Erasure Coding in Distributed Storage Systems

ZIGZAG CODES: MDS ARRAY CODES WITH OPTIMAL REBUILDING

ILIA SMAGLOY & LIRON BRONFMAN
Introduction

Motivation:
Reading a lot of information takes a lot of time.

Usually we only lose a single disk.
Basic Definitions

- **MDS** is a code which with \( r \) redundancy symbols, is able to reconstruct the original information if no more than \( r \) symbols are erased.

- Upon losing a **single** systemic disk- the fraction of the accessed information in the surviving node is called the **rebuilding ratio**.
The rebuilding ratio for MDS codes with 2 redundancies was proven to be between $\frac{1}{2}$ and $\frac{3}{4}$. Previous codes like EVENODD and X-Code have a ratio of $\frac{3}{4} + o(1)$ for $r = 2$.

The paper presents a family of MDS codes that with $r$ redundancies have a rebuilding ratio of $\frac{1}{r}$. We’ll focus on the scenario where $r = 2$. 

Introduction
Introduction

Example:

\[ n = 5, \ r = 2, k = n - r = 3. \text{Disk 1 was erased.} \]

Instead of reading all half of the information, we would only have to read half.
Introduction

- We’ll be focusing on the case $r = 2$.

- Later on we’ll also talk about generalizing the code construction.
Notations

- For a subset $X \subseteq M$, $\bar{X} = M \setminus X$.

- For two vectors $v = (v_1, ..., v_n), u = (u_1, ..., u_n)$, $v \cdot u = \sum_{i=1}^{n} v_i \cdot u_1$.

- $|v \setminus u| = \sum_{i: v_i = 1, u_i = 0} 1$ the number of coordinates where $v$ has 1, and $u$ has a 0.
Construction of a 2 redundancy array

Let $A = (a_{i,j})$ be an information array of size $p \times k$ over a finite field $F$.

We’ll assume $p = 2^m$.

We’ll add two parity columns and obtain a $(n = k + 2, k)$ an MDS array of size $p \times n$. 
Construction of a 2 redundancy array

We’ll perform the redundancy calculations over $F_q$.

For some $q > 2$, some power of some prime. In this case, it would be simplest to take $q = 3$. 
Construction of a 2 redundancy array

**Definition:** \( R_l = \{ a_{l,i} | 0 \leq i \leq k - 1 \} \)

**Definition:** \( r_l = \sum_{a \in R_l} \alpha_a a \), where \( \{ \alpha_a \} \subset F_q \).

We’ll choose, \( \alpha_a \equiv 1 \).
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

$p$

$k + 2$

$C_3$ is the parity column in $F_3$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

$p$

$k + 2$

$C_3$ is the parity column in $F_3$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

$k + 2$

$C_3$ is the parity column in $F_3$
**Example of a 2 redundancy array**

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

$p$

$k + 2$

$c_3$ is the parity column in $F_3$
Example of a 2 redundancy array

\[ \begin{array}{cccc|cc|ccc} \hline C_0 & C_1 & C_2 & C_3 & C_4 \\ \hline 1 & 1 & 0 & 2 & ? \\ 0 & 1 & 0 & 1 & ? \\ 0 & 1 & 1 & 2 & ? \\ 1 & 1 & 1 & 0 & ? \\ \hline \end{array} \]

\[ p \]

\[ k + 2 \]

\[ C_3 \text{ is the parity column in } F_3 \]
Construction of a 2 redundancy array

- **Definition:** Let $v \in F_2^m$ be a binary vector of length $m$. We’ll define a *zigzag permutation*

  $f_v: \{0,2^m - 1\} \rightarrow \{0,2^m - 1\}$

  $f_v(x) = x + v$

- **Example:** $m=2$, $v = (1,0)$, $x = (1,1)$.

  $f_{(1,0)}((1,1)) = (1,1) + (1,0) = (0,1)$
Construction of a 2 redundancy array

- We can also define the same function to work over numbers in $[0, p - 1]$ by converting them to binary vectors.

- Example: $m=2, \nu = (1,0), x = 3$.
  \[
  f_{(1,0)}(3) = 3 + (1,0) = (1,1) + (1,0) = (0,1) = 1
  \]

- Example: The permutation $f_{(1,0)}$ maps the numbers 0,1,2,3 to 2,3,0,1 respectively.
Construction of a 2 redundancy array

- Let \( \{ f_0, f_1, \ldots, f_{k-1} \} \) be a set of zigzag permutations over \([0, p - 1]\).

**Definition:** \( Z_l = \{ a_{i,j} | f_j(i) = l \} \).

**Example:** If \( p = 4 \) and \( f_0 = f_{(0,0)}, \ f_1 = f_{(1,0)}, \ f_2 = f_{(0,1)} \) then:
- \( Z_0 = \{ a_{0,0}, a_{2,1}, a_{1,2} \} \)
- \( Z_1 = \{ a_{1,0}, a_{3,1}, a_{0,2} \} \)
- \( Z_2 = \{ a_{2,0}, a_{0,1}, a_{3,2} \} \)
- \( Z_3 = \{ a_{3,0}, a_{1,1}, a_{2,2} \} \)
Construction of a 2 redundancy array

- The second parity column, $C_{k+1} = (z_0, \ldots, z_{p-1})^T$ is the zigzag parity column, where each $z_i = \sum_{a \in Z_i} \beta_a a$, where $\{\beta_a\} \subset F$.

We’ll see an explanation for the coefficients shortly.
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is some linear combination over the elements in $Z_i$. 

$\begin{align*}
Z_0 &= \{a_{0,0}, a_{2,1}, a_{1,2}\} \\
Z_1 &= \{a_{1,0}, a_{3,1}, a_{0,2}\} \\
Z_2 &= \{a_{2,0}, a_{0,1}, a_{3,2}\} \\
Z_3 &= \{a_{3,0}, a_{1,1}, a_{2,2}\}
\end{align*}$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>$Z_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$Z_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$Z_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$Z_3$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is some linear combination over the elements in $Z_i$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$  
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$  
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$  
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is some linear combination over the elements in $Z_i$.
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is some linear combination over the elements in $Z_i$. 

Example:

\[
\begin{array}{c|c|c|c|c|c}
| z_0 & a_{0,0} & a_{2,1} & a_{1,2} |
| z_1 & a_{1,0} & a_{3,1} & a_{0,2} |
| z_2 & a_{2,0} & a_{0,1} & a_{3,2} |
| z_3 & a_{3,0} & a_{1,1} & a_{2,2} |
\end{array}
\]
Construction of a 2 redundancy array

- We’ll look at the standard binary basis vectors of length \( m = \log_2 p \). We’ll notate \( e_0 = (0,0, ..., 0) \), \( e_1 = (1,0, ..., 0) \), \( e_i = (0, ..., 0,1,0, ..., 0) \).

- We’ll define \( u_i \) to be a binary vector such that \( u_i = \sum_{j=1}^{i} e_i \), meaning that for some \( i \), \( u_i = (1,1, ..., 1,0, ..., 0) \).
Construction of a 2 redundancy array

- Now we’re ready to define $\beta_{i,j}$.

**Definition:** $\beta_{i,j} = 2^{i\cdot u_j}$ in some field $F_q$ for $q > 2$.

- In the case of $r = 2$, we’ll use $F_3$, which means that $\beta_{i,j} \in \{1,2\}$. 
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = \]

\[ \beta_{1,0} = \]

\[ \beta_{1,1} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = \]

\[ \beta_{1,0} = \]

\[ \beta_{1,1} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

In the previous example:

\[
\beta_{0,j} = 2^0 \cdot u_j = 2^0 =
\]

\[
\beta_{1,0} =
\]

\[
\beta_{1,1} =
\]

\[
\beta_{1,2} =
\]
Construction of a 2 redundancy array

- In the previous example:
  \[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]
  \[ \beta_{1,0} = \]
  \[ \beta_{1,1} = \]
  \[ \beta_{1,2} = \]
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = 2^{0 \cdot u_j} = 2^0 = 1 \]

\[ \beta_{1,0} = 2^{1 \cdot (0,0)} = \]

\[ \beta_{1,1} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

- In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = \]

\[ \beta_{1,1} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

- In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]

\[ \beta_{1,1} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = 2^{0 \cdot j} = 2^0 = 1 \]

\[ \beta_{1,0} = 2^{1 \cdot (0,0)} = 2^0 = 1 \]

\[ \beta_{1,1} = 2^{1 \cdot (1,0)} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

- In the previous example:

\[ \beta_{0,j} = 2^{0} \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^{1} \cdot (0,0) = 2^0 = 1 \]

\[ \beta_{1,1} = 2^{1} \cdot (1,0) = 2^{(0,1) \cdot (1,0)} = \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]

\[ \beta_{1,1} = 2^1 \cdot (1,0) = 2^{(0,1) \cdot (1,0)} = 1 \]

\[ \beta_{1,2} = \]
Construction of a 2 redundancy array

- In the previous example:
  \[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]
  \[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]
  \[ \beta_{1,1} = 2^1 \cdot (1,0) = 2^{(0,1) \cdot (1,0)} = 1 \]
  \[ \beta_{1,2} = 2^1 \cdot (1,1) = \]
Construction of a 2 redundancy array

In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]

\[ \beta_{1,1} = 2^1 \cdot (1,0) = 2^{(0,1) \cdot (1,0)} = 1 \]

\[ \beta_{1,2} = 2^1 \cdot (1,1) = 2^{(0,1) \cdot (1,1)} = \]
Construction of a 2 redundancy array

- In the previous example:
  \[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]
  \[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]
  \[ \beta_{1,1} = 2^1 \cdot (1,0) = 2^{(0,1)} \cdot (1,0) = 1 \]
  \[ \beta_{1,2} = 2^1 \cdot (1,1) = 2^{(0,1)} \cdot (1,1) = 2^1 = \]
Construction of a 2 redundancy array

- In the previous example:

\[ \beta_{0,j} = 2^0 \cdot u_j = 2^0 = 1 \]

\[ \beta_{1,0} = 2^1 \cdot (0,0) = 2^0 = 1 \]

\[ \beta_{1,1} = 2^1 \cdot (1,0) = 2^{(0,1) \cdot (1,0)} = 1 \]

\[ \beta_{1,2} = 2^1 \cdot (1,1) = 2^{(0,1) \cdot (1,1)} = 2^1 = 2 \]
Construction of a 2 redundancy array

- For the rest of the $\beta$’s:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Construction of a 2 redundancy array

- For the rest of the $\beta$’s:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$  \quad z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}$
Construction of a 2 redundancy array

- For the rest of the $\beta$’s:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\} \quad z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}
\]

\[
Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\} \quad z_1 = a_{1,0} + a_{0,2} + 2a_{3,1}
\]
Construction of a 2 redundancy array

For the rest of the $\beta$’s:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$  \hspace{1cm} $Z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}$

$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$  \hspace{1cm} $Z_1 = a_{1,0} + a_{0,2} + 2a_{3,1}$

$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$  \hspace{1cm} $Z_2 = a_{0,1} + a_{2,0} + a_{3,2}$
Construction of a 2 redundancy array

- For the rest of the $\beta$’s:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$  \[ z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2} \]

$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$  \[ z_1 = a_{1,0} + a_{0,2} + 2a_{3,1} \]

$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$  \[ z_2 = a_{0,1} + a_{2,0} + a_{3,2} \]

$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$  \[ z_3 = a_{3,0} + a_{1,1} + 2a_{2,2} \]
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>$a_{0,0} + 2a_{2,1} + 2a_{1,2}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$a_{1,0} + a_{0,2} + 2a_{3,1}$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$a_{0,1} + a_{2,0} + a_{3,2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_{3,0} + a_{1,1} + 2a_{2,2}$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is an element in $F_3$
Example of a 2 redundancy array

\[
Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}
\]
\[
Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}
\]
\[
Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}
\]
\[
Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}
\]

Each \(z_i\) is an element in \(F_3\)
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>$a_{0,1} + a_{2,0} + a_{3,2}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_{3,0} + a_{1,1} + 2a_{2,2}$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$

$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$

$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$

$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is an element in $F_3$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$a_{3,0} + a_{1,1} + 2a_{2,2}$</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is an element in $F_3$
Example of a 2 redundancy array

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$Z_0 = \{a_{0,0}, a_{2,1}, a_{1,2}\}$
$Z_1 = \{a_{1,0}, a_{3,1}, a_{0,2}\}$
$Z_2 = \{a_{2,0}, a_{0,1}, a_{3,2}\}$
$Z_3 = \{a_{3,0}, a_{1,1}, a_{2,2}\}$

Each $z_i$ is an element in $F_3$
Reconstruction

**Definition:** For $v \in F_2^m$ such that $v \neq 0$, we’ll define the set of integers whose binary representation is orthogonal to $v$, meaning:

$$X_v = \{x \in F_2^m | x \cdot v = 0\}$$

**Example:** For $m = 2$ and $v = (1,0)$, then $X_v = \{0,1\}$. This is because $0 = (0,0)$, $1 = (0,1)$ and upon multiplication

$$(0,0) \cdot (1,0) = 0 = (0,1) \cdot (1,0).$$

How many elements does it have?
Reconstruction

- For $\nu = (0, ..., 0)$ we define $X_\nu = \{x \in F_2^m | x \cdot (1,1,...,1) = 0\}$. 

- We’ll notate $X_j = X_{\nu_j}$ and $f_j = f_{\nu_j}$ because we’re lazy.

- **Definition:** $S_r = \{a_{i,j} | i \in X_j\}$ and $S_z = \{a_{i,j} | i \notin X_j\}$.

- **Example:** $m = 2$, $\nu_0 = (0,0)$, $\nu_1 = (1,0)$, $\nu_2 = (0,1)$, $\nu_3 = (1,1)$
Reconstruction

- \( v_0 = (0,0), \ v_1 = (1,0), \ v_2 = (0,1) \)

- \( X_0 = \{0,3\}, \ X_1 = \{0,1\}, \ X_2 = \{0,2\} \)

- \( S_r = \{a_{0,0}, a_{3,0}, a_{0,1}, a_{1,1}, a_{0,2}, a_{2,2}\} \)

- \( S_z = \{a_{1,0}, a_{2,0}, a_{2,1}, a_{3,1}, a_{1,2}, a_{3,2}\} \)

<table>
<thead>
<tr>
<th>( c_0 )</th>
<th>( c_1 )</th>
<th>( c_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_{0,0} = 1 )</td>
<td>( a_{0,1} = 1 )</td>
<td>( a_{0,2} = 0 )</td>
</tr>
<tr>
<td>( a_{1,0} = 0 )</td>
<td>( a_{1,1} = 1 )</td>
<td>( a_{1,2} = 0 )</td>
</tr>
<tr>
<td>( a_{2,0} = 0 )</td>
<td>( a_{2,1} = 1 )</td>
<td>( a_{2,2} = 1 )</td>
</tr>
<tr>
<td>( a_{3,0} = 1 )</td>
<td>( a_{3,1} = 1 )</td>
<td>( a_{3,2} = 1 )</td>
</tr>
</tbody>
</table>
Reconstruction

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td>$a_{0,1} = 1$</td>
<td>$a_{0,2} = 0$</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>$a_{1,0} = 0$</td>
<td>$a_{1,1} = 1$</td>
<td>$a_{1,2} = 0$</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>$a_{2,0} = 0$</td>
<td>$a_{2,1} = 1$</td>
<td>$a_{2,2} = 1$</td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>$a_{3,0} = 1$</td>
<td>$a_{3,1} = 1$</td>
<td>$a_{3,2} = 1$</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

- Suppose $C_1$ got erased.
Reconstruction

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td></td>
<td></td>
<td>$a_{0,2} = 0$</td>
<td>2</td>
</tr>
<tr>
<td>$a_{1,0} = 0$</td>
<td></td>
<td></td>
<td>$a_{1,2} = 0$</td>
<td>1</td>
</tr>
<tr>
<td>$a_{2,0} = 0$</td>
<td></td>
<td>$a_{3,1} = 1$</td>
<td>$a_{2,2} = 1$</td>
<td>2</td>
</tr>
<tr>
<td>$a_{3,0} = 1$</td>
<td></td>
<td>$a_{3,1} = 1$</td>
<td>$a_{3,2} = 1$</td>
<td>0</td>
</tr>
</tbody>
</table>

- Suppose $C_1$ got erased.
Reconstruction

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td>X</td>
<td>$a_{0,2} = 0$</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>$a_{1,0} = 0$</td>
<td>o</td>
<td>$a_{1,2} = 0$</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>$a_{2,0} = 0$</td>
<td></td>
<td>$a_{2,2} = 1$</td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>$a_{3,0} = 1$</td>
<td></td>
<td>$a_{3,2} = 1$</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

- Suppose $C_1$ got erased.
- The values of $a_{0,1}$ and $a_{1,1}$ can be retrieved by looking at $r_0$ and $r_1$. 
Reconstruction

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td>$a_{1,0} = 0$</td>
<td>$a_{2,0} = 0$</td>
<td>$a_{3,0} = 1$</td>
<td></td>
</tr>
<tr>
<td>$a_{0,1} = 0$</td>
<td>$a_{1,1} = a_{3,1} = 1$</td>
<td>$a_{2,1} = a_{2,2} = 1$</td>
<td>$a_{3,2} = 1$</td>
<td></td>
</tr>
<tr>
<td>$a_{0,2} = 0$</td>
<td>$a_{1,2} = 0$</td>
<td>$a_{2,2} = 1$</td>
<td>$a_{3,2} = 1$</td>
<td></td>
</tr>
<tr>
<td>$2$</td>
<td>$z_0$</td>
<td>$1$</td>
<td>$z_1$</td>
<td></td>
</tr>
<tr>
<td>$2$</td>
<td>$z_2$</td>
<td>$1$</td>
<td>$z_3$</td>
<td></td>
</tr>
</tbody>
</table>

- Suppose $C_1$ got erased.
- The values of $a_{0,1}$ and $a_{1,1}$ can be retrieved by looking at $r_0$ and $r_1$.
- The values of $a_{2,1}$ and $a_{3,1}$ can be retrieved by looking at $z_0$ and $z_1$. 
Reconstruction

- The values of $a_{2,1}$ and $a_{3,1}$ can be retrieved by looking at $z_0$ and $z_1$.

This is because $f_1(2) = f_{v_1}(2) = 0$, meaning that for $a_{2,1}$ we’ll use $z_0$.

Similarly, $f_1(3) = f_{v_1}(3) = 1$, so for $a_{3,1}$ we’ll use $z_1$.

- As we’ve seen earlier,
  
  $zf_1(2) = z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}$
  
  $zf_1(3) = z_1 = a_{1,0} + 2a_{3,1} + a_{0,2}$.
Reconstruction

- This means that we need to solve the following equation to recover the missing values:

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

<table>
<thead>
<tr>
<th>(c_0)</th>
<th>(c_1)</th>
<th>(c_2)</th>
<th>(c_3)</th>
<th>(c_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_{0,0} = 1)</td>
<td>(a_{0,1} = 1)</td>
<td>(a_{0,2} = 0)</td>
<td>2</td>
<td>(z_0)</td>
</tr>
<tr>
<td>(a_{1,0} = 0)</td>
<td>(a_{1,1} = 1)</td>
<td>(a_{1,2} = 0)</td>
<td>1</td>
<td>(z_1)</td>
</tr>
<tr>
<td>(a_{2,0} = 0)</td>
<td></td>
<td>(a_{2,2} = 1)</td>
<td>2</td>
<td>(z_2)</td>
</tr>
<tr>
<td>(a_{3,0} = 1)</td>
<td>(a_{3,1} = 1)</td>
<td>(a_{3,2} = 1)</td>
<td>0</td>
<td>(z_3)</td>
</tr>
</tbody>
</table>

\[ r_0 = a_{0,0} + a_{0,1} + a_{0,2} \]
\[ r_1 = a_{1,0} + a_{1,1} + a_{1,2} \]
\[ z_{f_1(2)} = z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2} \]
\[ z_{f_1(3)} = z_1 = a_{1,0} + 2a_{3,1} + a_{0,2}. \]
Reconstruction

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td>$+1$</td>
<td>$a_{0,2} = 0$</td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>$a_{1,0} = 0$</td>
<td>$a_{1,1} = 0$</td>
<td>$a_{1,2} = 0$</td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>$a_{2,0} = 0$</td>
<td>$a_{2,2} = 1$</td>
<td>2</td>
<td>$z_2$</td>
<td></td>
</tr>
<tr>
<td>$a_{3,0} = 1$</td>
<td>$a_{3,1} = 0$</td>
<td>$a_{3,2} = 1$</td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

- This is only half of the remaining information!

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

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{0,0} = 1$</td>
<td></td>
<td></td>
<td>2</td>
<td>$z_0$</td>
</tr>
<tr>
<td>$a_{1,0} = 0$</td>
<td></td>
<td></td>
<td>1</td>
<td>$z_1$</td>
</tr>
<tr>
<td>$a_{2,0} = 0$</td>
<td></td>
<td></td>
<td>2</td>
<td>$z_2$</td>
</tr>
<tr>
<td>$a_{3,0} = 1$</td>
<td></td>
<td></td>
<td>0</td>
<td>$z_3$</td>
</tr>
</tbody>
</table>

This is only half of the remaining information!

We’ll see a proof of why this is true:

\[
\begin{align*}
    r_0 & = a_{0,0} + a_{0,1} + a_{0,2} \\
    r_1 & = a_{1,0} + a_{1,1} + a_{1,2} \\
    z_{f_1(2)} & = z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2} \\
    z_{f_1(3)} & = z_1 = a_{1,0} + 2a_{3,1} + a_{0,2}.
\end{align*}
\]
Lemma: Let $T \subset F^m$ be a set of vectors. We build $f_0, \ldots, f_m$ and $X_0, \ldots, X_m$ accordingly.

A) For any $u, v \in T$, to rebuild $v$ the number of accessed elements in node $u$ is $2^{m-1} + |f_v(X_v) \cap f_u(X_u)|$

B) if $v \neq 0$, then $|f_v(X_v) \cap f_u(X_v)| = \begin{cases} |X_v|, & |v \setminus u| = 0 \mod 2 \\ 0, & \text{otherwise} \end{cases}$
Rebuilding Ratio

A) For any $u, v \in T$, to rebuild $v$ the number of accessed elements in node $u$ is $2^{m-1} + |f_v(X_v) \cap f_u(X_v)|$

Proof:

We first access the rows such that $r \in X_v$. There are $2^{m-1}$ such rows, so $2^{m-1}$ nodes are accessed.

So, how many more?
Rebuilding Ratio

The remaining rows ($r \notin X_v$) we reconstruct with the Zig Zag redundancy. Which nodes do we need for that? Which new nodes do we need?

$$f_u^{-1}(f_v(X_v)) \setminus X_v$$
Rebuilding Ratio

So, we need to count:

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

So, now we have:

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

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

\[ = |f_u^{-1}(f_v(X_v)) \cap X_v| = |(f_v(X_v)) \cap f_u(X_v)| \]
Rebuilding Ratio

So, for now we know that in order to reconstruct \( v \) we need to access \( 2^{m-1} + |f_v(X_v) \cap f_u(X_v)| \) nodes from \( u \). But how many is that?

**Lemma:** B) if \( v \neq 0 \), then \(|f_v(X_v) \cap f_u(X_v)| = \begin{cases} |X_v|, & v \backslash u = 0 \mod 2 \\ 0, & \text{otherwise} \end{cases} \)
Rebuilding Ratio

B) if \( v \neq 0 \), then \( |f_v(X_v) \cap f_u(X_v)| = \begin{cases} |X_v|, & |v \setminus u| = 0 \mod 2 \\ 0, & \text{otherwise} \end{cases} \)

Proof:

Let’s notice that \( f_v(X_v) \) and \( f_u(X_v) \) are both co-sets of \( X_v \). Why?

**Definition:** Let \( v \in F_2^m \) be a binary vector of length \( m \). We’ll define a zigzag permutation

\[
f_v: [0, 2^m - 1] \rightarrow [0, 2^m - 1]
\]

\[
f_v(x) = x + v
\]
Rebuilding Ratio

So, they’re either the same co-set or two disjoint co-sets. When are the co-sets $v + X_v$ and $u + X_v$ the same?

When $v - u \in X_v$ \iff $(v - u) \cdot v = 2 \ 0$ \iff $\sum_{i=0}^{p-1} (v_i - u_i) \cdot v_i = 2 \ 0$

\iff $\sum_{i \ s.t. \ v_i = 1} (1 - u_i) = 2 \ 0$ \iff $\sum_{i \ s.t. \ v_i = 1, u_i = 0} 1 = 2 \ 0$ \iff $|v \setminus u| = 2 \ 0$
Rebuilding Ratio

So, we got that $f_v(X_v)$ and $f_u(X_v)$ are the same co-set iff $|v \setminus u| = 2 \mod 0$. Otherwise, they are disjoint co-sets. So we get,

$$|f_v(X_v) \cap f_u(X_v)| = \begin{cases} |X_v|, & |v \setminus u| = 0 \mod 2, \\ 0, & \text{otherwise} \end{cases},$$

like we wanted.

End of the lemma proof.
Rebuilding Ratio

Now, to the important part;

**Theorem:** We construct permutations $f_0, \ldots, f_m$ and sets $X_0, \ldots, X_m$ by the standard base vectors \( \{e_i\}_{i=1}^m \) of the binary field $F_2^m$ and $e_0 = (0, \ldots, 0)$ like we said. Then, the corresponding $(m + 3, m + 1)$ code has optimal rebuilding ratio of 0.5.

**Proof:** we can see that for each $i, j < m + 1$ $|e_i \setminus e_j| = 1$ and from the lemma we know that this means that for every $i$ we need to access $2^{m-1}$ nodes from each column $j$ s. t. $j \neq i$.

We can also see that we use only half of the redundancy columns.

In conclusion, in this construction we’ll always get 0.5 rebuilding ratio!
Now, we need to prove that this construction is MDS.

*Theorem:* Let \( u_i = \sum_{j=1}^{i} e_i \). \( \beta_{i,j} = 2^{i-u_j} \) in some field \( F_q \) for \( q > 2 \). The construction explained above is *MDS*. 
MDS property

- **Theorem:** Let $u_i = \sum_{j=1}^{i} e_i$. $\beta_{i,j} = 2^{i \cdot u_j}$ in some field $F_q$ for $q > 2$. The construction explained above is MDS.

- **Proof:** We’ll present the proof for 2 redundancies.
Let’s assume that disks $i, j$ fell, and that w.l.o.g $0 \leq i < j \leq m + 1$. There can be four different scenarios:
MDS property

- Let’s assume that disks $i, j$ fell, and that w.l.o.g $0 \leq i < j \leq m + 1$. There can be four different scenarios:

- A) $i = m, j = m + 1$. 

```
0 1 2    ...    m-1  m  m+1
```
MDS property

- Let’s assume that disks $i, j$ fell, and that w.l.o.g $0 \leq i < j \leq m + 1$. There can be four different scenarios:

- A) $i = m, j = m + 1$. 

![Diagram showing disks and their positions](image)
Let’s assume that disks $i, j$ fell, and that w.l.o.g $0 \leq i < j \leq m + 1$. There can be four different scenarios:

- **A) $i = m, j = m + 1$.**

We have access to all the systematic disks, so we’ll encode disks $i$ and $j$. 
MDS property

- Let’s assume that disks \( i, j \) fell, and that w.l.o.g \( 0 \leq i < j \leq m + 1 \). There can be four different scenarios:

- A) \( i = m, j = m + 1 \).

We have access to all the systematic disks, so we’ll encode disks \( i \) and \( j \).
MDS property

- B) $i < m, j = m + 1$. 
MDS property

- B) \( i < m, j = m + 1 \).
MDS property

- B) $i < m, j = m + 1$.

We still have access to the parity disk, so we’ll restore disk $i$.
MDS property

- B) \( i < m, j = m + 1 \).

We still have access to the parity disk, so we’ll restore disk \( i \).
MDS property

- B) $i < m$, $j = m + 1$.

We still have access to the parity disk, so we’ll restore disk $i$.

Now we have access to all systemic disks, so we’ll restore disk $j$. 
MDS property

- B) \( i < m, j = m + 1 \).

We still have access to the parity disk, so we’ll restore disk \( i \).

Now we have access to all systemic disks, so we’ll restore disk \( j \).
MDS property

- C) $i < m$, $j = m$. 
MDS property

- C) $i < m$, $j = m$. 
MDS property

- C) $i < m$, $j = m$.

We have access to the zigzag disk, so we’ll restore disk $i$. 
MDS property

- C) \( i < m, j = m \).

We have access to the zigzag disk, so we’ll restore disk \( i \).
MDS property

- C) \( i < m, j = m \).

We have access to the zigzag disk, so we’ll restore disk \( i \).

Now we have access to all systemic disks, so we’ll restore disk \( j \).
MDS property

- C) $i < m$, $j = m$.

We have access to the zigzag disk, so we’ll restore disk $i$.

Now we have access to all systemic disks, so we’ll restore disk $j$. 
MDS property

- D) $i, j < m$. 

![Diagram showing a sequence of columns and a gap at m and m+1](image-url)
MDS property

- D) $i, j < m$. 

![Diagram showing MDS property with cylinders and X symbols]
MDS property

- D) $i, j < m$.

We’ll go by each row $0 \leq r \leq p - 1$ and fix the erased values $a_{r,i}$ and $a_{r,j}$.
MDS property

- D) $i, j < m$.

We’ll go by each row $0 \leq r \leq p - 1$ and fix the erased values $a_{r,i}$ and $a_{r,j}$.

For each $r$, define $r' = r + e_i + e_j$. 
MDS property

- D) \( i, j < m \).

We’ll go by each row \( 0 \leq r \leq p - 1 \) and fix the erased values \( a_{r,i} \) and \( a_{r,j} \).

For each \( r \), define \( r' = r + e_i + e_j \).

Using this method we also recover \( a_{r',i}, a_{r',j} \).
MDS property

According to our construction we know that $a_{r,i}, a_{r,j} \in P_r$ and $a_{r',i}, a_{r',j} \in P_{r'}$. 
MDS property

- According to our construction we know that $a_{r,i}, a_{r,j} \in P_r$ and $a_{r',i}, a_{r',j} \in P_{r'}$.

- According to the definition $Z_l = \{ a_{i,j} | f_j(i) = l \}$, we know that $a_{r,i}, a_{r',j} \in Z_{r+e_i}$ and $a_{r,j}, a_{r',i} \in Z_{r+e_j}$.
MDS property

- According to our construction we know that $a_{r,i}, a_{r,j} \in P_r$ and $a_{r',i}, a_{r',j} \in P_{r'}$.

- According to the definition $Z_l = \{ a_{i,j} | f_j(i) = l \}$, we know that $a_{r,i}, a_{r',j} \in Z_{r+e_i}$ and $a_{r,j}, a_{r',i} \in Z_{r+e_j}$.

- Therefore, the only affected redundant symbols are $p_r, p_{r'}, z_{r+e_i}, z_{r+e_j}$.
Since the parity and zigzag disks are still accessible, we’ll get the values of $p_r^{\text{old}}, p_{r'}^{\text{old}}, z^{\text{old}}_{r+e_i}, z^{\text{old}}_{r+e_j}$. 
MDS property

- Since the parity and zigzag disks are still accessible, we’ll get the values of $p_r^{\text{old}}, p_r'^{\text{old}}, z_{r+e_i}^{\text{old}}, z_{r+e_j}^{\text{old}}$.

- We’ll calculate $p_r^{\text{new}}, p_r'^{\text{new}}, z_{r+e_i}^{\text{new}}, z_{r+e_j}^{\text{new}}$ with the remaining systematic symbols.
MDS property

- Since the parity and zigzag disks are still accessible, we’ll get the values of $p_r^{old}, p_r'^{old}, z_{r+e_i}^{old}, z_{r+e_j}^{old}$.

- We’ll calculate $p_r^{new}, p_r'^{new}, z_{r+e_i}^{new}, z_{r+e_j}^{new}$ with the remaining systematic symbols.

- Define:

  $$
  \begin{align*}
  \Delta_1 &= p_r^{old} - p_r^{new} \\
  \Delta_2 &= p_r'^{old} - p_r'^{new} \\
  \Delta_3 &= z_{r+e_i}^{old} - z_{r+e_i}^{new} \\
  \Delta_4 &= z_{r+e_j}^{old} - z_{r+e_j}^{new}
  \end{align*}
  $$
MDS property

- $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ are the changes between the previous parity and zigzag symbols.
MDS property

- $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ are the changes between the previous parity and zigzag symbols.

- They were only affected by the changes of $a_{r,i}, a_{r,j}, a_{r',i}, a_{r',j}$. 
MDS property

- $\Delta_1, \Delta_2, \Delta_3, \Delta_4$ are the changes between the previous parity and zigzag symbols.

- They were only affected by the changes of $a_{r,i}, a_{r,j}, a_{r',i}, a_{r',j}$.

- Therefore, we need to solve the following system:
MDS property

\[
\begin{align*}
\begin{cases}
    a_{r,i} + a_{r,j} &= \Delta_1 \\
    a_{r',i} + a_{r',j} &= \Delta_2 \\
    \beta_{r,i} a_{r,i} + \beta_{r',j} a_{r',j} &= \Delta_3 \\
    \beta_{r,j} a_{r,j} + \beta_{r',i} a_{r',i} &= \Delta_4
\end{cases}
\end{align*}
\]

\[
\begin{bmatrix}
    1 & 1 & 0 & 0 \\
    0 & 0 & 1 & 1 \\
    \beta_{r,i} & 0 & 0 & \beta_{r',j} \\
    0 & \beta_{r,j} & \beta_{r',i} & 0
\end{bmatrix}
\begin{bmatrix}
    a_{r,i} \\
    a_{r,j} \\
    a_{r',i} \\
    a_{r',j}
\end{bmatrix}
=
\begin{bmatrix}
    \Delta_1 \\
    \Delta_2 \\
    \Delta_3 \\
    \Delta_4
\end{bmatrix}
\]
MDS property

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\beta_{r,i} & 0 & 0 & \beta_{r',j} \\
0 & \beta_{r,j} & \beta_{r',i} & 0
\end{pmatrix}
\begin{pmatrix}
a_{r,i} \\
a_{r,j} \\
a_{r',i} \\
a_{r',j}
\end{pmatrix}
= 
\begin{pmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3 \\
\Delta_4
\end{pmatrix}
\]

This system only has a single solution when \( \frac{\beta_{r,i}}{\beta_{r',j}} \neq \frac{\beta_{r,j}}{\beta_{r',i}} \), meaning that

\[
\beta_{r,i} \beta_{r',i} \neq \beta_{r,j} \beta_{r',j}
\]
MDS property

- Meaning, that in order to restore the original values of rows $r$ and $r'$ we need to solve the system previously mentioned.
MDS property

- Meaning, that in order to restore the original values of rows $r$ and $r'$ we need to solve the system previously mentioned.

- This can be done iff $\beta_{r,i}\beta_{r',i} \neq \beta_{r,j}\beta_{r',j}$. 
MDS property

- Meaning, that in order to restore the original values of rows $r$ and $r'$ we need to solve the system previously mentioned.

- This can be done iff $\beta_{r,i}\beta_{r',i} \neq \beta_{r,j}\beta_{r',j}$.

- But according to the construction we gave, and because $i < j$

\[
\begin{align*}
\beta_{r,i}\beta_{r',i} &= 2^r \cdot u_i + r' \cdot u_i = 2^{(r+r')}u_i = 2^{(e_i+e_j) \cdot \sum_{k=1}^{i} e_k} = 2^{e_i \cdot e_i} = 2^1 = 2 \\
\beta_{r,j}\beta_{r',j} &= 2^r \cdot u_j + r' \cdot u_j = 2^{(r+r')}u_j = 2^{(e_i+e_j) \cdot \sum_{k=1}^{j} e_k} = 2^{e_i \cdot e_i + e_j \cdot e_j} = 2^0 = 1
\end{align*}
\]
MDS property

2 \neq 1
MDS property

\[ 2 \neq 1 \]

- Therefore, the matrix is solvable.
MDS property

\[ 2 \neq 1 \]

- Therefore, the matrix is solvable.

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\beta_{r,i} & 0 & 0 & \beta'_{r,j} \\
0 & \beta_{r,j} & \beta'_{r',i} & 0
\end{pmatrix}
\begin{pmatrix}
ar_{r,i} \\
ar_{r,j} \\
ar'_{r,i} \\
ar'_{r,j}
\end{pmatrix}
= 
\begin{pmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3 \\
\Delta_4
\end{pmatrix}
\]
MDS property

2 \neq 1

- Therefore, the matrix is solvable.
- Solve it and get the values of $a_{r,i}, a_{r,j}, a'_{r,i}, a'_{r,j}$.

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\beta_{r,i} & 0 & 0 & \beta'_{r,j} \\
0 & \beta_{r,j} & \beta'_{r,i} & 0
\end{bmatrix}
\begin{bmatrix}
a_{r,i} \\
a_{r,j} \\
a'_{r,i} \\
a'_{r,j}
\end{bmatrix}
= 
\begin{bmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3 \\
\Delta_4
\end{bmatrix}
\]
MDS property

- Repeat the process for every row $r$. 
MDS property

- Repeat the process for every row $r$.

- This way we’ll have recovered the values of disks $i$ and $j$. 
MDS property

- Repeat the process for every row $r$.

- This way we’ll have recovered the values of disks $i$ and $j$.

- This proves that we can recover from any two disk failures, proving that the construction is indeed a $(m + 2, m)$ MDS code.
MDS property

It’s worth noting is that the equations we’ve seen before will always yield results in \{0,1\}.
What software engineers need to know

- Recovering 2 systemic erasures is done by going through each row $r$, calculating $r'$, finding the corresponding $\Delta_i$'s and $\beta$'s, and solving the following system:

$$
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\beta_{r,i} & 0 & 0 & \beta_{r',j} \\
0 & \beta_{r,j} & \beta_{r',i} & 0
\end{pmatrix}
\begin{pmatrix}
a_{r,i} \\
a_{r,j} \\
a_{r',i} \\
a_{r',j}
\end{pmatrix} =
\begin{pmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3 \\
\Delta_4
\end{pmatrix}
$$
MDS example

- Let’s look at the example we had earlier:
MDS example

*Let’s look at the example we had earlier:*

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
## MDS example

Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
    z_0 &= a_{0,0} + 2a_{2,1} + 2a_{1,2} \\
    z_1 &= a_{1,0} + a_{0,2} + 2a_{3,1} \\
    z_2 &= a_{0,1} + a_{2,0} + a_{3,2} \\
    z_3 &= a_{3,0} + a_{1,1} + 2a_{2,2}
\end{align*}
\]
MDS example

Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Assume that disks 1 and 2 were erased.

$z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}$

$z_1 = a_{1,0} + a_{0,2} + 2a_{3,1}$

$z_2 = a_{0,1} + a_{2,0} + a_{3,2}$

$z_3 = a_{3,0} + a_{1,1} + 2a_{2,2}$
Let’s look at the example we had earlier:

Assume that disks 1 and 2 were erased.

\[
\begin{array}{c|c|c|c|c|c}
 C_0 & C_1 & C_2 & C_3 = p & C_4 = z \\
\hline
 1 & 0 & 0 & 2 & 0 \\
 0 & 1 & 0 & 1 & 2 \\
 0 & 1 & 1 & 2 & 2 \\
 1 & 1 & 1 & 0 & 1 \\
\end{array}
\]

\[
\begin{align*}
 z_0 &= a_{0,0} + 2a_{2,1} + 2a_{1,2} \\
 z_1 &= a_{1,0} + a_{0,2} + 2a_{3,1} \\
 z_2 &= a_{0,1} + a_{2,0} + a_{3,2} \\
 z_3 &= a_{3,0} + a_{1,1} + 2a_{2,2}
\end{align*}
\]
Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✗</td>
<td>✗</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>✗</td>
<td>✗</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>✗</td>
<td>✗</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Assume that disks 1 and 2 were erased.

Start with $r = 0$, meaning $r' = 3$.

$z_0 = a_{0,0} + 2a_{2,1} + 2a_{1,2}$

$z_1 = a_{1,0} + a_{0,2} + 2a_{3,1}$

$z_2 = a_{0,1} + a_{2,0} + a_{3,2}$

$z_3 = a_{3,0} + a_{1,1} + 2a_{2,2}$
MDS example

- \( p_0^{\text{old}} = 2, p_3^{\text{old}} = 0, z_2^{\text{old}} = 2, z_1^{\text{old}} = 2. \)
MDS example

- $p_0^{old} = 2$, $p_3^{old} = 0$, $z_2^{old} = 2$, $z_1^{old} = 2$.
- $p_0^{new} = 1$, $p_3^{new} = 1$, $z_2^{new} = 0$, $z_1^{new} = 0$. 
MDS example

- $p_0^{old} = 2, p_3^{old} = 0, z_2^{old} = 2, z_1^{old} = 2.$

- $p_0^{new} = 1, p_3^{new} = 1, z_2^{new} = 0, z_1^{new} = 0.$

- $\Delta_1 = 1, \Delta_2 = 2, \Delta_3 = 2, \Delta_4 = 2.$
MDS example

- $p_0^{old} = 2, p_3^{old} = 0, z_2^{old} = 2, z_1^{old} = 2.$

- $p_0^{new} = 1, p_3^{new} = 1, z_2^{new} = 0, z_1^{new} = 0.$

- $\Delta_1 = 1, \Delta_2 = 2, \Delta_3 = 2, \Delta_4 = 2.$

- $\beta_{0,1} = 1, \beta_{0,2} = 1, \beta_{3,1} = 2, \beta_{3,2} = 1.$
Therefore, we need to solve the system represented by the following product:

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 \\
0 & 1 & 2 & 0 \\
\end{bmatrix}
\begin{bmatrix}
a_{0,1} \\
a_{0,2} \\
a_{3,1} \\
a_{3,2} \\
\end{bmatrix}
= \begin{bmatrix}
1 \\
2 \\
2 \\
2 \\
\end{bmatrix}
\]
MDS example

Therefore, we need to solve the system represented by the following product:

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 \\
0 & 1 & 2 & 0
\end{pmatrix}
\begin{pmatrix}
a_{0,1} \\
a_{0,2} \\
a_{3,1} \\
a_{3,2}
\end{pmatrix}
= 
\begin{pmatrix}
1 \\
2 \\
2 \\
2
\end{pmatrix}
\]

The result is:

\[a_{0,1} = 1, a_{0,2} = 0, a_{3,1} = 1, a_{3,2} = 1\]
Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>X</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
MDS example

Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3 = p$</th>
<th>$C_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$a_{0,1} = 1, a_{0,2} = 0, a_{3,1} = 1, a_{3,2} = 1$
Let’s look at the example we had earlier:

<table>
<thead>
<tr>
<th>$c_0$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3 = p$</th>
<th>$c_4 = z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ a_{0,1} = 1, a_{0,2} = 0, a_{3,1} = 1, a_{3,2} = 1 \]
MDS example – very nice

It worked!
MDS example – very nice
Lengthening The Code

**Problem:** currently, for an info array with $p = 2^m$ rows we can only make an MDS code with out construction of length $(m + 3, m + 1)$ at most.

Solution: Duplication and Constant Weight Vector.
Duplication

Construction: For the construction we discussed before, we construct its $s$ — duplication the same, but this time the permutation set is a multiset

$$F = \{f_0, \ldots, f_{k-1}, f_0, \ldots, f_{k-1}, \ldots, f_0, \ldots, f_{k-1}\}$$

where each permutation has multiplicity of exactly $s$.

Property: For the construction we have shown earlier, the asymptotic ratio is $\frac{1}{2} \cdot \left(1 + \frac{1}{m}\right)$. 
Duplication

But the question stands – is it MDS?

Answer: We’ll need to expand our field of operations to $F_q, q \geq s + 1 + 1_{q\text{even}}$, with $a$ a primitive element. Then, in C’ the coefficients will be as follows:

$$\forall i, j \in [0, k - 1], t \in [0, s - 1] \alpha_{i,j}^{(t)} = 1, \beta_{i,j}^{(t)} = \begin{cases} a^{(t+1)(1-2\cdot1_q)}, u_j \cdot i = 0 \\ a^{t+1_q}, \text{else} \end{cases}$$
**Duplication**

**Theorem:** The \((s(m + 1), +2 s(m + 1))\) code is an MDS code.

**Proof:** Let’s check the different scenarios:

- The erased disks are \(i^{(t_1)}, i^{(t_2)}\).
  
  For the row \(r\), we have the same parity and same Zigzag,
  
  \[ \begin{bmatrix} 1 \\ \beta_{r,i}^{t_1} \\ 1 \\ \beta_{r,i}^{t_2} \end{bmatrix} \]
  
  so we need the matrix \( \begin{bmatrix} 1 \\ \beta_{r,i}^{t_1} \\ 1 \\ \beta_{r,i}^{t_2} \end{bmatrix} \) to be of full rank.

- The erased disks are \(i^{(t_1)}, j^{(t_2)}\). For \(0 \leq t_1, t_2 \leq s - 1\), \(0 \leq i < j \leq m - 1\). For the row \(r\), we’ll construct \(r' = r + e_i + e_j\) like before.
Duplication

But, like before, the second scenario requires \( \beta_{r,i}^{t_1} \cdot \beta_{r',i}^{t_1} \neq \beta_{r,j}^{t_2} \cdot \beta_{r',j}^{t_2} \) for it to have exactly one solution.

We can calculate and see that \( \beta_{r,i}^{t_1} \neq \beta_{r,i}^{t_1}, \beta_{r,j}^{t_2} = \beta_{r,j}^{t_2} \).

So, we understand that \( \beta_{r,i}^{t_1}, \beta_{r',i}^{t_1} \) are one of each option. So,
\[
\beta_{r,i}^{t_1} \cdot \beta_{r',i}^{t_1} = a^{(t_1+1)(1-2\cdot q)} \cdot a^{t_1+1} q \\
= a^{(2t_1+1)(1-q)}
\]

But, \( \beta_{r,j}^{t_2}, \beta_{r',j}^{t_2} \) are the same.

So, \( \beta_{r,j}^{t_2} \cdot \beta_{r',j}^{t_2} = a^{2x} \) for some \( x \neq 0 \).

The inequality is correct, so we indeed have only one solution and the code is MDS
Duplication

So, must we have that big of a field? Unfortunately, yes. Let’s prove that. Say we have an MDS $s$-duplication code. Let’s look at 2 info elements in row $i$, columns $j^{t_1}, j^{t_2}, s.t. t_1 \neq t_2$.

They are in the same row and ZigZag set.

So, like before we need the matrix

$$\begin{bmatrix} \alpha_{r,i}^{t_1} & \alpha_{r,i}^{t_2} \\ \beta_{r,i}^{t_1} & \beta_{r,i}^{t_2} \end{bmatrix}$$

to be of full rank.

Which means, $\forall t \in [0, s - 1]$

$$(\alpha_{r,i}^{t_1})^{-1} \cdot \beta_{r,i}^{t_1} \neq (\alpha_{r,i}^{t_2})^{-1} \cdot \beta_{r,i}^{t_2}$$

Which means we need $s$ different non zero elements in the field.

So, $q \geq s + 1$. 
Generalization of the code construction

- Where previously we had two redundancies, now we will work with $r$ redundancies.

- This means that instead of using binary vectors to construct the zigzag permutations, we will use $r$-ary vectors.
Generalization of the code construction

- **Reminder:** For \(v \in F_2^m\), we defined \(f_v : [0,2^m - 1] \rightarrow [0,2^m - 1]\) as the zigzag function such that \(f_v(x) = x + v\).

- **Definition:** For any \(0 \leq l \leq r - 1\) and \(v \in \mathbb{Z}_r^m\), define \(f_v^l : [0,r^m] \rightarrow [0,r^m]\) by

  \[
  f_v^l(x) = x + lv
  \]

We’ll notate \(f_i = f_{v_i}\)
Generalization of the code construction

Example: For $m = 2, r = 3, x = 4, l = 2, v_1 = (0,1)$

$$f_1^2(4) = f_{(0,1)}^2(4) = 4 + 2 \cdot (0,1) = (1,1) + (0,2) = (1,0) = 3$$

Calculations are performed over $Z_3$. 
Generalization of the code construction

- **Reminder:** For \( v \in F_2^m \), we defined \( X_v = \{ x \in F_2^m | x \cdot (1,1, ..., 1) = 0 \} \).

- **Definition:** For any \( 0 \leq l \leq r - 1 \) and \( v_i \in Z_r^m \), define
  \[
  X_i^l = \{ x \in [0,r^m - 1] | x \cdot v_i = r - l \}
  \]

- Calculations are performed over \( Z_r \).
Generalization of the code construction

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

$$f_u^{-l}f_v^{l}(X_v^l) \cap f_u^{-s}f_v^{s}(X_v^s) = \begin{cases} |X_v^s|, & (l - s)c_{v,u} = 0 \\ 0, & \text{otherwise} \end{cases}$$

For $s = 0$,

$$|f_u^{-l}f_v^{l}(X_v^l) \cap X_v^0| = \begin{cases} |X_v^0|, & \text{if} \ l c_{v,u} = 0 \\ 0, & \text{otherwise.} \end{cases}$$
Generalization of the code construction

**Theorem:** $|X_l^i| = r^{m-1}$

**Theorem:** Using the $r$-ary vectors $\{e_i\}_{i=1}^{m}$, construct the functions $\{f^l_0\}_{l=0}^{r-1}, \ldots, \{f^l_m\}_{l=0}^{r-1}$ and their corresponding sets $\{X^l_0\}_{l=0}^{r-1}, \ldots, \{X^l_m\}_{l=0}^{r-1}$ as previously described, where $X^l_0 = \{x \in Z^m_r | x \cdot (1, \ldots, 1) = l\}$.

The corresponding $(m + r + 1, m + 1)$ code has an optimal ratio of $\frac{1}{r}$. 

Generalization of the code construction

- **Theorem:** When $r = 3$, a field of size 4 is sufficient for making an MDS as described in the previous constructions.

- $r > 2$ erasures are unlikely, which makes the MDS property not very important for codes with $r$ redundancies. There is some motivation in using them because of their smaller rebuilding ratio.
Conclusions

**Advantages of zigzag codes:**

1. Recovery ratio of $\frac{1}{r}$.
2. Has an optimal update rate.
3. Has an optimal/close to optimal rebuilding ratio.

**Disadvantages of zigzag codes:**

1. Work over finite fields, which are computationally expensive (the bigger the field).
2. Extending to more than $m + 1$ columns is not easy and requires a bigger finite field.
Thanks For Listening