**IERG6120 Coding for Distributed Storage Systems** Lecture 13 - 01/11/2016

# Windows Azure LRC and Maximally Recoverable Codes

*Lecturer: Kenneth Shum* *Scribe: Qiaoqiao Zhou*

In this lecture, we go through the construction of a (6,2,2) Local Reconstruction Codes (LRC) in [1]. LRC is a new set of erasure codes designed for Windows Azure Storage system. The important benefits of LRC are that it reduces the bandwidth required for repair, while still allowing a significant reduction in storage overhead.

## 1 LRC

**Definition 1.** *We say that a linear $(n, k)$ code $C$ is a $(k, l, r)$ LRC if the following conditions are satisfied:*

- $n = k + l + r$ *and the normalized storage overhead is* $\frac{n}{k} = 1 + \frac{l+r}{k}$*;*
- *The $k$ information symbols are divided into $l$ groups of size $\frac{k}{l}$. For each such group there is one local parity-check symbol computed within group;*
- *There are $r$ global parity-check symbols computed from all the information symbols.*

From the definition, we have the following observation

**Observation 2.** *The key properties of a $(k, l, r)$ LRC are*

- *single information failure can be decoded from $\frac{k}{l}$ symbols;*
- *arbitrary failures up to $r + 1$ can be decoded.*

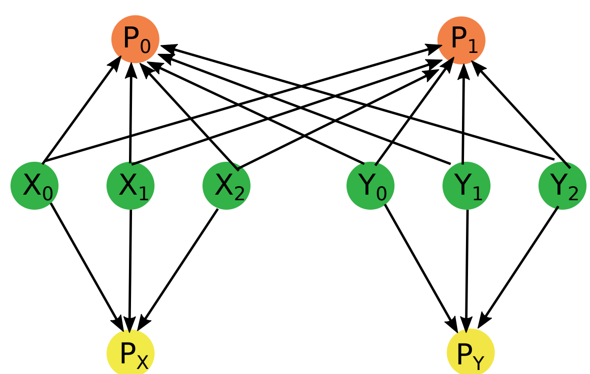


**Figure 1:** A (6,2,2) LRC Example

Consider a (6,2,2) LRC example shown in Fig 1 with 6 information symbols $\mathsf{X}_0, \mathsf{X}_1, \mathsf{X}_2, \mathsf{Y}_0, \mathsf{Y}_1, \mathsf{Y}_2$ and 4 parity-check symbols $\mathsf{P}_0, \mathsf{P}_1, \mathsf{P}_\mathsf{X}$ and $\mathsf{P}_\mathsf{Y}$. Symbols $\mathsf{P}_0$ and $\mathsf{P}_1$ are called *global* parity-check symbols and can be computed from *all* the information symbols as

$$\mathsf{P}_0 = \alpha_0\mathsf{X}_0 + \alpha_1\mathsf{X}_1 + \alpha_2\mathsf{X}_2 + \beta_0\mathsf{Y}_0 + \beta_1\mathsf{Y}_1 + \beta_2\mathsf{Y}_2 \tag{1}$$

$$\mathsf{P}_1 = \alpha_0^2\mathsf{X}_0 + \alpha_1^2\mathsf{X}_1 + \alpha_2^2\mathsf{X}_2 + \beta_0^2\mathsf{Y}_0 + \beta_1^2\mathsf{Y}_1 + \beta_2^2\mathsf{Y}_2, \tag{2}$$

Symbols $\mathsf{P_X}$ and $\mathsf{P_Y}$ are called *local* parity-check symbols. They are generated by dividing the message symbols into two equal size groups and computing one for each group as

$$\mathsf{P_X} = \mathsf{X_0} + \mathsf{X_1} + \mathsf{X_2} \tag{3}$$
$$\mathsf{P_Y} = \mathsf{Y_0} + \mathsf{Y_1} + \mathsf{Y_2} \tag{4}$$

Therefore, the generating matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_0 & \alpha_0^2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha_1 & \alpha_1^2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \alpha_2 & \alpha_2^2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \beta_0 & \beta_0^2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & \beta_1 & \beta_1^2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \beta_2 & \beta_2^2 \end{bmatrix} \tag{5}$$

From (5), we know that this is a code with minimum Hamming distance 4, which means it is capable to recover any 3 symbol erasures. Next, we will show that if we choose the values of $\alpha$'s and $\beta$'s appropriately, we can also tolerate some, but not all, erasure patterns consisting of 4 symbol erasures.

Before proceeding, we introduce some definitions.

**Definition 3.** *For a LRC, due to the special encoding structure, there are some erasure patterns that are inherently unrecoverable. They are called* ***information-theoretically non-decodable****. These are unrecoverable no matter how you pick the coefficients of the parity-check symbols. The other erasure patterns are* ***information-theoretically decodable****. These patterns are potentially recoverable, provided that the coefficients are chosen appropriately.*

For instance, say $\mathsf{X_0}, \mathsf{X_1}, \mathsf{X_2}, \mathsf{P_X}$ fails. This failure is non-decodable because there are only two parity-check symbols (global parity-check symbols) that can help to decode the 3 missing information symbols. The other local parity-check symbol $\mathsf{P_Y}$ is useless in this failure. It is impossible to decode 3 information symbols from merely 2 parity-check symbols, regardless of the coding equations. Therefore, this kind of failure is information-theoretically non-decodable. However, if $\mathsf{X_0}, \mathsf{X_1}, \mathsf{Y_0}, \mathsf{Y_1}$ fails, for this failure pattern, it is possible to construct coding equations such that it is equivalent to solving 4 unknowns using 4 linearly independent equations. Thus, information-theoretically decodable.

**Definition 4.** *If all information-theoretically decodable erasure patterns are indeed decodable, then the code is called a* ***maximally recoverable codes****.*

In the remaining, we determine the values of $\alpha$'s and $\beta$'s so that the $(6, 2, 2)$ LRC can decode all information-theoretically decodable 4 failure patterns, i.e., achieves the Maximally Recoverable property. We focus on non-trivial cases as follows:

1. None of the four parity-check symbols fails. The four failures are equally divided between group $\mathsf{X}$ and $\mathsf{Y}$. Consider $\mathsf{X_0}, \mathsf{X_1}, \mathsf{Y_0}, \mathsf{Y_1}$ fails. To maintain the recoverable property, we need the remaining

submatrix invertible.

$$
\begin{vmatrix}
0 & 0 & 1 & 0 & \alpha_0 & \alpha_0^2 \\
0 & 0 & 1 & 0 & \alpha_1 & \alpha_1^2 \\
1 & 0 & 1 & 0 & \alpha_2 & \alpha_2^2 \\
0 & 0 & 0 & 1 & \beta_0 & \beta_0^2 \\
0 & 0 & 0 & 1 & \beta_1 & \beta_1^2 \\
0 & 1 & 0 & 1 & \beta_2 & \beta_2^2
\end{vmatrix}
=
\begin{vmatrix}
1 & 0 & \alpha_0 & \alpha_0^2 \\
1 & 0 & \alpha_1 & \alpha_1^2 \\
0 & 1 & \beta_0 & \beta_0^2 \\
0 & 1 & \beta_1 & \beta_1^2
\end{vmatrix}
$$

$$
\begin{aligned}
&=
\begin{vmatrix}
1 & 0 & \alpha_0 & \alpha_0^2 \\
0 & 0 & \alpha_1 - \alpha_0 & \alpha_1^2 - \alpha_0^2 \\
0 & 1 & \beta_0 & \beta_0^2 \\
0 & 0 & \beta_1 - \beta_0 & \beta_1^2 - \beta_0^2
\end{vmatrix} \\
&= -
\begin{vmatrix}
\alpha_1 - \alpha_0 & \alpha_1^2 - \alpha_0^2 \\
\beta_1 - \beta_0 & \beta_1^2 - \beta_0^2
\end{vmatrix} \\
&= -(\alpha_1 - \alpha_0)(\beta_1 - \beta_0)
\begin{vmatrix}
1 & \alpha_1 + \alpha_0 \\
1 & \beta_1 + \beta_0
\end{vmatrix} \\
&= -(\alpha_1 - \alpha_0)(\beta_1 - \beta_0)(\alpha_0 + \alpha_1 + \beta_0 + \beta_1)
\end{aligned} \tag{6}
$$

Therefore, we need $\alpha_1 \neq \alpha_0, \beta_1 \neq \beta_0, \alpha_0 + \alpha_1 \neq \beta_0 + \beta_1$. Due to symmetry, the remaining cases can be handled similarly. To maintain the recoverable property, we require $\alpha$'s and $\beta$'s satisfy

$$
\begin{aligned}
\alpha_i \neq \alpha_j, \beta_i \neq \beta_j \quad \forall i \neq j \\
\alpha_i + \alpha_j \neq \beta_s + \beta_t \quad \forall i \neq j, \forall s \neq t
\end{aligned} \tag{7}
$$

2. Only one of $\mathsf{P_X}$ and $\mathsf{P_Y}$ fails. Assume $\mathsf{P_X}$ fails, For the remaining three failures, two are in group $\mathsf{Y}$ and one in group $\mathsf{X}$. For example, $\mathsf{X}_0, \mathsf{Y}_1, \mathsf{Y}_2$ and $\mathsf{P_X}$ fail. The remaining submatrix must be full-rank

$$
\begin{vmatrix}
0 & 0 & 0 & 0 & \alpha_0 & \alpha_0^2 \\
1 & 0 & 0 & 0 & \alpha_1 & \alpha_1^2 \\
0 & 1 & 0 & 0 & \alpha_2 & \alpha_2^2 \\
0 & 0 & 1 & 1 & \beta_0 & \beta_0^2 \\
0 & 0 & 0 & 1 & \beta_1 & \beta_1^2 \\
0 & 0 & 0 & 1 & \beta_2 & \beta_2^2
\end{vmatrix}
= -
\begin{vmatrix}
0 & \alpha_0 & \alpha_0^2 \\
1 & \beta_1 & \beta_1^2 \\
1 & \beta_2 & \beta_1^2
\end{vmatrix}
= -
\begin{vmatrix}
0 & \alpha_0 & \alpha_0^2 \\
1 & \beta_1 & \beta_1^2 \\
0 & \beta_2 - \beta_1 & \beta_2^2 - \beta_1^2
\end{vmatrix}
$$

$$
\begin{aligned}
&=
\begin{vmatrix}
\alpha_0 & \alpha_0^2 \\
\beta_2 - \beta_1 & \beta_2^2 - \beta_1^2
\end{vmatrix}
= \alpha_0(\beta_2 - \beta_1)
\begin{vmatrix}
1 & \alpha_0 \\
1 & \beta_2 + \beta_1
\end{vmatrix} \\
&= \alpha_0(\beta_2 - \beta_1)(\alpha_0 + \beta_1 + \beta_2)
\end{aligned} \tag{8}
$$

which require $\alpha_0 \neq 0, \beta_1 \neq \beta_2, \beta_1 + \beta_2 \neq \alpha_0$. Due to symmetry, the remaining cases can be handled similarly. To maintain the recoverable property, we require $\alpha$'s and $\beta$'s satisfy

$$
\begin{aligned}
\alpha_i \neq 0, \beta_i \neq 0 \quad \forall i \\
\alpha_i \neq \alpha_j, \beta_i \neq \beta_j \quad \forall i \neq j \\
\alpha_i + \alpha_j \neq \beta_s \quad \forall i \neq j, \forall s \\
\alpha_i \neq \beta_s + \beta_t \quad \forall i, \forall s \neq t
\end{aligned} \tag{9}
$$

3. Both $\mathsf{P_X}$ and $\mathsf{P_Y}$ fail. In addition, the remaining two failures are divided between group $\mathsf{X}$ and $\mathsf{Y}$. For

example, $\mathsf{X}_0, \mathsf{Y}_0, \mathsf{P}_\mathsf{X}, \mathsf{P}_\mathsf{Y}$ fail, the remaining submatrix

$$\begin{vmatrix} 0 & 0 & 0 & 0 & \alpha_0 & \alpha_0^2 \\ 1 & 0 & 0 & 0 & \alpha_1 & \alpha_1^2 \\ 0 & 1 & 0 & 0 & \alpha_2 & \alpha_2^2 \\ 0 & 0 & 0 & 0 & \beta_0 & \beta_0^2 \\ 0 & 0 & 1 & 0 & \beta_1 & \beta_1^2 \\ 0 & 0 & 0 & 1 & \beta_2 & \beta_2^2 \end{vmatrix} = \begin{vmatrix} \alpha_0 & \alpha_0^2 \\ \beta_0 & \beta_0^2 \end{vmatrix} = \alpha_0\beta_0(\alpha_0 + \beta_0) \tag{10}$$

which implies $\alpha_0 \neq \beta_0 \neq 0$. Due to symmetry, the remaining cases can be handled similarly. To maintain the recoverable property, we require $\alpha$'s and $\beta$'s satisfy

$$\alpha_i \neq 0, \beta_i \neq 0 \quad \forall i \qquad \text{and } \alpha_i \neq \beta_j, \quad \forall i, j \tag{11}$$

To ensure all the cases are recoverable, $\alpha$'s and $\beta$'s should satisfy the following conditions:

$$\begin{aligned} \alpha_i \neq 0, \beta_i \neq 0 \quad \forall i \\ \alpha_i \neq \beta_j, \quad \forall i, j \\ \alpha_i + \alpha_j \neq \beta_s + \beta_t \quad \forall i \neq j, \forall s \neq t \end{aligned} \tag{12}$$

One way to fulfill these conditions (12) is to assign to $\alpha$'s and $\beta$'s the element from a finite field GF($2^4$), which is produced by an irreducible polynomial (e.g. $f(x) = x^4 + x + 1$). Suppose the finite field elements can be represented by polynomials of the form $c_0 + c_1\gamma + c_2\gamma^2 + c_3\gamma^3$. Then, pick $\alpha_0, \alpha_1, \alpha_2 \in \{1, \gamma, 1+\gamma\}$ and $\beta_0, \beta_1, \beta_2 \in \{\gamma^2, \gamma^3, \gamma^2 + \gamma^3\}$ will satisfy the requirement (12).

**Exercise.** Consider a linear (8,5)-code over a finite field $\mathbb{F}$, defined by the following encoding structure: Symbols $x_0$, $x_1$, $x_2$, $y_0$, $y_1$ are information symbols. $p_h$ is a parity-check symbol computed by

$$p_h := ax_0 + bx_1 + cx_2 + dy_0 + ey_1$$

where $a, b, c, d, e$ are elements in $\mathbb{F}$. (The subscript "h" stands for "heavy". $p_h$ is a heavy parity-check symbol that depends on all information symbols, in contrast to local parity-check symbols to be defined next.)

$p_x$ and $p_y$ are local parity-check symbols computed by

$$p_x := x_0 + x_1 + x_2, \qquad p_y := y_0 + y_1 + p_h$$

The coded symbol can be presented in an array format:

| | | | |
|---|---|---|---|
| $x_0$ | $x_1$ | $x_2$ | $p_x$ |
| $y_0$ | $y_1$ | $p_h$ | $p_y$ |

This is a codes with locality 3. The code symbols in the first (resp. second) row belong to a simple parity-check code. If there is an erasure in the first (resp. second) row, we can recover the erased symbol by reading the other three symbols in the same row. Furthermore, if we choose the coefficients $a$ to $e$ appropriately, we are able to recover one more erasure on top of the two local erasures.

- Show that if there are three erasures in the same row, then it is not possible to recover the original information symbols.
- Show how to pick the coefficients $a$ to $e$ in order to correct any erasure pattern consisting of one erasure in a row and two erasures in the another row. (You need to specify the finite field $\mathbb{F}$, and explicitly write down the coefficients $a$ to $e$.)

# References


[1] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li and S. Yekhanin, "Erasure decoding in Windows Azure Storage," in *USENIX Annual Technical Conference*, Boston 2012.

[2] M. Blaum, J. L. Hafner and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," in *IEEE Trans. on Inform. Theory,* vol. 59, no. 7, pp.4510–4519, Jul. 2013.