Zigzag Codes: MDS Array Codes with Optimal Rebuilding

Itzhak Tamo, Student Member, IEEE, Zhiying Wang, Student Member, IEEE, and Jehoshua Bruck, Fellow, IEEE

Abstract—MDS array codes are widely used in storage systems to protect data against erasures. We address the rebuilding ratio problem, namely, in the case of erasures, what is the fraction of the remaining information that needs to be accessed in order to rebuild exactly the lost information? It is clear that when the number of erasures equals the maximum number of erasures that an MDS code can correct then the rebuilding ratio is 1 (access all the remaining information). However, the interesting and more practical case is when the number of erasures is smaller than the erasure correcting capability of the code. For example, consider an MDS code that can correct two erasures: What is the smallest amount of information that one needs to access in order to correct a single erasure? Previous work showed that the rebuilding ratio is bounded between $\frac{1}{4}$ and $\frac{2}{3}$, however, the exact value was left as an open problem. In this paper, we solve this open problem and prove that for the case of a single erasure, the rebuilding ratio is $\frac{1}{2}$. In general, we construct a new family of $r$-erasure correcting code, the rebuilding ratio is $\frac{1}{r}$. Hence, we can minimize the rebuilding ratio, then we also achieve optimal disk I/O, which is an important measurement in storage.

I. INTRODUCTION

Erasure-correcting codes are the basis of the ubiquitous RAID schemes for storage systems, where disks correspond to symbols in the code. Specifically, RAID schemes are based on MDS (maximum distance separable) array codes that can enable optimal storage and efficient encoding and decoding algorithms. With $r$ redundancy symbols, an MDS code is able to reconstruct the original information if no more than $r$ symbols are erased. An array code is a two dimensional array, where each column corresponds to a symbol in the code and is stored in a disk in the RAID scheme. We are going to refer to a disk/symbol as a node or a column interchangeably, and an entry in the array as an element. Examples of MDS array codes are EVENODD [1], [2], B-code [3], X-code [4], RDP [5], and STAR-code [6].

Suppose that some nodes are erased in a systematic MDS array code, we will rebuild them by accessing (reading) some information in the surviving nodes, all of which are assumed to be accessible. The fraction of the accessed information in the surviving nodes is called the rebuilding ratio, or simply ratio. If the nodes are erased, then the rebuilding ratio is 1 since we need to read all the remaining information. Is it possible to lower this ratio for less than $r$ erasures? Apparently, it is possible: Figure 1 shows an example of our new MDS code with 2 information nodes and 2 redundancy nodes, where every node has 2 elements, and operations are over the finite field of size 3. Consider the rebuilding of the first information node. It requires access to 3 elements out of 6 ($a$ rebuilding ratio of $\frac{1}{2}$), because $a = (a + b) - b$ and $c = (c + b) - b$.

It should be noted that the rebuilding ratio counts the amount of information accessed from the system. Therefore, if we can minimize the rebuilding ratio, then we also achieve optimal disk I/O, which is an important measurement in storage.

In practice, there is a difference between erasures of the information (also called systematic) and the parity nodes. An erasure of the former will affect the information access time since part of the raw information is missing, however erasure of the latter does not have such an effect, since the entire information is still accessible. Moreover, in most storage systems the number of parity nodes is quite small compared to the number of systematic nodes. Therefore, our constructions focus on optimal rebuilding ratio for the systematic nodes. The rebuilding of a parity node will require accessing all the information in the systematic nodes.

In [7], [8], a related problem is discussed: The nodes are assumed to be distributed and fully connected in a network, and the concept of repair bandwidth is defined as the minimum amount of data that needs to be transmitted over the network in order to rebuild the erased nodes. In contrast to our concept of rebuilding ratio a transmitted element of data can be a function of a number of elements that are accessed in the same node. In addition, in their general framework, an acceptable rebuilding is one that retains the MDS property and not necessarily rebuilds the original erased node, whereas, we restrict our...
solutions to exact rebuilding. It is clear that our framework is a special case of the general framework, hence, the repair bandwidth is a lower bound on the rebuilding ratio. Let $n$ be the total number of nodes and $k$ be the number of systematic nodes. Suppose a file of size $M$ is stored in an $(n,k)$ MDS code, where each node stores an information of size $M/k$. The number of redundancy/parity nodes is $r = n - k$, and in the rebuilding process all the surviving nodes are assumed to be accessible. A lower bound on the repair bandwidth for an $(n,k)$ MDS code was derived in [7]:

$$\frac{M}{k} \cdot \frac{n-1}{n-k}.$$  

It can be verified that Figure 1 matches this lower bound. Note that the above formula represents the amount of information, it should be normalized to reach the ratio. The normalized bandwidth compared to the size of the remaining array $\frac{M(n-1)}{k}$ is

$$\text{ratio} = \frac{1}{n-k} = \frac{1}{r}.$$  

(1)

A number of researchers addressed the repair bandwidth problem [7]–[16], however the constructed codes achieving the lower bound all have low code rate, i.e., $k/n < 1/2$. And it was shown by interference alignment in [14], [15] that this bound is asymptotically achievable for exact reconstructing and any $k,n$.

Instead of constructing MDS codes that can be easily rebuilt, a different approach of trying to rebuild existing families of MDS array codes was used in [17], [18]. The ratio of rebuilding a single systematic node was shown to be $\frac{1}{2} + o(1)$ for EVENODD, RDP, or X-code [1], [4], [5], all of which have 2 parities. However, based on the lower bound of (1) the ratio can be as small as 1/2. Moreover, related work on constructing codes with optimal rebuilding appeared independently in [19], [20]. Their constructions are similar to this work, but use larger finite field size.

Our main goal in this paper is to design $(n,k)$ MDS array codes with optimal rebuilding ratio, for arbitrary number of parities. We first consider the case of 2 parities. We assume that the code is systematic. In addition, we consider codes over some finite field $\mathbb{F}_2$ with an optimal update property, namely, when an information symbol which is an element from the field is rewritten, only the element itself and one element from each parity node needs an update. In total $r+1$ elements are updated. For an MDS code, this achieves the minimum reading/writing during writing of information. Hence, in the case of a code with 2 parities only 3 elements are updated. Under such assumptions, we will prove that every parity element is a linear combination of exactly one information element from each systematic column. We call this set of information elements a parity set. Moreover, the parity sets of a parity node form a partition of the information array.

For example, in Figure 1 the first parity node corresponds to parity sets $\{a,b\}, \{c,d\}$, which are elements in rows. We say this node is the row parity and each row of information forms a row set. The second parity node corresponds to parity sets $\{a,d\}, \{c,b\}$, which are elements in zigzag lines. We say that it is the zigzag parity and the parity set is called a zigzag set. For another example, Figure 2 shows a code with 3 systematic nodes and 2 parity nodes. Row parity $R$ is associated with row sets. Zigzag parity $Z$ is associated with sets of information elements with the same symbol. For instance the first element in column $R$ is a linear combination of the elements in the first row and in columns 0, 1, and 2. And the $Z$ in column $Z$ is a linear combination of all the $Z$ elements in columns 0, 1, and 2. We can see that each systematic column corresponds to a permutation of the four symbols. For instance, if read from top to bottom, column 0 corresponds to the permutation $[\bigtriangleup, \bigtriangledown, \lozenge, \spadesuit]$. In general, we will show that each parity relates to a set of a permutations of the systematic columns. Without loss of generality, we assume that the first parity node corresponds to identity permutations, namely, it is linear combination of rows.

It should be noted that in contrast to existing MDS array codes such as EVENODD and X-code, the parity sets in our codes are not limited to elements that correspond to straight lines in the array, but can also include elements that correspond to zigzag lines. We will demonstrate that this property is essential for achieving an optimal rebuilding ratio.

If a single systematic node is erased, we will rebuild each element in the erased node either by its corresponding row parity or zigzag parity, referred to as rebuild by row (or by zigzag). In particular, we access the row (zigzag) parity element, and all the elements in this row (zigzag) set, except the erased element. For example, consider Figure 2, suppose that the column labeled 1 is erased, then one can access the 8 shaded elements and rebuild its first two elements by rows, and the rest by zigzags. Namely, only half of the remaining elements are accessed. It can be verified that for the code in Figure 2, all the three systematic columns can be rebuilt by accessing half of the remaining elements. Thus the rebuilding ratio is 1/2, which is the lower bound expressed in (1).

The key idea in our construction is that for each erased node, the accessed row sets and the zigzag sets have a large intersection - resulting in a small number of accesses. Therefore it is crucial to find the permutations satisfying the above requirements. In this paper, we will present an optimal solution to this question by constructing permutations that are derived from binary vectors. This construction provides an optimal rebuilding ratio of 1/2 for any erasure of a systematic node. To generate the permutation over a set of integers from a binary vector, we simply add to each integer the vector and use the sum as the image of this integer. Here each integer is expressed as its binary expansion. For
example. Figure 3 illustrates how to generate the permutation on integers \( \{0, 1, 2, 3\} \) from the binary vector \( v = (1, 0) \). We first express each integer in binary: \((0,0),(0,1),(1,0),(1,1)\). Then add (mod 2) the vector \( v = (1,0) \) to each integer, and get \((1,0),(1,1),(0,0),(0,1)\). At last change each binary expansion back to integer and define it as the image of the permutation: 2, 3, 0, 1. Hence, 0, 1, 2, 3 are mapped to 2, 3, 0, 1 in this permutation, respectively. This simple technique for generating permutations is the key in our construction. We can generalize our construction for arbitrary \( r \) (number of parity nodes) by generating permutations using \( r \)-ary vectors. Our constructions are optimal in the sense that we can construct codes with \( r \) parities and rebuilding ratio of \( 1/r \).

So far we focused on the optimal rebuilding ratio, however, a code with two parity nodes should be able to correct two erasures. Namely, it needs to be an MDS code. We will prove that for a large enough field size the code can be made MDS. In addition, a key result we prove is that for a given number of rows, we have the maximum ratio and update. However, the length of the array is exponential to generate a larger number of permutations from binary vectors, and another is to use the same set of permutations multiple times.

In summary, the main contribution of this paper is the first explicit construction of systematic \((n,k)\) MDS array codes for any constant \( r = n - k \), which achieves optimal rebuilding ratio of \( 1/r \). Our codes have a number of useful properties:

- They are systematic codes, hence it is easy to retrieve information.
- They have high code rate \( k/n \), which is commonly required in storage systems.
- They have optimal update given a finite field \( \mathbb{F}_q \), namely, when an information element is updated, only \( r + 1 \) elements in the array need update.
- The rebuilding of a failed node requires no computation in each of the surviving nodes, and thus achieves optimal disk I/O.
- The encoding and decoding of the codes can be easily implemented for \( r = 2, 3 \), since the codes use small finite fields of size 3 and 4, respectively.
- They have optimal array size (maximum number of columns) among all systematic, optimal-update, and optimal-ratio codes. Moreover, we also have asymptotically optimal codes that have better array size.

They achieve optimal rebuilding ratio of \( 1/r \) when a single systematic erasure occurs.

The remainder of the paper is organized as follows. Section II constructs \((k+2,k)\) MDS array codes with optimal rebuilding ratio. Section III gives formal definitions and some general observations on MDS array codes. Section IV introduces ways to generate \((k+2,k)\) MDS array codes with larger number of columns. Section V generalizes the MDS code construction to an arbitrary number of parity columns. These generalized codes have properties that are similar to the \((k+2,k)\) MDS array codes, likewise some of them have optimal rebuilding ratio. Finally we provide concluding remarks in Section VI.

II. \((k+2,k)\) MDS ARRAY CODE CONSTRUCTIONS

Notations: In the rest of the paper, we are going to use \([i,j]\) to denote \([i, i+1, \ldots, j]\) and \([l]\) to denote \([1, 2, \ldots, l]\), for integers \( i \leq j \). And denote the complement of a subset \( X \subseteq M \) as \( \overline{X} = M \setminus X \). For a matrix \( A \), \( A^{T} \) denotes the transpose of \( A \). For a binary vector \( v = (v_1, \ldots, v_n) \) we denote by \( \overline{v} = (v_1 + 1 \mod 2, \ldots, v_n + 1 \mod 2) \) its complement vector. The standard vector basis of dimension \( m \) will be denoted as \( \{e_i\}_{i=1}^m \) and the zero vector will be denoted as \( 0_0 \). For two binary vectors \( v = (v_1, \ldots, v_m), u = (u_1, \ldots, u_m) \), the inner product is \( v \cdot u = \sum_{i=1}^{m} v_i u_i \mod 2 \). For two permutations \( f, g \), denote their composition by \( f \circ g \).

In this section we give the construction of MDS array code with two parities and optimal rebuilding ratio \( 1/2 \) for one erasure, which uses a finite field of optimal size 3.

Constructions

Let us define an MDS array code with 2 parities. Let \( A = (a_{ij}) \) be an information array of size \( p \times k \) over a finite field \( \mathbb{F}_q \), where \( i \in [0, p-1], j \in [0, k-1] \). We add to the array two parity columns and obtain an \( (n = k + 2, k) \) MDS code of array size \( p \times n \). Let the two parity columns be the row parity \( C_k = (r_0, r_1, \ldots, r_{p-1})^{T} \), and the zigzag parity \( C_{k+1} = (z_0, z_1, \ldots, z_{p-1})^{T} \). Let \( \{f_0, f_1, \ldots, f_{k-1}\} \) be zigzag permutations on \([0, p-1]\) associated with the systematic columns \([0, 1, \ldots, k-1]\). For any \( l \in [0, p-1] \), define the row set as the subset of information elements in the same row: \( R_l = \{a_{0,0}, a_{1,1}, \ldots, a_{l,k-1}\} \). The zigzag set is defined as elements in a zigzag line: \( Z_l = \{a_{ij}|f_l(i) = l\} \). Then define the row parity element as \( r_l = \sum_{a \in R_l} a \) and the zigzag parity element as \( z_l = \sum_{a \in Z_l} a \), for some sets of coefficients \( \{a_d\}, \{b_d\} \subseteq \mathbb{F} \). We can see that each parity element contains exactly one element from each systematic column, and we will show in Section III that this is equivalent to optimal update.

For example, in Figure 4 (a), we show three permutations on \([0,3]\). Therefore we have the 0th zigzag set \( Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\} \). The 0th row set is by default \( R_0 = \{a_{0,0}, a_{0,1}, a_{0,2}\} \). And in (b) we show the corresponding code. Columns \( C_0, C_1, C_2 \) are systematic columns. The row parity \( C_3 \) sums up elements in a row, and each element in the zigzag parity \( C_4 \) is a linear combination of the elements in some zigzag set. For instance, \( r_0 \) (or \( z_0 \)) is a linear combination of
elements in $R_0$ (or $Z_0$ respectively). Actually, this example is the code in Figure 2 with more details.

The **rebuilding ratio** is the average fraction of accessed elements in the surviving systematic and parity nodes while rebuilding one systematic node. A more specific definition will be given in the next section. In order to rebuild a systematic node, each erased element can be computed either by using its row set or by zigzag set. During the rebuilding process, an element is said to be rebuilt by row (zigzag), if we use the linear equation of its row (zigzag) set in order to compute its value. Solving this equation is done simply by accessing and reading in the surviving columns the values of the rest of the intermediates.

From the example in Figure 2, we know that in order to get low rebuilding ratio, we need to find zigzag sets $\{Z_i\}$ (and hence permutations $\{f_i\}$) such that the row and zigzag sets used in the rebuilding intersect as much as possible. Moreover, it is clear that the choice of the coefficients is crucial if we want to ensure the MDS property. Noticing that all elements and all coefficients are from some finite field, we would like to choose the coefficients such that the finite field size is as small as possible. So our construction of the code includes two steps:

1) Find zigzag permutations to minimize the ratio.
2) Assign the coefficients such that the code is MDS.

Next we generate zigzag permutations using binary vectors. We assume that the array has $p = 2^m$ rows.

In this section all the calculations for the indices are done over $\mathbb{F}_2$. By abuse of notation we use $x \in [0, 2^m - 1]$ both to represent the integer and its binary representation. It will be clear from the context which meaning is in use.

Let $v \in \mathbb{F}_2^m$ be a binary vector of length $m$. We define the permutation $f_v : [0, 2^m - 1] \to [0, 2^m - 1]$ by $f_v(x) = x + v$, where $x$ is represented in its binary representation.

For example when $m = 2$, $v = (1, 0)$, $x = 3$.

\[ f_{(1,0)}(3) = 3 + (1, 0) = (1, 1) + (1, 0) = (0, 1) = 1. \]

In other words, in order to get a permutation from $v$, we first write all integers in $[0, 2^m - 1]$ in binary expansion, then add vector $v$, and at last convert binary vectors back to integers. This procedure is illustrated in Figure 3. Thus we can see that the permutation $f_v$ in vector notation is $[2, 3, 0, 1]$. One can check that this is actually a permutation for any binary vector $v$. Next we present the code construction.

**Construction 1** Let $A$ be the information array of size $2^m \times k$. Let $T = \{v_0, v_1, \ldots, v_{k-1}\} \subseteq \mathbb{F}_2^m$ be a set of vectors of size $k$. For $v \in T$, we define the permutation $f_v : [0, 2^m - 1] \to [0, 2^m - 1]$ by $f_v(x) = x + v$. Construct the two parities as row and zigzag parities.

For example, in Figure 4 (a), the three permutations are generated by vectors $v_0 = (0, 0), v_1 = (1, 0), v_2 = (0, 1)$. In Figure 4 (b), the code is constructed with the row and the zigzag parities.

**Rebuilding Ratio**

Let us present the **rebuilding algorithm**: We define for a nonzero vector $v$, $X_v = \{x \in [0, 2^m - 1] : x \cdot v = 0\}$ as the set of integers whose binary representation is orthogonal to $v$. For example, $X_{(1,0)} = \{0, 1\}$. If $v$ is the zero vector we define $X_v = \{x \in \mathbb{F}_2^m : x \cdot (1, 1, \ldots, 1) = 0\}$. For ease of notation, denote the permutation $f_{v_j}$ as $f_j$ and the set $X_{v_j}$ as $X_j$. Assume column $j$ is erased, and define $S_r = \{a_{i,j} : i \in X_j\}$ and $S_z = \{a_{i,j} : i \not\in X_j\}$. Rebuild the elements in $S_r$ by rows and the elements in $S_z$ by zigzags.

**Example 1** Consider the code in Figure 4. Suppose node 1 (column $C_1$) is erased. Since $X_1 = X_{v_1} = X_{(1,0)} = \{0, 1\}$, we will rebuild $a_{0,1}, a_{1,1} \in S_r$ by row parity elements $r_0, r_1$, respectively. And rebuild $a_{2,1}, a_{3,1} \in S_z$ by zigzag parity elements $z_0, z_1$, respectively. In particular, we access the elements $a_{0,0}, a_{0,2}, a_{1,0}, a_{1,2}$, and the following four parity elements

\[
\begin{align*}
r_0 &= a_{0,0} + a_{0,1} + a_{0,2} \\
r_1 &= a_{1,0} + a_{1,1} + a_{1,2} \\
z_{f_1(2)} &= z_0 = a_{0,0} + 2a_{0,1} + 2a_{1,2} \\
z_{f_1(3)} &= z_1 = a_{1,0} + 2a_{3,1} + a_{0,2}.
\end{align*}
\]

Here $f_1(2) = f_{v_1}(2) = f_{(1,0)}(2) = 0$ and $z_{f_1(2)} = z_0$. Similarly, $f_1(3) = 1$ and $z_{f_1(3)} = z_1$. Note that each of the surviving node accesses exactly $\frac{1}{2}$ of its elements. Similarly, if node 0 is erased, we have $X_0 = \{0, 3\}$ so we rebuild $a_{0,0}, a_{3,0}$
by row and \( a_{1,0}, a_{2,0} \) by zigzag. Since \( X_2 = \{0, 2\} \), we rebuild \( a_{0,2}, a_{2,2} \) by row and \( a_{1,2}, a_{3,2} \) by zigzag in node 2. Rebuilding a parity node is easily done by accessing all the information elements.

**Theorem 1** Construct permutations \( f_0, \ldots, f_m \) and sets \( X_0, \ldots, X_m \) by the standard basis and the zero vector \( \{e_i\}_{i=0}^m \) as in Construction 1. Then the corresponding \((m + 3, m + 1)\) code has optimal ratio of \( \frac{1}{2} \).

Note that the code in Figure 4 is actually constructed as in Theorem 1. In order to prove Theorem 1, we first prove the following lemma. We represent each systematic node by the binary vector that generates its corresponding permutation. And define \( |v\setminus u| = \sum_{i: v_i = 1, u_i = 0} 1 \) as the number of coordinates at which \( v \) has a 1 but \( u \) has a 0.

**Lemma 2** (i) Let \( T \subseteq \mathbb{F}^m \) be a set of vectors. For any \( v, u \in T \), to rebuild node \( v \), the number of accessed elements in node \( u \) is

\[
|v\setminus u| = 2^m - |f_v(X_v) \cap f_u(X_v)|.
\]

(ii) If \( v \neq 0 \), then

\[
|f_v(X_v) \cap f_u(X_v)| = \begin{cases} |X_v|, & \text{mod } 2 = 0 \equiv 0 \\ |v\setminus u|, & \text{mod } 2 = 1 \equiv 1 \end{cases}.
\]

**Proof:** (i) In the rebuilding of node \( v \) the elements in rows \( X_v \) are rebuilt by rows, thus the row parity column accesses the values of the sum of rows \( X_v \). Therefore, the surviving node \( u \) also accesses its elements in rows \( X_v \). Hence, by now \( |X_v| = 2^m - 1 \) elements are accessed in node \( u \). The elements of node \( v \) in rows \( X_v \) are rebuilt by zigzags, thus the zigzag parity column accesses the values of the zigzag parity elements \( z_{f_v(\ell)} : \ell \in \mathbb{X}_v \), and each surviving systematic node accesses its elements that are contained in the corresponding zigzag sets, unless these elements were already accessed during the rebuilding by rows. The elements of node \( u \) in rows \( f_u^{-1}(f_v(X_v)) \) belong to zigzag sets \( \{Z_{f_v(\ell)} : \ell \in \mathbb{X}_v\} \), where \( f_u^{-1} \) is the inverse permutation of \( f_u \). Thus the extra elements node \( u \) needs to access are in rows \( f_u^{-1}(f_v(X_v)) \setminus X_v \).

\[
|f_u^{-1}(f_v(X_v)) \setminus X_v| = |f_u^{-1}(f_v(X_v)) \cap X_v| = 2^m - |f_u^{-1}(f_v(X_v)) \cup X_v| = 2^m - |f_u^{-1}(f_v(X_v))| + |X_v| - |f_u^{-1}(f_v(X_v)) \cap X_v| = |f_u^{-1}(f_v(X_v)) \cap X_v|,
\]

where we used the fact that \( f_v, f_u \) are bijections, and \( |X_v| = 2^m - 1 \).

(ii) Consider the group \( \mathbb{F}_2^m \times \mathbb{F}_2 \), and recall that \( f_v(X) = X + v = \{x + v: x \in X\} \). The sets \( f_v(X_v) = X_v + v \) and \( f_u(X_u) = X_u + u \) are cosets of the subgroup \( X_v \equiv \{v \in \mathbb{F}_2^m : v \cdot v = 0\} \), and they are either identical or disjoint. Moreover, they are identical iff \( v - u = 0 \) \( \text{mod } 2 \), and the result follows.

Let \( \{f_0, \ldots, f_{k-1}\} \) be a set of permutations over the set \([0, 2^m - 1]\) with associated subsets \( X_0, \ldots, X_{k-1} \subseteq [0, 2^m - 1] \), where each \( |X_i| = 2^{m-i} \). We say that this set is a set of orthogonal permutations if for any \( i, j \in [0, k - 1] \),

\[
\frac{|f_i(X_i) \cap f_j(X_j)|}{2^{m-i}} = \delta_{ij},
\]

where \( \delta_{ij} \) is the Kronecker delta. Assume the code was constructed by a set of orthogonal permutations. By Lemma 2 only half of the information is accessed (\( 2^{m-1} \) elements) in each of the surviving systematic columns during a rebuilding of a systematic column. Moreover, only \( 2^{m-1} \) elements are accessed from each parity node, too. Hence codes generated by orthogonal permutations have optimal rebuilding ratio \( \frac{1}{2} \).

Now we are ready to prove Theorem 1.

**Proof of Theorem 1:** Let \( i \neq j \), then since \( |e_i \setminus e_j| = 1 \), we get by Lemma 2 part (ii)

\[
|f_i(X_i) \cap f_j(X_i)| = 0.
\]

Moreover, \( f_i(X_0) = \{x + e_i : x \cdot (1, 1, \ldots, 1) = 0\} = \{y : y \cdot (1, 1, \cdots, 1) = 1\} \), and

\[
|f_0(X_0) \cap f_j(X_0)| = \{x : x \cdot (1, 1, \ldots, 1) = 0\} \cap \{y : y \cdot (1, 1, \cdots, 1) = 1\} = \emptyset.
\]

Hence the permutations \( f_0, \ldots, f_m \) are orthogonal permutations, and the ratio is \( \frac{1}{2} \) by Lemma 2 part (i).

Note that the optimal code can be shortened by removing some systematic columns and still retain an optimal ratio, i.e., for any \( k \leq m + 1 \) we have a code with optimal rebuilding.

**Finite Field Size**

Having found the set of orthogonal permutations, we need to specify the coefficients in the parities such that the code is MDS.

Consider the \((m + 3, m + 1)\) code \( C \) constructed by Theorem 1 and the vectors \( \{e_i\}_{i=0}^m \). Let \( \mathbb{F} \) be the finite field in use, where the information in row \( i \), column \( j \) is \( a_{ij} \in \mathbb{F} \). Let its row and zigzag coefficients be \( \alpha_{ij}, \beta_{ij} \in \mathbb{F} \) respectively. For a row set \( R_u = \{a_{ui_0}, a_{ui_1}, \ldots, a_{ui_m}\} \), the row parity is \( r_u = \sum_{i=0}^m a_{ui}a_{ui} \). For a zigzag set \( Z_u = \{a_{ui_0}, a_{ui_1}, \ldots, a_{ui_m}, a_{ui+e_{i+1}}\} \), the zigzag parity is \( z_u = \sum_{i=0}^m a_{ui+e_{i+1}}a_{ui+e_{i+1}} \).

Recall that the \((m + 3, m + 1)\) code is indeed MDS if we can recover the information from up to 2 columns erasures. It is clear that none of the coefficients \( \alpha_{ij}, \beta_{ij} \) can be zero. Moreover, if we assign all the coefficients as \( \alpha_{ij} = \beta_{ij} = 1 \) we get that in an erasure of two systematic columns the set of equations derived from the parity columns are linearly dependent and thus not solvable (the sum of the equations from the row parity and the sum of those from the zigzag parity will both be the sum of the entire information array).

Therefore the coefficients need to be from a field with more than one nonzero element, thus a field of size at least 3 is necessary.
Recall that we defined the permutations by binary vectors. This way of construction leads to special structure of the code. We are going to take advantage of it and assign the coefficients accordingly. Surprisingly $F_3$ is sufficient to correct two erasures.

**Construction 2** For the code $C$ in Theorem 1 over $F_3$, define $u_j = \sum_{i=0}^{l} e_i$ for $0 \leq j \leq m$. Assign row coefficients as $a_{ij} = 1$ for all $i, j$, and zigzag coefficients as

$$\beta_{ij} = 2^{i u_j}$$

(3)

where $i = (i_1, \ldots, i_m)$ is represented in binary and the calculation of the inner product in the exponent is done over $F_2$.

The coefficients in Figure 4 are assigned by Construction 2. For example

$$\beta_{3,1} = 2^{3 u_1} = 2^{(1,1),(1,0)} = 2^1 = 2.$$  

$$\beta_{3,2} = 2^{3 u_2} = 2^{(1,1),(1,1)} = 2^0 = 1.$$  

One can check that the code can tolerate any two erasures and hence is MDS.

The following theorem shows that the construction is MDS.

**Theorem 3** Construction 2 is an $(m+3, m+1)$ MDS code with optimal finite field size of 3.

**Proof:** It is easy to see that if at least one of the two erased columns is a parity column then we can recover the information. Hence we only need to show that we can recover from an erasure of any two systematic columns $i, j \in \{0, m\}$. In this scenario, we access the entire remaining information in the array. For $r \in \{0, 2^m - 1\}$ set $r' = r + e_i + e_j$, and recall that $a_{r_j} \in Z_l$ iff $l = r + e_i$, thus $a_{r_{ij}} = a_{r_{ij}} \in Z_1 + e_i$ and $a_{r_{ij}} = a_{r_{ij}} \in Z_2 + e_j$. From the two parity columns we need to solve the following equations

$$\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\beta_{r_{ij}} & 0 & \beta_{r'_{ij}} & a_{r_{ij}} \\
0 & \beta_{r'_{ij}} & a_{r'_{ij}} & 0
\end{bmatrix}
\begin{bmatrix}
a_{r_{ij}} \\
a_{r_{ij}} \\
a_{r_{ij}} \\
a_{r'_{ij}}
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}.$$  

(4)

Here $y_1, \ldots, y_4 \in F_3$ are the differences of the corresponding parity elements (the $r$-th, $r'$-th row parity, the $r + e_i$-th, $r + e_j$-th zigzag parity) after subtracting the weighted remaining elements in the row/zigzag sets. This set of equations is solvable iff

$$\beta_{r_{ij}} a_{r'_{ij}} \neq \beta_{r'_{ij}} a_{r_{ij}}.$$  

(5)

Note that the multiplicative group of $F_3 \setminus \{0\}$ is isomorphic to the additive group of $F_3$, hence multiplying two elements in $F_3 \setminus \{0\}$ is equivalent to summing up their exponent in $F_2$ when they are represented as a power of the primitive element of the field $F_3$. For columns $0 \leq i < j \leq m$ and rows $r, r'$ defined above, we have

$$\beta_{r_{ij}} a_{r'_{ij}} = 2^{r u_i + r' u_i} = 2^{(r + r') u_i} = 2^{(e_i + e_j) \sum_{l=0}^{i} e_l} = 2^{e_i} = 2.$$  

However in the same manner we derive that

$$\beta_{r_{ij}} a_{r'_{ij}} = 2^{(r + r') u_i} = 2^{(e_i + e_j) \sum_{l=0}^{i} e_l} = 2^{e_i + e_j} = 2^0 = 1.$$  

Hence (5) is satisfied and the code is MDS.

**Remark:** The above proof shows that $\beta_{r_{ij}} \neq \beta_{r'_{ij}}$ and $\beta_{r_{ij}} = \beta_{r'_{ij}}$ for $i < j$. And (5) is a necessary and sufficient condition for correcting erasure of columns $i$ and $j$ such that $v_i \neq v_j$.

It should be noted that in practice it is more convenient to use finite field $GF(2^s)$ for some integer $s$. In fact we can use a field of size 4 by simply modifying (3) to

$$\beta_{ij} = c^{i u_j},$$

where $c$ is a primitive element in $GF(4)$ and computations are done over $F_2$ in the exponent. It is obvious that this will not affect the proof of Theorem 3.

In addition to optimal ratio and optimal field size, we will show in the next section that the code in Theorem 1 is also of optimal array size, namely, it has the maximum number of columns, given the number of rows.

### III. Problem Settings and Properties

In this section, we prove some useful properties related to MDS array codes with optimal update.

Let $A = \{a_{ij}\}$ be an array of size $p \times k$ over a finite field $F$, where $i \in \{0, p - 1\}, j \in \{0, k - 1\}$, and each of its entries is an information element. Let $R = \{R_0, R_1, \ldots, R_{p-1}\}$ and $Z = \{Z_0, Z_1, \ldots, Z_{p-1}\}$ be two sets such that $R_i \cap Z_j = \emptyset$. Then for all $l \in \{0, p - 1\}$, define the row/zigzag parity element as $r_l = \sum_{a_{ij} \in R_l} a_{ij}$ and $z_l = \sum_{a_{ij} \in Z_l} a_{ij}$, for some sets of coefficients $\{a_{ij}\}, \{\beta_{ij}\} \subseteq F$. We call $R$ and $Z$ as the sets that generate the parity columns.

An MDS array code over $F_2$ with $r$ parities is said to be **optimal-update** if in the change of any information element only $r + 1$ elements are changed in the array. It is easy to see that $r + 1$ changes is the minimum possible number because if an information element appears only $r$ times in the array, then deleting at most $r$ columns will result in an unrecoverable $r$-eraser pattern and will contradict the MDS property. A small finite field size is desirable because we can update a small amount of information at a time if needed, and also get low computational complexity. Therefore we assume that the code is optimal-update, while we try to use the smallest possible finite field. When $r = 2$, only 3 elements in the code are updated when an information element is updated. Under this assumption, the following theorem characterizes the sets $R$ and $Z$.

**Theorem 4** For a $(k + 2, k)$ MDS code with optimal update, the sets $R$ and $Z$ are partitions of $A$ into $p$ equally sized sets of size $k$, where each set in $R$ or $Z$ contains exactly one element from each column.

**Proof:** Since the code is a $(k + 2, k)$ MDS code, each information element should appear at least once in each parity column $C_i, i \in \{0, k - 1\}$. However, since the code has optimal update, each element appears exactly once in each parity column.

Let $X \in R$, note that if $X$ contains two entries of $A$ from the systematic column $C_i, i \in \{0, k - 1\}$, then rebuilding is impossible if columns $C_j$ and $C_{j+1}$ are erased. Thus $X$ contains at most one entry from each column, therefore $|X| \leq k$. However each element of $A$ appears exactly once in each parity column,
Theorem 5

Let $A = (a_{ij})$ be an array of size $p \times k$ and the zigzag sets be $Z = \{Z_0, ..., Z_{p-1}\}$, then there exists a $(k+2, k)$ MDS array code for $A$ with $Z$ as its zigzag sets over the field $\mathbb{F}$ of size greater than $p(k-1) + 1$.

The proof is shown in Appendix A. The above theorem states that there exist coefficients such that the code is MDS, and thus we will focus first on finding proper zigzag permutations $\{f_i\}$. The idea behind choosing the zigzag sets is as follows: assume a systematic column $(a_{0,j_1}, a_{1,j_2}, ..., a_{p-j})^T$ is erased. Each element $a_{ij}$ is rebuilt either by row or by zigzag. The set $S = \{S_{0}, S_{1}, ..., S_{p-1}\}$ is called a rebuilding set for column $(a_{0,j_1}, a_{1,j_2}, ..., a_{p-j})^T$ if for each $i$, $S_i \subset \mathbb{F} \cup \mathbb{Z}$ and $a_{ij} \in S_i$. In order to minimize the number of accesses to rebuild the erased column, we need to minimize the size of

$$\bigcup_{i=0}^{p-1} S_i,$$

which is equivalent to maximizing the number of intersections between the sets $\{S_i\}_{i=0}^{p-1}$. More specifically, the intersections between the row sets in $S$ and the zigzag sets in $S$.

For a $(k+2, k)$ MDS code $C$ with $p$ rows define the rebuilding ratio $R(C)$ as the average fraction of accesses in the surviving systematic and parity nodes while rebuilding one systematic node, i.e.,

$$R(C) = \frac{\sum_{j=0}^{p} \min_{S_0, ..., S_{p-1} \text{ rebuilds } j} \left| \bigcup_{i=0}^{p-1} S_i \right|}{p(k+1)}.$$ 

Notice that in the two parity nodes, we access $p$ elements because each erased element must be rebuilt either by row or by zigzag, however $\bigcup_{i=0}^{p-1} S_i$ contains $p$ elements from the erased column. Thus the above expression is exactly the rebuilding ratio. Define the ratio function for all $(k+2, k)$ MDS codes with $p$ rows as

$$R(k) = \min _{C} R(C),$$

which is the minimal average portion of the array needed to be accessed in order to rebuild one erased column. By (1), we know that $R(k) \geq 1/2$. For example, the code in Figure 4 achieves the lower bound of ratio $1/2$, and therefore $R(3) = 1/2$. Moreover, we will see in Corollary 10 that $R(k)$ is almost $1/2$ for all $k$ and $p = 2^m$, where $m$ is large enough.

Thus far we have discussed the characteristics of an arbitrary MDS array code with optimal update. Next, let us look at our code in Construction 1.

Recall that by Theorem 5 this code can be an MDS code over a field large enough. The ratio of the constructed code will be proportional to the size of the union of the elements in the rebuilding set in (6). The following theorem gives the ratio for Construction 1 and can be easily derived from Lemma 2 part (i). Recall that given vectors $v_0, v_1, ..., v_{k-1}$, we write $f_i = f_{v_i}$ and $X_i = X_{v_i}$.

Theorem 6

The code described in Construction 1 and generated by the vectors $v_0, v_1, ..., v_{k-1}$ is a $(k+2, k)$ MDS array code with ratio

$$R = \frac{1}{2} + \frac{\sum_{j=0}^{p-1} \min_{f_i} |f_j(X_i) \cap f_j(X)|}{2^m k(k+1)}.$$ \hfill (7)

Note that different orthogonal sets of permutations can generate equivalent codes, hence we define equivalence of two sets of orthogonal permutations as follows. Let $F = \{f_1, f_2, ..., f_{k-1}, f_0\}$ be an orthogonal set of permutations over integers $[0, p-1]$, associated with subsets $X_1, X_2, ..., X_{k-1}, X_0$. And let $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_{k-1}, \sigma_0\}$ be another orthogonal set over $[0, p-1]$ associated with subsets $Y_1, Y_2, ..., Y_{k-1}, Y_0$. Then $F$ and $\Sigma$ are said to be equivalent if there exist permutations $g, h$ such that for all $i \in [0, k-1]$,

$$hfg = \sigma_i,$$

$$g^{-1}(X_i) = Y_i.$$

Note that multiplying $g$ on the right is the same as permuting the rows of the systematic nodes, and multiplying $h$ on the left permutes the rows of the second parity node. Therefore, codes constructed using $F$ or $\Sigma$ are essentially the same.

In particular, let us assume that the permutations are over integers $[0, 2^m-1]$, and the set of permutations $\Sigma$ and the subsets $Y_i$’s are the same as in Theorem 1: $\sigma_i = f_{v_i}$, $Y_i = \{x \in [0, 2^m-1]: x \cdot e_i = 0\}$, and $Y_0 = \{x \in [0, 2^m-1]: x \cdot (1, 1, ..., 1) = 0\}$. Next we show the optimal code in Theorem 1 is optimal in size, namely, it has the maximum number of columns given the number of rows. In addition any optimal-update, optimal-access code with maximum size is equivalent to the construction using standard basis vectors.

Theorem 7

Let $F$ be an orthogonal set of permutations over the integers $[0, 2^m-1]$,

(i) the size of $F$ is at most $m+1$;

(ii) if $|F| = m+1$ then it is equivalent to $\Sigma$ defined by the standard basis and zero vector.

Proof: We will prove it by induction on $m$. For $m = 0$ there is nothing to prove. (i) We first show that $|F| = k \leq m+1$. It is trivial to see that for any permutations $g, h$ on $[0, 2^m-1]$, the set $hFg = \{hfg, hfg^{-1}, ..., hfg^{m+1}\}$ is a code of size at most $m+1$. (ii) Then we show that the code is equivalent to the standard basis. Let $F$ be a code of size $m+1$. Then there exists an orthogonal set of permutations $\Sigma$ such that $F = \{\sigma_1, \sigma_2, ..., \sigma_{m+1}\}$ and $\Sigma$ is the same as in Theorem 1. Then we have $\sigma_i = f_{v_i}$, and $\Sigma$ is equivalent to the standard basis. Therefore, the code is equivalent to the standard basis.
\{hf_{0}, hf_{1}, ..., hf_{m-1}\} is also a set of orthogonal permutations with sets \(g^{-1}(X_0), g^{-1}(X_1), ..., g^{-1}(X_{k-1})\). Thus w.l.o.g. we can assume that \(f_0\) is the identity permutation and \(X_0 = [0, 2^{m-1} - 1]\). From the orthogonality we get that

\[
\bigcup_{i=1}^{k-1} f_i(X_0) \subseteq [2^{m-1}, 2^m - 1].
\]

We claim that for any \(i \neq 0, |X_i \cap X_0| = \frac{|X_0|}{m} = 2^{m-2} - 1\). Assume the contrary, thus if \(|X_i \cap X_0| > 2^{m-2}\), then for any distinct \(i, j \in [1, k-1]\), we get that

\[
f_j(X_i \cap X_0), f_i(X_i \cap X_0) \subseteq X_{0r},
\]

\[
|f_j(X_i \cap X_0)| = |f_i(X_i \cap X_0)| > 2^m - 2^m - 2 = \frac{|X_0|}{2}.
\]

From equations (8) and (9) we conclude that \(f_j(X_i \cap X_0) \cap f_i(X_i \cap X_0) \neq \emptyset\), which contradicts the orthogonality property. If \(|X_i \cap X_0| > 2^{m-2}\), the contradiction follows by a similar reasoning. Define the set of permutations \(F^* = \{f_i\}_{i=1}^{k-1}\) over the set of integers \([0, 2^{m-1} - 1]\) by \(f_i^*(x) = f_i(x) - 2^m - 1\), which is a set of orthogonal sets with sets \(X_i^* = \{X_i \cap X_0\}, i = 1, ..., k-1\). By induction \(k - 1 \leq m\) and the result follows.

(ii) Next we show that if \(|F| = m + 1\) then it is equivalent to \(\Sigma\) associated with \(\{Y_i\}\). Let \(F = \{f_1, f_2, ..., f_m, f_0\}\). Take two permutations \(g^i, h^i\) such that

\[
g^{-1}(X_1) = Y_1
\]

and \(h^i f_i g^i(Y_1) = Y_1\). Define \(f_i^* = h^i f_i g^i\) for all \(i \in [0, m]\). Then

\[
f_i(Y_1) = Y_1, f_i^*(Y_1) = Y_1.
\]

The new set of permutations \(\{f_i^*\}_{i=0}^{m}\) is also orthogonal with subsets \(\{g^{-1}(X_1)^m\}_{i=0}^{m}\), so \(f_i^*(Y_1) \cap f_i^*(Y_1) = \emptyset\). Hence for all \(i \neq 1\)

\[
f_i^*(Y_1) = Y_1 = [0, 2^{m-1} - 1],
\]

\[
f_i^*(Y_1) = Y_1 = [2^m - 1, 2^m - 1].
\]

By similar argument of part (i), we know \(\{f_2^*, ..., f_m^*, f_0^*\}\) restricted to \(Y_1\) (or to \(Y_1\)) is an orthogonal set of permutations, associated with subsets \(Y_1 \cap g^{-1}(X_i)\) (or with \(Y_1 \cap g^{-1}(X_i)\)), respectively, \(i \neq 1\). By the induction hypothesis, there exist permutations \(p, q\) over \(Y_1\) such that for \(i \neq 1\)

\[
\sigma_i = pf_i^*q,
\]

\[
g^{-1}(Y_1 \cap g^{-1}(X_1)) = Y_1 \cap Y_1,
\]

where \(\sigma_i, f_i^*\) are restricted to \(Y_1\). Similarly, there exist permutations \(r, s\) over \(Y_1\) such that for \(i \neq 1\)

\[
\sigma_i = rf_i^*s,
\]

\[
s^{-1}(Y_1 \cap g^{-1}(X_1)) = Y_1 \cap Y_1,
\]

where \(\sigma_i, f_i^*\) are restricted to \(Y_1\). Define permutation \(g''\) over \([0, 2^m - 1]\) as the union of \(g\) and \(s\): \(g''(x) = g(x)\) if \(x \in Y_1\), and \(g''(x) = s(x)\) if \(x \in Y_1\). Also define \(h''\) over \([0, 2^m - 1]\) as the union of \(p\) and \(r\). So \(g'', h''\) map \(Y_1\) (or \(Y_1\)) to itself. We will show that \(\{f_i\}_{i=0}^{m}\) is equivalent to \(\Sigma\) using \(g = g' g''\) and \(h = h'' h'\). For \(i \neq 1\), this is obvious from (10)(11). For \(i = 1\), we have

\[
g^{-1}(X_1) = g''^{-1}h''(X_1) = g''^{-1}(Y_1) = Y_1.
\]

We know \(\sigma_i = hf_{i} g\), for \(i \neq 1\). Let \(f = hf_{i} g\) and we will show \(f = \sigma_i\). By orthogonality \(f(Y_1) \cap \sigma_i(Y_1) = \emptyset\) for \(i \neq 1\). It is easy to see that for \(i \in [2, m]\), \(\sigma_i(Y_1) = Y_1\). Hence for \(i \in [2, m]\)

\[
f_i(Y_1) = Y_1, f_i(Y_1) = Y_1.
\]

Moreover, by construction \(f(Y_1) = h'' f_i g''(Y_1) = Y_1\), \(h'' f_1 g''(Y_1) = h'' f_1(Y_1) = h'' f_1(Y_1) = Y_1\), so

\[
f(Y_1) = Y_1, f_i(Y_1) = Y_1.
\]

Any integer \(x \in [0, 2^m - 1]\) can be written as the intersection of \(Y_i\) or \(Y'_i\), for all \(i \in [m]\), depending on its binary representation. For example, \(x = 1\) means \(\{x\} = \bigcap_{i=1}^{m} Y_i \cap Y_m\). For another example if \(x = 0\) then \(\{x\} = \bigcap_{i=1}^{m} Y_i\), and \(f(0) = f(\bigcap_{i=1}^{m} Y_i) = \bigcap_{i=1}^{m} Y_i \cap Y_m = (2^m - 1)\) by (12)(13) and since \(f\) is a bijection. Thus \(f(0) = 2^m - 1\). By a similar argument, \(f(x) = 2^m - 1 + x\) for all \(x\) and

\[
f = \sigma_1.
\]

Thus the proof is completed.

Note that by similar reasoning we can show that if \(|F| = m\), it is equivalent to \(\{\sigma_1, ..., \sigma_m\}\) defined by the standard basis.

Part (ii) of the above theorem says that if we consider codes with optimal update, optimal access, and optimal size, then they are equivalent to the standard-basis construction. In this sense, Theorem 1 gives the unique code. Moreover, if we find the smallest finite field for one code (as in Construction 2), there does not exist a code using a smaller field.

Part (i) of the above theorem implies that the number of rows has to be exponential in the number of columns in any systematic code with optimal ratio and optimal update. Notice that the code in Theorem 1 achieves the maximum possible number of columns, \(m + 1\). An exponential number of rows can be practical in some storage systems, since they are composed of dozens of nodes (disks) each of which has size in an order of gigabytes. However, a code may corresponds to only a small portion of each disk and we will need the flexibility of the array size. The following example shows a code of flatter array size with a cost of a small increase in the ratio.

Example 2 Let \(T = \{v \in \mathbb{F}_2^2 : \|v\|_1 = 3\}\) be the set of vectors with weight 3 and length \(m\). Notice that \(|T| = \binom{m}{3}\). Construct the code \(C\) by \(T\) according to Construction 1. Given \(v \in T, \{u \in T : \|v\| = 3\}\) is \(\binom{m-3}{3}\), which is the number of vectors with 1’s in different positions than \(v\). Similarly, \(\{|u \in T : \|v\| = 2\}\) is \(\binom{m-2}{2}\) and \(\{|u \in T : \|v\| = 1\}\) is \(3(m - 3)\). By Theorem 6 and Lemma 2, for large \(m\) the ratio is

\[
1 + \frac{2^m - 1}{2} + \frac{2^{m-1} m}{(m-3)(m+1)} \approx 1 + \frac{9}{2} + \frac{9}{2m}.
\]

Note that this code reaches the lower bound of the ratio as \(m\) tends to infinity, and has \(O(m^2)\) columns. More discussions on increasing the number of columns is presented in the next section.
IV. LENGTHENING THE CODE

As we mentioned, it is sometimes useful to construct codes with longer $k$ given the number of rows in the array. In this section we will provide two ways to reach this goal: we will first modify Example 2 and obtain an MDS code with a small finite field. Increasing the number of columns can also be done using code duplication (Theorem 9). In both methods, we sacrifice the optimal-ratio property for longer $k$, and the ratio is asymptotically optimal in both cases. Figure 5 summarizes the tradeoffs of different constructions. We will study the table in more details in the end of this section.

Constant Weight Vector

We will first give a construction based on Example 2 where all the binary vectors used have a constant weight. And we also specify the finite field size of the code.

Construction 3 Let $3|m$, and consider the following set of vectors $S \subseteq \mathbb{F}_2^n$: for each vector $v = (v_1, \ldots, v_m) \in S$, $\|v\|_1 = 3$ and $v_{i_1}, v_{i_2}, v_{i_3} = 1$ for some $i_1 \in [1, m/3], i_2 \in [m/3 + 1, 2m/3], i_3 \in [2m/3 + 1, m]$. For simplicity, we write $v = \{i_1, i_2, i_3\}$. Construct the $(k + 2, k)$ code as in Construction 1 using the set of vectors $S$, hence the number of systematic columns is $k = |S| = (\frac{m}{3})^3 = \frac{m^3}{27}$. For any $i \in [jm/3 + 1, (j + 1)m/3]$ and some $j = 0, 1, 2$, define a row vector $M_i = \sum_{\ell = jm/3 + 1}^{(j+1)m/3+1} e_\ell$. Then define a $m \times 3$ matrix

$$M_v = \begin{bmatrix} M_{i_1}^T & M_{i_2}^T & M_{i_3}^T \end{bmatrix}$$

for $v = \{i_1, i_2, i_3\}$. Let $a$ be a primitive element of $\mathbb{F}_9$. Assign the row coefficients as 1 and the zigzag coefficient for row $r$, column $v$ as $a^r$, where $t = rM_v \in \mathbb{F}_2^n$ (in its binary expansion).

For example, let $m = 6$, and $v = \{1, 4, 6\} = (1, 0, 0, 1, 0, 1) \in S$. The corresponding matrix is

$$M_v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^T.$$

For row $r = 26 = (0, 1, 1, 0, 1, 0)$, we have

$$t = rM_v = (0, 1, 1) = 3,$$

and the zigzag coefficient is $a^3$.

**Theorem 8** Construction 3 is a $(k + 2, k)$ MDS code with array size $2^m \times (k + 2)$ and $k = \frac{m^3}{27}$. Moreover, the rebuilding ratio is $\frac{1}{2} + \frac{3}{2m}$ for large $m$.

**Proof:** For each vector $v \in S$, there are $3(m/3 - 1)^2$ vectors $u \in S$ such that they have one 1 in the same location as $v$, i.e. $|v \setminus u| = 2$. Hence by Theorem 6 and Lemma 2, for large $m$ the ratio is

$$\frac{1}{2} + \frac{3((\frac{m}{9}) - 1)^2}{2^{m/9} + 1} \simeq \frac{1}{2} + \frac{9}{2m}.$$

Next we show that the MDS property of the code holds. Consider columns $u, v$ for some $u = \{i_1, i_2, i_3\} \neq v = \{j_1, j_2, j_3\}$ and $i_1, j_1 \in [1, m/3], i_2, j_2 \in [m/3 + 1, 2m/3], i_3, j_3 \in [2m/3 + 1, m]$. Consider rows $r$ and $r' = r + u + v$. The condition for the MDS property from (5) becomes

$$a^rM_u^T + a^{r'}M_v^T \mod 8 \neq a^{r''}M_w^T \mod 8 \quad (14)$$

where each vector of length 3 is viewed as an integer in $[0, 7]$ and the addition is the usual addition mod 8. Since $v \neq u$, let $l \in [1, 3]$ be the largest index such that $i_l \neq j_l$. W.l.o.g. assume that $i_l < j_l$, hence by the remark after Theorem 3

$$rM_{i_l}^T \neq r'M_{j_l}^T \quad (15)$$

and

$$rM_{j_l}^T = r'M_{i_l}^T. \quad (16)$$

Note that for all $l$, $l < t \leq 3$, $i_t \neq j_t$, then $r'M_{i_l}^T = (r + e_i + e_j)M_{i_l}^T = rM_{i_l}^T$, we have

$$rM_{i_l}^T = r'M_{i_l}^T = rM_{i_l}^T = \quad (17)$$

It is easy to infer from (15),(16),(17) that the l-th bit in the binary expansions of $rM_u^T + r'M_u^T \mod 8$ and $rM_u^T + r'M_u^T \mod 8$ don’t equal. Hence (14) is satisfied, and the result follows.

Notice that if we do mod 15 in (14) instead of mod 8, the proof still follows because 15 is greater than the largest possible sum in the equation. Therefore, a field of size 16 is also sufficient to construct an MDS code, and it is easier to implement in a storage system.

Construction 3 can be easily generalized to any constant $c$ such that it contains $O(m^c)$ columns and it uses any field of size at least $2^c + 1$. For simplicity assume that $c|m$, and simply construct the code using the set of vectors $\{v\} \subseteq \mathbb{F}_2^n$ such that $\|v\|_1 = c$, and for any $j \in [0, c - 1]$, there is a unique $i_j \in [jm/c + 1, (j + 1)m/c]$ and $v_{i_j} = 1$. Moreover, the finite field size of $2^{c+1}$ is also sufficient to make it an MDS code. When $c$ is odd the code has ratio of $\frac{1}{2} + \frac{3}{2m}$ for large $m$.

**Code Duplication**

Next we are going to duplicate the code to increase the number of columns in the constructed $(k + 2, k)$ MDS codes, such that $k$ does not depend on the number of rows, and the
Theorem 9

code, and columns 0 and the rest by rows. In $C$ is actually optimal.

The constructions so far assume that each zigzag permutation appears only once in the systematic columns. The key idea of code duplication is to use a multiset of permutations to define the zigzag parity. Let $C$ be a $(k+2,k)$ array code with $p$ rows, where the zigzag sets $\{Z_i\}_{i=1}^p$ are defined by the set of permutations $\{f_i\}_{i=1}^k$ acting on the integers $[0,p-1]$. For an integer $s$, an $s$-duplication code of $C$, denoted by $C'$, is an $(sk+2,sk)$ MDS code with zigzag permutations defined by duplicating the $k$ permutations $s$ times each. The formal definition and rebuilding algorithm are as follows. We are going to use superscripts to represent different copies of the ordinal code.

Construction 4 Define the multiset of permutations $F = \{f_0, \ldots, f_{k-1}, f_0, \ldots, f_{k-1}, \ldots, f_0, \ldots, f_{k-1}\}$, where each permutation $f_j$ has multiplicity $s$, for all $j \in [0,k-1]$. In order to distinguish different copies of the same permutation, denote the $t$-th $f_j$ as $f_j^{(t)}$. Let the $p \times sk$ information array be $[A^{(t)}]$, where $A^{(s-1)} = [a_{ij}^{(s-1)}], i \in [0,p-1]$, $j \in [0,k-1]$. Define the zigzag sets $Z_0, Z_1, \ldots, Z_{p-1}$ as $a_{ij}^{(1)} \in Z_t$ if $f_j^{(1)} = 1$. Notice that this definition is independent of $t$. For the $s$-duplication code $C'$, let the first parity still be the row parity, and the second parity be the zigzag parity according to the above zigzag sets. Denote the column corresponding to $f_j^{(1)}$ as column $j^{(0)}$, $0 \leq t \leq s-1$. Call the columns $\{j^{(t)} : j \in [0,k-1]\}$ the $t$-th copy of the original code.

Suppose in the optimal rebuilding algorithm of $C$ for column $i$, elements of rows $J = \{j_1, j_2, \ldots, j_s\}$ are rebuilt by zigzags, and the rest by rows. In $C'$, all the $s$ columns corresponding to $f_j$ are rebuilt in the same way: the elements in rows $J$ are rebuilt by zigzags, and the rest by rows.

In order to make the code MDS, the coefficients in the parities may be different from the original code $C$. An example of a 2-duplication of the code in Figure 4 is illustrated in Figure 6. Columns $0^{(0)}, 1^{(0)}, 2^{(0)}$ is the 0th copy of the original code, and columns $0^{(1)}, 1^{(1)}, 2^{(1)}$ is the 1st copy.

Theorem 9 If a $(k+2,k)$ code $C$ has rebuilding ratio $R(C)$, then its $s$-duplication code $C'$ has rebuilding ratio $R(C')(1 + \frac{1}{sk+1})$.

Proof: We will show that the rebuilding method in Construction 4 has rebuilding ratio of $R(C')(1 + \frac{1}{sk+1})$, and is actually optimal.

W.l.o.g. assume column $i^{(0)}$ is erased. Since column $i^{(1)}$, $t \in [1,s-1]$ corresponds to the same zigzag permutation as the erased column, for the erased element in the $t$-th row, no matter if it is rebuilt by row or by zigzag, we have to access the element in the $t$-th row and column $i^{(1)}$ (e.g. permutations $f_0^{(0)}, f_0^{(1)}$ and the corresponding columns $0^{(0)}, 0^{(1)}$ in Figure 6). Hence all the elements in column $i^{(t)}$ must be accessed. Moreover, the optimal way to access the other surviving columns can not be better than the optimal way to rebuild in the code $C$. Thus the proposed algorithm has optimal rebuilding ratio.

When column $i^{(0)}$ is erased, the average (over all $i \in [0,k-1]$) of the number of elements needed to be accessed in columns $i^{(t)}$, for all $t \in [0,k-1], t \neq i$ and $t \in [0,s-1]$ is

$$R(C)p(k+1) - p.$$ 

Here the term $-p$ corresponds to the access of the parity nodes in $C$. Moreover, we need to access all the elements in columns $i^{(t)}, 0 < t \leq s-1$, and access $p$ elements in the two parity columns. Therefore, the rebuilding ratio is

$$R(C') = \frac{s(R(C)p(k+1) - p) + (s-1)p + p}{p(sk+1)}$$

and the proof is completed.

Theorem 9 gives us the rebuilding ratio of the $s$-duplication of a code $C$ as a function of its rebuilding ratio $R(C)$. As a result, for the optimal-rebuilding ratio code in Theorem 1, the rebuilding ratio of its duplication code is slightly more than $1/2$, as the following corollary suggests.

Corollary 10 The $s$-duplication of the code in Theorem 1 has ratio $\frac{1}{2} + \frac{s+1}{4(s+1)},$ which is $\frac{1}{2} + \frac{1}{2(m+1)}$ for large $s$.

For example, we can rebuild the column $1^{(0)}$ in Figure 6 by accessing the elements in rows $[0,1]$ and in columns $0^{(0)}, 2^{(0)}, 0^{(1)}, 2^{(1)}, R, Z,$ and all the elements in column $1^{(1)}$. The rebuilding ratio for this code is $4/7$.

Using duplication we can have arbitrarily large number of columns, independent of the number of rows. Moreover the above corollary shows that it also has an almost optimal ratio. The next obvious question to be asked is: The duplication of which set of permutations will give the best asymptotic rebuilding ratio, when the number of duplications $s$ tends to infinity? The following theorem states that if we restrict ourselves to codes constructed using Construction 1 then the duplication of the permutations generated by the standard basis, gives the best asymptotic ratio. The proof is presented in Appendix B.

Theorem 11 The optimal asymptotic ratio among all codes constructed using duplication and Construction 1 is $\frac{1}{2}(1 + \frac{1}{m})$ and is achieved using the standard basis.

Next we address the problem of finding proper coefficients’ assignments in the parities in order to make the code MDS. Let $C'$ be the $s$-duplication of the optimal code of Theorem 1 and Corollary 10. Denote the coefficients for the element in row $i$ and column $j^{(t)}$ by $a_{ij}^{(t)}$ and $b_{ij}^{(t)}$, $0 \leq t \leq s-1$. Let $F_q$ be a field of size $q$ with primitive element $a$.

Construction 5 Let $F_q$ be a field of size at least $q \geq s + 1 + 1_q$, where

$$1_q = \begin{cases} 1, & q \text{ is even} \\ 0, & \text{else o.w.} \end{cases}$$
Figure 6. A 2-duplication of the code in Figure 4. The code has 6 information nodes and 2 parity nodes. The rebuilding ratio is 4/7.

Assign in $C'$, for any $i$, $j$, and $t \in [0, s - 1], a_{i,j}^{(t)} = 1,$

$$\beta_{i,j}^{(t)} = \begin{cases} a_{i,j}^{(t)(1-1-1-i)}, & \text{if } u_j, i = 1 \\ a_{i,j}^{1+i}, & \text{o.w.} \end{cases}$$

where $u_j = \sum_{i=0}^{j} e_i$.

Notice that the coefficients in each duplication have the same pattern as Construction 2 except that values 1 and 2 are replaced by $a^t$ and $a^{t+1}$ if $q$ is odd (or $a^{t+1}$ and $a^{-t-1}$ if $q$ is even).

**Theorem 12** Construction 5 is an $(s(m + 1) + 2, s(m + 1))$ MDS code.

**Proof:** Consider for example an $s$-duplication of the code in Theorem 1, by duplication of the code in Figure 4. The code has 24 and 68 columns (disks), and the field size needed can be 4 and 8 by Theorem 12, respectively. Both of these two sets of parameters are suitable for practical applications.

As mentioned in Theorem 11 the optimal construction yields a ratio of $1/2 + 1/2m$ by using duplication of the code in Theorem 1. However the field size is a linear function of the number of duplications of the code.

A comparison among the code constructed by the standard basis and zero vector (Theorem 1), by duplication of the standard basis and zero vector (Corollary 10), and by constant weight vectors (Construction 3) is shown in Figure 5. We can see that these three constructions provide a tradeoff among the rebuilding ratio, the number of columns, and the field size. If we want to access exactly 1/2 of the information during the rebuilding process, the standard basis construction is the only choice. If we are willing to sacrifice the rebuilding ratio for the sake of increasing the number of columns (disks) in the system, the other two codes are good options. Constant weight vectors technique has the advantage of a smaller field size over duplication, e.g., for $O(m^3)$ columns, the field of size 9 and $m$ is needed, respectively. However, duplication provides us a simple technique to have an arbitrary number of columns.
V. GENERALIZATION OF THE CODE CONSTRUCTION

In this section we generalize Construction 1 to an arbitrary number of parity nodes \( r = n - k \). We will construct an \((n,k)\) MDS array code, i.e., it can recover from up to \( r \) node erasures for arbitrary integers \( n,k \). We will show the code has optimal rebuilding ratio of \( 1/r \) when a systematic node is erased. When \( r = 3 \), we will prove that finite field size of 4 is sufficient for the code to be MDS. At last we will give an example of correcting multiple erasures.

As in the case for 2 parities, a file of size \( M \) is stored in the system, where each node (systematic or parity) stores a file of size \( \frac{M}{r} \). The \( k \) systematic nodes are stored in columns \([0,k-1]\). The \( i\)-th, \( 0 \leq i \leq r-1 \) parity node is stored in column \( k+i \), and is associated with zigzag sets \( \{Z^i_j : j \in [0,\ell - 1]\} \), where \( \ell \) is the number of rows in the array.

**Construction 6** Let \( A = (a_{ij}) \) be the information array of size \( r^m \times k \), for some integers \( k,m \). Let \( T = \{v_0,\ldots,v_{k-1}\} \subseteq \mathbb{Z}_r^m \) be a subset of vectors of size \( k \), where for each \( v = (v_1,\ldots,v_m) \in T \),

\[
gcd(v_1,\ldots,v_m,r) = 1, \tag{21}
\]

and \( \gcd \) is the greatest common divisor. For any \( l, 0 \leq l \leq r-1 \), and \( v \in T \) we define the permutation \( f^2_l : [0,r^m - 1] \rightarrow [0,r^m - 1] \) by \( f^2_l(x) = x + lv \), where by abuse of notation we use \( x \in [0,r^m - 1] \) both to represent the integer and its \( r \)-ary representation, and all the calculations are done over \( \mathbb{Z}_r \). For example, \( m = 2, r = 3, x = 4, l = 2, v = (0,1) \),

\[
f^2_2(4) = 4 + 2(0,1) = (1,1) + (0,2) = (1,0) = 3.
\]

One can check that the permutation \( f^2_l \) in a vector notation is \( [2,0,1,5,3,4,8,6,7] \). For simplicity denote the permutation \( f^2_l \) as \( f^2_l \) for \( v \in T \). For \( l \in [0,r^m - 1] \), we define the zigzag set \( Z^i_l \) in parity node \( i \), as the elements \( a_{ij} \) such that their coordinates satisfy \( f^2_l(i) = 1 \). In a rebuilding of systematic node \( i \) the elements in rows \( X^i_l = \{x : x \cdot v = r - 1\} \) are rebuilt by parity node \( l \), \( l \in [0,r-1] \), where the inner product in the definition is done over \( \mathbb{Z}_r \). For \( i \) we get that for any \( i \) and \( l, |X^i_l| = r^m - 1 \).

Note that similarly to Theorem 5, using a large enough field, the parity nodes described above form an \((n,k)\) MDS array code under appropriate selection of coefficients in the linear combinations of the zigzags.

Assume that the systematic column \( i \) is erased, what are the elements to be accessed in the systematic column \( j \neq i \) during the rebuilding process? By the construction, the elements of column \( i \) and rows \( X^j_l \) are rebuilt by the zigzags of parity \( l \). The indices of these zigzags are \( f^2_l(X^j) \). Therefore we need to access in the surviving systematic columns, all the elements that are contained in these zigzags. Specifically, the elements of systematic column \( j \) and rows \( f^2_l^{-1}(X^j) \) are contained in these zigzags, and therefore need to be accessed. In total, the elements to be accessed in systematic column \( j \) are

\[
\bigcup_{l=0}^{r-1} f^2_l^{-1}(X^j), \tag{22}
\]

The following lemma will help us to calculate the size of (22), and in particular to calculate the ratio of codes constructed by Construction 6.

**Lemma 14** For any \( v = (v_1,\ldots,v_m) \), \( u \in \mathbb{Z}_r^m \) and \( l,s \in [0,r-1] \) such that \( \gcd(v_1,\ldots,v_m,u) = 1 \), define \( c_{v,u} = v \cdot (v-u) - 1 \). Then

\[
|f^2_l^{-1} f^2_s(X^j) \cap f^2_s(X^j)| = \left\{ \begin{array}{ll} |X^j|, & (l-s)c_{v,u} = 0 \\ 0, & \text{o.w.} \end{array} \right. \]

In particular for \( s = 0 \) we get

\[
|f^2_l^{-1} f^2_0(X^j) \cap X^0| = \left\{ \begin{array}{ll} |X^0|, & \text{if } lc_{v,u} = 0 \\ 0, & \text{o.w.} \end{array} \right. \]

**Proof:** Consider the group \((\mathbb{Z}_r^m,+^\ast)\). Note that \( X^0 = \{x : x \cdot v = 0\} \) is a subgroup of \( \mathbb{Z}_r^m \) and \( X^0 = \{x : x \cdot v = r - 1\} \) is a coset. Therefore, \( X^0 = X^0_1 + a^0, X^0 = X^0 + a^0, \) for some \( a^0 \in X^0 \) and \( X^0 \). Hence \( f^2_l^{-1} f^2_s(X^0) = X^0 + a^0 + l(v-u) \) and \( f^2_l^{-1} f^2_0(X^0) = X^0 + a^0 + s(v-u) \) are cosets of \( X^0 \). So they are either identical or disjoint. Moreover they are identical if and only if

\[
a^0 - a^0 + (l-s)(v-u) \in X^0.
\]

i.e., \( (a^0 - a^0) + (l-s)(v-u) = 0 \). But by definition of \( X^0_1 \) and \( X^0 \), \( a^0 \cdot v = -l, a^0 \cdot v = -s \), so \((l-s) \cdot c_{v,u} = 0 \) and the result follows.

The following theorem gives the ratio for any code of Construction 6.

**Theorem 15** The ratio for the code constructed by Construction 6 and set of vectors \( T \) is

\[
\frac{\sum_{v \in T} \sum_{u \neq v \in T} 1}{\gcd(r,c_{v,u})} + \frac{|T|}{|T| - 1 + r},
\]

which also equals to

\[
\frac{1}{r} + \frac{\sum_{v \in T} \sum_{u \neq v \in T} |F_{v,u}(X^0_1) \cap X^0_u|}{|T|(|T| - 1 + r) r^m}.
\]

Where \( F_{v,u}(t) = f^2_l^{-1} f^2_s(t) \) for \( t \in X^0 \).

**Proof:** From any of the \( r \) parities, we access \( r^m - 1 \) elements during the rebuilding process of node \( v \). Therefore by (22), the fraction of the remaining elements to be accessed during the rebuilding is

\[
\frac{\sum_{u \neq v \in T} |f^2_l^{-1} f^2_s(X^j)| + r \cdot r^m}{(|T| - 1 + r) r^m}.
\]

Averaging over all the systematic nodes, the ratio is

\[
\frac{\sum_{v \in T} \sum_{u \neq v \in T} |f^2_l^{-1} f^2_s(X^j)| + |T| r^m}{|T|(|T| - 1 + r) r^m}.
\]

From Lemma 14, and noticing that

\[
|\{i : ic_{v,u} = 0 \text{ mod } r\}| = \gcd(r,c_{v,u}),
\]

we get

\[
|f^2_l^{-1} f^2_s(X^j)| = r^m - 1 \times r / \gcd(r,c_{v,u}),
\]
Theorem 16 The set \( \{ \{ f_j^l X_j^l \}_{l=0}^{r-1} \} \) together with the sets \( \{ \{ X_j^l \}_{l=0}^{r-1} \} \) is a family of orthogonal permutations. Moreover the corresponding \((m+1+r, m+1)\) code has optimal ratio of \(r\).

Proof: For \( 1 \leq i \neq j \leq m \), \( c_{ij} = c_i \cdot (c_i - c_j) - 1 = 0 \), hence by Lemma 14 for any \( l \in [0, r-1] \)

\[ f_j^{-1} f_i(X_i^l) \cap X_i^l = X_i^l, \]
and (25) is satisfied. For \( 1 \leq i \leq m \), and all \( 0 \leq l \leq r-1 \),

\[ f_0^{-1} f_i^l(X_i^l) = f_i^l(\{ v : v \cdot (1, ..., 1) = 1 \}) = \{ v + le_i : v \cdot (1, ..., 1) = 1 \} = \{ v : v \cdot (1, ..., 1) = 0 \} = X_0^l. \]

Therefore, \( f_0^{-1} f_i^l(X_i^l) \cap X_i^l = X_0^l \), and (25) is satisfied. Similarly,

\[ f_i^{-1} f_0^l(X_0^l) = f_i^{-1}(\{ v : v \cdot (1, ..., 1) = 1 \}) = \{ v - le_i : v \cdot (1, ..., 1) = 1 \} = \{ v : v \cdot (1, ..., 1) = 0 \} = X_0^l. \]

Hence again (25) is satisfied and this is a family of orthogonal permutations, and the result follows.

Surprisingly, one can infer from the above theorem that changing the number of parities from 2 to 3 adds only one node to the system, but reduces the rebuilding ratio from 1/2 to 1/3 in the rebuilding of any systematic code.

The example in Figure 7 shows a code with 3 systematic nodes and 3 parity nodes constructed by Theorem 16 with \( m = 2 \). The code has an optimal ratio of 1/3. For instance, if column \( C_1 \) is erased, accessing rows \( \{0, 1, 2\} \) in the remaining nodes will be sufficient for rebuilding.

Similar to the 2 parity case, the following theorem shows that Theorem 16 achieves the optimal number of columns. In other words, the number of rows has to be exponential in the number of columns in any systematic MDS code with optimal ratio, optimal update, and \( r \) parities. This follows since any such optimal code is constructed from a family of orthogonal permutations.

Theorem 17 Let \( \{ \{ f_j^l \}_{l=0}^{r-1} \} \) be a family of orthogonal permutations over the integers \([0, r^m-1]\) together with the sets \( \{ \{ X_j^l \}_{l=0}^{r-1} \} \), \( k \leq m + 1 \).

Proof: We prove it by induction on \( m \). When \( m = 0 \), it is trivial that \( k \leq 1 \). Now suppose we have a family of orthogonal permutations \( \{ \{ f_j^l \}_{l=0}^{r^m-1} \} \) over \([0, r^m-1]\), and we will show \( k \leq m + 1 \). Recall that orthogonality is equivalent to (25). Notice that for any permutations \( g, h_0, ..., h_{r-1} \), the sets of permutations \( \{ h_i f_0^l g \}_{l=0}^{r^m-1}, \{ h_i f_0^l g \}_{l=0}^{r^m-1} \) are still a family of orthogonal permutations with sets \( \{ g^{-1}(X_0^l) \}_{l=0}^{r^m-1}, \{ g^{-1}(X_{k-1}^l) \}_{l=0}^{r^m-1} \). This is because

\[ h_i f_0^l g(X_0^l) = h_i f_0^l(X_0^l) = h_i f_0^l g(X_0^l). \]

Therefore, \( w.l.o.g \), we can assume \( X_0^l = [r^m-(l+1)r^{m-1}-1] \), and \( f_0^l \) is the identity permutation, for \( 0 \leq l \leq r-1 \).

Let \( 1 \leq i \neq j \leq k, l \in [0, r-1] \) and define

\[ A = f_i^l(X_i^l) = f_i^l(X_i^l), \]
\[ B = f_j^l(X_j^l) \cap X_0^l, \]
\[ C = f_i^l(X_i^l). \]

Therefore, \( B, C \) are subsets of \( A \), and their complements in \( A \) are

\[ A \setminus B = f_i^l(X_i^l \setminus X_0^l), \]
\[ A \setminus C = f_i^l(X_i^l \setminus X_0^l). \]
we conclude that

$$f_i^{j}(X_0^j) = f_i^{j}(X_0^j) = X_0^j$$

Thus, in order to construct an MDS code with the optimal ratio 1/3, we need to consider a parity node code with optimal ratio 1/3. For any $f_i^{j}$, $f_i^{j}$ is a permutation on $\{X_1^j\}$, hence

$$A \cap B \cap C \subseteq X_0^j.$$ (28)

From (27), (28) we conclude that $B = C = A \cap X_0^j$, i.e.,

$$f_i^{j}(X_0^j \cap X_0^j) = f_i^{j}(X_0^j \cap X_0^j).$$

For each $l \in [0, r - 1], l \in [1, k - 1]$ define $f_i^{j}(X_0^j) = f_i^{j}(X_0^j) - lr^{-m-1}$ and $X_0^l = X_0^j \cap X_0^l$ then, we have

$$f_i^{j}([0, r^{-m-1} - 1]) = f_i^{j}(X_0^l) - lr^{-m-1} = X_0^j - lr^{-m-1} = [0, r^{-m-1} - 1],$$

(30)

From (26), (27) and (28), we can conclude that $f_i^{j}$ is a bijection, hence

$$X_0^j = X_0^l.$$ (31)

Similarly, for any $j \neq 0, f_i^{j}(X_0^j) = f_i^{j}(X_0^j) = X_0^j,$

hence

$$A \cap B \cap C \subseteq X_0^j.$$ (28)

From (27), (28) we conclude that $B = C = A \cap X_0^j$, i.e.,

$$f_i^{j}(X_0^j \cap X_0^j) = f_i^{j}(X_0^j \cap X_0^j).$$

For each $l \in [0, r - 1], l \in [1, k - 1]$ define $f_i^{j}(X_0^j) = f_i^{j}(X_0^j) - lr^{-m-1}$ and $X_0^l = X_0^j \cap X_0^l$ then, we have

$$f_i^{j}([0, r^{-m-1} - 1]) = f_i^{j}(X_0^l) - lr^{-m-1} = X_0^j - lr^{-m-1} = [0, r^{-m-1} - 1],$$

(30)

From (26), (27) and (28), we can conclude that $f_i^{j}$ is a bijection, hence

$$X_0^j = X_0^l.$$ (31)

Similarly, for any $j \neq 0, f_i^{j}(X_0^j) = f_i^{j}(X_0^j) = X_0^j,$

hence

$$A \cap B \cap C \subseteq X_0^j.$$ (28)

From (27), (28) we conclude that $B = C = A \cap X_0^j$, i.e.,

$$f_i^{j}(X_0^j \cap X_0^j) = f_i^{j}(X_0^j \cap X_0^j).$$

For each $l \in [0, r - 1], l \in [1, k - 1]$ define $f_i^{j}(X_0^j) = f_i^{j}(X_0^j) - lr^{-m-1}$ and $X_0^l = X_0^j \cap X_0^l$ then, we have

$$f_i^{j}([0, r^{-m-1} - 1]) = f_i^{j}(X_0^l) - lr^{-m-1} = X_0^j - lr^{-m-1} = [0, r^{-m-1} - 1],$$

(30)

From (26), (27) and (28), we can conclude that $f_i^{j}$ is a bijection, hence

$$X_0^j = X_0^l.$$ (31)
Let the generator matrix of the code be

\[ G' = \begin{bmatrix}
    I & & & & \\
    & I & & & \\
    & & \ddots & & \\
    & & & I & \\
    P_0 & \cdots & P_m & & \\
    P_0^2 & \cdots & P_m^2 & & 
\end{bmatrix}_{(m+4) \times m+1} \]  

Here each block matrix is of size $3^m \times 3^m$, and $P_j^2$ represents the square of the matrix $P_j$.

**Theorem 18** When $r = 3$, a field of size 4 is sufficient to make the code MDS using Construction 7.

The proof is shown in Appendix C. The main idea is similar to the case of two parities: utilizing the special structure of the permutations. For example, the coefficients of the parities in Figure 7 are assigned as Construction 7. One can check that the system is protected against any three erasures.

A natural generalization of the rebuilding problem, is what happens if multiple erasures of systematic nodes occur, i.e., $e > 1$. Our goal is to rebuild these nodes simultaneously from the information in the surviving nodes. It should be noted that this model is a bit different from the distributed repair problem, where the recovery of each node is done separately. If there are $e$ erasures, $1 \leq e \leq r$, the lower bound for repair bandwidth is

\[ \frac{e}{r} \]

and so is the lower bound for the rebuilding ratio. It turns out that the code constructed in Theorem 16 has also optimal rebuilding ratio for any number of erasures $1 \leq e \leq r$. For more details see [25]. In the following we give an example of an optimal rebuilding in the case of two erasures.

**Example 3** Consider the code in Figure 7 with $r = 3$. Assume that $e = 2$ and columns $C_0, C_1$ were erased. Access rows $\{0, 1, 3, 4, 6, 7\}$ in columns $C_2, C_3$, rows $\{1, 2, 4, 5, 7, 8\}$ in column $C_4$, and rows $\{2, 0, 5, 3, 8, 6\}$ in column $C_5$. One can check that the accessed elements are sufficient to rebuild the two erased columns $C_0, C_1$, and the rebuilding ratio is $2/3 = e/r$. It can be shown that an optimal rebuilding can be done for any two systematic node erasures.

## VI. Concluding Remarks

In this paper, we described explicit constructions of the first known systematic $(n,k)$ MDS array codes with $n - k$ equal to some constant, such that the amount of information needed to rebuild an erased column equals to $1/(n - k)$, matching the information-theoretic lower bound. While the codes are new and interesting from a theoretical perspective, they also provide an exciting practical solution, specifically, when $n - k = 2$, our zigzag codes are the best known alternative to RAID-6 schemes. RAID-6 is the most prominent scheme in storage systems for combating disk failures [1]-[6]. Our new zigzag codes provide a RAID-6 scheme that has optimal update (important for write efficiency), small finite field size (important for computational efficiency) and optimal access of information for rebuilding - cutting the current rebuilding time by a factor of two.

We note that one can add redundancy for the sake of lowering the rebuilding ratio. For instance, one can use three parity nodes instead of two. The idea is that the third parity is not used for protecting data from erasures, since in practice, three concurrent failures are unlikely. However, with three parity nodes, we are able to rebuild a single failed node by accessing only $1/3$ of the remaining information (instead of $1/2$). An open problem is to construct codes that can be extended in a simple way, namely, codes with three parity nodes such that the first two nodes ensure a rebuilding ratio of $1/2$ and the third node further lowers the ratio to $1/3$. Hence, we can first construct an array with two parity nodes and when needed, extend the array by adding an additional parity node to obtain additional improvement in the rebuilding ratio.

Another future research direction is to consider the ratio of read accesses in the case of a write (update) operation. For example, in an array code with two parity nodes, in order to update a single information element, one needs to read at least three elements and write three elements, because we need to know the values of the old information and old parities and compute the new parity elements (by subtracting the old information from the parity and adding the new information). However, an interesting observation, in our optimal code construction with two parity nodes, is if we update all the information in the first column and the rows in the top half of the array (see Figure 4), we do not need to read for computing the new parities, because we know the values of all the information elements needed for computing the parities. These information elements take about half the size of the entire array. So in a storage system we can cache the information to be written until most of these elements need to be updated (we could arrange the information in a way that these elements are often updated at the same time), hence, the ratio between the number of read operations and the number of new information elements is relatively very small. Clearly, we can use a similar approach for any other systematic column. In general, given $r$ parity nodes, we can avoid redundant read operations if we update about $1/r$ of the array.

## Appendix A

**Proof of Theorem 5**

Theorem 5 states that if the finite field is large enough, we can make a code constructed by permutations MDS. We rewrite the theorem here.

**Theorem 5** Let $A = (a_{ij})$ be an array of size $p \times k$ and the zigzag sets be $Z = \{Z_0, \ldots, Z_{p-1}\}$, then there exists a $(k + 2, k)$ MDS array code for $A$ with $Z$ as its zigzag sets over the field $\mathbb{F}$ of size greater than $p(k - 1) + 1$.

In order to prove Theorem 5, we use the well known Combinatorial Nullstellensatz by Alon [21]:

**Theorem 19** (Combinatorial Nullstellensatz) [21, Th 1.2] Let $\mathbb{F}$ be an arbitrary field, and let $f(x_1, \ldots, x_n)$ be a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. Suppose the degree of $f$ is $\deg(f) = \sum_{i=1}^n t_i$, where each $t_i$ is a nonnegative integer, and suppose the
coefficient of \( \prod_{i=1}^{q} x_i^{j_i} \) in \( f \) is nonzero. Then, if \( S_1, \ldots, S_n \) are subsets of \( F \) with \( |S_i| > t_i \), there are \( s_1 \in S_1, s_2 \in S_2, \ldots, s_q \in S_q \) so that

\[
f(s_1, \ldots, s_q) \neq 0.
\]

**Proof of Theorem 5:** Assume the information of \( A \) is given in a column vector \( W \) of length \( pk \), where column \( i \in [0, k - 1] \) of \( A \) is in the row set \( \{(ip, (i + 1)p - 1)\} \) of \( W \). Each systematic node \( i, i \in [0, k - 1] \), can be represented as \( Q_iW \) where \( Q_i = [p_{ip, p}I_{p_{ip, p}}, 0_{p_{ip, p}[k-1]}] \).

Moreover define \( Q_k = [p_{ip, p}I_{p_{ip, p}}, \ldots, I_{p_{ip, p}}, Q_{k+1} = [x_0 p_{0}, x_1 P_1, \ldots, x_k P_k] \) where the \( P_i \)'s are permutation matrices (not necessarily distinct) of size \( p \times p \), and the \( x_i \)'s are variables, such that \( C_k = Q_k W, C_{k+1} = Q_{k+1} W \). The permutation matrix \( P_i = (p_i)^{j_i} \) is defined as \( p_i^{j_i} = 1 \) if and only if \( a_{m,i} \in Z_i \). In order to show that there exists such MDS code, it is sufficient to show that there is an assignment for the intermediates \( \{x_i\} \) in the field \( F \), such that for any set of integers \( \{s_1, s_2, \ldots, s_k\} \subseteq [0, k + 1] \) the matrix \( Q = [Q_{s_1}^T, Q_{s_2}^T, \ldots, Q_{s_k}^T] \) is of full rank. It is easy to see that if the parity column \( C_{k+1} \) is erased i.e., \( k + 1 \notin \{s_1, s_2, \ldots, s_k\} \) then \( Q \) is of full rank. If \( k \notin \{s_1, s_2, \ldots, s_k\} \) and \( k + 1 \in \{s_1, s_2, \ldots, s_k\} \) then \( Q \) is of full rank if none of the \( x_i \)'s equals to zero. The last case is when both \( k, k + 1 \in \{s_1, s_2, \ldots, s_k\} \), i.e., there are \( 0 \leq i < j \leq k - 1 \) such that \( i, j \notin \{s_1, s_2, \ldots, s_k\} \).

It is easy to see that in that case \( Q \) is of full rank if and only if the submatrix

\[
B_{ij} = \begin{pmatrix}
  x_i P_i \\
  x_j P_j \\
\end{pmatrix}
\]

is of full rank. This is equivalent to \( \det(B_{ij}) \neq 0 \). Note that \( \deg(\det(B_{ij})) = p \) and the coefficient of \( x_i^p \) is \( \det(P_i) \in \{1, -1\} \). Define the polynomial

\[
T = T(x_0, x_1, \ldots, x_{k-1}) = \prod_{0 \leq i < j \leq k-1} \det(B_{ij}),
\]

and the result follows if there are elements \( a_{0, a_1, \ldots, a_{k-1}} \in F \) such that \( T(a_0, a_1, \ldots, a_{k-1}) \neq 0 \). \( T \) is of degree \( p^{k^2} \) and the coefficient of \( \prod_{i=0}^{k-1} x_i^{k(1-k-i)} \) is \( \prod_{i=0}^{k-1} \det(P_i)^{k-k-i} \neq 0 \). Set for any \( t, S_l = F \setminus 0 \) in Theorem 19, and the result follows. \( \Box \)

**Appendix B**

**Proof of Theorem 11**

Here we will prove that the duplication code of standard basis has optimal rate. The theorem is restated here.

**Theorem 11** The optimal asymptotic ratio among all codes constructed using duplication and Construction 1 is \( \frac{1}{2}(1 + \frac{1}{m}) \) and is achieved using the standard basis.

In order to prove the theorem, we need to define a related graph and to prove an extra theorem and lemma. Define the directed graph \( D_m = D_m(V, E) \) as \( V = \{v \in F_m^2 : v \neq 0\} \), and \( E = \{(v_1, v_2) : |v_1| = 1 \mod 2\} \). Hence the vertices are the nonzero binary vectors of length \( m \), and there is a directed edge from \( v_1 \) to \( v_2 \) if \( [v_2 - v_1] \) is odd. Let \( H \) be an induced subgraph of \( D_m \) on a subset of \( V \). Let \( S \) and \( T \) be two disjoint subsets of vertices of \( H \). We define the density between \( S \) and \( T \) to be \( d_{S,T} = \frac{E_{S,T}}{2|S||T|} \), and the density of the set \( S \) to be \( d_S = \frac{E_S}{2|S|} \), where \( E_S \) is the number of edges with both of its endpoints in \( S \), and \( E_{S,T} \) is the number of edges incident with a vertex in \( S \) and a vertex in \( T \).

For example, suppose the vertices of \( H \) are the vectors \( (0, 0, 0, 1), (0, 0, 1, 1), (1, 1, 1, 0), (1, 0, 0, 0) \). The graph \( H \) is shown in Figure 8. The density of the graph is \( d_H = 7/16 \), and for \( S = (1, 0, 0, 0), T = \{(0, 0, 1, 1), (1, 1, 1, 0)\} \) the density is \( d_{S,T} = 1/2 \). Denote by \( C(H) \) the code constructed using the vertices of \( H \), and Construction 1. In this example the code \( C(H) \) has four systematic disks and is encoded using the permutations generated by the four vectors of \( H \). Duplication of the code \( C(H ) \), \( s \) times, namely duplicating \( s \) times the permutations generated by the vectors of \( H \) will yield a code with \( 4s \) systematic disks and two parities.

Let \( v_1, v_2 \) be two vertices of \( H \), such that there is a directed edge from \( v_1 \) to \( v_2 \). By Lemma 2 we know that this edge means \( f_{v_2}(X_{v_2}) \cap f_{v_1}(X_{v_1}) = \emptyset \), therefore only half of the information from the column corresponding to \( v_1 \) is accessed and read while rebuilding the column corresponding to \( v_2 \). Note that this observation is also correct for a \( s \)-duplication code of \( C(H) \). Namely, if we rebuild any column corresponding to a copy of \( v_2 \), only half of the information is accessed in any of the columns corresponding to a copy of \( v_1 \). Intuitively, when the number of copies \( s \) is large, the density of the graph captures how often such savings will occur. The following theorem shows that the asymptotic ratio of any code constructed using Construction 1 and duplication is a function of the density of the corresponding graph \( H \).

**Theorem 20** Let \( H \) be an induced subgraph of \( D_m \). Let \( C_s(H) \) be the \( s \)-duplication of the code constructed using the vertices of \( H \) and Construction 1. Then the asymptotic ratio of \( C_s(H) \) is

\[
\lim_{s \to \infty} R(C_s(H)) = 1 - \frac{d_H}{2}
\]

**Proof:** Let the set of vertices and edges of \( H \) be \( V(H) = \{v_i\} \) and \( E(H) \) respectively. Denote by \( v_i^{(1)}, v_i \in V(H), l \in [0, s - 1] \), the \( l \)-th copy of the column corresponding to the vertex \( v_i \). In the rebuilding of column \( v_i^{(l)}, l \in [0, s - 1] \) each remaining systematic column \( v_i^{(l)} \) should be selected to access all of its \( 2^m \) elements unless \( |v_i| = 1 \) and in that case it only has to access \( 2^{m-1} \) elements. Hence the total amount of accessed information for rebuilding this column is

\[
(s|V(H)| - 1)2^m - \deg^+(v_i)s2^{m-1},
\]
where $\deg^+$ is the indegree of $v_i$ in the induced subgraph $H$. Averaging over all the columns in $C_s(H)$ we get the ratio:

$$ R(C_s(H)) = \frac{\sum_{v_i \in C_s(H)} (s|V(H)| - 1)2^m - \deg^+(v_i)s2^{m-1}}{s|V(H)|(s|V(H)| + 1)2^m} $$

$$ = \frac{s|V(H)|(s|V(H)| - 1)2^m - s^2\sum_{v_i \in V(H)} \deg^+(v_i)2^{m-1}}{s|V(H)|(s|V(H)| + 1)2^m} $$

Hence

$$ \lim_{s \to \infty} R(C_s(H)) = 1 - \frac{|E(H)|}{2|V(H)|^2} = 1 - \frac{d_H}{2}. $$

We conclude from Theorem 20 that the asymptotic ratio of any code using duplication and a set of binary vectors $\{v_i\}$ is a function of the density of the induced subgraph on this set of vertices. Hence the induced subgraph of $D_m$ with maximal density corresponds to the code with optimal asymptotic ratio. It is easy to check that the induced subgraph with its vertices as the standard basis $\{e_i\}_{i=1}^m$ has density $\frac{m-1}{m}$. In fact this is the maximal possible density among all the induced subgraphs and therefore it gives a code with the best asymptotic ratio, but in order to show it we need the following technical lemma.

**Lemma 21** Let $D = D(V, E)$ be a directed graph and $S, T$ be a partition of $V$, i.e., $S \cap T = \emptyset, S \cup T = V$, then

$$ d_V \leq \max \{d_S, d_T, d_{ST}\} $$

**Proof:** Note that $d_V = \frac{|S|^2d_S + |T|^2d_T + 2S|T|d_{ST}}{|V|^2}$. W.l.o.g assume that $d_S \geq d_T$ therefore if $d_S \geq D_{ST}$,

$$ d_V = \frac{|S|^2d_S + |T|^2d_T + 2S|T|d_{ST}}{|V|^2} $$

$$ \leq \frac{|S|^2d_S + |T|^2d_S - |T|^2d_S + |T|^2d_T + 2S|T|d_{ST}}{|V|^2} $$

$$ = \frac{d_S(|S| + |T|)^2 - |T|^2(d_S - d_T)}{|V|^2} $$

$$ \leq d_S. $$

If $d_{ST} \geq \max \{d_S, d_T\}$ then

$$ d_V = \frac{|S|^2d_S + |T|^2d_T + 2S|T|d_{ST}}{|V|^2} $$

$$ \leq \frac{|S|^2d_{ST} + |T|^2d_{ST} + 2S|T|d_{ST}}{|V|^2} $$

$$ = d_{ST} $$

and the result follows.

Now we are ready to prove the optimality of the duplication of the code using the standard basis, if we assume that the number of copies $s$ tends to infinity. We will show that for any induced subgraph $H$ of $D_m$, $d_H \leq \frac{m-1}{m}$. Hence the optimal asymptotic ratio among all codes constructed using duplication and Construction 1 is

$$ 1 - \frac{1}{m} \frac{m-1}{m} = \frac{1}{2}(1 + \frac{1}{m}) $$

and is achieved using the standard basis.

**Proof of Theorem 11:** We say that a binary vector is an even (odd) vector if it has an even (odd) weight. For two binary vectors $v_1, v_2$, $|v_2\backslash v_1|$ being odd is equivalent to

$$ 1 = v_2 \cdot v_1 = v_2 \cdot ((1, \ldots, 1) + v_1) = \|v_2\|_1 + v_2 \cdot v_1. $$

Hence, one can check that when $v_1, v_2$ have the same parity, there are either no edges or 2 edges between them. Moreover, when their parities are different, there is exactly one edge between the two vertices.

When $m = 1$, the graph $D_1$ has only one vertex and the only nonempty induced subgraph is itself. $d_H = d_{D_1} = 0 = \frac{m-1}{m}$.

When $m = 2$, the graph $D_2$ has three vertices and one can check that the induced subgraph with maximum density contains $v_1 = (1, 0), v_2 = (0, 1)$, and the density is $\frac{1}{2} = \frac{m-1}{m}$. For $m > 2$, assume to the contrary that there exists a subgraph of $D_m$ with density greater than $\frac{m-1}{m}$. Let $H$ be a subgraph of $D_m$ with minimum number of vertices among all subgraphs with maximal density. Hence for any subset of vertices $S \subseteq V(H)$, we have $d_S < d_H$. Therefore from Lemma 21 we conclude that for any nontrivial partition $S, T$ of $V(H)$, $d_{ST} \leq d_{S,T}$. If $H$ contains both even and odd vectors, denote by $S$ and $T$ the set of even and odd vectors of $H$ respectively. Since between any even and any odd vertex there is exactly one directed edge we get that $d_H \leq d_{S,T} = \frac{1}{2}$. However

$$ \frac{1}{2} < \frac{m-1}{m} < d_H, $$

and we get a contradiction. Thus $H$ contains only odd vectors or even vectors.

Let $V(H) = \{v_1, \ldots, v_k\}$. If this set of vectors is independent then $k \leq m$ and the outgoing degree for each vertex $v_i$ is at most $k - 1$ hence $d_H = \frac{|E(H)|}{|V(H)|^2} \leq \frac{k(k-1)}{k^2} \leq \frac{m-1}{m}$ and we get a contradiction. Hence assume that the dimension of the subspace spanned by these vectors in $\mathbb{F}_2^m$ is $i < k$ where $v_1, v_2, \ldots, v_i$ are basis for it. Define $S = \{v_1, \ldots, v_i\}, T = \{v_{i+1}, \ldots, v_k\}$. The following two cases show that the density cannot be higher than $\frac{m-1}{m}$.

**$H$ contains only odd vectors:** Let $u \in T$. Since $u \in \text{span}\{S\}$ there is at least one $v \in S$ such that $u \cdot v \neq 0$ and thus $(u, v), (v, u) \notin E(H)$, therefore the number of directed edges between $u$ and $S$ is at most $2(l - 1)$ for all $u \in T$, which means

$$ d_H \leq d_{S,T} \leq \frac{2(l - 1)|T|}{2|S||T|} = \frac{l - 1}{l} \leq \frac{m-1}{m} $$

and we get a contradiction.

**$H$ contains only even vectors:** Since the $v_i$’s are even the dimension of $\text{span}\{S\}$ is at most $m - 1$ (since for example $(1, 0, \ldots, 0) \notin \text{span}\{S\}$) thus $l \leq m - 1$. Let $H^*$ be the induced subgraph of $D_{m+1}$ with vertices $V(H^*) = \{(1, v_i)|v_i \in V(H)\}$. It is easy to see that all the vectors of $H^*$ are odd, $\{(1, v_i), (1, v_j)\} \in E(H^*)$ if and only if $(v_i, v_j) \in E(H)$, and the dimension of $\text{span}\{V(H^*)\}$ is at most $l + 1 \leq m$. Having already proven
the case for odd vectors, we conclude that
\[
d_H = d_{H^r} \leq \frac{\dim(\text{span}\{V(H^r)\}) - 1}{\dim(\text{span}\{V(H^r)\})}
\]
\[
\leq \frac{l + 1 - 1}{l + 1} = \frac{m - 1}{m},
\]
and we get a contradiction.

**APPENDIX C**

**PROOF OF THEOREM 18**

Next discuss the finite field of a code with three parities. The key idea of the proof is that if three erasures happen, we do not try to solve for all of the unknown elements at the same time, but solve a few linear equations at a time. No matter which columns are erased, we can always rearrange the ordering of the unknown elements and the equations, such that the coefficient matrix of the linear equations has a common format. Therefore, as long as this format is an invertible matrix, we know the code is MDS.

**Theorem 18** When \( r = 3 \), a field of size \( 4 \) is sufficient to make the code MDS using Construction 7.

**Proof:** We need to show we can rebuild three erasures, with \( x \) erasures of systematic nodes, and \( 3 - x \) of the parities, for \( x = 1, 2, 3 \). It is easy to see that when \( c \) is a nonzero coefficient, we can rebuild from one systematic and two parity erasures.

In case of two systematic erasures, suppose information columns \( i, j \) and parity column 2 are erased, \( 0 \leq i < j \leq k - 1 \). We will show that instead of solving equations involving all the unknown elements, we only need to solve 6 linear equations at a time. In order to recover the elements in row \( v \), consider the set of rows in the erased columns:
\[
W(v) = v + \text{span}\{e_i - e_j\}.
\]
We call \( v \) a starting point. \( W(v) \) contains 3 elements and altogether there are 6 unknown elements in the two columns \( i, j \). Notice that elements in rows \( W(v) \) and column \( i, j \) are mapped to elements in rows \( W(v) \) and parity 0. Also for parity 1 they are mapped to rows \( W(v) + e_i = W(v) + e_j \), which are equal because they are both cosets of \( \text{span}\{e_i - e_j\} \) and \( v + e_i \) is a member in both cosets. Therefore, by accessing rows \( W(v) \) in the surviving information nodes and parity 0, and rows \( W(v) + e_i \) in parity 1, we get 6 equations on these 6 unknowns.

For example, in Figure 7 columns \( C_1, C_2, C_3 \) are erased, then \( i = 1, j = 2 \). And consider the starting point \( v = (1, 0) \), which is 3 as an integer. Then \( W(v) = v + \text{span}\{e_2 - e_1\} = \{(1, 0), (2, 2), (0, 1)\} \), or \( \{3, 8, 1\} \) written as integers. Similarly, \( W(v) + e_j = W(v) + e_1 = \{4, 6, 2\} \) as integers. The 6 elements in rows \( W(v) \) in columns \( C_1, C_2 \) are \( \{a_1, a_2, a_3, a_3^2, a_3^2, a_2, a_2, a_2, a_2, a_2, a_2, a_2, a_2\} \). They are mapped to rows \( W(v) \) in parity 0 (column \( C_4 \)) and to rows \( W(v) + e_j \) in parity 1 (column \( C_5 \)). Therefore, we can solve for the 6 unknowns at a time.

Writing in matrix form, we need to solve the linear equations \( Gx = y \), where \( x \) is the \( 6 \times 1 \) unknown vector, \( y \) is a vector of size \( 6 \times 1 \), and \( G \) is a \( 6 \times 6 \) matrix. \( G \) can be written as
\[
\begin{pmatrix}
\text{info } i, W(v) & \text{info } j, W(v) \\
\text{parity } 0, W(v) & \text{parity } 1, W(v) + e_j \\
\end{pmatrix}
\]
and each submatrix here is of size \( 3 \times 3 \). The first 3 columns in \( G \) correspond to column \( i \), the last 3 columns correspond to column \( j \). The first 3 rows in \( G \) correspond to parity 0, the last 3 rows correspond to parity 1. We wrote the corresponding column and row indices at the top and on the left of the matrix. Since parity 0 is the row sum, the first 3 rows of \( G \) are two \( 3 \times 3 \) identity matrices.

Now we reorder the row and columns of \( G \) and show that \( \det(G) = \det(B - A) = 0 \). For \( t = 0, 1 \), order the elements in cosets \( W(v) + te_j \) as \( (v + te_j, v + te_j) + (e_i - e_j), (v + te_j) + 2(e_i - e_j) \). What are \( A \) and \( B \)? For row \( u \in W(v) \) in column \( i \), it is mapped to row \( u + e_i = (u + e_j) + (e_i - e_j) \) in parity 1. So \( A \) corresponds to a cyclic shift permutation. Suppose \( u = v + g(e_i - e_j) \), \( g = 0, 1, 2 \), then the coefficient is determined by
\[
u \sum_{i=1}^{j} e_i = v \sum_{i=1}^{j} e_i + g.
\]
According to (32), the coefficient is \( c \) if \( u \sum_{i=1}^{j} e_i = 0 \), and is 1 otherwise. For only one value of \( g \in \{0, 2\} \), the above expression is 0. Therefore we have
\[
A = \begin{bmatrix}
a_1 & a_1 \\
a_2 & a_3 \\
\end{bmatrix}
\]
with \( a_1 a_2 a_3 = c \). Similarly, row \( u \) in column \( j \) is mapped to \( u + e_j \) in parity 1. So \( B \) corresponds to diagonal matrix. And the coefficient is determined by
\[
u \sum_{i=1}^{j} e_i = v \sum_{i=1}^{j} e_i + g = v \sum_{i=1}^{i} e_i,
\]
which is a constant for \( W(v) \). Hence
\[
B = \begin{bmatrix}
b & b \\
b & b \\
\end{bmatrix}
\]
with \( b = 1 \) or \( c \). Now
\[
\det(G) = \det(B - A) = \det \begin{bmatrix}
b & -a_1 \\
-a_2 & b \\
-a_3 & b \\
\end{bmatrix}
\]
\[
= b^3 - a_1 a_2 a_3 = b^3 - c.
\]
The above value is \( 1 - c \) or \( c^3 - c \). If \( c \neq 0 \), and \( c^2 \neq 1 \), then \( \det(G) \neq 0 \). When \( c \) is a primitive element in \( GF(4) \), the above conditions are satisfied.

For example, if in Figure 7 we erase columns \( C_1, C_2, C_6 \) and take the staring point \( v = (1, 0) \), then \( W(v) \) is ordered
as $(3, 8, 1)$ and $W(v) + e_j$ is ordered as $(4, 6, 2)$. It is easy to check that

$$A = \begin{bmatrix} 1 & c \\ 1 & 1 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$ 

Similarly, if column $i, j$ and parity 0 are erased, we can show

$$\det(G) = \det(B - A)AB \neq 0.$$ 

When column $i, j$ and parity 1 are erased, we have

$$\det(G) = \det(B^2 - A^2)$$

$$= \det(b^2 - a_1a_3)$$

$$= \det(b^2 - a_1a_2 - a_2a_3)$$

$$= b^6 - (a_1a_2a_3)^2 = b^6 - c^2.$$ 

When $b = 1$ or $c$, the above value is $1 - c^2$ or $c^6 - c^2$. So we need $c \neq 0$, $c^2 \neq 1$, $c^4 \neq 1$. Again, for a finite field of size 4, these conditions are satisfied. Hence we can rebuild any two systematic and one parity erasures.

Suppose 3 systematic columns $i, j, l$ are erased, and $1 \leq i < j < l \leq k$. We will show that each time we need to solve 27 equations, and then reduce it to the case of two systematic erasures. In order to rebuild any row $v$ in these three columns, consider the following set of 9 rows (and therefore 27 unknown elements): 

$$V = v + \text{span}\{e_i - e_l, e_j - e_l\}.$$ 

These unknowns correspond to rows $V$ in parity 0. In parity 1, they correspond to rows $V + e_i = V + e_l$, which are equal to each other since they are cosets of span$\{e_i - e_l, e_j - e_l\}$ and $v + e_i$ is a member in all of them. Similarly, the unknowns correspond to rows $V + 2e_i = V + 2e_l = V + 2e_t$ in parity 2. Altogether we have 27 parity elements (equations).

Next we write these equations in a matrix form:

$$Gx = y,$$

where $G, y$ are coefficients, $x$ are unknowns. We are going to show that $\det(G) \neq 0$. Order the set span$\{e_i - e_l, e_j - e_l\}$ arbitrarily as $(v_0, v_1, \ldots, v_8)$. And order the coset $V + t e_l$ as $(v + t e_l + v_0, v + t e_l + v_1, \ldots, v + t e_l + v_8)$, for $t = 0, 1, 2$.

Now the coefficient matrix $G$ will be

$$\begin{array}{cccc}
\text{parity 0}, V & \text{info } i, V & \text{info } j, V & \text{info } l, V \\
\text{parity 1}, V + e_i & 1 & 1 & 1 \\
\text{parity 2}, V + 2e_l & A' & B' & C' \\
& A^2 & B^2 & C^2 \\
\end{array}$$

where each sub-block is of size $9 \times 9$. The first, second, and third block rows correspond to parity 0,1, and 2, respectively. And the first, second and third block columns correspond to erased column $i, j, l$ respectively. Since parity 0 is row sum, the first block rows contain identity submatrices. What is $C'$ for parity 1? By Construction 6, row $u$ in column $l$ corresponds to row $u + e_l$ in parity 1. So $C'$ should be diagonal. By (32) the values in $C$ are determined by $u \cdot \sum_{i=1}^{l} e_i$. And for some constants $g, h$, we have $u = v + g(e_i - e_l) + h(e_i - e_l)) \in V$, and thus $u \cdot \sum_{i=1}^{l} e_i = (v + g(e_i - e_l) + h(e_i - e_l)) \sum_{i=1}^{l} e_i = v \sum_{i=1}^{l} e_i + g - g + h = v \sum_{i=1}^{l} e_i$ is a constant for $V$. So $C' = I$ or $cI$.

for a primitive element $c$. Now notice that $C'$ is commutative with $A'$ and $B'$, we have $\det(G) = \det(B' - A') \det(C' - A') \det(C' - B')$ (without commutativity, this equation may not hold). Moreover, since $V$ is the union of $W(v), W(v + e_l - e_l), W(v + 2(e_l - e_l))$, and

$$\det(B' - A') = \det \begin{bmatrix} I \\ A' \\ B' \\ C' \end{bmatrix},$$

we know that $\det(B' - A')$ is simply the multiplication of three determinants in (34) with starting point $v, v + e_l - e_l, v + 2(e_l - e_l)$, which are always nonzero. Similarly, we can conclude that $\det(C' - A'), \det(C' - B')$ are also nonzero. Hence the code an correct any three erasures and is an MDS code.

REFERENCES


D. S. Papailiopoulos, A. G. Dimakis, and V. R. Cadambe, “Repair optimal erasure codes through hadamard designs”, in *ISIT*, 2011.


Itzhak Tamo was born in Israel in 1981. He received the B.A. and B.Sc. degrees in 2008 from the Mathematics Department and the Electrical and Computer Engineering Department, respectively, Ben-Gurion University, Israel. He is now a doctoral student with the Department of Electrical and Computer Engineering, Ben-Gurion University. His research interests include algebraic coding, combinatorial structures, and finite group theory.

Zhiying Wang received the B.Sc. degree in Information Electronics and Engineering from Tsinghua University, Beijing, China, in 2007 and M. Sc. degree in Electrical Engineering from California Institute of Technology, Pasadena, USA, in 2009. She is now a Ph.D. candidate with the department of Electrical Engineering, California Institute of Technology. Her research focuses on information theory, coding theory, with an emphasis on coding for storage devices and systems.

Jehoshua Bruck (S86-M89-SM93-F01) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1989.

He is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology, Pasadena, CA. His extensive industrial experience includes working with IBM Almaden Research Center, as well as cofounding and serving as the Chairman of Rainfinity, acquired by EMC in 2005; and XtremIO, acquired by EMC in 2012. His current research interests include information theory, coding theory, with an emphasis on coding for storage devices and systems.

Dr. Bruck is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award, and the IBM Outstanding Technical Achievement Award. His papers were recognized in journals and conferences, including winning the 2010 IEEE Communications Society Best Student Paper Award in Signal Processing and Coding for Data Storage for his paper on codes for limited-magnitude errors in flash memories, the 2009 IEEE Communications Society Best Paper Award in Signal Processing and Coding for Data Storage for his paper on rank modulation for flash memories, the 2005 A. Schellkunoff Transactions Prize Paper Award from the IEEE Antennas and Propagation Society for his paper on signal propagation in wireless networks, and the 2003 Best Paper Award in the 2003 Design Automation Conference for his paper on cyclic combinational circuits.