Distributed Storage Codes through Hadamard Designs

Dimitris S. Papailiopoulos and Alexandros G. Dimakis
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
Email:{papailio, dimakis}@usc.edu

***Abstract*—In distributed storage systems that employ coding, the issue of minimizing the total *repair bandwidth* required to exactly regenerate a storage node after a failure arises. This repair bandwidth depends on the structure of the storage code and the repair strategies used to restore the lost data. Minimizing it, requires that undesired data during a repair align in the smallest possible spaces, using the concept of interference alignment (IA). Here, we introduce a new representation for the symbol extension IA scheme of Cadambe *et al* in terms of points on lattices. Then, we combine new clues for perfect alignment and fundamental properties of Hadamard matrices to construct a new explicit *near* maximum-distance separable (MDS) storage code with favorable repair properties.**

Specifically, we build an explicit $(k+2,k)$ storage code over $\mathbb{GF}(3)$, whose single systematic node failures can be repaired with bandwidth that matches exactly the theoretical minimum cut-set bound. Moreover, the repair of single parity node failures generates at most the same repair bandwidth as any systematic node failure. Finally, we prove that this new storage code is near MDS, that is, $k+\frac{1}{2}$ encoded blocks suffice to retrieve the entire information.

I. Introduction

In recent years, the demand for large scale data storage has increased significantly with applications demanding seamless storage, access, and security for massive amounts of data. When the deployed storage nodes of an array are individually unreliable, as is the case in modern data centers or peer-to-peer networks, redundancy through coding can be introduced to offer reliability against node failures. However, increased reliability does not come for free: one has to address the challenge of maintaining an encoded representation when erasures occur. To maintain the same redundancy when a storage node leaves the system, a new node has to join the array, access some existing nodes, and regenerate the contents of the departed node. In its most general form this problem is known as the *Code Repair Problem* [3], [1].

The interest in the code repair problem, and specifically in designing (n,k) erasure codes for storage, stems from the fact that there exists a fundamental minimum repair bandwidth needed to regenerate a lost node, that is roughly a $\frac{1}{\min(n-k,k)}$ fraction of the encoded data object. Recently, there has been specific interest in MDS storage codes since they offer maximum reliability for a given storage capacity, for example check [2]. However, most practical solutions for storage use off-the-shelf existing MDS codes (such as Reed Solomon codes), where the naïve repair of a single node failure requires the *entire* information to be downloaded.

Designing optimal MDS storage codes, i.e., ones achieving minimum repair bandwidth, seems to be challenging especially for rates $\frac{k}{n} \geq \frac{1}{2}$. Recent works by [11] and [12] used the symbol extension technique of [4] to establish that there exist asymptotically optimal MDS storage codes, that come arbitrarily close to the theoretic minimum repair bandwidth for all n, k. However, these asymptotic schemes are impractical due to the arbitrarily large field size and the fast growing file sub-packetization, even for fixed n, k. Explicit and practical designs for optimal MDS storage codes are constructed roughly for rates $\frac{k}{n} \leq \frac{1}{2}$ [5]- [10] and [13], and most of them are based upon the concept of interference alignment. Interestingly, as of now no explicit (near) MDS storage code constructions exist with (near) optimal repair properties for the high data rate regime .

Our Contribution: In this work we introduce a new high-rate explicit $(k+2,k)$ storage code over $\mathbb{GF}(3)$. Our storage code exploits fundamental properties of Hadamard designs and a new representation for the symbol extension technique of Cadambe *et al* [4]. We observe that the columns of the matrices generated by the asymptotic scheme of [4] can be represented as points on a lattice. This representation gives hints for specific problem parameters under which *perfect* alignment can be attained with fixed file subpacketization. Our storage code has these parameters tuned and perfect alignment is attained for fixed in k file subpacketization.

Code Properties: Assuming that the file to be encoded has size M, each of the $k+2$ storage nodes stores a coded block of size $\frac{M}{k}$. Repairing a single systematic node failure costs $\frac{k+1}{2k}M$ in repair communication bandwidth, matching the theoretic minimum. Additionally, single parity node failures generate (at most) the same $\frac{k+1}{2k}M$ repair bandwidth. Finally, we show that the storage code is near

systematic node	systematic data
1	$\mathbf{f}_1$
$\vdots$	$\vdots$
k	$\mathbf{f}_k$
parity node	parity data
a	$\mathbf{A}_1^T\mathbf{f}_1 + \ldots + \mathbf{A}_k^T\mathbf{f}_k$
b	$\mathbf{B}_1^T\mathbf{f}_1 + \ldots + \mathbf{B}_k^T\mathbf{f}_k$

Fig. 1. A $(k+2, k)$ CODED STORAGE ARRAY.

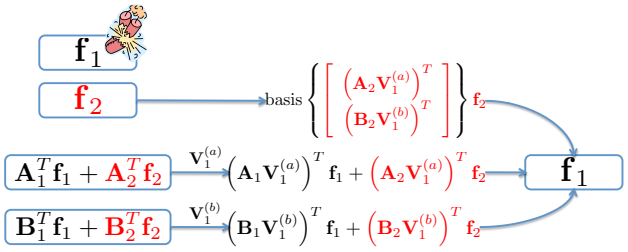

Fig. 2. Repair of a $(4, 2)$ code.

MDS. The contents of any k storage nodes and at most half the contents of an extra node can reconstruct the entire information.[1]

II. DISTRIBUTED STORAGE CODES WITH 2 PARITY NODES

In this section, we consider the code repair problem for storage codes with 2 parity nodes. After we lay down the model for repair, we continue with introducing the two fundamental ideas and tools upon which our code is built.

Let a file of size $M = 2k\beta$ denoted by the vector $\mathbf{f} \in \mathbb{F}^{2k\beta}$ be partitioned in k parts $\mathbf{f} = \left[\mathbf{f}_1^T \ldots \mathbf{f}_k^T\right]^T$, each of size 2β, where 2β denotes the subpacketization factor, $\beta \in \mathbb{N}^*$.[2] We wish to store this file with rate $\frac{k}{k+2}$ across k systematic and 2 parity storage units each having storage capacity $\frac{M}{k} = 2\beta$. To achieve this level of redundancy, the file is encoded using a $(k+2, k)$ distributed storage code. The structure of the storage array is given in Fig. 1, where $\mathbf{A}_i$ and $\mathbf{B}_i$ are $2\beta \times 2\beta$ matrices of coding coefficients used by the parity nodes a and b, respectively, to "mix" the contents of the ith file piece $\mathbf{f}_i$. Observe that the code is in systematic form: k nodes store the k parts of the file, $\mathbf{f}_1, \ldots, \mathbf{f}_k$, and each of the 2 parity nodes stores a linear combination of the k file pieces.

Remark 1: A storage code has the MDS property when any possible k-collections of coded blocks can be used to retrieve the file $\mathbf{f}$. This is equivalent to requiring all $\binom{n}{k}$ k-collections of blocks picked from $\mathbf{A}_1^T\mathbf{f}_1 + \ldots + \mathbf{A}_k^T\mathbf{f}_k$, $\mathbf{B}_1^T\mathbf{f}_1 + \ldots + \mathbf{B}_k^T\mathbf{f}_k$, and $\mathbf{f}_s$, $s \in \{1, \ldots, k\}$, to be linearly independent.[3]

To maintain the same level of redundancy when a node fails or leaves the system, the code repair process has to take place to exactly restore the lost data in a *newcomer* storage component. Let for example a systematic node $i \in \{1, \ldots, k\}$ fail. Then, a newcomer joins the storage network, connects to the remaining nodes, and has to download sufficient data to reconstruct $\mathbf{f}_i$. Observe that the missing piece $\mathbf{f}_i$ exists as a term of a linear combination *only* at each parity node, as seen in Fig. 1. It is obvious that to regenerate it, the newcomer has to download from the parity nodes at least the size of what was lost, i.e., 2β linearly independent data elements. The download contents from the parity nodes can be represented as a stack of 2β equations

$$\begin{bmatrix} \mathbf{y}_i^{(a)} \\ \mathbf{y}_i^{(b)} \end{bmatrix} \triangleq \underbrace{\begin{bmatrix} \left(\mathbf{A}_i\mathbf{V}_i^{(a)}\right)^T \\ \left(\mathbf{B}_i\mathbf{V}_i^{(b)}\right)^T \end{bmatrix}\mathbf{f}_i}_{\text{useful data}} + \sum_{u=1,u\neq i}^{k} \underbrace{\begin{bmatrix} \left(\mathbf{A}_u\mathbf{V}_i^{(a)}\right)^T \\ \left(\mathbf{B}_u\mathbf{V}_i^{(b)}\right)^T \end{bmatrix}\mathbf{f}_u}_{\text{interference by } \mathbf{f}_u}, \quad (1)$$

where $\mathbf{y}_i^{(a)}, \mathbf{y}_i^{(b)} \in \mathbb{F}^\beta$ are the equations downloaded from parity nodes a and b respectively, and $\mathbf{V}_i^{(a)}, \mathbf{V}_i^{(b)} \in \mathbb{F}^{2\beta\times\beta}$ are the *repair matrices* used to mix the parity contents.[4] Observe that retrieving $\mathbf{f}_i$ from (1) is equivalent to solving an underdetermined set of 2β equations in the $2k\beta$ unknowns of $\mathbf{f}$, with respect to only the 2β desired unknowns of $\mathbf{f}_i$. However, this is not possible due to the additive *interference* components in the received equations created by the undesired unknowns $\mathbf{f}_u$, $u \neq i$, as noted in (1). Hence, additional data need to be downloaded from the systematic nodes, which will "replicate" the interference terms and will be subtracted from the downloaded equations. To erase a single interference term, a download of a basis of equations that generates the corresponding interference suffices. Eventually, when all undesired terms are subtracted, a full rank system of 2β equations in 2β uknowns has to be formed. Thus, it can be proven that the *repair bandwidth* to exactly regenerate systematic node i is given by

$$\gamma_i = 2\beta + \sum_{u=1,u\neq i}^{k} \text{rank}\left(\left[\mathbf{A}_u\mathbf{V}_i^{(a)}\ \mathbf{B}_u\mathbf{V}_i^{(b)}\right]\right),$$

where the sum rank term is the aggregate of interference dimensions. This is why interference alignment plays a key role: the lower the interference dimensions are, the less repair data need to be downloaded. We would like to note that the theoretical minimum repair bandwidth of any node for optimal $(k+2, k)$ MDS codes is exactly $(k+1)\beta$; this corresponds to each interference spaces having rank

[1] This near MDS property is reminiscent of a Fountain-like code property: $k(1+\epsilon)$ coded symbols could give back the initial k information symbols with certain high probability.

[2] $\mathbb{F}$ denotes the finite field, over which all operation are performed.

[3] In this work, we only consider coding matrices that are diagonal, hence we will drop the transpose notation in the rest of the text.

[4] In this section we consider that the newcomer downloads the same amount of information from both parities, although in general this not need to be the case. For example, during the repair of a parity node for the code that we present in Section IV, the newcomer downloads asymmetrically from the remaining nodes.

β. An abstract example of a code repair instance for a $(4, 2)$ storage code is given in Fig. 2.

Apparently, γ_i heavily depends on both the storage code and repair matrices design. In the following, we provide some new and old tools that assist us in designing an explicit $(k+2, k)$ near MDS storage code (along with repair matrices), whose repair bandwidth of any node is exactly $(k+1)\beta$.

III. The Lattice and Hadamard Matrix Toolbox

A. Dots on a lattice: Taking care of perfect alignment

Let an 8×4 matrix $\mathbf{V} = [\mathbf{w}\ \mathbf{X}_1\mathbf{w}\ \mathbf{X}_2\mathbf{w}\ \mathbf{X}_1\mathbf{X}_2\mathbf{w}]$, where $\mathbf{X}_1$ and $\mathbf{X}_2$ are diagonal matrices, not a multiple of each other, and $\mathbf{w}$ is the all ones vector of length 4. Moreover, consider the product

$$\mathbf{X}_1\mathbf{V} = \left[\mathbf{X}_1\mathbf{w}\ \mathbf{X}_1^2\mathbf{w}\ \mathbf{X}_1\mathbf{X}_2\mathbf{w}\ \mathbf{X}_1^2\mathbf{X}_2\mathbf{w}\right]. \tag{2}$$

If we wish to check the number of recurring vectors in $[\mathbf{V}\ \ \mathbf{X}_1\mathbf{V}]$, i.e. the level of "alignment" of the space elements, then observe that we need to only check which tuples of powers of $\mathbf{X}_1$ and $\mathbf{X}_2$ "generate" each column vector. Observe, that using a point representation and checking the intersection between sets $\{(0,0),(0,1),(1,0),(1,1)\}$ for $\mathbf{V}$ and $\{(1,0),(1,1),(2,0),(2,1)\}$ for $\mathbf{X}_1\mathbf{V}$, is equivalent to enumerating duplicate columns in $[\mathbf{V}\ \ \mathbf{X}_1\mathbf{V}]$. The need in enumerating overlapping vectors is especially useful when analyzing the symbol extension technique of [4]. In the following, we formalize this points on a lattice representation.

Let a set of $2\beta \times 2\beta$ diagonal matrices $\{\mathbf{X}_s\}_{s=1}^k$ over some field $\mathbb{F}$, such that no matrix in $\{\mathbf{X}_s\}_{s=1}^k$ can be written as a scaled product $c\prod_{s=1}^k \mathbf{X}_s^{x_s}$ of the matrices in that set, i.e. the matrices form a "base" of all possible products. Moreover, let a full-rank $2\beta \times \beta$ matrices $\mathbf{V}_i$ whose columns are the elements of the set

$$\mathcal{V}_i = \left\{ \prod_{s=1, s\neq i}^{k} \mathbf{X}_s^{x_s}\mathbf{w} :\ x_s \in \{0, \ldots, \Delta-1\} \right\}, \tag{3}$$

where $\Delta^{k-1} = \beta$, and β is picked such that Δ is an integer allowing exactly β vectors in $\mathcal{V}_i$. Then, we define a lattice representation of $\mathbf{V}$ as the arrangement of β points in the following set

$$\mathcal{L}(\mathbf{V}_i) \triangleq \left\{ \sum_{s=1, s\neq i}^{k} x_s\mathbf{e}_s;\ x_s \in \{0, \ldots, \Delta-1\} \right\}, \tag{4}$$

where $\mathbf{e}_i$ is the i-th basis vector of $\mathbb{R}^k$. Observe that the i-th dimension of the space where $\mathcal{L}(\mathbf{V}_i)$ lies in, indicates the possible exponents x_i of $\mathbf{X}_i$. For this definition, we state some properties that will become useful in the following developments.

(L.1) The number of distinct points in $\mathcal{L}(\mathbf{V}_i)$ equals the rank of $\mathbf{V}_i$, i.e., $|\mathcal{L}(\mathbf{V}_i)| = \text{rank}(\mathbf{V}_i)$.

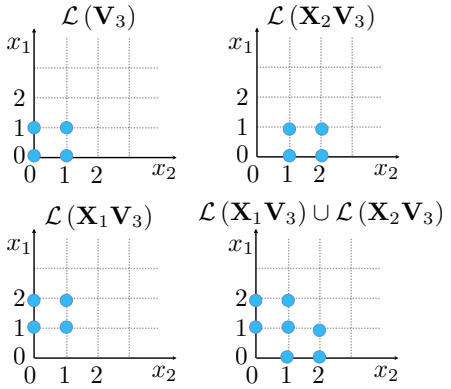

Fig. 3. Here we have $k = 3$, $\beta = 4$, and $\Delta = 2$. Moreover, $\mathcal{L}(\mathbf{V}_3) = \{(0,0,0),(0,1,0),(1,0,0),(1,1,0)\}$, $\mathcal{L}(\mathbf{X}_1\mathbf{V}_3) = \{(1,0,0),(1,1,0),(2,0,0),(2,1,0)\}$, and $\mathcal{L}(\mathbf{X}_2\mathbf{V}_3) = \{(0,1,0),(0,2,0),(1,1,0),(1,2,0)\}$.

(L.2) The products $\mathbf{X}_j\mathbf{V}_i$, $j \neq i$, and $\mathbf{X}_i\mathbf{V}_i$ map to

$$\mathcal{L}(\mathbf{X}_j\mathbf{V}) = \left\{ (x_j{+}1)\mathbf{e}_j + \sum_{s=1, s\neq i}^{k} x_s\mathbf{e}_s;\ x_s \in \{0, \ldots, \Delta{-}1\} \right\}$$

and $\mathcal{L}(\mathbf{X}_i\mathbf{V}) = \left\{ e_i{+}\sum_{i=1, s\neq i}^{k} x_i\mathbf{e}_i;\ x_s \in \{0, \ldots, \Delta{-}1\} \right\}$, respectively.

(L.3) The cardinality of a union of sets, equals the rank of the concatenation of the corresponding matrices, i.e.,

$$\left| \bigcup_{l=1}^{K} \mathcal{L}(\mathbf{X}_{i_l}\mathbf{V}_i) \right| = \text{rank}\left([\mathbf{X}_{i_1}\mathbf{V}_i \ldots \mathbf{X}_{i_K}\mathbf{V}_i]\right),$$

for some integer $K \geq 1$ and a collection of indeces $i_j \in \{1, \ldots, N\}$.

Remark 2: Observe that $\mathcal{L}(\mathbf{X}_j\mathbf{V}_i)$ corresponds to a single position shift of $\mathcal{L}(\mathbf{V}_i)$ towards greater integer values along the axis x_j and $\mathcal{L}(\mathbf{X}_i\mathbf{V}_i)$ "positions" $\mathcal{L}(\mathbf{V}_i)$ in a higher dimensional space of vectors not overlapping with any shifts of $\mathcal{L}(\mathbf{V}_i)$.

In Fig. 2, we give an illustrative example for $k = 3$, $\beta = 4$, and $\Delta = 2$, where $\mathbf{V}_3$ is the same as the motivating example in the beginning of this section assuming base matrices $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$.

Now, for our interference alignment interests, we observe that if we could design the $\mathbf{X}_i$ matrices and pick the parameter Δ, then favorable properties can occur. For example, consider diagonal matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ over the reals, whose diagonal elements are arbitrarily selected from $\{\pm 1\}$ and assume that $\Delta = 1$. Then, $\mathbf{X}_1^2 = \mathbf{X}_2^2 = \mathbf{X}_1^0 = \mathbf{X}_2^0$, due to which we obtain

$$\begin{aligned}\mathcal{L}(\mathbf{X}_1\mathbf{V}) &= \{x_1\mathbf{e}_1{+}, x_2\mathbf{e}_2;\ x_1, x_2 \in \{0,1\}\} \\ &= \mathcal{L}(\mathbf{X}_2\mathbf{V}) = \mathcal{L}(\mathbf{V}).\end{aligned}$$

The same holds for diagonals over $\mathbb{GF}(3)$. Generally, we can see that by enforcing the diagonal matrices to have a "cyclic property" we can achieve perfect alignment. We thus obtain the following lemma.

Lemma 1: For diagonal matrices, $\mathbf{X}_1, \ldots, \mathbf{X}_k$, whose elements are either selected as *i)* $(q-1)$-th roots of unity, or *ii)* elements from $\mathbb{GF}(q)$, we have that $\mathbf{X}_s^{q-1} = \mathbf{X}_s^0$, for all $s \in \{1, \ldots, k\}$, and $\mathcal{L}(\mathbf{X}_j \mathbf{V}_i) = \mathcal{L}(\mathbf{V}_i)$, for all $i \in \{1, \ldots, k\} \backslash j$.

We use this simple Lemma, to generate storage coding matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ having similar properties such that during repairs perfect alignment is possible.

B. Hadamard Designs: Taking care of the full-rank property

Let a $2\beta \times 2\beta$ Hadamard matrix $\mathbf{H}_{2\beta}$ of the Sylvester's construction over $\mathbb{GF}(3)$

$$\mathbf{H}_{2\beta} = \begin{bmatrix} \mathbf{H}_\beta & \mathbf{H}_\beta \\ \mathbf{H}_\beta & 2\mathbf{H}_\beta \end{bmatrix}, \tag{5}$$

with $\mathbf{H}_1 = 1$. Then, we have the following properties for this matrix.

(H.1) $\mathbf{H}_{2\beta}$ is full-rank with mutually orthogonal columns.
(H.2) Any two columns of $\mathbf{H}_{2\beta}$ differ in β positions.
(H.3) The columns of $\mathbf{H}_{2\beta}$ are the elements of the set

$$\mathcal{H}_{2\beta} = \left\{ \prod_{i=1}^{\log_2(2\beta)} \mathbf{X}_i^{x_i} \mathbf{w} : x_i \in \{0,1\} \right\}, \tag{6}$$

where $\mathbf{X}_i = \mathbf{I}_{2^{i-1}} \otimes \text{blkdiag}\left(\mathbf{I}_{\frac{\beta}{2^{i-1}}}, 2\mathbf{I}_{\frac{\beta}{2^{i-1}}}\right)$ is a $2\beta \times 2\beta$ diagonal matrix.

The proofs of these properties can be found in the Appendix.

For an illustrative example of property (H.3) consider the following

$$\begin{aligned} \mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 \\ 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} &= \begin{bmatrix} 1 & 1 & & \\ 1 & 2 & & \\ 1 & 1 & \mathbf{X}_1 & \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \\ 1 & 2 & & \end{bmatrix} \\ \text{and } \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} &= \begin{bmatrix} 1 & & \\ 1 & & \\ 1 & \mathbf{X}_2 & \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ 1 & & \end{bmatrix} \\ \Leftrightarrow \mathbf{H}_4 &= [\mathbf{w} \;\; \mathbf{X}_2\mathbf{w} \;\; \mathbf{X}_1\mathbf{w} \;\; \mathbf{X}_2\mathbf{X}_1\mathbf{w}]. \end{aligned} \tag{7}$$

Interestingly, due to property (H.3), we can represent $\mathbf{H}_{2\beta}$ as

$$\mathcal{L}(\mathbf{H}_{2\beta}) = \left\{ \sum_{s=1}^{\log(2\beta)} x_s \mathbf{e}_s ;\; x_s \in \{0,1\} \right\}. \tag{8}$$

In the following, based on the aforementioned properties of the Hadamard designs, we introduce a storage code that during repair useful data spaces are full rank.

IV. A New Storage Code

We introduce a $(k+2, k)$ near MDS storage code over $\mathbb{GF}(3)$, with subpacketization factor $\beta = 2^{k-1}$ and coding matrices

$$\mathbf{A}_i = \mathbf{I}_{2\beta}, \;\; \mathbf{B}_i = \mathbf{X}_i, \tag{9}$$

where $\mathbf{X}_i = \mathbf{I}_{2^{i-1}} \otimes \text{blkdiag}\left(\mathbf{I}_{\frac{\beta}{2^{i-1}}}, 2\mathbf{I}_{\frac{\beta}{2^{i-1}}}\right)$, $i \in \{1, \ldots, k\}$, are the same matrices defined in (H.3). When systematic node $i \in \{1, \ldots, k\}$ fails, the columns of the repair matrices $\mathbf{V}_i^{(a)} = \mathbf{V}_i^{(b)} = \mathbf{V}_i$ are the elements of the set

$$\mathcal{V}_i = \left\{ \prod_{s=1, s\neq i}^{k} \mathbf{X}_s^{x_s} \mathbf{w} : x_s \in \{0,1\} \right\}. \tag{10}$$

For the repair of parity node failures, please refer to subsection IV.B. We continue by proving the repair properties of our storage code.

A. Repairing the systematic parts of the code

Let systematic node i fail. First, observe that the repair matrix $\mathbf{V}_i$ is full column rank since it is a collection of β distinct columns from $\mathcal{H}_{2\beta}$. More precisely observe that

$$\begin{aligned} \text{rank}(\mathbf{V}_i) &= |\mathcal{L}(\mathbf{V}_i)| \\ &= \left| \left\{ \sum_{s=1, s\neq i}^{k} x_i \mathbf{e}_i ;\; x_i \in \{0,1\} \right\} \right| = \beta. \end{aligned} \tag{11}$$

Now, observe that our storage coding matrices are drawn over $\mathbb{GF}(3)$, hence they have the cyclic property presented in Lemma 1. Then, it is easy to prove that the interference spaces span the minimum dimensions possible.

Lemma 2: The interference spaces during the exact repair of a systematic node $i \in \{1, \ldots, k\}$ have rank β, i.e.

$$\text{rank}([\mathbf{V}_i \;\; \mathbf{X}_j \mathbf{V}_i]) = \beta, \tag{12}$$

for any $j \in \{1, \ldots, k\} \backslash i$.

Proof: Due to Lemma 1 we have that $\mathcal{L}(\mathbf{X}_j \mathbf{V}_i) = \mathcal{L}(\mathbf{V}_i)$, hence $\text{rank}([\mathbf{V}_i \; \mathbf{X}_j \mathbf{V}_i]) = |\mathcal{L}(\mathbf{V}_i)| = \beta$. □

Again, observe that the selection of the coding and repair matrices is done in such a way that the repair matrices are half the columns of a 2β Hadamard matrix. Therefore, we have the following lemma.

Lemma 3: The desired data space during the exact repair of a single systematic node $i \in \{1, \ldots, k\}$ is full-rank, i.e.

$$\text{rank}([\mathbf{V}_i \;\; \mathbf{X}_i \mathbf{V}_i]) = 2\beta. \tag{13}$$

Proof: Observe that

$$\mathcal{L}(\mathbf{X}_i \mathbf{V}_i) = \left\{ e_i + \sum_{s=1, s\neq i}^{k} x_i \mathbf{e}_i ;\; x_i \in \{0,1\} \right\}. \tag{14}$$

Hence,

$$\begin{aligned} \mathcal{L}(\mathbf{V}_i) \cup \mathcal{L}(\mathbf{X}_i \mathbf{V}_i) &= \left\{ \sum_{s=1, s\neq i}^{k} x_i \mathbf{e}_i ;\; x_i \in \{0,1\} \right\} \\ &\quad \bigcup \left\{ e_i + \sum_{s=1, s\neq i}^{k} x_i \mathbf{e}_i ;\; x_i \in \{0,1\} \right\} \\ &= \mathcal{L}(\mathbf{H}_{2\beta}). \end{aligned} \tag{15}$$

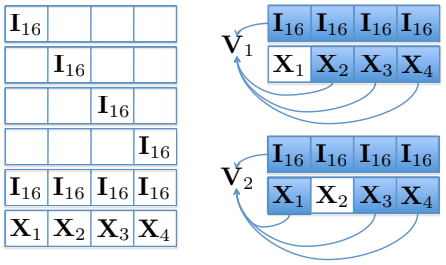

Fig. 4. The coding and repair matrices of the $(6, 4)$ code. We illustrate the "absorbing" properties of the repair matrices, for the repair of systematic node 1 and 2. The columns spaces of the repair matrices are invariant to the corresponding blue blocks.

Therefore, the elements of the desired data matrix are the elements of the set $\mathcal{H}_{2\beta}$, hence $[\mathbf{V}_i \;\; \mathbf{X}_i\mathbf{V}_i]$ is full rank. □
Due to Lemma 2 and 3 we obtain the following.

Theorem 1: A single systematic node i of the code can be repaired with communication bandwidth

$$\gamma_i = (k+1)\beta = \frac{k+1}{2k}M, \tag{16}$$

for all $i \in \{1, \ldots, k\}$.
For illustrative purposes, Fig. 4 depicts a $(6, 4)$ code of our new construction, along with the properties of some repair matrices.

B. Repairing the parity nodes

In this subsection we consider the case of a single parity node repair and prove that it generetes at most the repair bandwidth of a single systematic repair.

Theorem 2: The repair bandwidth to repair a single parity node failure of the code is at most $(k+1)\beta$.[5]

Proof: Let parity node a fail. Then, the newcomer uses the $2\beta \times 2\beta$ repair matrix $\mathbf{V}_a^{(b)} = \mathbf{X}_1$ to multiply parity node b and downloads

$$\mathbf{X}_1 \left(\sum_{i=1}^{k} \mathbf{X}_1 \mathbf{f}_i \right) = \mathbf{f}_1 + \sum_{i=2}^{k} \mathbf{X}_1\mathbf{X}_i\mathbf{f}_i. \tag{17}$$

Observe, that the component corresponding to systematic part $\mathbf{f}_1$ is the same as the contents of the lost parity node a. Interestingly, using (H.2) we observe that each of the remaining blocks, $\mathbf{X}_1\mathbf{X}_i\mathbf{f}_i$ have exactly β positions identical to the corresponding β positions of $\mathbf{f}_i$, that is one of the rest $k-1$ terms that needs to be reconstructed. This is due to the fact that the diagonal elements of matrices $\mathbf{X}_1\mathbf{X}_i$ and $\mathbf{I}_{2\beta}$ are the elements of some two columns of of $\mathbf{H}_{2\beta}$; hence, they differ in β positions due to (H.2). Therefore, the newcomer has to download from systematic node $i \in \{2, \ldots, k\}$, the β entries that parity a's contents $\mathbf{f}_i$ differ from $\mathbf{X}_1\mathbf{X}_j\mathbf{f}_i$. More specifically, let

$$\mathcal{I}_i = \left\{ l; l \in \{1, \ldots, 2\beta\} \text{ and } [\mathbf{X}_1\mathbf{X}_i]_{l,l} \neq 1 \right\} \tag{18}$$

[5]By "at most" we mean that this result is proved using an achievable scheme, however, we do not prove that it is optimal.

be the set of β indeces at which the diagonal elements of $\mathbf{X}_1\mathbf{X}_i$ and $\mathbf{I}_{2\beta}$ differ. Then, the $2\beta \times \beta$ repair matrix used to download contents from systematic node $i \in \{2, \ldots, k\}$ is

$$\mathbf{V}_i^{(a)} = \left([\mathbf{I}_{2\beta}]_{\mathcal{I}_i,:} \, \mathbf{X}_1\mathbf{X}_i \right)^T. \tag{19}$$

That way, the newcomer will have to download 2β equations from parity node b and β equations from each of the $(k-1)$ systematic nodes $i\{2, \ldots, k\}$; thus, it can reconstruct the lost piece by exactly downloading $(k+1)\beta$. The repair of parity node b can be performed in an according fashion. □

C. Data collectors and the MDS property

The MDS property of storage codes can be tested by using the concept of *Data Collectors* (DCs). A storage code satisfies the MDS property if any DC that connects to any k out of n storage nodes and retrieve the file $\mathbf{f}$. To test this perfect file reconstruction property for each DC, a matrix is associated with it.

For the case of storage nodes with 2 parity nodes we test this property for DCs that connect to either *i)* $k-1$ systematic nodes and 1 parity, or *ii)* $k-2$ systematic and the 2 parity nodes. For these cases we have the following theorems for our new code.

Theorem 3: The matrix of a DC connecting to $k-1$ systematic and 1 parity node is full rank M, i.e. the DC can perfectly reconstruct the file. Morover, the matrix of a DC connecting to $k-2$ systematic and 2 parity nodes has rank $M-\beta$, i.e. the DC cannot perfectly reconstruct $\mathbf{f}$ and needs to connect to an extra node to download β additional equations that assist perfect file reconstruction.

Corollary 1: For our code, a DC that connects to $k+1$ nodes can perfectly reconstruct $\mathbf{f}$ by downloading $k + \frac{1}{2}$ coded blocks.

D. Appendix

Proof of (H.1): First observe that $\mathbf{H}_{2\beta} = \mathbf{H}_{2\beta}^T$. Moreover,

$$\begin{aligned}
\mathbf{H}_{2\beta}\mathbf{H}_{2\beta}^T = \mathbf{H}_{2\beta}\mathbf{H}_{2\beta} &= \begin{bmatrix} 2\mathbf{H}_\beta\mathbf{H}_\beta & \mathbf{0}_{\beta\times\beta} \\ \mathbf{0}_{\beta\times\beta} & 2\mathbf{H}_\beta\mathbf{H}_\beta \end{bmatrix} \\
&= 2\left(\mathbf{I}_2 \otimes \mathbf{H}_\beta\mathbf{H}_\beta\right) && \pmod 3 \\
&= 2\left(\mathbf{I}_2 \otimes 2\left(\mathbf{I}_2 \otimes \mathbf{H}_\beta\mathbf{H}_{\frac{\beta}{2}}\right)\right) && \pmod 3 \\
&= 4\left(\mathbf{I}_4 \otimes \mathbf{H}_{\frac{\beta}{2}}\mathbf{H}_{\frac{\beta}{2}}\right) && \pmod 3 \\
&\vdots \\
&= 2\beta \cdot \left(\mathbf{I}_{2\beta} \otimes \mathbf{H}_1\mathbf{H}_1\right) && \pmod 3 \\
&= 2\beta \cdot \mathbf{I}_{2\beta}. && \pmod 3
\end{aligned}$$

Therefore, the rank of $\mathbf{H}_{2\beta}$ is 2β and its columns are mutually orthogonal. □

Proof of (H.2): Due to (H.1) and for any two distinct columns of $\mathbf{H}_{2\beta}$, we have

$$\begin{aligned}\left[\mathbf{H}_{2\beta}\right]_{:,i}^{T}\left[\mathbf{H}_{2\beta}\right]_{:,j} &= 0 \pmod 3\\ \sum_{k=1}^{2\beta}\left[\mathbf{H}_{2\beta}\right]_{k,i}\left[\mathbf{H}_{2\beta}\right]_{k,j} &= 0 \pmod 3\\ \beta\cdot 1+\beta\cdot 2 &= 0 \pmod 3. \end{aligned} \tag{20}$$

This is equivalent to the fact that β elements are equal and β differ, which proves the property. □

Proof of (H.3): Let a $2\beta\times 2\beta$ diagonal matrix

$$\mathbf{X}_i = \mathbf{I}_{2^{i-1}}\otimes \text{blkdiag}\left(\mathbf{I}_{\frac{\beta}{2^{i-1}}}, 2\mathbf{I}_{\frac{\beta}{2^{i-1}}}\right) \tag{21}$$

defined for $i=\{1,\ldots,\log_2(2\beta)\}$. $\mathbf{X}_i$ is a diagonal matrix, whose elements is a series of alternating 1s and 2s, starting with $\frac{\beta}{2^{i-1}}$ 1s that flip to 2s and back every $\frac{\beta}{2^{i-1}}$ positions. Observe that we can expand $\mathbf{H}_{2\beta}$ in the following way

$$\begin{aligned}\mathbf{H}_{2\beta} &= \begin{bmatrix}\mathbf{H}_\beta & \mathbf{H}_\beta\\ \mathbf{H}_\beta & 2\mathbf{H}_\beta\end{bmatrix}\\ &= \begin{bmatrix}\underbrace{\mathbf{1}_{2\times 1}\otimes\mathbf{H}_\beta}_{\mathbf{F}_1} & \mathbf{X}_1\left(\mathbf{1}_{2\times 1}\otimes\mathbf{H}_\beta\right)\end{bmatrix}. \end{aligned} \tag{22}$$

We proceed in the same manner by expanding all "smaller" $\mathbf{H}_i$s

$$\begin{aligned}\mathbf{F}_1 &= \mathbf{1}_{2\times 1}\otimes\left[\mathbf{1}_{2\times 1}\otimes\mathbf{H}_{\frac{\beta}{2}} \quad \mathbf{X}_1\left(\mathbf{1}_{2\times 1}\otimes\mathbf{H}_{\frac{\beta}{2}}\right)\right]\\ &= \begin{bmatrix}\underbrace{\mathbf{1}_{2^2\times 1}\otimes\mathbf{H}_{\frac{\beta}{2^1}}}_{\mathbf{F}_2} & \mathbf{X}_2\left(\mathbf{1}_{2^2\times 1}\otimes\mathbf{H}_{\frac{\beta}{2^1}}\right)\end{bmatrix}\\ \mathbf{F}_2 &= \begin{bmatrix}\underbrace{\mathbf{1}_{2^3\times 1}\otimes\mathbf{H}_{\frac{\beta}{2^2}}}_{\mathbf{F}_3} & \mathbf{X}_3\left(\mathbf{1}_{2^3\times 1}\otimes\mathbf{H}_{\frac{\beta}{2^2}}\right)\end{bmatrix}\\ &\vdots\\ \mathbf{F}_{\log_2(2\beta)-1} &= \left[\mathbf{1}_{2\beta\times 1} \quad \mathbf{X}_{\log_2(2\beta)}\mathbf{1}_{2\beta\times 1}\right], \end{aligned} \tag{23}$$

where $\mathbf{F}_i$ is a $2\beta\times\frac{2\beta}{2^i}$ matrix. Thus,

$$\begin{aligned}\text{span}\left(\mathbf{H}_{2\beta}\right) &= \text{span}\left(\left[\mathbf{F}_1 \;\; \mathbf{X}_1\mathbf{F}_1\right]\right)\\ &= \text{span}\left(\left[\mathbf{F}_2 \;\; \mathbf{X}_2\mathbf{F}_2 \;\; \mathbf{X}_1\mathbf{F}_2 \;\; \mathbf{X}_1\mathbf{X}_2\mathbf{F}_2\right]\right)\\ &\vdots\\ &= \text{span}\left(\left\{\prod_{i=1}^{\log_2(2\beta)}\mathbf{X}_i^{x_i}\mathbf{w} : x_i\in\{0,1\}\right\}\right), \end{aligned}$$

which proves the property. □

Proof of Theorem 3: We begin by showing the first part of Theorem 3. For this case, we consider a DC that connects to systematic nodes $\{1,\ldots,k-1\}$ and parity node a. The determinant of the corresponding DC matrix is

$$\begin{aligned}&\det\left(\left[\begin{array}{ccc|c}\mathbf{I}_{2\beta} & \ldots & \mathbf{0}_{2\beta\times 2\beta} & \mathbf{0}_{2\beta\times 2\beta}\\ \vdots & & \vdots & \vdots\\ \mathbf{0}_{2\beta\times 2\beta} & \ldots & \mathbf{I}_{2\beta} & \mathbf{0}_{2\beta\times 2\beta}\\ \hline \mathbf{A}_1 & \ldots & \mathbf{A}_{k-1} & \mathbf{A}_k\end{array}\right]\right)\\ &= \det\left(\mathbf{A}_k\right)\neq 0, \end{aligned} \tag{24}$$

since $\mathbf{A}_k$ is a full rank diagonal matrix. AccordinglyI, we can prove that any DC that connects to $k-1$ systematic and 1 parity node can reconstruct $\mathbf{f}$, since the download information has M linearly independent equations.

We continue with proving the second part of Theorem 3. Here, we consider the DC that connects to systematic nodes $\{1,\ldots,k-2\}$ and both parity nodes. The corresponding matrix will be

$$\left[\begin{array}{ccc|cc}\mathbf{I}_{2\beta} & \ldots & \mathbf{0}_{2\beta\times 2\beta} & \mathbf{0}_{2\beta\times 2\beta} & \mathbf{0}_{2\beta\times 2\beta}\\ \vdots & & \vdots & \vdots & \\ \mathbf{0}_{2\beta\times 2\beta} & \ldots & \mathbf{I}_{2\beta} & \mathbf{0}_{2\beta\times 2\beta} & \mathbf{0}_{2\beta\times 2\beta}\\ \hline \mathbf{I}_{2\beta} & \ldots & \mathbf{I}_{2\beta} & \mathbf{I}_{2\beta} & \mathbf{I}_{2\beta}\\ \mathbf{X}_1 & \ldots & \mathbf{X}_{k-2} & \mathbf{X}_{k-1} & \mathbf{X}_k\end{array}\right]. \tag{25}$$

Observe that the leftmost $2(k-2)\beta$ columns of the matrix in (25) are linearly independent, due to the upper-left identity block. Moreover, the leftmost $2(k-2)\beta$ columns are linearly independent with the rightmost 4β each other, using an analogous argument. Hence, we need to only check the rank of the submatrix

$$\begin{bmatrix}\mathbf{I}_{2\beta} & \mathbf{I}_{2\beta}\\ \mathbf{X}_{k-1} & \mathbf{X}_k\end{bmatrix}. \tag{26}$$

In general, for any DC matrix of this type, we need to check the rank of

$$\begin{bmatrix}\mathbf{I}_{2\beta} & \mathbf{I}_{2\beta}\\ \mathbf{X}_i & \mathbf{X}_j\end{bmatrix}, \tag{27}$$

for $i,j\in\{1,\ldots,k\}$ and $i\neq j$. Observe that the following holds

$$\begin{aligned}&\text{rank}\left(\begin{bmatrix}\mathbf{I}_{2\beta} & \mathbf{I}_{2\beta}\\ \mathbf{X}_i & \mathbf{X}_j\end{bmatrix}\right)\\ &= \text{rank}\left(\begin{bmatrix}\mathbf{I}_{2\beta} & \mathbf{I}_{2\beta}\\ \mathbf{X}_i & \mathbf{X}_j\end{bmatrix}\begin{bmatrix}\mathbf{I}_{2\beta} & -\mathbf{I}_{2\beta}\\ \mathbf{0}_{2\beta\times 2\beta} & \mathbf{I}_{2\beta}\end{bmatrix}\right)\\ &= \text{rank}\left(\begin{bmatrix}\mathbf{I}_{2\beta} & 0\\ \mathbf{X}_i & \mathbf{X}_j-\mathbf{X}_i\end{bmatrix}\right) \qquad (28)\\ &= 2\beta+\text{rank}\left(\mathbf{X}_j-\mathbf{X}_i\right)=3\beta. \qquad (29)\end{aligned}$$

This is due to the fact that $\mathbf{X}_j-\mathbf{X}_i$ has β zero columns due to (H.2). Hence, the rank of the matrix in (25) is equal to $M-\beta$ and a DC of this type has access to $M-\beta$ linearly independent equations. This means that an extra β need to be downloaded from an additional systematic node. Therefore, $k+\frac{1}{2}$ coded blocks, i.e. $M+\beta$ equations, are required for file reconstruction. □

REFERENCES

[1] The Coding for Distributed Storage wiki `http://tinyurl.com/storagecoding`

[2] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in raid architectures," in *IEEE Transactions on Computers*, 1995.

[3] A. G. Dimakis, P. G. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," IEEE Transactions on Information Theory, to appear.

[4] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the K user interference channel," *IEEE Transactions on Information Theory*, 54(8), pp. 3425–3441, Aug. 2008.

[5] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, Seoul, Korea, July 2009.

[6] D. Cullina, A. G. Dimakis, and T. Ho, "Searching for minimum storage regenerating codes," In *Allerton Conference on Control, Computing, and Communication*, Urbana-Champaign, IL, September 2009.

[7] Rashmi K.V., N. B. Shah, P. V. Kumar, and K. Ramchandran "Exact regenerating codes for distributed storage," In *Allerton Conference on Control, Computing, and Communication*, Urbana-Champaign, IL, September 2009 (preprint available at http://arxiv.org/abs/0906.4913).

"""

[8] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. IEEE ITW*, Jan. 2010 (preprint available at http://arxiv.org/abs/0908.2984).

[9] C. Suh and K. Ramchandran, "Exact Regeneration Codes for Distributed Storage Repair Using Interference Alignment," *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, June 2010 (Preprint available at http://arxiv.org/abs/1001.0107v2).

[10] Y. Wu. "A construction of systematic MDS codes with minimum repair bandwidth," Submitted to *IEEE Transactions on Information Theory*, Aug. 2009. Preprint available at http://arxiv.org/abs/0910.2486.

[11] V. Cadambe, S. Jafar, and H. Maleki, "Distributed data storage with minimum storage regenerating codes - exact and functional repair are asymptotically equally efficient," in *IEEE International Workshop on Wireless Network Coding (WiNC) ,2010. (preprint available at http://arxiv.org/abs/1004.4299)*, Apr 2010.

[12] C. Suh and K. Ramchandran, "On the Existence of Optimal Exact-Repair MDS Codes for Distributed Storage", Apr. 2010. Preprint available online at http://arxiv.org/abs/1004.4663

[13] K. 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," submitted to IEEE Transactions on Information Theory, arXiv:1005.4178 [cs.IT].