Repair Optimal Erasure Codes through Hadamard Designs

Dimitris S. Papailiopoulos[†], Alexandros G. Dimakis[†], and Viveck R. Cadambe[*]
[†]Electrical Engineering, University of Southern California
Email:{papailio, dimakis}@usc.edu
[*]Research Laboratory of Electronics, Massachusetts Institute of Technology
Email: viveck@mit.edu [*]

October 10, 2012

Abstract

In distributed storage systems that employ erasure coding, the issue of minimizing the total *communication* required to exactly rebuild 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. Designing high-rate maximum-distance separable (MDS) codes that achieve the optimum repair communication has been a well-known open problem. Our work resolves, in part, this open problem. In this work, we use Hadamard matrices to construct the first explicit 2-parity MDS storage code with optimal repair properties *for all* single node failures, including the parities. Our construction relies on a novel method of achieving *perfect* interference alignment over finite fields with a finite number of symbol extensions. We generalize this construction to design m-parity MDS codes that achieve the optimum repair communication for single systematic node failures.

1 Introduction

Distributed storage systems have reached such a massive scale that recovery from failures is now part of regular operation rather than a rare exception [5]. Large-scale distributed storage systems need to tolerate multiple failures, both to provide high availability and to prevent data loss. Erasure coding is particularly attractive since it provides higher reliability without requiring a large number of data replicas that increase cost as data grows. Storage applications where erasure coding techniques are being currently deployed include distributed file systems like Facebook's coded Hadoop, Google Colossus, and Microsoft Azure [33]. Peer-to-peer storage systems like Cleversafe and Wuala (see e.g. [2, 4]) are also typically using Reed-Solomon or more sophisticated storage erasure codes.

One central issue in coded distributed storage systems is the code repair problem: How to optimize the maintenance of encoded representation when failures occur. To maintain the same redundancy when a storage node fails or leaves the system, a *newcomer* node has to join the array, access some existing nodes, and exactly reproduce the contents of the departed node. In its most general form this problem is known as the *Exact Code Repair Problem* [2, 3]. There are several metrics that can be optimized during repair: the total information read from existing disks during repair [9, 10], the total information communicated in the distributed storage network (repair bandwidth [3]), or locality, the total number of disks accessed for each repair [6, 11, 30–34].

[*]A preliminary version of this work was presented in [1].

Currently, the most well-understood metric is that of repair bandwidth. In this work we are particularly interested in constructing (n, k)-MDS storage codes that are optimal with respect to the repair bandwidth. The information theoretic bounds for repair bandwidth were specified in [3] and shown to be asymptotically, and in some cases exactly tight for all values of n, k in a series of recent papers [4,15,18,20–22]. Beyond MDS codes, [3] demonstrated a tradeoff between storage and repair bandwidth, and code constructions for other points of this tradeoff are under active investigation, see e.g. [4,22,27,40]. On this tradeoff, the minimum storage point is achieved by MDS erasure codes with optimal repair, also known as Minimum Storage Regenerating (MSR) codes.

For code rates $k/n \leq 1/2$, explicit MSR codes were designed by Shah *et al.* [18], Rashmi *et al.* [22], and Suh *et al.* [17]. For the high-rate regime however, with the exception of the special cases where $k = 2, 3,$ [13, 14, 17, 18], the only known complete constructions [20, 21] require arbitrarily large file sizes (symbol extensions) and field order. These constructions use the symbol extension interference alignment technique of [12] to establish that there exist MDS storage codes, that come arbitrarily close to (but do not exactly match) the information theoretic lower bound of the repair bandwidth for all n, k. These asymptotic constructions are impractical due to the arbitrarily large finite field order and the fast growing file size that is required, even for small values of n and k.

Our Contribution: We introduce the first explicit high-rate $(k + 2, k)$-MDS storage code with optimal repair communication. Our storage code exploits fundamental properties of Hadamard designs and perfect interference alignment instances that can be understood through the use of a lattice representation of the symbol extension technique of Cadambe *et al.* [12, 20, 21].

Independently of this work, there has recently been a substantial progress in designing high-rate explicit MSR codes. Tamo et al. [23] and Cadambe et al. [25] designed MDS codes for any (n, k) parameters that have optimal repair for the systematic nodes. In fact, while several code constructions exist for repairing systematic nodes optimally [23–25], the existence of codes which can optimally repair parity nodes as well remained an open problem. The advantage of our work is that all n nodes are optimally repaired and the disadvantage is that our construction is currently only optimal for $n - k = 2$. In parallel to and independently from our results, Wang et al. [26] have constructed codes which are optimal for the repair of all nodes. The construction of [26] is different from ours, though exploration of underlying connections is a possible direction for future work.

Our key technical contribution is a scheme that achieves *perfect* interference alignment with a finite number of extensions that we present in Section III. This was developed in [29] and used in a 2 parity storage code with optimal repair for k nodes and near optimal repair of 2 nodes, that can handle any single node failure. We use a combinatorial view of different interference alignment schemes using a framework we call *dots-on-a-lattice*. Hadamard matrices are shown to be crucial in achieving finite perfect alignment while ensuring the full-rank of desired subspaces. In Sections V and VI, we prove the repair bandwidth optimality of our code construction. In Section VII, we give explicit conditions on the MDS property of the code and show that a finite field of order greater than or equal to $2k + 3$ suffices to satisfy them.

Finally, in Section VIII, we present m-parity MDS code constructions based on Hadamard designs that achieve optimal repair for systematic node failures, but are suboptimal for parity node repairs. The MDS property for these designs is probabilistically guaranteed by choosing random constants multiplying the coding matrices.[1] In Section IX, we show that our m-parity codes are equivalent to codes that involve permutation matrices, in the manner of [23] and [25], under a similarity transformation of the coding matrices.

[1] All codes presented in this work are for the case where during a node repair all $d = n - 1$ surviving nodes participate in the process. Generalizations where only subsets of the surviving nodes participate in repair have been pursued for special cases of (n, k) in literature (See for example, [2–4, 22, 39, 40])

2 MDS Storage Codes with 2 Parity Nodes

In this section, we consider the code repair problem for MDS storage codes with 2 parity nodes. After we lay down the model for repair, we continue with introducing our code construction.

Let a file of size $M = kN$ denoted by the vector $\mathbf{f} \in \mathbb{F}_q^{kN}$ be partitioned in k parts $\mathbf{f} = \left[\mathbf{f}_1^T \dots \mathbf{f}_k^T\right]^T$, each of size N, where N denotes the *subpacketization* factor[2], $\frac{N}{2} \in \mathbb{N}^*$.[3] We encode $\mathbf{f}$ using an $(n = k+2, k)$ code and store it across k systematic and 2 parity storage units, each having storage capacity $\frac{M}{k} = N$. Hence, the effective coding rate is $R = \frac{k}{k+2}$. We require that our code is MDS, i.e., the encoded storage array is resilient to up to any 2 node erasures. A storage code has the MDS property when any collection of k storage nodes can reconstruct the file $\mathbf{f}$.

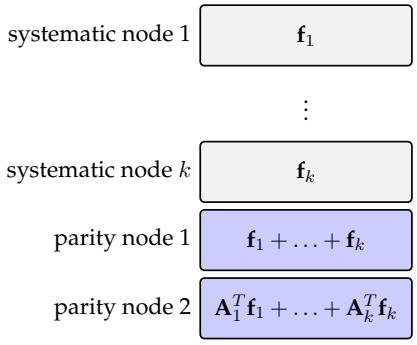

Figure 1: A $(k+2, k)$ encoded storage array.

In Fig. 1, we provide a generic representation of a 2-parity MDS encoded storage array. The first k storage nodes store the systematic file parts. Without loss of generality, the first parity stores the sum of all k systematic parts $\mathbf{f}_1 + \ldots + \mathbf{f}_k$ and the second parity stores a linear combination of them $\mathbf{A}_1^T\mathbf{f}_1 + \ldots + \mathbf{A}_k^T\mathbf{f}_k$.[4] Here, $\mathbf{A}_i$ denotes an $N \times N$ matrix of coding coefficients used by the second parity node to "scale and mix" the contents of the ith file piece $\mathbf{f}_i$, $i \in [k]$, where $[N] = \{1, \ldots, N\}$. This representation is a systematic one: k nodes store uncoded file pieces and each of the 2 parities stores a linear combination of the k file parts.

When a node of the $(k+2, k)$ encoded array fails, a *newcomer* node joins the system, downloads sufficient information from the remaining $d = n - 1 = k + 1$ nodes and regenerates what was lost. Hence, when a failure occurs, the *Code Repair* process is initiated to exactly regenerate the lost coded data in a *newcomer* storage component. See Fig. 2, for a sketch of the repair of a generic $(4, 2)$-MDS code.

We now consider that a systematic node $i \in [k]$ fails. Then, a newcomer storage node joins the storage network, it connects to the remaining nodes, and has to download sufficient data to reconstruct $\mathbf{f}_i$. Observe that the lost systematic part $\mathbf{f}_i$ exists *only* as a term of a linear combination at each parity node, as seen in Fig. 1. Since $\mathbf{f}_i$ has size N, to (linearly) regenerate it, the newcomer has to download from the parity nodes at least N linearly independent equations. Assuming that it downloads the same amount of data from both parities, the downloaded contents can be represented as a

[2] A larger file can be cut into pieces of size M where coding is performed independently on these pieces. If splitting the file in smaller pieces leaves the last piece having size less than M, then we can zero-pad it and encode it without storing the zero-padded blocks.

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

[4] The MDS property and the repair bandwidth of a code are invariant under a change of basis [22].

stack of N equations

$$\begin{bmatrix} \mathbf{p}_i^{(1)} \\ \mathbf{p}_i^{(2)} \end{bmatrix} \triangleq \begin{bmatrix} \left(\mathbf{V}_i^{(1)}\right)^T \mathbf{f}_1 + \ldots + \left(\mathbf{V}_i^{(1)}\right)^T \mathbf{f}_k \\ \left(\mathbf{V}_i^{(2)}\right)^T \mathbf{A}_1^T \mathbf{f}_1 + \ldots + \left(\mathbf{V}_i^{(2)}\right)^T \mathbf{A}_k^T \mathbf{f}_k \end{bmatrix} = \underbrace{\begin{bmatrix} \left(\mathbf{V}_i^{(1)}\right)^T \\ \left(\mathbf{A}_i \mathbf{V}_i^{(2)}\right)^T \end{bmatrix} \mathbf{f}_i}_{\text{useful data}} + \sum_{s=1, s\neq i}^{k} \underbrace{\begin{bmatrix} \left(\mathbf{V}_i^{(1)}\right)^T \\ \left(\mathbf{A}_s \mathbf{V}_i^{(2)}\right)^T \end{bmatrix} \mathbf{f}_s}_{\text{interference by } \mathbf{f}_s}, \tag{1}$$

where $\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)} \in \mathbb{F}_q^{\frac{N}{2}}$ are the equations downloaded from the first and second parity node, respectively, and $\mathbf{V}_i^{(1)}, \mathbf{V}_i^{(2)} \in \mathbb{F}_q^{N \times \frac{N}{2}}$ are *repair matrices*. Each repair matrix is used to mix the N parity equations to form $\frac{N}{2}$ "repair equations." Retrieving $\mathbf{f}_i$ from Eq. (1) is equivalent to solving an underdetermined set of N equations in the kN unknowns of $\mathbf{f}$, with respect to the N desired unknowns of $\mathbf{f}_i$. However, this is not possible due to $k-1$ additive *interference* components in the received equations generated by the undesired unknowns $\mathbf{f}_s$, $s \neq i$, as noted in Eq. (1). These $k-1$ interference terms are combined with the desired data and need to be canceled. Therefore, the newcomer needs to download additional data from the remaining $k-1$ systematic nodes. These new equations will be used to replicate and cancel the interference terms from the downloaded parity equations.

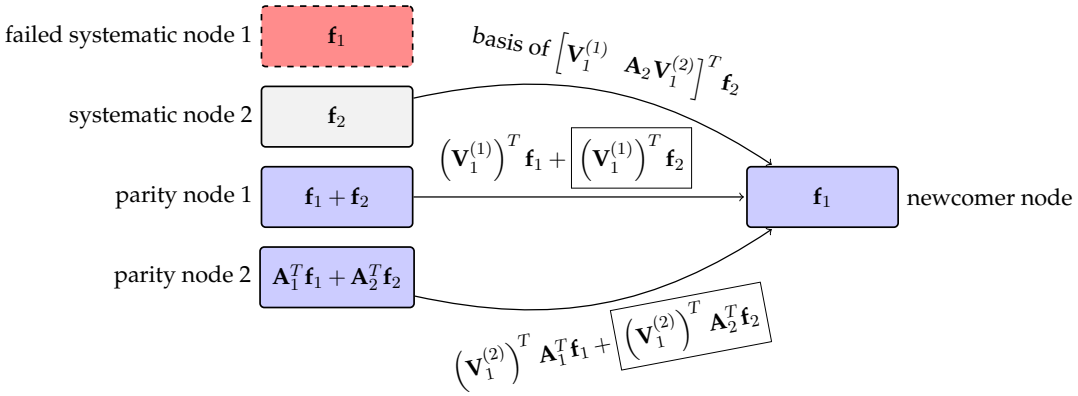

Figure 2: The code-repair problem for a $(4, 2)$ code. Here, $\mathbf{A}_1$ and $\mathbf{A}_2$ are $N \times N$ coding matrices and each $\mathbf{f}_1$ and $\mathbf{f}_2$ has size N. Let the first node fail. Then, a newcomer node joins the system and downloads data from the 3 remaining nodes to regenerate $\mathbf{f}_1$. The useful information is mixed with the undesired part $\mathbf{f}_2$ in both data blocks downloaded from the two parities. The interference parts appear inside a box. To retrieve $\mathbf{f}_1$, the interference terms need to be erased. For that a basis of these terms needs to be downloaded by systematic node 2. Then, the newcomer can erase the interference. Note that for succesful regeneration of $\mathbf{f}_1$ we also require that the matrix $[\mathbf{V}_1^{(1)} \quad \mathbf{A}_2\mathbf{V}_1^{(2)}]$ has full-rank N.

To cancel a single interference term of Eq. (1) that has size N, it suffices to download a basis of equations that can generate it. For example, to erase the interference component generated by the file part $\mathbf{f}_s$, the newcomer needs to connect to systematic node s and download a number of linear equations in $\mathbf{f}_s$ that can generate it. This number is equal to $\text{rank}\left(\begin{bmatrix} \left(\mathbf{V}_i^{(1)}\right)^T \\ \left(\mathbf{A}_s \mathbf{V}_i^{(2)}\right)^T \end{bmatrix}\right)$ which satisfies

$$\frac{N}{2} \leq \text{rank}\left(\begin{bmatrix} \left(\mathbf{V}_i^{(1)}\right)^T \\ \left(\mathbf{A}_s \mathbf{V}_i^{(2)}\right)^T \end{bmatrix}\right) \leq N. \tag{2}$$

Hence, in terms of downloaded equations, this is exactly the repair-bandwidth required to delete the interference term caused by $\mathbf{f}_s$. The lower bound in Eq. (2) comes from the fact that $\frac{N}{2}$ linearly independent equations need to be downloaded from each of the parities, thus rank $\left(\mathbf{V}_i^{(1)}\right) =$ rank $\left(\mathbf{V}_i^{(2)}\right) = \frac{N}{2}$. To erase all interference terms, the newcomer needs to download an aggregate of $\sum_{s=1,s\neq i}^{k}$ rank$([\mathbf{V}_i^{(1)} \ \mathbf{A}_s\mathbf{V}_i^{(2)}])$ equations from the remaining $k-1$ systematic nodes. We note that posterior to erasing interference terms, we require that the remaining N equations in the N unknowns of $\mathbf{f}_i$ are a full-rank system N, i.e., rank$([\mathbf{V}_i^{(1)} \ \mathbf{A}_i\mathbf{V}_i^{(2)}]) = N$. Again, please see Fig. 2 for a generic example of a $(4,2)$-MDS storage code repair instance. Hence, we can state the repair problem of a systematic node i as a rank constrained, rank minimization one, performed over $\mathbb{F}_q$.

$$\begin{aligned}\mathcal{R}_i\text{: } \min_{\mathbf{V}_i^{(1)},\mathbf{V}_i^{(2)}} & \sum_{s=1,s\neq i}^{k} \text{rank}\left(\left[\mathbf{V}_i^{(1)} \ \mathbf{A}_s\mathbf{V}_i^{(2)}\right]\right) \\ \text{s.t.: rank}\left(\left[\mathbf{V}_i^{(1)} \ \mathbf{A}_i\mathbf{V}_i^{(2)}\right]\right) &= N.\end{aligned} \tag{3}$$

Remark 1. *From [3] it is known that the theoretical minimum repair bandwidth, for any single node repair of an optimal (linear or nonlinear) $(k+2,k)$-MDS code is exactly $\frac{1}{2}$ times the number of remaining equations in the system, i.e., $(k+1)\frac{N}{2}$. This bound is proven using cut-set bounds on an infinite flow graph. Here, we provide an interpretation of this bound in terms of linear codes by calculating the minimum possible sum of ranks in $\mathcal{R}_i$. Since each repair matrix has to have full column rank of $\frac{N}{2}$ to be a feasible solution, the minimum rank each interference term can possibly have is $\frac{N}{2}$. This aggregates in a minimum repair bandwidth of $N+(k-1)\frac{N}{2} = (k+1)\frac{N}{2}$ repair equations. Interestingly, linear codes suffice to asymptotically achieve this bound [20], [21].*

Although the theoretical minimum repair bandwidth has been established in the literature and asymptotically optimal schemes that achieve it with a finite block length have been constructed, high-rate MDS codes that achieve it has been a challenging open problem. The difficulty in designing optimal MDS storage codes lies in a threefold requirement: *i)* the code has to satisfy the MDS property, *ii)* systematic nodes of the code have to be optimally repaired, and *iii)* parity nodes of the code have to be optimally repaired. Currently, there exist MDS codes for the low-rate regime, i.e., $\frac{k}{n} \leq \frac{1}{2}$, for which all nodes can be optimally repaired [17,18,22]. For the high data rate regime, Tamo *et al.* [23] and Cadambe *et al.* [25] presented the first MDS codes where any systematic node failure can be optimally repaired. Prior to this work, there did not exist MDS storage codes of arbitrarily high-rate that can optimal repair *any* node.

In the following, we present an explicit, high-rate $(k+2,k)$-MDS storage code. Our code achieves the minimum repair bandwidth bound for the repair of *any* single systematic or parity node failure. Before we proceed with the construction itself, we will state the intuition behind our code constructions and the tools that we use. Motivated by asymptotic alignment schemes, we use similar concepts induced by a combinatorial explanation of interference alignment in terms of dots on lattices. In contrast to the asymptotic interference alignment codes of [20] and [21], here, instead of letting randomness choose the coding matrices, we select particular constructions based on Hadamard matrices that achieve *exact* interference alignment in finite symbol extensions. In Section V, we prove the optimal repair of systematic nodes, in Section VI we show the optimal repair of parity nodes, and in Section VII we state explicit conditions for the MDS property.

3 Dots-on-a-lattice and Hadamard Designs

In this section, we simplify our ultimate goal of finding codes with minimal repair bandwidth defined in problem $\mathcal{R}_i$ of Section II. On simplifying the problem at hand, we will explain our Hadamard matrix based design which lies at the heart of the code constructions described in Eq. (20). Consider $\mathcal{R}_i$ and let us say that node $i = 1$ fails. Then, we would like to repair it by downloading the minimum amount of information, i.e., we aim to minimize the repair bandwidth which is equivalent to minimizing

$$\sum_{s=2}^{k} \text{rank}\left([\mathbf{A}_s\mathbf{V}_1 \;\; \mathbf{V}_1]\right). \tag{4}$$

Observe that here, and in the construction that follows, when we repair a systematic node, we use the same repair matrix $\mathbf{V}_i$ for all parities.

To minimize the repair bandwidth required to regenerate node $i = 1$, we would like to maximize the overlap (alignment) of $\mathbf{V}_1$ and $\mathbf{A}_s\mathbf{V}_1$ for $s \neq 1$. Ideally, we would like to have all the columns of $\mathbf{A}_s\mathbf{V}_1$ to lie in the column space of $\mathbf{V}_1$, so that the rank of $[\mathbf{A}_s\mathbf{V}_1 \;\; \mathbf{V}_1]$ is as small as the rank of $\mathbf{V}_1$. In other words, we would like $\mathbf{V}_1$ to be an *invariant subspace* of $\mathbf{A}_s,\ s \in [k]\backslash 1$

$$\text{colspan}(\mathbf{V}_1) = \text{colspan}(\mathbf{A}_2\mathbf{V}_1) = \ldots = \text{colspan}(\mathbf{A}_k\mathbf{V}_1) \tag{5}$$

so that $\mathbf{V}_1$ completely overlaps with all $\mathbf{A}_s\mathbf{V}_1$, for $s \neq 1$, as desired. This idea is central to our constructions. In this section, we pursue a simpler problem whose solution captures the aforementioned idea. We attempt to find *two* matrices which we denote by $\mathbf{T}_1$ and $\mathbf{T}_2$ and a matrix $\mathbf{V}$ which is invariant to both $\mathbf{T}_1, \mathbf{T}_2$.

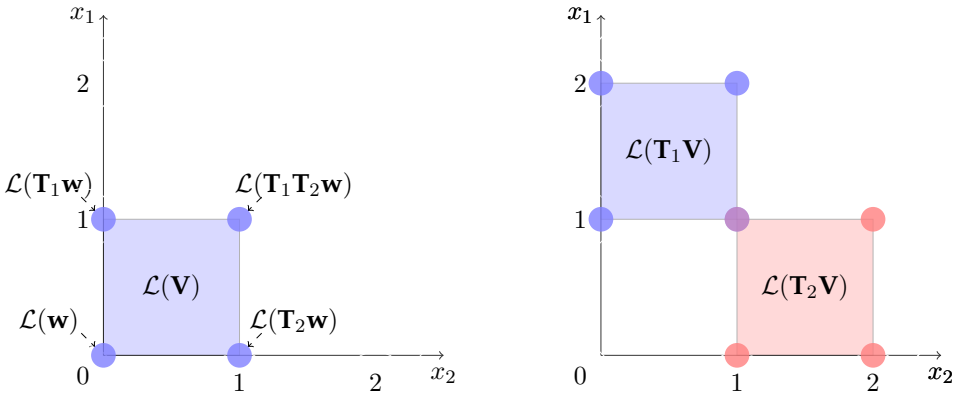

Figure 3: We use an $\mathcal{L}$ map from vectors generated as $\mathbf{T}_1^{x_1}\mathbf{T}_2^{x_2}\mathbf{w}$ to (x_1, x_2) lattice points on $\mathbb{Z}^2$. We first represent $\mathbf{V}$ as dots-on-a-lattice, where $\mathbf{V} = [\mathbf{w} \;\; \mathbf{T}_1\mathbf{w} \;\; \mathbf{T}_2\mathbf{w} \;\; \mathbf{T}_1\mathbf{T}_2\mathbf{w}]$ and $\mathcal{L}(\mathbf{V}) = \{(0,0),(1,0),(0,1),(1,1)\}$. We also depict the representation of the matrix $[\mathbf{T}_1\mathbf{V} \;\; \mathbf{T}_2\mathbf{V}]$ as dots-on-a-lattice, i.e., $\mathcal{L}([\mathbf{T}_1\mathbf{V} \;\; \mathbf{T}_2\mathbf{V}]) = \mathcal{L}(\mathbf{T}_1\mathbf{V}) \cup \mathcal{L}(\mathbf{T}_2\mathbf{V}) = \{(1,0),(2,0),(1,1),(2,1),(0,1),(1,1),(0,2),(1,2)\}$. Observe that if there existed a "wrap-around" cyclic property on the $\mathbf{T}_1, \mathbf{T}_2$ matrices, then the sets of dots $\mathcal{L}(\mathbf{T}_2\mathbf{V})$ and $\mathcal{L}(\mathbf{T}_2\mathbf{V})$ could potentially overlap.

We now consider two arbitrary $N \times N$ full-rank matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ that commute. We wish to construct a full-rank matrix $\mathbf{V}$, with at most $\frac{N}{2}$ columns, such that the span of $\mathbf{T}_1\mathbf{V}$ aligns as much as possible with the span of $\mathbf{T}_2\mathbf{V}$: we have to pick $\mathbf{V}$ such that it minimizes the dimensions of the union

of the two spans, that is the rank of $[\mathbf{T}_1\mathbf{V} \quad \mathbf{T}_2\mathbf{V}]$. How can we construct such a matrix? Assume that we start with one vector with nonzero entries, i.e., $\mathbf{V} = \mathbf{w}$, and for simplicity we let it be the all-ones vector. Then in the general case, $\mathbf{T}_1\mathbf{w}$ and $\mathbf{T}_2\mathbf{w}$ have zero intersection which is not desired. However, we can augment $\mathbf{V}$ such that it has as columns the elements of the set $\{\mathbf{w}, \mathbf{T}_1\mathbf{w}, \mathbf{T}_2\mathbf{w}, \mathbf{T}_1\mathbf{T}_2\mathbf{w}\}$. This idea of augmenting the set $\mathbf{V}$ in this manner to increase the overlap is presented in [12]. Observe that each vector $\mathbf{T}_1^{x_1}\mathbf{T}_2^{x_2}\mathbf{w}$ of $\mathbf{V}$ can be represented by the power tuple (x_1, x_2). This helps us visualize $\mathbf{V}$ as a set of dots on the 2-dimensional integer lattice as shown in Fig. 3.

For this new selection of $\mathbf{V}$, we have

$$\mathbf{T}_1\mathbf{V} = \begin{bmatrix}\mathbf{T}_1\mathbf{w} & \mathbf{T}_1^2\mathbf{w} & \mathbf{T}_1\mathbf{T}_2\mathbf{w} & \mathbf{T}_1^2\mathbf{T}_2\mathbf{w}\end{bmatrix} \tag{6}$$

$$\text{and } \mathbf{T}_2\mathbf{V} = \begin{bmatrix}\mathbf{T}_2\mathbf{w} & \mathbf{T}_2\mathbf{T}_1\mathbf{w} & \mathbf{T}_2^2\mathbf{w} & \mathbf{T}_1\mathbf{T}_2^2\mathbf{w}\end{bmatrix}. \tag{7}$$

The intersection of the spans of these two matrices is now nonzero: the matrix $[\mathbf{T}_1\mathbf{V} \quad \mathbf{T}_2\mathbf{V}]$ has rank 7 instead of the maximum possible of 8. This happens because the vector $\mathbf{T}_1\mathbf{T}_2\mathbf{w}$ is repeated in both matrices $\mathbf{T}_1\mathbf{V}$ and $\mathbf{T}_2\mathbf{V}$. In Fig. 3, we illustrate this concatenation, in terms of dots on $\mathbb{Z}^2$, where the intersection between the two spans is manifested as an overlap of dots. Observe how matrix multiplication of $\mathbf{T}_1$ and $\mathbf{T}_2$ with the vectors in $\mathbf{V}$ is pronounced through the dots representation: the dots representations of $\mathbf{T}_1\mathbf{V}$ and $\mathbf{T}_2\mathbf{V}$ matrices are shifted versions of $\mathcal{L}(\mathbf{V})$ along the x_1 and x_2 axes.

However, the $\mathbf{T}_i$ matrices (which in our case are coding matrices) are free to design under some specific constraints (which in our case is the MDS property of the code). Therefore, we can try to construct explicit matrices such that

$$\text{span}(\mathbf{T}_1\mathbf{V}) = \text{span}(\mathbf{T}_2\mathbf{V}), \tag{8}$$

as we required in Eq. (5). Interestingly, perfect and finite symbol extension interference alignment instances are possible when we enforce the dots representation of the $\mathbf{V}$ matrix to wrap-around itself and have a cyclic property. This wrap-around property is crucial in enabling perfect alignment of spaces. We will see that this property is obtained when the elements of the matrices are m^{th} roots of unity, i.e.,

$$\mathbf{T}_i^m = \mathbf{I}_N. \tag{9}$$

To see that, we formally state the dots-on-a-lattice representation. Let a map $\mathcal{L}$ from a matrix with r columns, each generated as $\mathbf{T}_1^{x_1}\mathbf{T}_2^{x_2}\mathbf{w}$, to a set of r points, such that the column $\mathbf{T}_1^{x_1}\mathbf{T}_2^{x_2}\mathbf{w}$ maps to the point (x_1, x_2). Then, we have for $\mathbf{V}$

$$\mathcal{L}(\mathbf{V}) \triangleq \{x_1\mathbf{e}_1 + x_2\mathbf{e}_2;\ x_1, x_2 \in \{0, \ldots, m-1\}\}, \tag{10}$$

where $\mathbf{e}_i$ is the i-th column of the identity matrix. Using this representation, the products $\mathbf{T}_1\mathbf{V}$ and $\mathbf{T}_2\mathbf{V}$ map to

$$\mathcal{L}(\mathbf{T}_1\mathbf{V}) = \left\{(x_1+1)\mathbf{e}_1 + x_2\mathbf{e}_2 : x_1, x_2 \in \{0, \ldots, m-1\}\right\}$$

$$\text{and } \mathcal{L}(\mathbf{T}_2\mathbf{V}) = \left\{x_1\mathbf{e}_1 + (x_2+1)\mathbf{e}_2 : x_1, x_2 \in \{0, \ldots, m-1\}\right\}.$$

For perfect alignment, we have to design the $\mathbf{T}_i$ matrices such that $\mathcal{L}(\mathbf{T}_1\mathbf{V}) = \mathcal{L}(\mathbf{T}_2\mathbf{V})$. A sufficient set of conditions for perfect span intersection is that the power tuples of $\mathbf{V}$, $\mathbf{T}_1\mathbf{V}$, and $\mathbf{T}_2\mathbf{V}$ perfectly intersect. Consider for example the following condition: $\mathcal{L}(\mathbf{T}_1\mathbf{V}) = \mathcal{L}(\mathbf{V})$, i.e.,

$$\left\{(x_1+1)\mathbf{e}_1 + x_2\mathbf{e}_2 : x_1, x_2 \in [m]\right\} = \left\{x_1\mathbf{e}_1 + x_2\mathbf{e}_2 : x_1, x_2 \in [m]\right\}.$$

The above condition means that when we multiply $\mathbf{T}_i^{m-1}$ with $\mathbf{T}_i$, the new product maps to $\mathbf{I}$, i.e., the $\mathbf{T}_i^{x_i}$ has a cyclic "power periodicity" of m. Hence, the aforementioned perfect alignment conditions are satisfied when the matrix powers "wrap around" upon reaching their modulus, m, i.e., when the additions $x_1 + 1$ and $x_2 + 1$ are performed modulo m. This wrap-around property is obtained when the $\mathbf{T}_1$ and $\mathbf{T}_2$ are diagonals of m^{th} roots of unity

$$\mathbf{T}_1^m = \mathbf{T}_1^0 = \mathbf{T}_2^m = \mathbf{T}_2^0 = \mathbf{I}_N. \tag{11}$$

Remark 2. *Observe that for diagonal matrices $\mathbf{T}_1, \mathbf{T}_2$, where the diagonal elements are m-th roots of unity, the operator $\mathcal{L}$ defines an isomorphism, between the elements of the group $\mathbf{T}_1^{x_1}\mathbf{T}_2^{x_2}\mathbf{w}$, under the left-hand-side matrix product operation with $\mathbf{T}_1, \mathbf{T}_2$ matrices, and the elements of $\mathbb{F}_m^2$ under addition.*

Arbitrary selection of $\mathbf{T}_1, \mathbf{T}_2$ with diagonals as roots of unity is not sufficient to ensure the full-rank property of $\mathbf{V}$. To hint on a general procedure which outputs "good" matrices, we see an example where we tune our parameters such that $\mathbf{V}$ has orthogonal columns. We would like to note that although we consider orthogonality at this time, linear independence of the columns of $\mathbf{V}$ always suffices. Let us briefly consider the case where $m = 2$ and $N = 2^3$, for which we choose

$$\mathbf{T}_1 = \text{diag}\left(\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}\right) \text{ and } \mathbf{T}_2 = \text{diag}\left(\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}\right). \tag{12}$$

For these matrices, $\mathbf{V}$ has $m^2 = 4$ orthogonal columns

$$\mathbf{V} = [\mathbf{w} \quad \mathbf{T}_1\mathbf{w} \quad \mathbf{T}_2\mathbf{w} \quad \mathbf{T}_1\mathbf{T}_2\mathbf{w}] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{13}$$

and $\mathbf{T}_1\mathbf{V} = [\mathbf{T}_1\mathbf{w} \quad \mathbf{w} \quad \mathbf{T}_1\mathbf{T}_2\mathbf{w} \quad \mathbf{T}_2\mathbf{w}]$, $\mathbf{T}_2\mathbf{V} = [\mathbf{T}_2\mathbf{w} \quad \mathbf{T}_1\mathbf{T}_2\mathbf{w} \quad \mathbf{w} \quad \mathbf{T}_1\mathbf{w}]$, have fully overlapping spans. We observe that for the additional matrix

$$\mathbf{T}_3 = \text{diag}\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}\right) \tag{14}$$

we have that $[\mathbf{V} \quad \mathbf{T}_3\mathbf{V}] = \mathbf{H}_8$, where $\mathbf{H}_8$ is the 8×8 Hadamard matrix. In the following, we generalize the above observations and show that Hadamard designs provide the conditions for perfect alignment and linear independence for our problem.

Let $m = 2$, $N = 2^L$, and $\mathbf{X}_i = \mathbf{I}_{2^{i-1}} \otimes \text{blkdiag}\left(\mathbf{I}_{\frac{N}{2^i}}, -\mathbf{I}_{\frac{N}{2^i}}\right)$, for $i \in [L]$, and consider the set[5]

$$\mathcal{H}_N = \left\{ \left(\prod_{i=1}^{L} \mathbf{X}_i^{x_i} \right) \mathbf{w} : x_i \in \{0, 1\} \right\}. \tag{15}$$

Then, we have the following key lemma.

[5]In our general code constructions, we set L to be k.

Lemma 1. *Let an $N \times N$ Hadamard matrix of the Sylvester's construction [38]*

$$\mathbf{H}_N \triangleq \begin{bmatrix} \mathbf{H}_{\frac{N}{2}} & \mathbf{H}_{\frac{N}{2}} \\ \mathbf{H}_{\frac{N}{2}} & -\mathbf{H}_{\frac{N}{2}} \end{bmatrix}, \tag{16}$$

with $\mathbf{H}_1 = 1$. Then, $\mathbf{H}_N$ is full-rank with mutually orthogonal columns, that are the N elements of $\mathcal{H}_N$.

Proof. The proof can be found in the Appendix. □

Example To illustrate the connection between the Hadamard matrix $\mathbf{H}_N$ and its dots-on-a-lattice representation $\mathcal{H}_N$ we "decompose" the Hadamard matrix of order 4

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = [\mathbf{w} \;\; \mathbf{X}_2\mathbf{w} \;\; \mathbf{X}_1\mathbf{w} \;\; \mathbf{X}_2\mathbf{X}_1\mathbf{w}], \tag{17}$$

where $\mathbf{X}_1 = \text{diag}\begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$, $\mathbf{X}_2 = \text{diag}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$, and $\mathbf{w} = \mathbf{1}_{4\times 1}$. Due to the commutativity of $\mathbf{X}_1$ and $\mathbf{X}_2$, the columns of $\mathbf{H}_4$ are also the elements of $\mathcal{H}_4 = \{\mathbf{w}, \mathbf{X}_1\mathbf{w}, \mathbf{X}_2\mathbf{w}, \mathbf{X}_1\mathbf{X}_2\mathbf{w}\}$.

Now, we will construct a matrix $\mathbf{V}_i$, whose columns are generated using the product-structure of $\mathcal{H}_N$, with the only difference that we omit a single $\mathbf{X}_i$ matrix from its construction. That is, we generate $\mathbf{V}_i$ with columns in the set

$$\mathcal{V}_i = \left\{ \left(\prod_{s=1, s\neq i}^{L} \mathbf{X}_s^{x_s} \right) \mathbf{w} : x_s \in \{0,1\} \right\}. \tag{18}$$

The space of $\mathbf{V}_i$ is invariant with respect to all $\mathbf{X}_s$, $s \neq i$, since the corresponding lattice representation wraps around itself due to $\mathbf{X}_s^2 = \mathbf{I}_N$, that is

$$\mathcal{L}(\mathbf{V}_i) = \mathcal{L}(\mathbf{X}_s\mathbf{V}_i), \; \forall s \neq i \Leftrightarrow \text{colspan}(\mathbf{V}_i) = \text{colspan}(\mathbf{X}_s\mathbf{V}_i), \; \forall s \neq i. \tag{19}$$

Additionally, we have

$$\mathcal{L}(\mathbf{X}_i\mathbf{V}_i) = \left\{ \mathbf{e}_i + \sum_{s=1, s\neq i}^{L} x_s\mathbf{e}_s : x_s \in \{0,1\} \right\},$$

and we observe that $\mathcal{L}(\mathbf{X}_i\mathbf{V}_i) \cap \mathcal{L}(\mathbf{V}_i) = \emptyset$, i.e., $\mathcal{L}(\mathbf{V}_i)$ does not include any points with nonzero x_i coordinates. Then, due to the orthogonality of elements within $\mathcal{H}_N$, we have

$$|\mathcal{L}(\mathbf{V}_i)| = |\mathcal{L}(\mathbf{X}_s\mathbf{V}_i)| = \text{rank}(\mathbf{V}_i) = \text{rank}(\mathbf{X}_i\mathbf{V}_i) = N/2,$$

for any $i \neq s$. Hence, we obtain the following lemma for the set $\mathcal{H}_N$ and its associated $\mathcal{L}$ map.

Lemma 2. *For any $i, j \in [L]$ we have that*

$$\textit{rank}([\mathbf{V}_i \;\; \mathbf{X}_j\mathbf{V}_i]) = |\mathcal{L}(\mathbf{V}_i) \cup \mathcal{L}(\mathbf{X}_j\mathbf{V}_i)| = \begin{cases} N, & i = j, \\ \frac{N}{2}, & i \neq j. \end{cases}$$

In Fig. 4, we give an illustrative example of the aforementioned definitions and properties. For $N = 2^3$, we consider $\mathbf{H}_8$ and $\mathbf{V}_3$ along with the matrix product $\mathbf{X}_2\mathbf{V}_3$ and their corresponding lattice representations.

To conclude this section, we have showed that starting from Hadamard matrices, we could obtain $\mathbf{X}_i$ and $\mathbf{V}_i$ matrices that have the perfect alignment properties of Eq. (5) required by our repair optimization problem. We will use these $\mathbf{X}_i$ matrices to build the coding matrices of our repair optimal code.

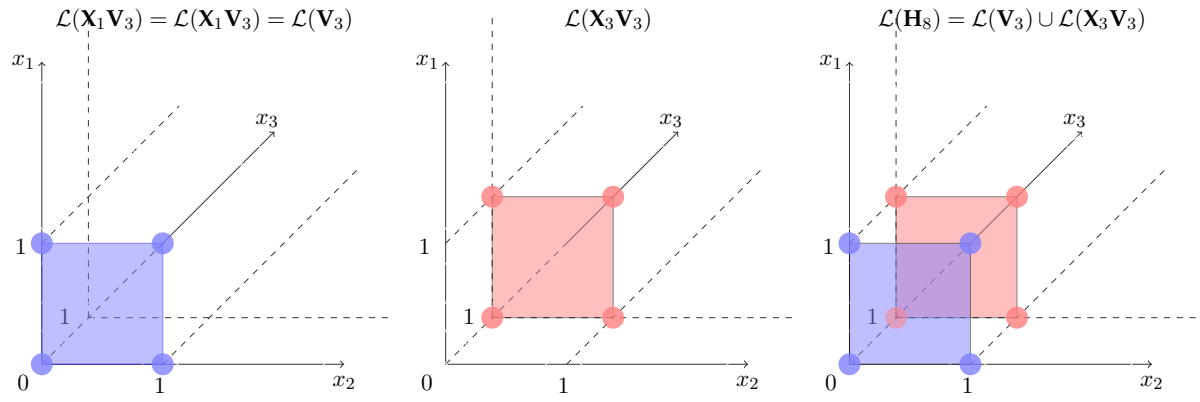

Figure 4: We set $N = 8$ and show the dots representation of $\mathbf{V}_3$, $\mathbf{X}_1\mathbf{V}_3$, $\mathbf{X}_2\mathbf{V}_3$, $\mathbf{X}_3\mathbf{V}_3$, and $\mathbf{H}_8$. Observe that we have perfect overlap for the cases of the $\mathbf{V}_3$, $\mathbf{X}_1\mathbf{V}_3$, and $\mathbf{X}_2\mathbf{V}_3$ matrices, which however do not intersect with $\mathbf{X}_3\mathbf{V}_3$. Hence, a union of the two distinct lattice point sets produces the lattice points of $\mathcal{L}(\mathbf{H}_8)$.

$$\mathbf{A}_1 = \text{diag}\left(9\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\\-1\\-1\\-1\\-1\\-1\\-1\\-1\\-1\end{bmatrix} + 7\begin{bmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\end{bmatrix} + \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\right), \quad \mathbf{A}_2 = \text{diag}\left(5\begin{bmatrix}1\\1\\1\\1\\-1\\-1\\-1\\-1\\1\\1\\1\\1\\-1\\-1\\-1\\-1\end{bmatrix} + 2\begin{bmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\end{bmatrix} + \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\right), \quad \mathbf{A}_3 = \text{diag}\left(9\begin{bmatrix}1\\1\\-1\\-1\\1\\1\\-1\\-1\\1\\1\\-1\\-1\\1\\1\\-1\\-1\end{bmatrix} + 4\begin{bmatrix}1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\\1\\-1\end{bmatrix} + \begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\right)$$

Figure 5: The coding matrices of a repair optimal $(5, 3)$-MDS code over $\mathbb{F}_{11}$

4 A Repair Optimal 2-Parity MDS Code

Code Construction 1. *Let the file size be $M = k2^{k+1}$ and let a $(k+2, k)$-MDS storage code with coding matrices*

$$\mathbf{A}_i = a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N, \ i \in [k], \tag{20}$$

where $N = 2^{k+1}$,

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

and a_i, b_i satisfy $a_i^2 - b_i^2 = -1$, for all $i \in [k]$.[6]

Theorem 1. *There exists a finite field $\mathbb{F}_q$ of prime and odd characteristic and order $q \geq 2k + 3$ and explicit, non-zero constants $a_i, b_i \in \mathbb{F}_q$, $\forall i \in [k]$, such that the $(k+2, k)$ storage code in Eq. (20) is a repair optimal MDS storage code.*

[6] We use -1 to denote the field element $q - 1$ in $\mathbb{F}_q$.

In Fig. 3, we give the coding matrices of a $(5,3)$-MDS code over $\mathbb{F}_{11}$ based on our construction. We would like to note that the field over which we construct our codes needs to have prime characteristic that is not equal to 2. This requirement is posed due to the fact that in a field whose characteristic is 2, the element -1 be equal to 1 and all our coding matrices will be equal to identity.

Remark 3. *The code constructions presented here have generator matrices that are as sparse as possible over $\mathbb{F}_q$, since any additional sparsity would violate the MDS property. This creates the additional benefit of minimum update complexity (at least over $\mathbb{F}_q$), when elements of the stored data object change.*

5 Optimal Systematic Node Repair

In this section, we show that our code in Eq. (20) has optimal systematic node repairs.

Let systematic node $i = 1$ of the code in Eq. (20) fail. We will construct $\mathbf{V}_1$ such that it is *a common invariant subspace* of $\mathbf{X}_2, \mathbf{X}_3, \ldots, \mathbf{X}_{k+1}$, i.e., if

$$\text{colspan}(\mathbf{V}_1) = \text{colspan}(\mathbf{X}_2\mathbf{V}_1) = \ldots = \text{colspan}(\mathbf{X}_{k+1}\mathbf{V}_1), \tag{22}$$

then, $\mathbf{V}_1$ would completely overlap with $\mathbf{A}_s\mathbf{V}_1$, for $s \neq 1$, as desired. To see this, note that $\mathbf{A}_s\mathbf{V}_1 = a_s\mathbf{X}_s\mathbf{V}_1 + b_s\mathbf{X}_{k+1}\mathbf{V}_1 + \mathbf{V}_1$. Therefore, every column vector of $\mathbf{A}_s\mathbf{V}_1, s \neq 1$ is the sum of three column vectors: one from the span of $\mathbf{X}_s\mathbf{V}_1$, one from the span of $\mathbf{X}_{k+1}\mathbf{V}_1$ and one from $\mathbf{V}_1$ itself. If $\mathbf{V}_1$ is invariant to both $\mathbf{X}_s$ and $\mathbf{X}_{k+1}$, then every column vector of $\mathbf{A}_s\mathbf{V}_1$ is a sum of three column vectors from the span of $\mathbf{V}_1$ and therefore lies in the span of $\mathbf{V}_1$. Hence, if we satisfy Eq. (22), $\mathbf{A}_s\mathbf{V}_1$ will lie in the span of $\mathbf{V}_1$.

Consider now the repair of systematic node $i \in [k]$ of the code in Eq. (20). The coding matrix $\mathbf{A}_i$ corresponding to the lost systematic piece $\mathbf{f}_i$, holds one matrix, that is, $\mathbf{X}_i$, which is unique among all other coding matrices, $\mathbf{A}_s$, $s \in [k]\backslash i$. We pick the columns of the repair matrix as a set of $\frac{N}{2}$ vectors whose lattice representation is invariant to all $\mathbf{X}_s$ matrices but to one key matrix: the unique $\mathbf{X}_i$ component of $\mathbf{A}_i$. We construct the $N \times \frac{N}{2}$ repair matrix $\mathbf{V}_i$ whose columns are the elements of the set

$$\mathcal{V}_i = \left\{ \left(\prod_{s=1, s\neq i}^{k+1} \mathbf{X}_s^{x_s} \right) \mathbf{w} : x_s \in \{0,1\} \right\}. \tag{23}$$

This repair matrix is used to multiply both the contents of parity node 1 and 2, that is, $\mathbf{V}_i^{(1)} = \mathbf{V}_i^{(2)} = \mathbf{V}_i$. During the repair, the useful (desired signal) space populated by $\mathbf{f}_i$ is

$$[\mathbf{V}_i \;\; \mathbf{A}_i\mathbf{V}_i] \tag{24}$$

and the interference space due to file part $\mathbf{f}_s$, $s \neq i$, is

$$[\mathbf{V}_i \;\; \mathbf{A}_s\mathbf{V}_i]\,. \tag{25}$$

Remember that an optimal solution to $\mathcal{R}_i$ requires the useful space to have rank N and each of the interference spaces rank $\frac{N}{2}$. Observe that the following holds for each of the interference spaces

$$\begin{aligned} \frac{N}{2} &\leq \text{rank}\left([\mathbf{V}_i \;\; (a_s\mathbf{X}_s + b_s\mathbf{X}_{k+1} + \mathbf{I}_N)\,\mathbf{V}_i]\right) \\ &\leq |\mathcal{L}(\mathbf{V}_i) \cup \mathcal{L}(\mathbf{X}_s\mathbf{V}_i) \cup \mathcal{L}(\mathbf{X}_{k+1}\mathbf{V}_i)| = |\mathcal{L}(\mathbf{V}_i)| = \frac{N}{2}, \end{aligned} \tag{26}$$

for $s \neq i$, since

$$\mathcal{L}(\mathbf{X}_s\mathbf{V}_i) = \mathcal{L}(\mathbf{V}_i), s \in [k+1]\backslash i, \tag{27}$$

due to Lemma 2. Then, for the useful data space we have

$$\begin{aligned} N &\geq \text{rank}\left([\mathbf{V}_i \quad \mathbf{A}_i\mathbf{V}_i]\right) = \text{rank}\left([\mathbf{V}_i \quad (a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N)\,\mathbf{V}_i]\right) \\ &\overset{(*)}{=} \text{rank}\left([\mathbf{V}_i \quad \mathbf{X}_i\mathbf{V}_i]\right) = |\mathcal{L}\left(\mathbf{V}_i\right) \cup \mathcal{L}\left(\mathbf{X}_i\mathbf{V}_i\right)| \\ &= |\mathcal{L}\left(\mathbf{H}_N\right)| = N, \end{aligned} \tag{28}$$

for any $a_i \neq 0$, where $(*)$ comes from the fact that $(a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N)\,\mathbf{V}_i$ is a linear combination of columns from $\mathbf{V}_i$, $\mathbf{X}_{k+1}\mathbf{V}_i$, and $\mathbf{X}_i\mathbf{V}_i$. However, the column spaces of $\mathbf{V}_i$ and $\mathbf{X}_{k+1}\mathbf{V}_i$ are identical, thus the column space of $(a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N)\,\mathbf{V}_i$ can be generated by linear combinations of $\mathbf{X}_i\mathbf{V}_i$ and $\mathbf{V}_i$. Moreover, $[\mathbf{V}_i \quad (a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N)\,\mathbf{V}_i]$ has $\mathbf{V}_i$ within its columns, thus its span is the same as the span of $[\mathbf{V}_i \quad \mathbf{X}_i\mathbf{V}_i]$.

Therefore, by using $\mathbf{V}_i$ as a repair matrix, we are able to generate the minimum amount of interference and at the same time satisfy the full-rank constraint of $\mathcal{R}_i$. Hence, the repair matrix in Eq. (23) is an optimal solution for $\mathcal{R}_i$ and systematic node i can be optimally repaired by downloading $(k+1)\frac{N}{2}$ data equations, for all $i \in [k]$.

In Fig. 6, we present a more pictorial repair example where we consider the $(5, 3)$ case of our code in Eq. (20). We sketch the structure (or the generator matrix) of our code, where the a_i and b_i scalars and the $\mathbf{I}_{16}$ matrices in the $\mathbf{A}_i$ blocks are not mentioned for simplicity. In each $\mathbf{A}_i$ block of the second parity corresponds a unique "key" $\mathbf{X}_i$ matrix. During the repair of node 3, we use the repair matrix $\mathbf{V}_3$ that is defined in Eq. (23). Observe that matrices $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_4$ are used to construct $\mathbf{V}_3$. Hence, $\mathbf{V}_3$ is an invariant subspace of $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$. That way, interference aligns on the subspace $\mathbf{V}_3$, and the useful space $[\mathbf{V}_3\ \mathbf{X}_3\mathbf{V}_3]$ spans all N dimensions, since subspaces $\mathbf{V}_3$ and $\mathbf{X}_3\mathbf{V}_3$ are linearly independent. The alignment and full-rank properties are also exhibited, by the dots-on-a-lattice representation of the matrices.

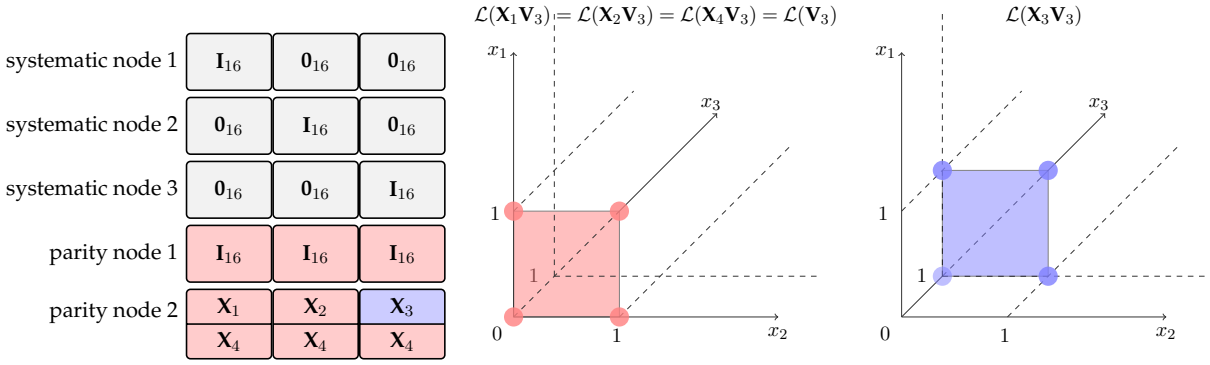

Figure 6: We illustrate some properties of the repair of node 3 in our $(6, 3)$ code. The red boxes in the coding matrix denote the matrices for which $\mathbf{V}_3$ is an invariant subspace. We also depict the dots-on-a-lattice representation of the repair spaces. The products of $\mathbf{V}_3$ with $\mathbf{X}^{x_1}$, $\mathbf{X}_2^{x_2}$ have the same lattice representation, whereas $\mathbf{X}_3^{x_3}\mathbf{V}_3$, $x_3 \neq 0$, correspond to a disjoint sets of lattice points

6 Optimal Parity Repair

Performing optimal repair of parity nodes is conceptually more challenging than performing optimal repair of systematic nodes. This is because, as we will see in the following, repair space properties that hold during systematic node repairs, have to hold even after a change of basis. For the low-rate regime, $\frac{k}{n} \leq \frac{1}{2}$ repair optimal codes for both systematic and parity nodes have been discovered in [21, 22]. However, the problem remained open for the high-rate regime. In particular, it is not known whether the systematic repair optimal codes of [23–25] admit optimal (or even non-trivially efficient) parity repair. We will explore this problem in this section. In particular, we will describe the additional properties to be satisfied by the coding matrices to ensure optimal parity repair (in addition to the properies that guarantee optimal systematic repair).

The code that we define in Eq. (20) admits optimal parity repair due to the fact that it satisfies all the space requirements that are analogous to the systematic repair setting. To see that we will rewrite our code in a new basis using a simple change of variables as the one in Fig. 7, where the parity nodes are transformed to "look like" systematic ones. This renaming of nodes provides the details under which parity repairs are understood in the systematic node repair framework.

systematic node 1	$\mathbf{f}_1$	$\mathbf{y}_1 - \mathbf{y}_2 - \mathbf{y}_3$	$\mathbf{y}'_1 - \mathbf{y}'_2 - \mathbf{y}'_3$
systematic node 1	$\mathbf{f}_2$	$\mathbf{y}_2$	$\mathbf{y}'_2$
systematic node k	$\mathbf{f}_3$	$\mathbf{y}_3$	$\mathbf{y}'_3$
parity node 1	$\mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_1$	$\mathbf{y}_1$	$\mathbf{A}''_1\mathbf{y}'_1 + \mathbf{A}''_2\mathbf{y}'_2 + \mathbf{A}''_3\mathbf{y}'_3$
parity node 2	$\mathbf{A}_1^T\mathbf{f}_1 + \mathbf{A}_2^T\mathbf{f}_2 + \mathbf{A}_3^T\mathbf{f}_3$	$\mathbf{A}'_1\mathbf{y}_1 + \mathbf{A}'_2\mathbf{y}_2 + \mathbf{A}'_3\mathbf{y}_3$	$\mathbf{y}'_1$

Figure 7: A change of variables for a $(5, 3)$-MDS code. We use it to represent a code in a way that we can treat the parity nodes as systematic nodes. These representations are equivalent with each other, in the sense that the code maintains its distance and repair properties.

As we will see in the following, the key ingredient of our construction that "unlocks" optimal repair for the first parity is the inclusion of the identity matrix in each $\mathbf{A}_i$. The same goes for the $\mathbf{X}_{k+1}$ matrix and the repair of the second parity. Both these additionally included matrices refine the parity repair process such that optimality is feasible. Selecting appropriate constants a_i and b_i is also essential to our developments.

6.1 Repairing the first parity

Let the first parity node fail. We make a change of variables to obtain a new representation for our code in Eq. (20), where the first parity is a systematic node in an equivalent representation. We start with our $(k + 2, k)$-MDS storage code of Eq. (20)

$$\begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N & \dots & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{I}_N & \dots & \mathbf{0}_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \dots & \mathbf{I}_N \\ \mathbf{I}_N & \mathbf{I}_N & \dots & \mathbf{I}_N \\ \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_k \end{bmatrix} \mathbf{f} \tag{29}$$

and make the following change of variables

$$\sum_{i=1}^{k} \mathbf{f}_i = \mathbf{y}_1, \tag{30}$$

$$\mathbf{f}_s = \mathbf{y}_s,\ s \in \{2, \ldots, k\}. \tag{31}$$

We solve Eq. (30) and Eq. (31) for $\mathbf{f}_1$ in terms of the $\mathbf{y}_i$ variables and obtain

$$\mathbf{f}_1 = \mathbf{y}_1 - \sum_{s=2}^{k} \mathbf{y}_s. \tag{32}$$

Then, we plug (31) and (32) in Eq. (29), to have the equivalent representation

$$\begin{bmatrix} \mathbf{I}_N & -\mathbf{I}_N & \ldots & -\mathbf{I}_N \\ \mathbf{0}_N & \mathbf{I}_N & \ldots & \mathbf{0}_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \ldots & \mathbf{I}_N \\ \mathbf{I}_N & \mathbf{0}_N & \ldots & \mathbf{0}_N \\ \mathbf{A}_1 & \mathbf{A}_2 - \mathbf{A}_1 & \ldots & \mathbf{A}_k - \mathbf{A}_1. \end{bmatrix} \mathbf{y}, \tag{33}$$

where $\mathbf{y} = \left[\mathbf{y}_1^T \ldots \mathbf{y}_k^T\right]^T \in \mathbb{F}_q^{kN}$. The first parity node of the code in Eq. (20) now corresponds to the node which contains $\mathbf{y}_1$ in the aforementioned representation. The coding matrices under this new representation are

$$\mathbf{A}_1 = a_1 \mathbf{X}_1 + b_1 \mathbf{X}_{k+1} + \mathbf{I}_N, \tag{34}$$

$$\mathbf{A}_s - \mathbf{A}_1 = a_s \mathbf{X}_s + (b_s - b_1)\mathbf{X}_{k+1} - a_1 \mathbf{X}_1, \tag{35}$$

for $s \in \{2, \ldots, k\}$. In contrast to the systematic node repair process, in the following we use a repair matrix of a slightly different structure. We construct the repair matrix $\mathbf{V}_a$ with columns in the set

$$\mathcal{V}_a = \left\{ \left(\prod_{s=2}^{k+1} (\mathbf{X}_1 \mathbf{X}_s)^{x_s} \right) \mathbf{w} : x_s \in \{0, 1\} \right\}. \tag{36}$$

Observe that this set is also a subset of $\mathcal{H}_N$. Then, to repair the node of (33) that contains $\mathbf{y}_1$ (i.e., the one that corresponds to the first parity node of (29)) we download $\mathbf{X}_1 \mathbf{V}_a$ times the contents of the first parity in Eq. (33) and $\mathbf{V}_a$ times the contents of the second parity. Hence, during this repair, the useful space is spanned by

$$[\mathbf{X}_1 \mathbf{V}_a \ \ \mathbf{A}_1 \mathbf{V}_a] \tag{37}$$

and the interference space due to the "transformed file part" $\mathbf{y}_s$, $s \in \{2, \ldots, k\}$, is

$$[\mathbf{X}_1 \mathbf{V}_a \ \ (\mathbf{A}_s - \mathbf{A}_1)\mathbf{V}_a]. \tag{38}$$

Before we proceed, observe that the following rank conditions hold

$$\mathcal{L}(\mathbf{X}_1 \mathbf{X}_s \mathbf{V}_a) = \mathcal{L}(\mathbf{V}_a) = \left\{ \left(\sum_{s=2}^{k+1} x_s \pmod 2 \right) \mathbf{e}_1 + \sum_{s=2}^{k+1} x_s \mathbf{e}_s;\ x_s \in \{0, 1\} \right\} \tag{39}$$

$$\Leftrightarrow \mathcal{L}(\mathbf{X}_1 \mathbf{V}_a) = \mathcal{L}(\mathbf{X}_s \mathbf{V}_a) = \left\{ \left(1 + \sum_{s=2}^{k+1} x_s \pmod 2 \right) \mathbf{e}_1 + \sum_{s=2}^{k+1} x_s \mathbf{e}_s;\ x_s \in \{0, 1\} \right\} \tag{40}$$

$$\Rightarrow \mathcal{L}(\mathbf{X}_{s_1} \mathbf{V}_a) = \mathcal{L}(\mathbf{X}_{s_2} \mathbf{V}_a), \tag{41}$$

for any $s, s_1, s_2 \in [k+1]$. The above equations imply that

$$\mathcal{L}\left(\mathbf{V}_a\right) \cup \mathcal{L}\left(\mathbf{X}_1\mathbf{V}_a\right) = \left\{ \sum_{s=1}^{k+1} x_s \mathbf{e}_s;\ x_s \in \{0,1\} \right\} = \mathcal{L}\left(\mathbf{H}_N\right). \tag{42}$$

Therefore, we have the following for each of the interference spaces

$$\begin{aligned} \frac{N}{2} &\leq \operatorname{rank}\left(\left[\mathbf{X}_1\mathbf{V}_a \quad \left(a_s\mathbf{X}_s + (b_s - b_1)\mathbf{X}_{k+1} - a_1\mathbf{X}_1\right)\mathbf{V}_a\right]\right) \\ &\leq \left|\mathcal{L}\left(\mathbf{X}_1\mathbf{V}_a\right) \cup \mathcal{L}\left(\mathbf{X}_s\mathbf{V}_a\right) \cup \mathcal{L}\left(\mathbf{X}_{k+1}\mathbf{V}_a\right)\right| \\ &= \left|\mathcal{L}\left(\mathbf{X}_1\mathbf{V}_a\right)\right| = \frac{N}{2}. \end{aligned} \tag{43}$$

Moreover, for the useful data space we have

$$\begin{aligned} \operatorname{rank}\left(\left[\mathbf{X}_1\mathbf{V}_a \quad \left(a_1\mathbf{X}_1 + b_1\mathbf{X}_{k+1} + \mathbf{I}_N\right)\mathbf{V}_a\right]\right) &= \operatorname{rank}([\mathbf{X}_1\mathbf{V}_a \ \ \mathbf{V}_a]) \\ &= \left|\mathcal{L}\left(\mathbf{V}_a\right) \cup \mathcal{L}\left(\mathbf{X}_1\mathbf{V}_a\right)\right| = \left|\mathcal{L}\left(\mathbf{H}_N\right)\right| = N, \end{aligned} \tag{44}$$

due to the same arguments as in Eq. (28). Thus, we can perform optimal repair of the node containing $\mathbf{y}_1$ in Eq. (33), which is equivalent to optimally repairing the first parity of our code in Eq. (20).

6.2 Repairing the second parity

Here, we have an additional step. We will first transform our coding matrices of (20) (via a symbol remapping) to an equivalent code, whose code matrix has a structure identical to the structure of the original code matrix. The difference is that in the new coding matrices, the last node looks like the first parity node of the original code structure, and the first parity node looks identical to the last node. This representation will allow us to prove the repair properties of the second parity as we did for the first in Subsection VI-A. Without loss of generality, we can multiply any non-zero coding column block that multiplies the ith file part with a full-rank matrix $\mathbf{B}_i$ and maintain the same code and repair properties, as shown in [22]. We can rewrite the non-zero coded blocks in the following manner

$$\begin{bmatrix} \mathbf{I}_N \\ \mathbf{A}_i \end{bmatrix} \mathbf{f}_i = \begin{bmatrix} \mathbf{I}_N \\ \mathbf{A}_i \end{bmatrix} \mathbf{B}_i\mathbf{B}_i^{-1}\mathbf{f}_i = \begin{bmatrix} \mathbf{B}_i \\ \mathbf{A}_i\mathbf{B}_i \end{bmatrix} \mathbf{f}_i' \tag{45}$$

where $\mathbf{B}_i$ is invertible and $\mathbf{f}_i' = \mathbf{B}_i^{-1}\mathbf{f}_i$. Then, If we can repair $\mathbf{f}_i'$, we can also repair $\mathbf{f}_i$ by multiplying $\mathbf{f}_i'$ with $\mathbf{B}_i$. This does not incur any additional repair bandwidth compared to the repair of $\mathbf{f}_i'$.

In the following derivations, we use the fact that $\mathbf{X}_s^2 = \mathbf{I}_N$, for any $s \in [k+1]$. We multiply the i-th block of (20) with $a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N$ to obtain

$$\begin{aligned} \begin{bmatrix} \mathbf{I}_N \\ a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N \end{bmatrix} &\equiv \begin{bmatrix} a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N \\ \left(a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N\right)\left(a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N\right) \end{bmatrix} = \begin{bmatrix} a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N \\ \left(a_i\mathbf{X}_i + \mathbf{I}_N\right)^2 - b_i^2\mathbf{I}_N \end{bmatrix} \\ &\equiv \begin{bmatrix} a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N \\ 2a_i\mathbf{X}_i + (a_i^2 - b_i^2 + 1)\mathbf{I}_N. \end{bmatrix} \overset{(*)}{\equiv} \begin{bmatrix} a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N \\ 2a_i\mathbf{X}_i \end{bmatrix}, \end{aligned} \tag{46}$$

where in $(*)$ we use the fact that $a_i^2 - b_i^2 + 1 = 0$. We continue by multiplying the i-th column block with $(a_i)^{-1}\mathbf{X}_i$ to obtain

$$\begin{bmatrix} a_i\mathbf{X}_i - b_i\mathbf{X}_{k+1} + \mathbf{I}_N \\ 2a_i\mathbf{X}_i \end{bmatrix} \equiv \begin{bmatrix} \mathbf{I}_N - a_i^{-1}b_i\mathbf{X}_{k+1}\mathbf{X}_i + a_i^{-1}\mathbf{X}_i \\ 2\mathbf{I}_N \end{bmatrix}. \tag{47}$$

Hence,

$$2\mathbf{A}_i^{-1} = \mathbf{I}_N - a_i^{-1} b_i \mathbf{X}_{k+1} \mathbf{X}_i + a_i^{-1} \mathbf{X}_i),\ i \in [k]. \tag{48}$$

Then, we rewrite our original code as

$$\begin{bmatrix} 2\mathbf{A}_1^{-1} & \mathbf{0}_N & \dots & \mathbf{0}_N \\ \mathbf{0}_N & 2\mathbf{A}_2^{-1} & \dots & \mathbf{0}_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \dots & 2\mathbf{A}_k^{-1} \\ 2\mathbf{A}_1^{-1} & 2\mathbf{A}_2^{-1} & \dots & 2\mathbf{A}_k^{-1} \\ 2\mathbf{I}_N & 2\mathbf{I}_N & \dots & 2\mathbf{I}_N \end{bmatrix} \mathbf{f}', \tag{49}$$

where $\mathbf{f}'$ is a full row-rank transformation of $\mathbf{f}$. We will focus on the following code matrix

$$\begin{bmatrix} \mathbf{I}_N & \mathbf{0}_N & \dots & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{I}_N & \dots & \mathbf{0}_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \dots & \mathbf{I}_N \\ 2\mathbf{A}_i^{-1} & 2\mathbf{A}_i^{-1} & \dots & 2\mathbf{A}_i^{-1} \\ \mathbf{I}_N & \mathbf{I}_N & \dots & \mathbf{I}_N \end{bmatrix} \mathbf{f}'. \tag{50}$$

To repair our second parity, we can use the same repair matrices used for the repair of the second parity of the above construction, with a slight manupulation. We simply need to multiply the repair matrices corresponding to systematic parts with $2^{-1}\mathbf{A}_i^{-1}$.

We proceed in the same manner that we handled the first parity repair. We make a change of variables such that the second parity becomes a systematic node in a new representation

$$\sum_{i=1}^{k} \mathbf{f}'_i = \mathbf{y}'_1 \tag{51}$$

and obtain the equivalent form

$$\begin{bmatrix} \mathbf{I}_N & -\mathbf{I}_N & \dots & -\mathbf{I}_N \\ \mathbf{0}_N & \mathbf{I}_N & \dots & \mathbf{0}_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_N & \mathbf{0}_N & \dots & \mathbf{I}_N \\ 2\mathbf{A}_1^{-1} & 2\mathbf{A}_2^{-1} - 2\mathbf{A}_1^{-1} & \dots & 2\mathbf{A}_k^{-1} - 2\mathbf{A}_1^{-1} \\ \mathbf{I}_N & \mathbf{0}_N & \dots & \mathbf{0}_N. \end{bmatrix} \mathbf{y}', \tag{52}$$

where

$$2\mathbf{A}_1^{-1} = \mathbf{I}_N - a_1^{-1} b_1 \mathbf{X}_{k+1} \mathbf{X}_1 + a_1^{-1} \mathbf{X}_1, \tag{53}$$

$$2\mathbf{A}_s^{-1} - 2\mathbf{A}_1^{-1} = a_s^{-1} \mathbf{X}_s - a_s^{-1} b_s \mathbf{X}_{k+1} \mathbf{X}_s + a_1^{-1} b_1 \mathbf{X}_{k+1} \mathbf{X}_1 - a_1^{-1} \mathbf{X}_1. \tag{54}$$

Then, the parity node which corresponds to systematic node 1 here, can be repaired by using $\mathbf{V}_b$ with columns in the set

$$\mathcal{V}_b = \left\{ \left(\mathbf{X}_{k+1}^{x_{k+1}} \prod_{s=1}^{k} (\mathbf{X}_1 \mathbf{X}_s)^{x_s} \right) \mathbf{w} : x_{k+1}, x_s \in \{0,1\} \right\}. \tag{55}$$

Again, the following equations hold

$$\mathcal{L}(\mathbf{X}_{k+1}\mathbf{V}_b) = \mathcal{L}(\mathbf{V}_b) = \left\{ \left(\sum_{s=2}^{k} x_s \pmod 2 \right) \mathbf{e}_1 + \sum_{s=2}^{k+1} x_s \mathbf{e}_s;\ x_s \in \{0,1\} \right\}, \tag{56}$$

$$\mathcal{L}(\mathbf{X}_{s_1}\mathbf{V}_b) = \mathcal{L}(\mathbf{X}_{s_2}\mathbf{V}_b) = \left\{ \left(1 + \sum_{s=2}^{k} x_s \pmod 2 \right) \mathbf{e}_1 + \sum_{s=2}^{k+1} x_s \mathbf{e}_s;\ x_s \in \{0,1\} \right\}, \tag{57}$$

$$\text{and } \mathcal{L}(\mathbf{X}_{s_1}\mathbf{X}_{k+1}\mathbf{V}_b) = \mathcal{L}(\mathbf{X}_{s_1}\mathbf{V}_b), \tag{58}$$

for all $s_1, s_2 \in [k]$. Hence, we have for the interfence space generated by component $\mathbf{y}'_s$, $s \in \{2, \ldots, k\}$

$$\begin{aligned}\frac{N}{2} &\leq \text{rank}\left(\left[\mathbf{X_1}\mathbf{V}_b \;\; (\hat{\mathbf{A}}_s - \hat{\mathbf{A}}_1)\mathbf{V}_b\right)\right) \\ &\leq |\mathcal{L}(\mathbf{X}_s\mathbf{V}_b) \cup \mathcal{L}(\mathbf{X}_1\mathbf{V}_b) \cup \mathcal{L}(\mathbf{X}_{k+1}\mathbf{X}_1\mathbf{V}_b) \cup \mathcal{L}(\mathbf{X}_{k+1}\mathbf{X}_s\mathbf{V}_b)\,| \\ &= |\mathcal{L}(\mathbf{X}_1\mathbf{V}_b) \cup \mathcal{L}(\mathbf{X}_s\mathbf{V}_b)| = \frac{N}{2}. \end{aligned} \tag{59}$$

Moreover, the useful space is full-rank

$$\text{rank}\left(\left[\mathbf{X_1}\mathbf{V}_b \;\; \left(\mathbf{I}_N - a_1^{-1}b_1\mathbf{X}_{k+1}\mathbf{X}_1 + a_1^{-1}\mathbf{X}_1\right)\mathbf{V}_b\right]\right) = \text{rank}\left([\mathbf{X_1}\mathbf{V}_b \;\; \mathbf{V}_b]\right) = N. \tag{60}$$

Thus, we can perform optimal repair of the second parity of the code in Eq. (20), with repair bandwidth $(k+1)\frac{N}{2}$.

7 The MDS Property

In this section we give explicit conditions on the a_i, b_i constants, for all $i \in [k]$, and the order of the finite field $\mathbb{F}_q$, for which the code in Eq. (20) is MDS. We discuss the MDS property using the notion of data collectors (DCs), in the same manner that it was used in [3]. A DC can be considered as an external user that can connect and has complete access to the contents of some subset of k nodes. A storage code where each node expends $\frac{M}{k}$ worth of storage, has the MDS property when all possible $\binom{n}{k}$ DCs can decode the file $\mathbf{f}$. We can show that testing the MDS property is equivalent to checking the rank of a specific matrix associated with each DC. This DC matrix is the vertical concatenation of the k stacks of equations stored by the nodes that the DC connects to. If all $\binom{n}{k}$ DC matrices are full-rank, then we declare that the storage code has the MDS property.

We start with a DC that connects to systematic nodes $\{1\ldots, k-1\}$ and the first parity node. The determinant of the corresponding DC matrix is

$$\det\left(\left[\begin{array}{ccc|c} \mathbf{I}_N & \ldots & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \vdots & & \vdots & \vdots \\ \mathbf{0}_{N\times N} & \ldots & \mathbf{I}_N & \mathbf{0}_{N\times N} \\ \hline \mathbf{I}_N & \ldots & \mathbf{I}_N & \mathbf{I}_N \end{array}\right]\right) = \det(\mathbf{I}_N) \neq 0, \tag{61}$$

since $\mathbf{I}_N$ is a full-rank diagonal matrix. We continue by considering a DC that connects to systematic nodes $\{1\ldots, k-1\}$ and the second parity node. For that we have

$$\det\left(\left[\begin{array}{ccc|c} \mathbf{I}_N & \ldots & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \vdots & & \vdots & \vdots \\ \mathbf{0}_{N\times N} & \ldots & \mathbf{I}_N & \mathbf{0}_{N\times N} \\ \hline \mathbf{A}_1 & \ldots & \mathbf{A}_{k-1} & \mathbf{A}_k \end{array}\right]\right) = \det(\mathbf{A}_k) \neq 0, \tag{62}$$

due to $\mathbf{A}_k$ being full-rank.

Finally, we consider DCs that connect to $k-2$ systematic nodes and both parity nodes. Let a DC that connects to systematic node

$$\{1\ldots, k-2\}$$

and the two parities. The corresponding DC matrix is

$$\left[\begin{array}{ccc|cc} \mathbf{I}_N & \ldots & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \vdots & & \vdots & \vdots & \\ \mathbf{0}_{N\times N} & \ldots & \mathbf{I}_N & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \hline \mathbf{I}_N & \ldots & \mathbf{I}_N & \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{A}_1 & \ldots & \mathbf{A}_{k-2} & \mathbf{A}_{k-1} & \mathbf{A}_k \end{array}\right]. \tag{63}$$

The leftmost $(k-2)N$ columns of the matrix in Eq. (63) are linearly independent, due to the upper-left identity block. Moreover, the leftmost $(k-2)N$ columns are linearly independent with the rightmost $2N$, using an analogous argument, if and only if that the right most $2N$ columns are linearly independent. Hence, we need to only check the rank of the sub-matrix

$$\begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{A}_{k-1} & \mathbf{A}_k \end{bmatrix}. \tag{64}$$

In the general case, a DC that connects to some $k-2$ subset of systematic nodes and the two parities has a corresponding matrix where the following block needs to be full-rank so that the MDS property can be satisfied

$$\begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{A}_i & \mathbf{A}_j \end{bmatrix}, \tag{65}$$

for $i, j \in [k]$ and $i \neq j$. The code is MDS when for all $i, j \in [k]$

$$\begin{aligned} &\text{rank}\left(\begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N & a_j\mathbf{X}_j + b_j\mathbf{X}_{k+1} + \mathbf{I}_N \end{bmatrix}\right) \\ &= \text{rank}\left(\begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N & a_j\mathbf{X}_j + b_j\mathbf{X}_{k+1} + \mathbf{I}_N \end{bmatrix} \times \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{0}_{N\times N} & -\mathbf{I}_N \end{bmatrix}\right) \\ &= \text{rank}\left(\begin{bmatrix} \mathbf{I}_N & 0 \\ a_i\mathbf{X}_i + b_i\mathbf{X}_{k+1} + \mathbf{I}_N & a_i\mathbf{X}_i - a_j\mathbf{X}_j + (b_i - b_j)\mathbf{X}_{k+1} \end{bmatrix}\right) \\ &= N + \text{rank}\left(a_i\mathbf{X}_i - a_j\mathbf{X}_j + (b_i - b_j)\mathbf{X}_{k+1}\right) = 2N, \end{aligned} \tag{66}$$

which is true if

$$\text{rank}\left(a_i\mathbf{X}_i - a_j\mathbf{X}_j + (b_i - b_j)\mathbf{X}_{k+1}\right) = N. \tag{67}$$

Since the diagonal elements of $\mathbf{X}_i$ are $\{\pm 1\}$, the previous requirement yields the following lemma.

Lemma 3. *The code in Eq. (20) is MDS when*

$$i)\ \ a_i - a_j + (b_i - b_j) \neq 0, \tag{68}$$
$$ii)\ \ a_i + a_j - (b_i - b_j) \neq 0, \tag{69}$$
$$iii)\ \ a_i - a_j - (b_i - b_j) \neq 0, \tag{70}$$
$$\textit{and } iv)\ \ a_i + a_j + (b_i - b_j) \neq 0, \tag{71}$$

for all $i \neq j \in [k]$.

Now, remember that our initial constraint on the a_i and b_i constants was

$$a_i^2 - b_i^2 = -1 \Leftrightarrow (a_i - b_i)(a_i + b_i) = -1. \tag{72}$$

One solution to the previous equation is the following

$$a_i - b_i = x_i, \tag{73}$$
$$a_i + b_i = -x_i^{-1}. \tag{74}$$

If we input the above solution to (72), then the MDS equations (71) become

$$\begin{aligned} a_i - a_j + (b_i - b_j) &= a_i + b_i - (a_i + b_j) \\ &= -x_i^{-1} + x_j^{-1} \neq 0 \\ &\Leftrightarrow x_i^{-1} \neq x_j^{-1}, \end{aligned} \tag{75}$$

$$\begin{aligned} a_i + a_j - (b_i - b_j) &= a_i - b_i + a_j + b_j \\ &= x_i - x_j^{-1} \neq 0 \\ &\Leftrightarrow x_i \neq x_j^{-1}, \end{aligned} \tag{76}$$

$$\begin{aligned} a_i - a_j - (b_i - b_j) &= a_i - b_i - (a_j - b_j) \\ &= x_i - x_j \neq 0 \\ &\Leftrightarrow x_i \neq x_j, \end{aligned} \tag{77}$$

$$\begin{aligned} a_i + a_j + (b_i - b_j) &= a_i + b_i + a_j - b_j \\ &= -x_i^{-1} + x_j \neq 0 \\ &\Leftrightarrow x_i^{-1} \neq x_j. \end{aligned} \tag{78}$$

The above conditions can be equivalently stated as

$$x_i \neq x_j \text{ and } x_i x_j \neq 1, \tag{79}$$

for any $i \neq j \in [k]$.

Then, consider a field $\mathbb{F}_q$ of size q. The set of x_is that satisfies our MDS requirements, is such in which no two elements are inverses of each other. Note that the non-zero elements of a finite field can be partitioned into two equal sized partitions, where the multiplicative inverse of an element from the first partition lies in the second partition (and vice-versa). If we additionally do not consider $x_i \in \{1, q-1\}$, then we are left with $\frac{q-3}{2}$ elements. Therefore, we can consider a field of prime characteristic p, such as its order q has the property

$$k \leq \frac{q-3}{2} \Leftrightarrow q \geq 2k+3 \tag{80}$$

and obtain $x_1, \ldots, x_k$ such that our requirements are satisfied. Then, the elements a_i and b_i, for all $i \in [k]$, can be obtained through the following equations

$$a_i = 2^{-1} x_i - 2^{-1} x_i^{-1}, \tag{81}$$

$$b_i = -2^{-1} x_i - 2^{-1} x_i^{-1}. \tag{82}$$

Observe that the above solutions yield $a_i \neq 0$ (that is needed for successful repair), for all $i \in [k]$, when $x_i \notin \{0, 1, q-1\}$. Therefore a prime characteristic field of order greater than, or equal to $2k+3$ always suffices to obtain the MDS property.

8 Generalizing to more than 2 parities

8.1 m-parity codes with optimal systematic repair

We generalize the Hadamard design construction of Section IV and of the code in [29], to construct $(k+m, k)$-MDS storage codes for file sizes $M = km^k$. Our constructions are based on a generalization

of the Sylvester construction for complex Hadamard matrices that use m^{th} roots of unity. We generate these matrices as

$$\mathbf{H}_{m^k} = \mathbf{H}_m \otimes \mathbf{H}_{m^{k-1}}, \tag{83}$$

where $\mathbf{H}_m$ is the m-point Discrete Fourier Transform matrix over a finite field. For example, for $m = 3$ and $\mathbb{F}_7$, we have

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \rho & \rho^2 \\ 1 & \rho^2 & \rho \end{bmatrix} \text{ and } \mathbf{H}_9 = \begin{bmatrix} \mathbf{H}_3 & \mathbf{H}_3 & \mathbf{H}_3 \\ \mathbf{H}_3 & \rho\mathbf{H}_3 & \rho^2\mathbf{H}_3 \\ \mathbf{H}_3 & \rho^2\mathbf{H}_3 & \rho\mathbf{H}_3 \end{bmatrix}, \tag{84}$$

where $\rho = 2$. Then, we consider the set

$$\mathcal{H}_{m^k} = \left\{ \left(\prod_{i=1}^{k} \mathbf{X}_i^{x_i} \right) \mathbf{w} : x_i \in \{0, 1, \ldots, m-1\} \right\}, \tag{85}$$

where $\mathbf{w} = \mathbf{1}_{m^k \times 1}$ and

$$\mathbf{X}_i = \mathbf{I}_{m^{i-1}} \otimes \text{blkdiag}\left(\mathbf{I}_{\frac{N}{m^i}}, \rho \mathbf{I}_{\frac{N}{m^i}}, \ldots, \rho^{m-1} \mathbf{I}_{\frac{N}{m^i}} \right). \tag{86}$$

Here, ρ denotes an m^{th} root of unity which yields

$$\mathbf{X}_i^m = \mathbf{I}_{m^k}. \tag{87}$$

As with the $m = 2$ case, there is a one-to-one correspondence between the elements of the set $\mathcal{H}_{m^k}$ and the columns of $\mathbf{H}_{m^k}$. The general m proof for that property follows the same manner of the $m = 2$ case, thus we omit it.

Remark 4. *Observe that for the full-rank property of the spaces in the $n - k = 2$ code construction we required $\mathbf{H}_N^T \mathbf{H}_N = \mathbf{I}_N$, for $N = 2^{k+1}$, i.e., that the useful spaces have orthogonal columns. However, linear independece suffices for our purposes. In our m-parity code construction, we only require that $\mathbf{H}_{m^k}$ is full-rank.*

We proceed by ebstablishing that there exist fields in which $\mathbf{H}_{m^k}$ is full-rank. First observe that

$$\mathbf{H}_{m^k} = \underbrace{\mathbf{H}_m \otimes \ldots \otimes \mathbf{H}_m}_{k \text{ times}}. \tag{88}$$

Hence,

$$|\mathbf{H}_{m^k}| = \prod_{i=1}^{k} |\mathbf{H}_m|^m. \tag{89}$$

Therefore, $\mathbf{H}_{m^k}$ is full-rank, if and only if $|\mathbf{H}_m| \neq 0$. Observe, that since $\mathbf{H}_m$ is the m-point DFT matrix, then it follows the Vandermonde structure, and its determinant is given by

$$|\mathbf{H}_m| = \prod_{1 \leq i < j \leq m} (\alpha_i - \alpha_j), \tag{90}$$

with $\alpha_i = \rho^{i-1}$ and $i = 1, \ldots, m$. Hence, the full-rank property is obtained when $\rho^i \neq \rho^j$ for all $i \neq j \in \{0, \ldots, q-1\}$. Therefore, to maintain the full-rank property of $\mathbf{H}_{m^k}$, the finite field over which we operate should be chosen such that all m^{th} roots of unity are distinct. The number of distinct m^{th} roots of unity over a finite field $\mathbb{F}_q$ is given by the number of (distinct) solutions of the equation $x^m = 1$. This is equal to the order of the cyclic group that generates m^{th} roots of unity within the multiplicative group of $\mathbb{F}_q$. This subgroup has order m, i.e., our $\mathbf{H}_m$ matrix is full-rank, when m divides $q - 1$ [35], i.e., when $q = C \cdot m + 1$, for some non-negative C.

Code Construction 2. *Our $(k+m,k)$-MDS code encodes a file* $\mathbf{f}$ *of size $M = km^k$ in the manner of*

$$\begin{bmatrix} \mathbf{I}_{km^k} \\ \mathbf{A}^{(k,m)} \end{bmatrix} \mathbf{f}, \tag{91}$$

where

$$\mathbf{A}^{(k,m)} = \begin{bmatrix} \mathbf{I}_{m^k} & \mathbf{I}_{m^k} & \dots & \mathbf{I}_{m^k} \\ \lambda_{1,1}\mathbf{X}_1 & \lambda_{1,2}\mathbf{X}_2 & \dots & \lambda_{1,k}\mathbf{X}_k \\ \lambda_{2,1}\mathbf{X}_1^2 & \lambda_{2,2}\mathbf{X}_2^2 & \dots & \lambda_{2,k}\mathbf{X}_k^2 \\ \vdots & & & \\ \lambda_{m-1,k}\mathbf{X}_1^{m-1} & \lambda_{m-1,2}\mathbf{X}_2^{m-1} & \dots & \lambda_{m-1,k}\mathbf{X}_k^{m-1} \end{bmatrix}, \tag{92}$$

with $\lambda_{i,j} \in \mathbb{F}_q$.

8.1.1 Optimal repair of the systematic nodes

For this code, let systematic node $i \in [k]$ fail. Then, to repair it we construct the repair matrix $\mathbf{V}_i$ that has as columns the elements of the set

$$\mathcal{V}_i = \left\{ \left(\prod_{s=1, s\neq i}^{k} \mathbf{X}_s^{x_s} \right) \mathbf{w} : x_s \in \{0, 1, \dots, m-1\} \right\}. \tag{93}$$

This matrix is used to multiply the contents of each of the parity nodes. Here, the useful space during the repair is given by

$$\begin{bmatrix} \mathbf{V}_i & \mathbf{X}_i\mathbf{V}_i & \mathbf{X}_i^2\mathbf{V}_i & \dots & \mathbf{X}_i^{m-1}\mathbf{V}_i \end{bmatrix} \tag{94}$$

and the interference space generated by systematic component $j \neq i$ is spanned by

$$\begin{bmatrix} \mathbf{V}_i & \mathbf{X}_j\mathbf{V}_i & \mathbf{X}_j^2\mathbf{V}_i & \dots & \mathbf{X}_j^{m-1}\mathbf{V}_i \end{bmatrix}. \tag{95}$$

Due to the modulus-m property of the powers of the $\mathbf{X}_i$ matrices, we obtain the following under the lattice representation

$$\mathcal{L}\left(\mathbf{V}_i\right) = \mathcal{L}\left(\mathbf{X}_j^l \mathbf{V}_i\right) \text{ and } \mathcal{L}\left(\mathbf{X}_i^{l_1}\mathbf{V}_i\right) \cap \mathcal{L}\left(\mathbf{X}_i^{l_2}\mathbf{V}_i\right) = \emptyset, \tag{96}$$

for any $j \in [k] \neq i$, and $l, l_1, l_2 \in \{0, \dots, m-1\}$, with $l_1 \neq l_2$. The above property and the fact that the elements of $\mathcal{H}_{m^k}$ are linearly independent lead us to the following lemma.

Lemma 4. *For any $i, j \in [k]$ we have that*

$$\begin{aligned} \textit{rank}(\begin{bmatrix} \mathbf{V}_i & \mathbf{X}_j\mathbf{V}_i & \mathbf{X}_j^2\mathbf{V}_i & \dots & \mathbf{X}_j^{m-1}\mathbf{V}_i \end{bmatrix}) &= \left| \mathcal{L}(\mathbf{V}_i) \cup \mathcal{L}\left(\mathbf{X}_j\mathbf{V}_i\right) \cup \mathcal{L}\left(\mathbf{X}_j^2\mathbf{V}_i\right) \cup \ldots \cup \mathcal{L}\left(\mathbf{X}_j^{m-1}\mathbf{V}_i\right) \right| \\ &= \begin{cases} m^k, & i = j, \\ m^{k-1}, & i \neq j. \end{cases} \end{aligned} \tag{97}$$

By Lemma 4 we see that each of the $k-1$ interference terms is confined within m^{k-1} dimensions and the full-rank property of the useful space is maintained. This is equivalent to stating that we can repair a single systematic node failure by downloading exactly $m^k + (k-1)m^{k-1} = (n-1)m^{k-1}$ equations, which matches exactly the optimal repair bandwidth of [3].

In Fig. 8, we give a sketch of the generator matrix of a $(6,3)$ systematic-repair optimal code. While optimally repairable $(6,3)$ codes have been discovered previously [4, 15, 18, 20–22], we use these parameters merely for exposition. Each parity block is associated with a specific key matrix $\mathbf{X}_i$. This, when considering the repair of node 3, allows a selection of $\mathbf{V}_3$ that is an invariant subspace to all matrices but to the key ones $\mathbf{X}_3$ and $\mathbf{X}_3^2$ which multiply the desired and lost file piece. This $\mathbf{V}_3$ enables perfect alignment of interference in m^2 dimensions, while ensuring a full-rank m^3 useful space.

systematic node 1	$\mathbf{I}_{16}$	$\mathbf{0}_{16}$	$\mathbf{0}_{16}$
systematic node 2	$\mathbf{0}_{16}$	$\mathbf{I}_{16}$	$\mathbf{0}_{16}$
systematic node 3	$\mathbf{0}_{16}$	$\mathbf{0}_{16}$	$\mathbf{I}_{16}$
parity node 1	$\mathbf{I}_{16}$	$\mathbf{I}_{16}$	$\mathbf{I}_{16}$
parity node 2	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$
parity node 2	$\mathbf{X}_1^2$	$\mathbf{X}_2^2$	$\mathbf{X}_3^2$

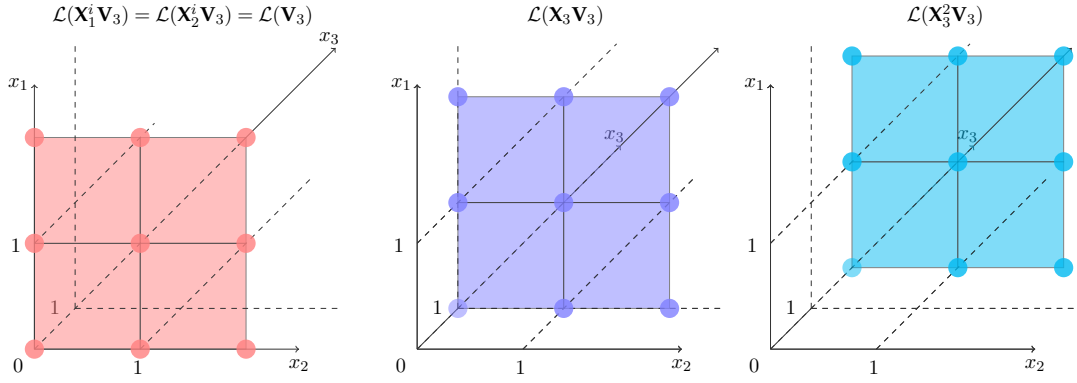

Figure 8: We illustrate some properties of the repair of node 3 in our $(6,3)$ code. The red boxes in the coding matrix denote the matrices for which $\mathbf{V}_3$ is an invariant subspace. We also we depict the dots-on-a-lattice representation of the repair spaces. The dots of any of the products of $\mathbf{V}_3$ with $\mathbf{X}^{x_1}, \mathbf{X}_2^{x_2}$ have the same lattice representation, whereas $\mathbf{X}_3^{x_3}\mathbf{V}_3$, $x_3 \neq 0$, correspond to a disjoint sets of lattice points.

8.1.2 The MDS property

We establish the MDS property of our m-parity codes in a probabilistic way: we show that when we select the $\lambda_{i,j}$ variables uniformly at random over a sufficiently large finite field of prime characteristic and order $q = mC + 1$, then the code is MDS with probability arbitrarily close to 1. This is shown using the Schwartz-Zippel lemma [36, 37] on a nonzero polynomial on $\lambda_{i,j}$s induced by the products

of all possible DC matrix determinants.

Let a DC of the code in Eq. (91) that connects to $k-P$ systematic nodes and P parities. For simplicity consider that this is the DC that is connected to the last $k-P$ systematic nodes and the first P parity nodes. The induced determinant of the corresponding DC matrix will be zero if the following determinant is zero

$$\det\left(\begin{bmatrix} \mathbf{I}_{m^k} & \mathbf{I}_{m^k} & \dots & \mathbf{I}_{m^k} \\ \lambda_{1,1}\mathbf{X}_1 & \lambda_{1,2}\mathbf{X}_2 & \dots & \lambda_{1,k}\mathbf{X}_k \\ \lambda_{2,1}\mathbf{X}_1^2 & \lambda_{2,2}\mathbf{X}_2^2 & \dots & \lambda_{2,k}\mathbf{X}_k^2 \\ \vdots & & & \\ \lambda_{P-1,1}\mathbf{X}_1^{P-1} & \lambda_{P-1,2}\mathbf{X}_2^{P-1} & \dots & \lambda_{P-1,k}\mathbf{X}_k^{P-1} \\ \hline \mathbf{0}_{(k-P)m^k\times Pm^k} & & \mathbf{I}_{(k-P)m^k} & \end{bmatrix}\right) \tag{98}$$

$$= |\mathbf{I}_{(k-P)m^k}| \det\left(\begin{bmatrix} \mathbf{I}_{m^k} & \mathbf{I}_{m^k} & \dots & \mathbf{I}_{m^k} \\ \lambda_{1,1}\mathbf{X}_1 & \lambda_{1,2}\mathbf{X}_2 & \dots & \lambda_{1,P}\mathbf{X}_P \\ \lambda_{2,1}\mathbf{X}_1^2 & \lambda_{2,2}\mathbf{X}_2^2 & \dots & \lambda_{2,P}\mathbf{X}_P^2 \\ \vdots & & & \\ \lambda_{P-1,1}\mathbf{X}_1^{P-1} & \lambda_{P-1,2}\mathbf{X}_2^{P-1} & \dots & {}_{P-1,P}\mathbf{X}_P^{P-1} \end{bmatrix}\right). \tag{99}$$

Since each of the $\mathbf{X}_i$ matrices is diagonal, each column of the matrix in the right hand side of (99) has exactly P nonzero elements. These Pm^k columns can be considered to belong into m^k groups of columns. Each column of a specific group has identical non-zero support with any other vector in that group. Then, any two columns within a block

$$\begin{bmatrix} \mathbf{I}_{m^k} \\ \lambda_{1,i}\mathbf{X}_i \\ \lambda_{2,i}\mathbf{X}_i^2 \\ \vdots \\ \lambda_{P-1,i}\mathbf{X}_i^{P-1} \end{bmatrix} \tag{100}$$

are orthogonal since their nonzero supports have zero overlap. Hence, a linear dependence will only exist among columns of a given non-zero support. We can then rewrite the matrix determinant of (99) as

$$\det\left(\mathbf{P}\begin{bmatrix} \mathbf{B}_1 & \mathbf{0}_{m^k\times m^k} & \dots & \mathbf{0}_{m^k\times m^k} \\ \mathbf{0}_{m^k\times m^k} & \mathbf{B}_2 & \dots & \mathbf{0}_{m^k\times m^k} \\ \vdots & & & \\ \mathbf{0}_{m^k\times m^k} & \dots & \mathbf{0}_{m^k\times m^k} & \mathbf{B}_P \end{bmatrix}\mathbf{P}\right) = |\mathbf{P}|^2\prod_{i=1}^{P}|\mathbf{B}_i|, \tag{101}$$

where $\mathbf{P}$ is the permutation matrix that groups the columns of the matrix according to their non-zero support, and $\mathbf{B}_i$ is a $P\times P$ full matrix of the form

$$\begin{bmatrix} \rho_{i_1,j_1}\lambda_{i_1,j_1} & \rho_{i_1,j_2}\lambda_{i_1,j_2} & \dots & \rho_{i_1,j_P}\lambda_{i_1,j_P} \\ \rho_{i_2,j_1}\lambda_{i_2,j_1} & \rho_{i_2,j_2}\lambda_{i_2,j_2} & \dots & \rho_{i_2,j_P}\lambda_{i_2,j_P} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{i_P,j_1}\lambda_{i_P,j_1} & \rho_{i_P,j_2}\lambda_{i_P,j_2} & \dots & \rho_{i_P,j_P}\lambda_{i_P,j_P} \end{bmatrix}, \tag{102}$$

where ρ_{i_1,j_1} is some m^{th} root of unity, the indeces depend on i, and no $\lambda_{i,j}$ appears more than once within each matrix. We intend to show that the determinant of the above matrix is not the zero-polynomial. To do so, setting $\lambda_{i_l,j_l}=1, \lambda_{i_l,j_{l'}}=0$ for $l\neq l'$ will make the above matrix a diagonal matrix, whose determinant is $\prod_{l=1}^{P}\rho_{i_l,j_l}$ which is clearly not zero. Therefore, the polynomial formed by the above determinant cannot be the zero polynomial. Accordingly, we can compute the determinant of each DC in this way. In the same manner, each of them will be a nonzero polynomial in $\lambda_{i,j}$. The product of all these determinants will as well be a nonzero polynomial in $\lambda_{i,j}$ of some degree D. By the Schwartz-Zippel lemma, we know that when we draw $\lambda_{i,j}$ uniformly at random over a field

of order q, this induced polynomial is zero with probability less than or equal to $\frac{D}{q}$. Hence, the MDS property is satisfied with probability arbitrarily close to 1, for sufficiently large finite fields, whose order is $q = m \cdot C + 1$ and C is a free nonzero variable[7] that can scale to infinity.

9 Connection to Permutation-Matrix Based Codes

Here we investigate an interesting connection between our systematic-repair optimal codes of Section VIII and the permutation-matrix based codes presented in [23] and [25]. A similar connection was exploited under a subspace interference alignment framework in [25]. Under a similarity transformation, our codes are equivalent to ones with coding matrices picked as specific permutation matrices. Multiplying the column space of an $\mathbf{X}_i$ matrix of our construction with the Hadamard matrix $\mathbf{H}_{m^k}$, yields a matrix that is a permutation of the columns of the Hadamard matrix

$$\mathbf{H}_{m^k}^{-1}\mathbf{X}_i\mathbf{H}_{m^k} = \mathbf{H}_{m^k}^{-1}\mathbf{H}_{m^k}\mathbf{P}_i = \mathbf{P}_i, \tag{103}$$

where $\mathbf{P}_i$ is some permutation matrix. This is due to the fact that the elements of $\mathcal{H}_{m^k}$ wrap around when multiplied with the $\mathbf{X}_i$ matrices. More precisely $\mathcal{L}(\mathbf{H}_{m^k}) = \mathcal{L}(\mathbf{X}_i\mathbf{H}_{m^k})$ for any i. Hence, we rewrite the $\mathbf{A}^{(m,k)}$ matrix of (91) as

$$\begin{aligned}
\left(\mathbf{I}_m \otimes \mathbf{H}_{m^k}^{-1}\right)\mathbf{A}^{(k,m)}\left(\mathbf{I}_k \otimes \mathbf{H}_{m^k}\right) &= \left(\mathbf{I}_m \otimes \mathbf{H}_{m^k}^{-1}\right)\begin{bmatrix} \mathbf{I}_{m^k}\mathbf{H}_{m^k} & \mathbf{I}_{m^k}\mathbf{H}_{m^k} & \dots & \mathbf{I}_{m^k}\mathbf{H}_{m^k} \\ \lambda_{1,1}\mathbf{X}_1\mathbf{H}_{m^k} & \lambda_{1,2}\mathbf{X}_2\mathbf{H}_{m^k} & \dots & \lambda_{1,k}\mathbf{X}_k\mathbf{H}_{m^k} \\ \lambda_{2,1}\mathbf{X}_1^2\mathbf{H}_{m^k} & \lambda_{2,2}\mathbf{X}_2^2\mathbf{H}_{m^k} & \dots & \lambda_{2,k}\mathbf{X}_k^2\mathbf{H}_{m^k} \\ \vdots & & & \\ \lambda_{m-1,k}\mathbf{X}_1^{m-1}\mathbf{H}_{m^k} & \lambda_{m-1,2}\mathbf{X}_2^{m-1}\mathbf{H}_{m^k} & \dots & \lambda_{m-1,k}\mathbf{X}_k^{m-1}\mathbf{H}_{m^k} \end{bmatrix} \\
&= \left(\mathbf{I}_m \otimes \mathbf{H}_{m^k}^{-1}\right)\begin{bmatrix} \mathbf{H}_{m^k} & \mathbf{H}_{m^k} & \dots & \mathbf{H}_{m^k} \\ \lambda_{1,1}\mathbf{H}_{m^k}\mathbf{P}_{1,1} & \lambda_{1,2}\mathbf{H}_{m^k}\mathbf{P}_{1,2} & \dots & \lambda_{1,k}\mathbf{H}_{m^k}\mathbf{P}_{1,k} \\ \lambda_{2,1}\mathbf{H}_{m^k}\mathbf{P}_{2,1} & \lambda_{2,2}\mathbf{H}_{m^k}\mathbf{P}_{2,2} & \dots & \lambda_{2,k}\mathbf{H}_{m^k}\mathbf{P}_{2,k} \\ \vdots & & & \\ \lambda_{m-1,k}\mathbf{H}_{m^k}\mathbf{P}_{m-1,1} & \lambda_{m-1,2}\mathbf{H}_{m^k}\mathbf{P}_{m-1,2} & \dots & \lambda_{m-1,k}\mathbf{H}_{m^k}\mathbf{P}_{m-1,k} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I}_{m^k} & \mathbf{I}_{m^k} & \dots & \mathbf{I}_{m^k} \\ \lambda_{1,1}\mathbf{P}_{1,1} & \lambda_{1,2}\mathbf{P}_{1,2} & \dots & \lambda_{1,k}\mathbf{P}_{1,k} \\ \lambda_{2,1}\mathbf{P}_{2,1} & \lambda_{2,2}\mathbf{P}_{2,2} & \dots & \lambda_{2,k}\mathbf{P}_{2,k} \\ \vdots & & & \\ \lambda_{m-1,k}\mathbf{P}_{m-1,1} & \lambda_{m-1,2}\mathbf{P}_{m-1,2} & \dots & \lambda_{m-1,k}\mathbf{P}_{m-1,k} \end{bmatrix},
\end{aligned} \tag{104}$$

where $\mathbf{P}_{i,j}$ is a permutation matrix. The systematic nodes of this equivalent $(k+m,k)$-MDS code can be optimally repaired using the repair matrices $\mathbf{V}_i\mathbf{H}_i^{-1}$, where $\mathbf{V}_i$ has the columns of the set $\mathcal{V}_i = \left\{\prod_{s=1,s\neq i}^{k}\mathbf{X}_s^{x_s}\mathbf{w} : x_s \in \{0,1,\dots,m-1\}\right\}$. This is true since the rank properties of the corresponding useful and interference spaces are invarianteto full-rank column transformations. Interestingly, this connection is two-way.

We find the connection manifested by the above equivalence examples to be of specific interest. Further investigation on it may lead to more understanding of the repair optimal high-rate designs. For the $\frac{k}{n} \leq \frac{1}{2}$ rate regime, there are several results and explicit codes for all parameters (n,k,d) where interference alignment has been fundamental in these solutions [17,18,22]. Formulation of an interference alignment framework for optimally repairing all nodes of a high-rate MDS code remains an interesting open problem.

[7]Note that the existence of infinite primes q of the form $m \cdot C + 1$ is guaranteed by Dirichlet's theorem on arithmetic progressions [42].

10 Conclusions

We presented the first explicit, high-rate, $(k+2,k)$ erasure MDS storage code that achieves optimal repair bandwidth for any single node failure, including the parities. Our construction is based on perfect interference alignment properties offered by Hadamard designs. Moreover, we generalize our constructions to erasure codes with m-parities that achieve optimally repair of the systematic parts. Interestingly, the size of the code for the high-rate regime is exponential in the parameter k. Fundamental limits on the size of the code for a given repair bandwidth is an interesting and ongoing area of future work [41].

11 Acknowledgement

The authors would like to thank Changho Suh for insightful discussions.

Appendix

Proof of Lemma 1: It is easy to show that the matrices are orthogonal [38]. Observe that $\mathbf{H}_N = \mathbf{H}_N^T$ and

$$\begin{aligned}\mathbf{H}_N\mathbf{H}_N^T = \mathbf{H}_N\mathbf{H}_N = \begin{bmatrix} 2\mathbf{H}_{\frac{N}{2}}\mathbf{H}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}\times\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}\times\frac{N}{2}} & 2\mathbf{H}_{\frac{N}{2}}\mathbf{H}_{\frac{N}{2}} \end{bmatrix} &= 2\left(\mathbf{I}_2 \otimes \mathbf{H}_{\frac{N}{2}}\mathbf{H}_{\frac{N}{2}}\right) \\ &= 2\left(\mathbf{I}_2 \otimes 2\left(\mathbf{I}_2 \otimes \mathbf{H}_{\frac{N}{4}}\mathbf{H}_{\frac{N}{4}}\right)\right) \\ &= 4\left(\mathbf{I}_4 \otimes \mathbf{H}_{\frac{N}{4}}\mathbf{H}_{\frac{N}{4}}\right) \\ &\vdots \\ &= N\cdot(\mathbf{I}_N \otimes \mathbf{H}_1\mathbf{H}_1) = N\cdot\mathbf{I}_N. \end{aligned} \tag{105}$$

We also have that $N \neq 0 \pmod q$, for $q > 2$, thus the rank of $\mathbf{H}_N$ is N and its columns are mutually orthogonal. Then, let an $N \times N$ diagonal matrix

$$\mathbf{X}_i = \mathbf{I}_{2^{i-1}} \otimes \text{blkdiag}\left(\mathbf{I}_{\frac{N}{2^i}}, -\mathbf{I}_{\frac{N}{2^i}}\right) \tag{106}$$

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

$$\mathbf{H}_N = \begin{bmatrix} \mathbf{H}_{\frac{N}{2}} & \mathbf{H}_{\frac{N}{2}} \\ \mathbf{H}_{\frac{N}{2}} & -\mathbf{H}_{\frac{N}{2}} \end{bmatrix} = \begin{bmatrix} \underbrace{\mathbf{1}_{2\times1} \otimes \mathbf{H}_{\frac{N}{2}}}_{\mathbf{F}_1} & \mathbf{X}_1\left(\mathbf{1}_{2\times1} \otimes \mathbf{H}_{\frac{N}{2}}\right) \end{bmatrix}. \tag{107}$$

We proceed in the same manner by expanding all "smaller" $\mathbf{H}_{\frac{N}{2^i}}$s

$$\begin{aligned}
\mathbf{F}_1 =& \mathbf{1}_{2\times 1} \otimes \left[\mathbf{1}_{2\times 1} \otimes \mathbf{H}_{\frac{N}{2^2}} \quad \mathbf{X}_2 \left(\mathbf{1}_{2\times 1} \otimes \mathbf{H}_{\frac{N}{2^2}}\right)\right] = \left[\underbrace{\mathbf{1}_{2^2\times 1} \otimes \mathbf{H}_{\frac{N}{2^2}}}_{\mathbf{F}_2} \quad \mathbf{X}_3 \left(\mathbf{1}_{2^2\times 1} \otimes \mathbf{H}_{\frac{N}{2^2}}\right)\right] \\
\mathbf{F}_2 =& \left[\underbrace{\mathbf{1}_{2^3\times 1} \otimes \mathbf{H}_{\frac{N}{2^3}}}_{\mathbf{F}_3} \quad \mathbf{X}_3 \left(\mathbf{1}_{2^3\times 1} \otimes \mathbf{H}_{\frac{N}{2^3}}\right)\right] \\
&\vdots \\
\mathbf{F}_{\log_2(N)-1} =& \left[\mathbf{1}_{N\times 1} \quad \mathbf{X}_{\log_2(N)}\mathbf{1}_{N\times 1}\right], \qquad (108)
\end{aligned}$$

where $\mathbf{F}_i$ is an $N \times \frac{N}{2^i}$ matrix and $\mathbf{F}_{\log_2(N)} = \mathbf{w}$ is the all ones vector of length N. Thus,

$$\begin{aligned}
\text{span}\left(\mathbf{H}_N\right) &= \text{span}\left(\left[\mathbf{F}_1 \quad \mathbf{X}_1\mathbf{F}_1\right]\right) = \text{span}\left(\left[\mathbf{F}_2 \quad \mathbf{X}_2\mathbf{F}_2 \quad \mathbf{X}_1\mathbf{F}_2 \quad \mathbf{X}_1\mathbf{X}_2\mathbf{F}_2\right]\right) \\
&\vdots \\
&= \text{span}\left(\left\{\left(\prod_{i=1}^{\log_2(N)} \mathbf{X}_i^{x_i}\right)\mathbf{w} : x_i \in \{0,1\}\right\}\right), \qquad (109)
\end{aligned}$$

which proves the final part of Lemma 1. □

References

[1] D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, "Repair optimal erasure codes through Hadamard designs," In *Allerton Conf. on Control, Comp., and Comm.*, Urbana-Champaign, IL, September 2011.

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

[3] A. G. Dimakis, P. G. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *IEEE Trans. on Inform. Theory*, vol. 56, pp. 4539 – 4551, Sep. 2010.

[4] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," in *IEEE Proceedings*, vol. 99, pp. 476 – 489, Mar. 2011.

[5] S. Ghemawat, H. Gobioff, and S.-T. Leung "The Google file system," in proc. *ACM Symp. on Op. Sys. Principles (SOSP)*, Oct., 2003.

[6] O. Khan, R. Burns, J. Plank, and C. Huang, "In search of I/O-optimal recovery from disk failures," to appear in *Hot Storage 2011, 3rd Workshop on Hot Topics in Storage and File Systems*, Portland, OR, Jun., 2011.

[7] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: a quantitiative comparison," *in Proc. IPTPS*, 2002.

[8] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in raid architectures," in *IEEE Trans. on Computers*, vol. 44, pp. 192 – 202 Feb. 1995.

[9] Z. Wang, A. G. Dimakis, and J. Bruck, "Rebuilding for array codes in distributed storage systems," in *proc. Workshop on the Application of Communication Theory to Emerging Memory Technologies (ACTEMT)*, 2010.

[10] L. Xiang, Y. Xu, J.C.S. Lui, and Q. Chang, "Optimal recovery of single disk failure in RDP code storage systems" in *proc. ACM SIGMETRICS (2010) international conference on Measurement and modeling of computer systems*

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

[12] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the K user interference channel," *IEEE Trans. on Inform. Theory*, vol. 54, pp. 3425–3441, Aug. 2008.

[13] 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, Jul. 2009.

[14] D. Cullina, A. G. Dimakis, and T. Ho, "Searching for minimum storage regenerating codes," In *Allerton Conf. on Control, Comp., and Comm.*, Urbana-Champaign, IL, September 2009.

[15] K.V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran "Explicit construction of optimal exact regenerating codes for distributed storage," In *Allerton Conf. on Control, Comp., and Comm.*, Urbana-Champaign, IL, September 2009.

[16] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. 2010 IEEE Information Theory Workshop (ITW)*, Jan. 2010.

[17] C. Suh and K. Ramchandran, "Exact-Repair MDS Code Construction Using Interference Alignment," in *IEEE Trans. on Inform. Theory*, vol. 57, pp. 1425 - 1442, Mar. 2011.

[18] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: necessity and code constructions," in *IEEE Trans. on Inform. Theory*, Dec. 2011 .

[19] Y. Wu. "A construction of systematic MDS codes with minimum repair bandwidth," in *IEEE Trans. on Inform. Theory*, vol. 57, pp. 3738 – 3741 May 2011.

[20] V. Cadambe, S. Jafar, and H. Maleki, "Distributed data storage with minimum storage regenerating codes - exact and functional repair are asymptotically equally efficient," in *2010 IEEE Intern. Workshop on Wireless Network Coding (WiNC)*, Apr 2010.

[21] 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

[22] 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," in *IEEE Trans. on Inform. Theory*, vol. 57, pp. 5227 – 5239, Jul. 2011.

[23] I. Tamo, Z. Wang, and J. Bruck "MDS Array Codes with Optimal Rebuilding," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT) 2001*, St. Petersburg, Russia, Aug. 2011.

[24] V. R. Cadambe, C. Huang and J. Li, "Permutation codes: optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT) 2001*, St. Petersburg, Russia, Aug. 2011.

[25] V. R. Cadambe, C. Huang, S. A. Jafar and J. Li, "Optimal Repair of MDS Codes in Distributed Storage via Subspace Interference Alignment" *arxiv pre-print 2011*, preprint available at http://arxiv.org/abs/1106.1250

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

[27] K. W. Shum and Y. Hu, "Exact minimum-repair-bandwidth cooperative regenerating codes for distributed storage systems," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT) 2001*, St. Petersburg, Russia, Aug. 2011.

[28] D. S. Papailiopoulos and A. G. Dimakis, "Distributed storage Codes Meet Multiple-Access Wiretap Channels," *in Proc. 2010 48th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, Illinois, Sep. 2010.

[29] D. S. Papailiopoulos and A. G. Dimakis, "Distributed storage Codes through Hadamard designs," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT) 2011*, St. Petersburg, Russia, Aug. 2011.

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

[31] D. S. Papailiopoulos, J. Luo, A. G. Dimakis, C. Huang, and J. Li, "Simple regenerating codes: network coding for cloud storage", in *IEEE International Conference on Computer Communications (Infocom) 2012, Miniconference*.

[32] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword elements," Preprint available at http://arxiv.org/abs/1106.3625.

[33] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," *USENIX ATC*, Boston, MA, Jun. 2012.

[34] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT) 2012*, Boston, MA, Jul. 2012.

[35] R. Lidl and H. Niederreiter, *Finite Fields (Encyclopedia of Mathematics and its Applications)*, Cambridge University Press, 2008.

[36] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "A Random Linear Network Coding Approach to Multicast," *IEEE Trans. on Inform. Theory*, vol. 52 (10), pp. 4413–4430, Oct. 2006.

[37] R. Motwani and P. Raghavan, "Randomized algorithms", *ACM Comput. Surv.*, vol. 28, pp. 33–37, Mar. 1996.

[38] J. J. Sylvester. "Thoughts on inverse orthogonal matrices, simultaneous sign successions, and tessellated pavements in two or more colours, with applications to Newton's rule, ornamental tile-work, and the theory of numbers." *Philosophical Magazine*, 34, pp. 461–475, 1867.

[39] El Rouayheb, S. and Ramchandran, K.; , "Fractional repetition codes for repair in distributed storage systems," Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on , vol., no., pp.1510-1517, Sept. 29 2010-Oct. 1 2010

[40] Shah, N.B. and Rashmi, K.V. and Vijay Kumar, P. and Ramchandran, K.; , "Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff," Information Theory, IEEE Transactions on , vol.58, no.3, pp.1837-1852, March 2012

[41] Cadambe, V.R. and Cheng Huang and Jin Li and Mehrotra, S.; , "Polynomial length MDS codes with optimal repair in distributed storage," Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on , vol., no., pp.1850-1854, 6-9 Nov. 2011

[42] P. G. L. Dirichlet, "Beweis des Satzes, dass jede unbegrenzte arithmetische Progression, deren erstes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält," Abhand. Ak. Wiss. Berlin 48, 1837.