 liber8tion code," *The International Journal of High Performance Computing and Applications*, vol. 23, pp. 242–251, August 2009.

[4] L. Xu and J. Bruck, "X-code: Mds array codes with optimal encoding," *IEEE Transactions on Information Theory,*, vol. 45, pp. 272 –276, jan 1999.

[5] C. Huang and L. Xu, "Star : An efficient coding scheme for correcting triple storage node failures," *Computers, IEEE Transactions on*, vol. 57, pp. 889 –901, July 2008.

[6] V. R. Cadambe, S. Jafar, and H. Maleki, "Distributed data storage with minimum storage regenerating codes - exact and functional repair are asymptotically equally efficient," *CoRR*, vol. abs/1004.4299, April 2010. http://arxiv.org/abs/1004.4299.

[7] C. Suh and K. Ramchandran, "On the existence of optimal exact-repair mds codes for distributed storage," *CoRR*, vol. abs/1004.4663, April 2010. http://arxiv.org/abs/1004.4663.

[8] V. R. Cadambe, C. Huang, and J. Li, "Permutation code: Optimal exact-repair of a single failed node in MDS code based distributed storage systems," *Proceedings of IEEE Symposium on Information Theory (ISIT)*, July 2011.

[9] I. Tamo, Z. Wang, and J. Bruck, "MDS array codes with optimal rebuilding," *Proceedings of IEEE Symposium on Information Theory (ISIT)*, July 2011.

[10] I. Tamo, Z. Wang, and J. Bruck, "MDS array codes with optimal rebuilding," *CoRR*, vol. abs/1103.3737, 2011. http://arxiv.org/abs/1103.3737.

[11] V. R. Cadambe, C. Huang, S. A. Jafar, and J. Li, "Optimal repair of MDS codes in distributed storage via subspace interference alignment," 2011. Preprint available on author's website, http://newport.eecs.uci.edu/∼vcadambe.

[12] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Distributed storage codes through hadamard designs," *To be presented in ISIT 2011*, july 2011.

[13] N. Alon and M. Luby, "A linear time erasure-resilient code with nearly optimal recovery," *Information Theory, IEEE Transactions on*, vol. 42, pp. 1732 –1736, nov 1996.

[14] G. Bresler and D. Tse, "3 user interference channel: Degrees of freedom as a function of channel diversity," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pp. 265 –271, Oct. 2009.

[15] G. Bresler, D. Cartwright, and D. Tse, "Settling the feasibility of interference alignment for the mimo interference channel: the symmetric square case," *CoRR*, vol. abs/1104.0888, 2011.

[16] M. Razaviyayn, G. Lyubeznik, and Z.-Q. Luo, "On the degrees of freedom achievable through interference alignment in a mimo interference channel," *CoRR*, vol. abs/1104.0992, 2011.

[17] A. Dimakis, P. Godfrey, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *IEEE INFOCOM*, pp. 2000 –2008, may 2007.

[18] Y. Wu and A. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *IEEE International Symposium on Information Theory*, pp. 2276 –2280, 28 2009-july 3 2009.

[19] C. Suh and K. Ramchandran, "Exact regeneration codes for distributed storage repair using interference alignment," *CoRR*, vol. abs/1001.0107, 2010. http://arxiv.org/abs/1001.0107.

[20] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction," *CoRR*, vol. abs/1005.4178, 2010. http://arxiv.org/abs/1005.4178.

[21] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramachandran, "Explicit codes minimizing repair bandwidth for distributed storage," *CoRR*, vol. abs/0908.2984, 2009. http://arxiv.org/abs/0908.2984.

[22] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *arxiv.org*, vol. abs/1004.4438, 2010. http://arxiv.org/abs/1004.4438.

[23] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," *CoRR*, vol. abs/1107.1627, July 2011. http://arxiv.org/abs/1107.1627.

[24] V. Cadambe and S. Jafar, "Interference alignment and the degrees of freedom of the k user interference channel," *IEEE Trans. on Information Theory*, vol. 54, pp. 3425–3441, Aug. 2008.

[25] C. Suh and D. Tse, "Interference alignment for cellular networks," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pp. 1037 –1044, Sept. 2008.

[26] D. S. Papailiopoulos and A. G. Dimakis, "Distributed storage codes through hadamard designs," *2011 IEEE Symposium on Information Theory (ISIT)*, July 2011.

[27] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length codes for optimal repair in distributed storage," 2011. In Preparation.