Coding for Distributed Storage Systems via Rank-Metric Codes

Natalia Silberstein
Technion - Israel Institute of Technology

Based on joint works with
Ankit Singh Rawat, O. Ozan Koyluoglu, and Sriram Vishwanath
University of Texas at Austin
Distributed storage system

source

storage node 1

storage node 2

storage node n
Distributed storage system

source

storage node 1

storage node 2

storage node n

data collector
Distributed storage system

source

storage node 1

storage node 2

storage node n
Overview

• Coding for distributed storage systems (DSS):
  – Erasure codes
  – Regenerating codes
  – Locally repairable codes

• New optimal locally repairable codes (LRCs)
• LRCs with local minimum storage regeneration
• Error resilience in DSS
Coding for DSS-example

(4,2) MDS code (erasure code)
Coding for DSS-example

(4,2) MDS code (erasure code)

data collector
Reliability of erasure codes vs. replication

Replication
Reliability of erasure codes vs. replication
Erasure codes for DSS

• Using an \((n, k)\) erasure (MDS) code:
  – Partition the original data of size \(\mathcal{M}\) into \(k\) packets
  – Generate \(n\) packets
  – Store each packet in a different node

• To reconstruct the data use any \(k\) packets
  (MDS property)
Node repair process

- If only one node fails, how to rebuild the redundancy?
Node repair process

- If only one node fails, how to rebuild the redundancy?
- Naïve method: to reconstruct the whole data from \( k \) nodes
Node repair process

• If only one node fails, how to rebuild the redundancy?
• Naïve method: to reconstruct the whole data from $k$ nodes
• It can be done much more efficiently!

File of size $\mathcal{M}$

storage node 1

storage node 2

storage node $n$
Two goals for codes design for DSS

Reducing the repair bandwidth (the amount of data downloaded from the nodes during a node repair process)

Achieving locality (reducing the number of nodes participating in a node repair process)
Two families of codes for DSS

Reducing the **repair bandwidth** (the amount of data downloaded from the nodes during a node repair process)

– Regenerating codes

Achieving **locality** (reducing the number of nodes participating in a node repair process)

– Locally repairable codes
System parameters and notations

- $M$ - size of file
- $n$ - number of nodes
- $k$ - number of nodes sufficient to reconstruct a file
- $\alpha$ - storage per node
- *repair bandwidth* - the amount of data downloaded from the nodes during a node repair process
- *locality* - number of nodes participating in a node repair process
Regenerating Codes
System parameters and notations

- $\mathcal{M}$ - size of file
- $n$ - number of nodes
- $k$ - number of nodes sufficient to reconstruct a file
- $\alpha$ - storage per node
- **repair bandwidth** - the amount of data downloaded from the nodes during a node repair process
- **locality** - number of nodes participating in a node repair process
System model

File of size $\mathcal{M}$

storage node 1

storage node 2

storage node $n$

Newcomer node 2

data collector

$\alpha$

$\beta$

$d$

$k$
System model

File of size $\mathcal{M}$

storage node 1

storage node 2

storage node $n$

Newcomer node 2

$\beta d =$ repair bandwidth

$\alpha$

$\beta$

$\alpha$

$\beta$

$k$

data collector
Storage-repair bandwidth tradeoff

• **Tradeoff** between storage $\alpha$ and repair bandwidth $\gamma = \beta d$ was studied based on network coding approach. [*]

• **MSR**(minimum storage) code: $(\alpha, \gamma) = \left(\frac{M}{k'}, \frac{Md}{k(d-k+1)}\right)$

• **MBR**(minimum bandwidth) code: $(\alpha, \gamma) = \left(\frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k}\right)$

[A. Dimakis et al.,”Network coding for distributed storage systems, 2007”]
Storage-repair bandwidth tradeoff

- **Tradeoff** between storage $\alpha$ and repair bandwidth $\gamma = \beta d$ was studied based on network coding approach. [*]
- **MSR** (minimum storage) code: $(\alpha, \gamma) = \left( \frac{M}{k}, \frac{Md}{k(d-k+1)} \right)$
- **MBR** (minimum bandwidth) code: $(\alpha, \gamma) = \left( \frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k} \right)$

**Example:** $M=20$ MB, $k=20$, $n=25$

- Reed-Solomon code: $\alpha=1$ MB, $\beta=20$ MB
- Minimum storage (MSR): $\alpha=1$ MB, $\beta=4.8$ MB
- Minimum repair bandwidth (MBR): $\alpha=1.66$ MB, $\beta=1.66$ MB

[A. Dimakis et al., ”Network coding for distributed storage systems, 2007”]
Vector (Array) codes for DSS

A linear $[n, \mathcal{M}, d_{\text{min}}, \alpha]_q$ vector (array) code $C$ of length $n$ over $\mathbb{F}_q$:

- a subspace of $\mathbb{F}_q^{\alpha n}$ of dimension $\mathcal{M}$
- the symbols $c_i$ of a codeword $c \in C$ belong to $\mathbb{F}_q^\alpha$
- the minimum distance $d_{\text{min}}$ is the minimum Hamming distance over $\mathbb{F}_q^\alpha$. 
Vector (Array) codes for DSS

A linear $[n, \mathcal{M}, d_{\text{min}}, \alpha]_q$ vector (array) code $C$ of length $n$ over $\mathbb{F}_q$:

- a subspace of $\mathbb{F}_q^{\alpha n}$ of dimension $\mathcal{M}$
- the symbols $c_i$ of a codeword $c \in C$ belong to $\mathbb{F}_q^\alpha$
- the minimum distance $d_{\text{min}}$ is the minimum Hamming distance over $\mathbb{F}_q^\alpha$.

An $[n, \mathcal{M}, d_{\text{min}}, \alpha]_q$ array code is called an $(n, k, \alpha)$ MDS array code if $\alpha | \mathcal{M}$, $k = \mathcal{M} / \alpha$ and $d_{\text{min}} = n - k + 1$
Vector (Array) codes for DSS

A linear \([n, \mathcal{M}, d_{\text{min}}, \alpha]_q\) vector (array) code \(C\) of length \(n\) over \(\mathbb{F}_q\):

- a subspace of \(\mathbb{F}_q^{\alpha n}\) of dimension \(\mathcal{M}\)
- the symbols \(c_i\) of a codeword \(c \in C\) belong to \(\mathbb{F}_q^\alpha\)
- the minimum distance \(d_{\text{min}}\) is the minimum Hamming distance over \(\mathbb{F}_q^\alpha\).

An \([n, \mathcal{M}, d_{\text{min}}, \alpha]_q\) array code is called an \((n, k, \alpha)\) MDS array code if \(\alpha | \mathcal{M}\), \(k = \mathcal{M}/\alpha\) and \(d_{\text{min}} = n - k + 1\)

To store a file \(f\) of size \(\mathcal{M}\) on a DSS, \(f\) is first encoded to a codeword \(c\) of an \([n, \mathcal{M}, d_{\text{min}}, \alpha]_q\) code and then each symbol of \(c\) is stored on a different node.
Example: MSR Zigzag code [*]

Systematic nodes

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>

Parity nodes

Parameters: $\mathcal{M} = 12$, $n = 5$, $k = 3$, $\alpha = 4$, $\beta = 2$, $d = 4$

Example: MSR Zigzag code [*]

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td></td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td></td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td></td>
<td>$c_3$</td>
<td>$a_3+b_3+c_3$</td>
<td>$a_3+b_1+c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td></td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>

Systematic nodes

Parity nodes

Parameters: $\mathcal{M} = 12$, $n = 5$, $k = 3$, $\alpha = 4$, $\beta = 2$, $d = 4$

Example: MSR Zigzag code [*]

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
<td>$a_3+b_3+c_3$</td>
<td>$a_3+b_1+c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>

Systematic nodes

parity nodes

Parameters: $\mathcal{M} = 12$, $n = 5$, $k = 3$, $\alpha = 4$, $\beta = 2$, $d = 4$

Example: MSR Zigzag code [*]

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>

Systematic nodes

parity nodes

Parameters: $\mathcal{M} = 12$, $n = 5$, $k = 3$, $\alpha = 4$, $\beta = 2$, $d = 4$

### Example: MSR Zigzag code

**Systematic nodes**

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
</tr>
</tbody>
</table>

**Parity nodes**

<table>
<thead>
<tr>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>

**Parameters:**

- $M = 12$, $n = 5$, $k = 3$
- $\alpha = 4$, $\beta = 2$, $d = 4$

Example: MSR Zigzag code [*]

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td></td>
<td>$c_2$</td>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>

Systematic nodes

parity nodes

Parameters: $M = 12$, $n = 5$, $k = 3$, $\alpha = 4$, $\beta = 2$, $d = 4$

Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left( \frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k} \right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2Md}{2kd-k^2+k}, \frac{2Md}{2kd-k^2+k}\right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[Example illustration with (10,9) MDS code]

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left( \frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k} \right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2Md}{2kd-k^2+k}, \frac{2Md}{2kd-k^2+k}\right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left( \frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k} \right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \theta=1\)

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left(\frac{2Md}{2kd-k^2+k'}, \frac{2Md}{2kd-k^2+k}\right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Example: MBR Code [*]

- MBR point: \((\alpha_{MBR}, \gamma_{MBR}) = \left( \frac{2Md}{2kd-k^2+k}, \frac{2Md}{2kd-k^2+k} \right)\)
- Example: \(M=9, k=3, n=5, \alpha=d=4, \beta=1\)

[Example MBR Code]

\[ \begin{align*}
X_1 & \quad Y_1 \\
X_2 & \quad Y_2 \\
X_3 & \quad Y_3 \\
X_4 & \quad Y_4 \\
X_5 & \quad Y_5 \\
X_6 & \quad Y_6 \\
X_7 & \quad Y_7 \\
X_8 & \quad Y_8 \\
X_9 & \quad Y_9 \\
\end{align*} \]

\((10,9)\) MDS code

Uncoded repair!

[K.V. Rashmi, N.B. Shah, P.V.Kumar, K.Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage”, 2009]
Locally Repairable Codes
System parameters and notations

- $\mathcal{M}$ - size of file
- $n$ - number of nodes
- $k$ - number of nodes sufficient to reconstruct a file
- $\alpha$ - storage per node
- repair bandwidth - the amount of data downloaded from the nodes during a node repair process
- locality - number of nodes participating in a node repair process
Locally repairable codes (LRCs)

An $[n, M, d_{\text{min}}, \alpha]_q$ code $C$ has $(r, \delta)$ locality, if for each symbol $c_i \in \mathbb{F}_q^\alpha$ of a codeword, there exists a set of coordinates $\Gamma(i)$, s.t.

- $i \in \Gamma(i)$
- $|\Gamma(i)| \leq r + \delta - 1$
- Minimum distance of $C|_{\Gamma(i)}$ is at least $\delta$

A code that satisfies these properties is called $(r, \delta, \alpha)$LRC.

\[ \begin{array}{c}
\text{Local group} \\
\text{Local code}
\end{array} \]
Locally repairable codes (LRCs)

An \([n, M, d_{\min}, \alpha]_q\) code \(C\) has \((r, \delta)\) locality, if for each symbol \(c_i \in \mathbb{F}_q^\alpha\) of a codeword, there exists a set of coordinates \(\Gamma(i)\), s.t.

- \(i \in \Gamma(i)\)
- \(|\Gamma(i)| \leq r + \delta - 1\)
- Minimum distance of \(C|_{\Gamma(i)}\) is at least \(\delta\)

A code that satisfies these properties is called \((r, \delta, \alpha)\)LRC.

**LRCs for DSS:** A node \(i\) can be locally repaired by connecting to \(r\) other nodes of \(\Gamma(i)\), even if \(\delta - 2\) other nodes fail.
Upper bound on $d_{\text{min}}$ for LRCs

**Theorem 1.** For an $(r, \delta, \alpha)$ LRC $C$ over $\mathbb{F}$ of length $n$ and dimension $\mathcal{M}$ we have

$$d_{\text{min}}(C) \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil + 1 - \left( \left\lceil \frac{\mathcal{M}}{r\alpha} \right\rceil - 1 \right)(\delta - 1)$$

- This theorem establishes a trade off between
  - Node failure resilience ($d_{\text{min}}$)
  - Per node storage ($\alpha$)
Known $d_{\text{min}}$-optimal LRCs

- **Scalar** $(r, \delta, \alpha = 1)$ LRCs
  - The existence of optimal codes is known when
    
    $$(r + \delta - 1)|n \text{ and } |F| = MN^M \quad [*],[**]$$

  - An explicit construction for optimal codes is known for
    
    $$n = \left\lceil \frac{k}{r} \right\rceil (r + \delta - 1) \text{ and } |F| \geq n \quad [**]$$

- **Vector** $(r, \delta, \alpha)$ LRCs
  - An explicit construction for optimal codes is known for
    
    $$\alpha = r + \delta - 1, (r + \delta - 1)|n \text{ and } |F| \geq n \quad [\diamond],[\diamond \diamond]$$

[*] Gopalan, C. Huang, H. Simitchi, and S. Yekhanin, ``On the locality of codeword symbols'', 2011
New LRCs via rank-metric (MRD) codes

• Attain the upper bound on minimum distance
• Explicit optimal codes for wide range of parameters
• Special case \( \alpha = 1 \) (scalar codes)
  – New explicit codes over smaller fields, for parameters, where only existence was known
  – Codes without restriction \((r + \delta - 1)|n\)

Rank-metric codes

• Let $\mathbb{F}_{q^m}$ be extension field of $\mathbb{F}_q$

• Since $\mathbb{F}_{q^m} \cong \mathbb{F}_q^m$, a vector $\mathbf{v} = (v_1, ..., v_N) \in \mathbb{F}_{q^m}^N$ can be represented by an $m \times N$ matrix $\mathbf{V}$ over $\mathbb{F}_q$. 

\[
\mathbb{F}_{q^m} \quad \mathbb{F}_q^m \quad \mathbf{v} \quad \mathbf{V} 
\]
Rank-metric codes

• Let $\mathbb{F}_{q^m}$ be extension field of $\mathbb{F}_q$

• Since $\mathbb{F}_{q^m} \cong \mathbb{F}_q^m$, a vector $\mathbf{v} = (v_1, \ldots, v_N) \in \mathbb{F}_{q^m}^N$ can be represented by an $m \times N$ matrix $V$ over $\mathbb{F}_q$

• rank of $\mathbf{v} \in \mathbb{F}_{q^m}^N$ is defined by $\text{rank}(\mathbf{v}) = \text{rank}(V)$

• rank distance of $\mathbf{v}, \mathbf{u}$ is defined by $d_R(\mathbf{v}, \mathbf{u}) = \text{rank}(V - U)$
Rank-metric codes

- An $[N, K, D]_{qm}$ rank-metric code $C \subset \mathbb{F}_{qm}^N$ is a linear code over $\mathbb{F}_{qm}$ of length $N$, dimension $K$ and minimum rank distance $D$.
- If $D = N - K + 1$, the code $C$ is called maximum rank distance (MRD).

[ P.Delsarte, “Bilinear forms over a finite field, with applications to coding theory,” 1978 ]
R. M. Roth, “Maximum-rank array codes and their application to crisscross error correction,” 1991.]
Rank-metric codes

- An \([N, K, D]_{qm}\) rank-metric code \(C \subset \mathbb{F}_{qm}^N\) is a linear code over \(\mathbb{F}_{qm}\) of length \(N\), dimension \(K\) and minimum rank distance \(D\).
- If \(D = N - K + 1\), the code \(C\) is called **maximum rank distance (MRD)**.

**Gabidulin MRD codes** - analog of Reed-Solomon codes for rank metric.

- A codeword in a Gabidulin code \(C^{Gab} \ (m \geq N)\) is:
  \[c = (f(g_1), \ldots, f(g_N))\]

- where \(g_i \in \mathbb{F}_{qm}\) are linearly independent over \(\mathbb{F}_q\), \(1 \leq i \leq N\).
- \(f(x) \in \mathbb{F}_{qm}[x]\) is a **linearized polynomial** of \(q\)-degree at most \(K - 1\).
Rank-metric codes

- An $[N, K, D]_{qm}$ rank-metric code $C \subseteq \mathbb{F}_{qm}^N$ is a linear code over $\mathbb{F}_{qm}$ of length $N$, dimension $K$ and minimum rank distance $D$
- If $D = N - K + 1$, the code $C$ is called maximum rank distance (MRD)
- Gabidulin MRD codes - analog of Reed-Solomon codes for rank metric
  - A codeword in a Gabidulin code $C^{Gab}$ ($m \geq N$) is
    $$
c = (f(g_1), \ldots, f(g_N))$$
  - where $g_i \in \mathbb{F}_{qm}$ are linearly independent over $\mathbb{F}_q$, $1 \leq i \leq N$
  - $f(x) \in \mathbb{F}_{qm}[x]$ is a linearized polynomial of $q$-degree at most $K - 1$

Linearized polynomial of $q$-degree $n$: $f(x) = \sum_{i=0}^{n} a_i x^{q^i}$, $a_i \in \mathbb{F}_{qm}$

$f(b_1 \alpha_1 + b_2 \alpha_2) = b_1 f(\alpha_1) + b_2 f(\alpha_2)$ for $b_1, b_2 \in \mathbb{F}_q, \alpha_1, \alpha_2 \in \mathbb{F}_{qm}$
Rank-metric codes

- An $[N, K, D]_{qm}$ rank-metric code $C \subseteq \mathbb{F}_{qm}^N$ is a linear code over $\mathbb{F}_{qm}$ of length $N$, dimension $K$ and minimum rank distance $D$
- If $D = N - K + 1$, the code $C$ is called maximum rank distance (MRD)
- Gabidulin MRD codes - analog of Reed-Solomon codes for rank metric
  - A codeword in a Gabidulin code $C^{Gab}$ ($m \geq N$) is
    $$c = (f(g_1), \ldots, f(g_N))$$
  - where $g_i \in \mathbb{F}_{qm}$ are linearly independent over $\mathbb{F}_q$, $1 \leq i \leq N$
  - $f(x) \in \mathbb{F}_{qm}[x]$ is a linearized polynomial of $q$-degree at most $K - 1$

An MRD code with minimum distance $D$ can correct any $D - 1$ rank erasures and $(D - 1)/2$ rank errors
Construction of \((r, \delta, \alpha)\) LRCs via Gabidulin codes

\[ f \in \mathbb{F}_{q^m}^\mathcal{M} \]

\([N, \mathcal{M}, D]_q m\) Gabidulin code

- \(c_1\)
- \(c_2\)
- \(\ldots\)
- \(c_{r\alpha}\)
- \(c_{r\alpha+1}\)
- \(c_{r\alpha+2}\)
- \(\ldots\)
- \(c_{2r\alpha}\)
- \(\ldots\)
- \(c_{N-r'\alpha+1}\)
- \(\ldots\)
- \(c_N\)

\(r\alpha \leq r'\alpha \leq r\alpha\)
Construction of \((r, \delta, \alpha)\) LRCs via Gabidulin codes

\[
f \in \mathbb{F}_{q^m}^M
\]

\([N, \mathcal{M}, D]_{q^m}\) Gabidulin code

\[
\begin{array}{ccccccc}
1 & \ldots & r \\
c_1 & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_2 & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_\alpha & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\begin{array}{ccccccc}
1 & \ldots & r \\
c_{r\alpha+1} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_{r\alpha+2} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_{2r\alpha} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\begin{array}{ccccccc}
1 & \ldots & r' \\
c_{N-r'+1} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_{N-r'+2} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\vdots & \vdots & \cdot & \cdot & \cdot & \cdot & \cdot \\
c_{N} & \ldots & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{array}
\]

\[
\begin{array}{ccccccc}
group 1 \\
group 2 \\
group \left[\frac{n}{r+\delta-1}\right]
\end{array}
\]
Construction of \((r, \delta, \alpha)\) LRCs via Gabidulin codes

\[ f \in \mathbb{F}^M_{q^m} \]

\([N, M, D]_{q^m}\) Gabidulin code

\[ r\alpha \]

\[ r\alpha \rightarrow \]

MDS array codes over \(\mathbb{F}_q\)

\[
\begin{array}{cccc}
c_1 & c_2 & \cdots & c_{r\alpha} \\
c_{r\alpha+1} & c_{r\alpha+2} & \cdots & c_{2r\alpha} \\
& & \vdots & \\
c_{r\alpha+\alpha} & \cdots & c_{N-\delta+\alpha} & c_{N-\delta+1} \\
\end{array}
\]

\[ \delta - 1 \]

\[ \delta - 1 \]

\[ \delta - 1 \]

\[
\begin{array}{cccc}
1 & \cdots & r & \alpha \\
c_1 & \cdots & c_{r\alpha} \\
c_2 & \cdots & c_{r\alpha+1} \\
& \vdots & \vdots & \\
c_{\alpha} & \cdots & c_{r\alpha+\alpha} & c_{N-\delta+\alpha} \\
\end{array}
\]
group 1

\[
\begin{array}{cccc}
1 & \cdots & r & \alpha \\
& & \vdots & \\
& & \vdots & \\
c_{r\alpha+\alpha} & \cdots & c_{2r\alpha} & c_{N-\delta+1} \\
\end{array}
\]
group 2

\[
\begin{array}{cccc}
1 & \cdots & r' & \alpha \\
c_{N-r'+1} & \cdots & c_{N-r'+\alpha} \\
c_{N-r'+2} & \cdots & c_{N-\delta+1} \\
& \vdots & \vdots & \\
c_{N-\delta+\alpha} & \cdots & c_{N} \\
\end{array}
\]
group \([n/(r + \delta - 1)]\)
**$d_{\text{min}}$-optimality**

**Theorem.** Let $C^{\text{loc}}$ be an $(r, \delta, \alpha)$ LRC over $\mathbb{F}$ of length $n$, minimum distance $d_{\text{min}}$ and dimension $\mathcal{M}$, obtained by this construction. Then

- If $(r + \delta - 1)|n$ then $C^{\text{loc}}$ is $d_{\text{min}}$-optimal with $|\mathbb{F}| = q^m$, $m \geq N = \frac{n}{r+\delta-1} r \alpha$, $q \geq r + \delta - 1$.

- If $n(\text{mod } r + \delta - 1) - (\delta - 1) > \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil \text{ (mod } r) > 0$ then $C^{\text{loc}}$ is $d_{\text{min}}$-optimal with

  $|\mathbb{F}| = q^m$, $m \geq N = \left(n - (\delta - 1) \left(\left\lfloor \frac{n}{r+\delta-1} \right\rfloor + 1\right)\right) \alpha$, $q \geq r + \delta - 1$. 
Example: optimal scalar LRC

$(r, \delta, \alpha) = (4,2,1)$

file

$\begin{array}{cccccccccc}
    f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & f_7 & f_8 & f_9 \\
\end{array}$

over $\mathbb{F}_{5^{11}}$

$[11,9,3]_{5^{11}}$ Gabidulin code

$\begin{array}{cccc}
    c_1 & c_2 & c_3 & c_4 \\
    c_5 & c_6 & c_7 & c_8 \\
    c_9 & c_{10} & c_{11} \\
\end{array}$

MDS codes over $\mathbb{F}_5$

$\begin{array}{cccc}
    c_1 & c_2 & c_3 & c_4 & p_1 \\
    c_5 & c_6 & c_7 & c_8 & p_2 \\
    c_9 & c_{10} & c_{11} & p_3 \\
\end{array}$

group 1
group 2
group 3
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $M = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$. 

\begin{tabular}{|c|c|c|c|c|c|}
\hline
$c_1$ & $c_2$ & $c_3$ & $c_4$ & $p_1$ \\
\hline
\end{tabular} 

\begin{tabular}{|c|c|c|c|c|c|}
\hline
$c_5$ & $c_6$ & $c_7$ & $c_8$ & $p_2$ \\
\hline
\end{tabular} 

\begin{tabular}{|c|c|c|c|c|}
\hline
$c_9$ & $c_{10}$ & $c_{11}$ & $p_3$ \\
\hline
\end{tabular} 

\text{group 1} \hspace{2cm} \text{group 2} \hspace{2cm} \text{group 3}
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\min} \leq 4$. We prove that $d_{\min} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); \ g_1, ... g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\min} \leq 4$. We prove that $d_{\min} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); g_1, ..., g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); \ g_1, \ldots g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4);$
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8);$
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$
Example: optimal scalar LRC

• $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $M = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.

• $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.

• $c_i = f(g_i); g_1, ... g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$

• $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$

• Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent over $\mathbb{F}_q$ elements (Any 3 for group 3)
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $M = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.
- $f(x) \in F_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); g_1, \ldots, g_{11} \in F_{q^{11}}$ linearly independent over $F_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$
- Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent over $F_q$ elements (Any 3 for group 3)
- Any 4 symbols from group 1, 4 symbols of group 2 and 3 symbols of group 3 constitute a codeword of a Gabidulin code of distance $D = 3$
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $M = 9$ and minimum distance $d_{\min} \leq 4$. We prove that $d_{\min} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); g_1, ... g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$; $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$; $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$
- Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent over $\mathbb{F}_q$ elements (Any 3 for group 3)
- Any 4 symbols from group 1, 4 symbols of group 2 and 3 symbols of group 3 constitute a codeword of a Gabidulin code of distance $D = 3$
- Any $i$ erasures inside a group correspond to $i - 1$ rank erasures
Example: optimal scalar LRC

• $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.

• $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.

• $c_i = f(g_i); \ g_1, \ldots g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$.

• $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$

• Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent over $\mathbb{F}_q$ elements (Any 3 for group 3).

• Any 4 symbols from group 1, 4 symbols of group 2 and 3 symbols of group 3 constitute a codeword of a Gabidulin code of distance $D = 3$

• Any $i$ erasures inside a group correspond to $i - 1$ rank erasures
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $M = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i)$; $g_1, ..., g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$
- Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent elements (Any 3 for group 3)
- Any $i$ erasures inside a group correspond to $i - 1$ rank erasures

1 node erasure in each group: equivalent to 0 rank erasures

28
Example: optimal scalar LRC

- $C_{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\text{min}} \leq 4$. We prove that $d_{\text{min}} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); g_1, ... g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
  $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
  $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$
- Any 4 symbols of the group 1 or 2 are evaluations of $f(x)$ in 4 linearly independent over $\mathbb{F}_q$ points (Any 3 for group 3)
- Any 4 symbols from group 1, 4 symbols of group 2 and 3 symbols of group 3 constitute a codeword of a Gabidulin code of distance $D = 3$
- Any $i$ erasures inside a group correspond to $i - 1$ rank erasures

2 node erasures in one group, 1 in another group: equivalent to 1 rank erasure, which can be corrected by a code with rank distance 3
Example: optimal scalar LRC

- $C^{loc}$ is a $(r = 4, \delta = 2, \alpha = 1)$ LRC of length $n = 14$, dimension $\mathcal{M} = 9$ and minimum distance $d_{\min} \leq 4$. We prove that $d_{\min} = 4$.
- $f(x) \in \mathbb{F}_{q^{11}}[x]$ is a linearized polynomial of $q$-degree 8.
- $c_i = f(g_i); g_1, ... g_{11} \in \mathbb{F}_{q^{11}}$ linearly independent over $\mathbb{F}_q$
- $p_1 = f(g_1) + f(g_2) + f(g_3) + f(g_4) = f(g_1 + g_2 + g_3 + g_4)$;
- $p_2 = f(g_5) + f(g_6) + f(g_7) + f(g_8) = f(g_5 + g_6 + g_7 + g_8)$;
- $p_3 = f(g_9) + f(g_{10}) + f(g_{11}) = f(g_9 + g_{10} + g_{11})$

3 node erasures in one group (the worst case): equivalent to 2 rank erasures, which can be corrected by a code with rank distance 3

- Any $i$ erasures inside a group correspond to $i - 1$ rank erasures
Example: optimal vector LRC

\( (r, \delta, \alpha) = (3, 3, 4) \)

\( f \) of size 28 over \( \mathbb{F}_{q^{36}} \)

\([36, 28, 9]_{q^{36}} \) Gabidulin code

(5,3,4)-MDS array code over \( \mathbb{F}_q \)
Example: optimal vector LRC

- $C^{loc}$ is $(r = 3, \delta = 3, \alpha = 4)$ LRC of length $n = 15$, dimension $M = 28$ and minimum distance $d_{\text{min}} \leq 5$. We prove that $d_{\text{min}} = 5$. 

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$a_5$</td>
<td>$a_9$</td>
<td>$p_{11}^a$</td>
<td>$p_{12}^a$</td>
<td></td>
</tr>
<tr>
<td>$a_2$</td>
<td>$a_6$</td>
<td>$a_{10}$</td>
<td>$p_{21}^a$</td>
<td>$p_{22}^a$</td>
<td></td>
</tr>
<tr>
<td>$a_3$</td>
<td>$a_7$</td>
<td>$a_{11}$</td>
<td>$p_{31}^a$</td>
<td>$p_{32}^a$</td>
<td></td>
</tr>
<tr>
<td>$a_4$</td>
<td>$a_8$</td>
<td>$a_{12}$</td>
<td>$p_{41}^a$</td>
<td>$p_{42}^a$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$</td>
<td>$b_5$</td>
<td>$b_9$</td>
<td>$p_{11}^b$</td>
<td>$p_{12}^b$</td>
<td></td>
</tr>
<tr>
<td>$b_2$</td>
<td>$b_6$</td>
<td>$b_{10}$</td>
<td>$p_{21}^b$</td>
<td>$p_{22}^b$</td>
<td></td>
</tr>
<tr>
<td>$b_3$</td>
<td>$b_7$</td>
<td>$b_{11}$</td>
<td>$p_{31}^b$</td>
<td>$p_{32}^b$</td>
<td></td>
</tr>
<tr>
<td>$b_4$</td>
<td>$b_8$</td>
<td>$b_{12}$</td>
<td>$p_{41}^b$</td>
<td>$p_{42}^b$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1$</td>
<td>$c_5$</td>
<td>$c_9$</td>
<td>$p_{11}^c$</td>
<td>$p_{12}^c$</td>
<td></td>
</tr>
<tr>
<td>$c_2$</td>
<td>$c_6$</td>
<td>$c_{10}$</td>
<td>$p_{21}^c$</td>
<td>$p_{22}^c$</td>
<td></td>
</tr>
<tr>
<td>$c_3$</td>
<td>$c_7$</td>
<td>$c_{11}$</td>
<td>$p_{31}^c$</td>
<td>$p_{32}^c$</td>
<td></td>
</tr>
<tr>
<td>$c_4$</td>
<td>$c_8$</td>
<td>$c_{12}$</td>
<td>$p_{41}^c$</td>
<td>$p_{42}^c$</td>
<td></td>
</tr>
</tbody>
</table>
Example: optimal vector LRC

- $C^{loc}$ is $(r = 3, \delta = 3, \alpha = 4)$ LRC of length $n = 15$, dimension $M = 28$ and minimum distance $d_{\text{min}} \leq 5$. We prove that $d_{\text{min}} = 5$.
- Any $\delta - 1 + i = 2 + i$ node erasures in a local group correspond to $\alpha i = 4i$ rank erasures.
- Any 4 node erasures correspond to at most 8 rank erasures, which can be corrected by a Gabidulin code of rank distance 9.
LRCs with local regeneration
System parameters and notations

- $\mathcal{M}$ - size of file
- $n$ - number of nodes
- $k$ - number of nodes sufficient to reconstruct a file
- $\alpha$ - storage per node

**repair bandwidth** - the amount of data downloaded from the nodes during a node repair process

**locality** - number of nodes participating in a node repair process
LRCs with local regeneration

• To repair a failed node, all the data stored on \( r \) nodes in the same local group is downloaded (repair bandwidth \( r \alpha \))

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>( a_5 )</td>
<td>( a_9 )</td>
<td>( p_{11}^a )</td>
<td>( p_{12}^a )</td>
<td></td>
</tr>
<tr>
<td>( a_2 )</td>
<td>( a_6 )</td>
<td>( a_{10} )</td>
<td>( p_{21}^a )</td>
<td>( p_{22}^a )</td>
<td></td>
</tr>
<tr>
<td>( a_3 )</td>
<td>( a_7 )</td>
<td>( a_{11} )</td>
<td>( p_{31}^a )</td>
<td>( p_{32}^a )</td>
<td></td>
</tr>
<tr>
<td>( a_4 )</td>
<td>( a_8 )</td>
<td>( a_{12} )</td>
<td>( p_{41}^a )</td>
<td>( p_{42}^a )</td>
<td></td>
</tr>
</tbody>
</table>
LRCs with local regeneration

• To repair a failed node, all the data stored on \( r \) nodes in the same local group is downloaded (repair bandwidth \( r\alpha \))

• To reduce the repair bandwidth, a newcomer node downloads \( \beta \leq \alpha \) symbols from a set of \( d \geq r \) nodes from the same group

\[
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 \\
\hline
a_1 & a_5 & a_9 & p_{11}^a & p_{12}^a \\
a_2 & a_6 & a_{10} & p_{21}^a & p_{22}^a \\
a_3 & a_7 & a_{11} & p_{31}^a & p_{32}^a \\
a_4 & a_8 & a_{12} & p_{41}^a & p_{42}^a \\
\end{array}
\]
Gabidulin code

MSR codes over $\mathbb{F}_q$

MSR-LRC

file over $\mathbb{F}_{q^m}$

Zigzag Code

group 1

Zigzag Code

group 2

Zigzag Code

group $\lfloor n/(r + \delta - 1) \rfloor$
\textbf{MBR-LRC}

file over $\mathbb{F}_{q^m}$

Gabidulin code

... 

\textbf{MBR codes over $\mathbb{F}_q$}

\begin{itemize}
  \item group 1
  \item group 2
  \item group $[n/(r + \delta - 1)]$
\end{itemize}
MSR-LRC and MBR-LRC

- MSR-LRC[*] and MBR-LRC [**]
  obtained by the construction based on rank-metric codes achieve
- Maximum resilience (i.e., optimal minimum distance);
- Maximum capacity (i.e., maximum rate).


Error Resilience in Distributed Storage
Propagation of errors
Propagation of errors
Propagation of errors
**Challenge:**

A single corrupted node can infect many others.
Two models of adversarial errors

- **Static errors:**
  an adversary replaces the content of an affected node *only once*.

- **Dynamic errors:**
  an adversary may replace the content of an affected node *any number of times*. 
Static errors model
Static errors model

• Single corrupted node may infect many others

• Error has a large Hamming weight ($\leq n$)
Solution: use another metric!

- **Observation:** Error rank is bounded

\[
\text{Error rank is bounded by } 1 + e^y_1 + Ae^y_n + Be^y_2
\]
Solution: use another metric!

- **Observation:** Error rank is bounded

- **Rank-metric codes** can correct such errors
Construction of $t$-errors correcting MSR code[*]

System parameters and notations

- $\mathcal{M}$ - size of file
- $n$ - number of nodes
- $k$ - number of nodes sufficient to reconstruct a file
- $\alpha$ - storage per node
- **repair bandwidth** - the amount of data downloaded from the nodes during a node repair process
- **locality** - number of nodes participating in a node repair process
- $t$ – number of errors
Construction of $t$-errors correcting MSR code

\[ f \in \mathbb{F}_q^\mathcal{M} \]

\[[N = k\alpha, \mathcal{M}, D = 2t\alpha + 1]_{q^N} \text{ Gabidulin code}\]

\[
\begin{array}{cccccccc}
  c_1 & c_2 & \ldots & c_\alpha & c_{\alpha+1} & c_{\alpha+2} & \ldots & c_{k\alpha} \\
\end{array}
\]

\[(n, k\alpha, k, \alpha, \beta, d)_{\mathbb{F}_q} \text{ MSR code}\]

\[
\begin{array}{cccc}
  1 & \ldots & k & n-k \\
  c_1 & \ldots & & \\
  c_2 & \ldots & & \\
  \vdots & \vdots & & \\
  c_\alpha & \ldots & c_{k\alpha} & \\
\end{array}
\]
$t$-errors correcting codes for DSS

• The same idea can be applied to obtain
  – $t$-errors correcting MBR codes
  – $t$-errors correcting LRC

• $t$-errors correcting MSR and LRC have the maximal possible rate!
Example: $t = 1$ (static)

- Let file of size $\mathcal{M} = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, \mathcal{M} = 4, D = 9]_{q^N}$ MRD code to form a codeword $c = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5,3,4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$b_2$</td>
<td>$c_2$</td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3$</td>
<td>$b_3$</td>
<td>$c_3$</td>
<td>$a_3+b_3+c_3$</td>
<td>$a_3+b_1+c_4$</td>
</tr>
<tr>
<td>$a_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>
Example: \( t = 1 \) (static)

- Let file of size \( \mathcal{M} = 4 \) over \( \mathbb{F}_{q^{12}} \), \( n = 5, k = 3, \alpha = 4, t = 1 \)
- We use \([N = 12, \mathcal{M} = 4, D = 9]_{q^N}\) MRD code to form a codeword \( c = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]\)
- Then we apply the \((5,3,4)_{q}\) zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1 + e_1)</td>
<td>(b_1)</td>
<td>(c_1)</td>
<td>(a_1 + b_1 + c_1)</td>
<td>(a_1 + 2b_3 + 2c_2)</td>
</tr>
<tr>
<td>(a_2 + e_2)</td>
<td>(b_2)</td>
<td>(c_2)</td>
<td>(a_2 + b_2 + c_2)</td>
<td>(a_2 + 2b_4 + c_1)</td>
</tr>
<tr>
<td>(a_3 + e_3)</td>
<td>(b_3)</td>
<td>(c_3)</td>
<td>(a_3 + b_3 + c_3)</td>
<td>(a_3 + b_1 + c_4)</td>
</tr>
<tr>
<td>(a_4 + e_4)</td>
<td>(b_4)</td>
<td>(c_4)</td>
<td>(a_4 + b_4 + c_4)</td>
<td>(a_4 + b_2 + 2c_3)</td>
</tr>
</tbody>
</table>
Example: $t = 1$ (static)

- Let file of size $\mathcal{M} = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, \mathcal{M} = 4, D = 9]_{q^N}$ MRD code to form a codeword $\mathbf{c} = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5,3,4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1+e_1$</td>
<td>$b_1$</td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2+e_2$</td>
<td></td>
<td>$c_2$</td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3+e_3$</td>
<td></td>
<td>$c_3$</td>
<td>$a_3+b_3+c_3$</td>
<td>$a_3+b_1+c_4$</td>
</tr>
<tr>
<td>$a_4+e_4$</td>
<td>$b_4$</td>
<td>$c_4$</td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>
Example: $t = 1$ (static)

- Let file of size $\mathcal{M} = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, \mathcal{M} = 4, D = 9]_{q^N}$ MRD code to form a codeword $\mathbf{c} = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5,3,4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1+e_1$</td>
<td></td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2+e_2$</td>
<td>$b_1$</td>
<td></td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3+e_3$</td>
<td></td>
<td>$c_2$</td>
<td></td>
<td>$a_3+b_3+c_4$</td>
</tr>
<tr>
<td>$a_4+e_4$</td>
<td>$b_2$</td>
<td>$c_3$</td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>
Example: $t = 1$ (static)

- Let file of size $M = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, M = 4, D = 9]_{q^N}$ MRD code to form a codeword $c = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5,3,4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1 + e_1$</td>
<td>$b_1 - e_1$</td>
<td>$c_1$</td>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2 + e_2$</td>
<td>$b_2 - e_2$</td>
<td>$c_2$</td>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3 + e_3$</td>
<td>$b_3 - 2^{-1}e_1$</td>
<td>$c_3$</td>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4 + e_4$</td>
<td>$b_4 - 2^{-1}e_2$</td>
<td>$c_4$</td>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>
Example: $t = 1$ (static)

- Let file of size $\mathcal{M} = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, \mathcal{M} = 4, D = 9]_{q^N}$ MRD code to form a codeword $c = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5,3,4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1+e_1$</td>
<td>$b_1-e_1$</td>
<td>$c_1$</td>
<td>$a_1+b_1+c_1$</td>
<td>$a_1+2b_3+2c_2$</td>
</tr>
<tr>
<td>$a_2+e_2$</td>
<td>$b_2-e_2$</td>
<td>$c_2$</td>
<td>$a_2+b_2+c_2$</td>
<td>$a_2+2b_4+c_1$</td>
</tr>
<tr>
<td>$a_3+e_3$</td>
<td>$b_3-2^{-1}e_1$</td>
<td>$c_3$</td>
<td>$a_3+b_3+c_3$</td>
<td>$a_3+b_1+c_4$</td>
</tr>
<tr>
<td>$a_4+e_4$</td>
<td>$b_4-2^{-1}e_2$</td>
<td>$c_4$</td>
<td>$a_4+b_4+c_4$</td>
<td>$a_4+b_2+2c_3$</td>
</tr>
</tbody>
</table>

\[ c + (e_1, e_2, e_3, e_4)[I\ B\ 0] \]
Example: $t = 1$ (static)

- Let file of size $\mathcal{M} = 4$ over $\mathbb{F}_{q^{12}}$, $n = 5$, $k = 3$, $\alpha = 4$, $t = 1$
- We use $[N = 12, \mathcal{M} = 4, D = 9]_{q^N}$ MRD code to form a codeword $c = [a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4]$
- Then we apply the $(5, 3, 4)_q$ zigzag code

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1 + e_1$</td>
<td>$b_1 - e_1$</td>
<td>$c_1$</td>
<td>$a_1 + b_1 + c_1$</td>
<td>$a_1 + 2b_3 + 2c_2$</td>
</tr>
<tr>
<td>$a_2 + e_2$</td>
<td>$b_2 - e_2$</td>
<td>$c_2$</td>
<td>$a_2 + b_2 + c_2$</td>
<td>$a_2 + 2b_4 + c_1$</td>
</tr>
<tr>
<td>$a_3 + e_3$</td>
<td>$b_3 - 2^{-1}e_1$</td>
<td>$c_3$</td>
<td>$a_3 + b_3 + c_3$</td>
<td>$a_3 + b_1 + c_4$</td>
</tr>
<tr>
<td>$a_4 + e_4$</td>
<td>$b_4 - 2^{-1}e_2$</td>
<td>$c_4$</td>
<td>$a_4 + b_4 + c_4$</td>
<td>$a_4 + b_2 + 2c_3$</td>
</tr>
</tbody>
</table>

$D = 9$

$\text{DC}$

$c + (e_1, e_2, e_3, e_4)[I\ B\ 0]$
Dynamic errors model
Dynamic errors model

• **Dynamic errors:**
  an adversary may replace the content of an affected node *any number of times.*
Dynamic errors model

- **Dynamic errors:**
  
an adversary may replace the content of an affected node **any number of times.**
Dynamic errors model

• **Dynamic errors:**
  an adversary may replace the content of an affected node **any number of times.**
Dynamic errors model

• Dynamic errors:

an adversary may replace the content of an affected node any number of times.
Dynamic errors model

- **Dynamic errors:**
  an adversary may replace the content of an affected node **any number of times.**
Dynamic errors model

- **Dynamic errors:**
  
an adversary may replace the content of an affected node **any number of times.**
Dynamic errors model

- **Dynamic errors:**
  an adversary may replace the content of an affected node **any number of times.**
Dynamic errors model

• **Problem:** A corrupted node can introduce an error of *unbounded* rank
Dynamic errors model

• **Problem:** A corrupted node can introduce an error of unbounded rank

• **Solution:** [*]
  – Combination of our coding scheme with the subspace signatures

Restricting error rank

- **The idea:** to restrict rank of the error that an attacked node can cause in the system
- **Node** $i$: $\alpha$ symbols over $\mathbb{F}_{q^N}$ are considered as $\alpha$ vectors of length $N$ over $\mathbb{F}_q$ which span a subspace $V_i$
- **Node** $i$ always sends the data contained in $V_i$
- We enforce a corrupted node $j$ to send vectors only from $V_j$ by using **subspace signatures**
Subspace signatures approach

• A subspace signature [*] of the content of each node is stored in a trusted verifier.

• Trusted verifier checks the data sent by a node against the stored subspace signature.

• Data reconstruction:
  – If $s \leq t$ nodes fail the test
    
    $s\alpha$ rank erasures and $(t - s)\alpha$ rank errors
  – The original file can be reconstructed if an MRD code has rank distance
    
    $D = 2t\alpha + 1 \geq 2(t - s)\alpha + s\alpha + 1$

[F.Zhao, T. Kalker, M. Medard, and K.J.Han, “Signatures for content distribution with network coding,” 2007]
Conclusion

Rank-metric codes for DSS:
• Optimal locally repairable codes
• LRCs with local regeneration
• Error-correcting codes
Conclusion

Rank-metric codes for DSS:

• Optimal locally repairable codes
• LRCs with local regeneration
• Error-correcting codes (active eavesdropper)

• Secure DSS against passive eavesdropper [*]

Thank you!