9-2005

New Efficient MDS Array Codes for RAID Part I: Reed-Solomon-like Codes for Tolerating Three Disk Failures

Gui-Liang FENG
University of Louisiana at Lafayette

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Feng Bao
Singapore Management University, fbao@smu.edu.sg

Jia-Chen SHEN
University of Louisiana at Lafayette

DOI: https://doi.org/10.1109/TC.2005.150

New Efficient MDS Array Codes for RAID. Part I. Reed-Solomon-like Codes for Tolerating Three Disk Failures

Gui-Liang FENG; Center for Adv. Comput. Studies, Louisiana Univ., Lafayette, LA, USA;

Deng, R.H.; Feng BAO; SHEN, Jia-Chen.

Published in *IEEE Transactions on Computers*, 2005, 54 (9), 1071-1080.
doi:10.1109/TC.2005.150

SECTION I
INTRODUCTION

Many applications, particularly in a business environment, need highly available and reliable multiple hard disks to store huge amounts of data. A new technique called Redundant Arrays of Inexpensive Disks (RAID) can be employed to satisfy this requirement [1]. RAID is widely used in many companies, universities, and government organizations. However, the disks in RAID may fail in a few years because of random damage and other reasons. To protect the data in RAID, constructing erasure codes for tolerating multiple disk failures is very important.

In order to retrieve the information lost in r failed (erased) disks, we need at least r redundant disks (in coding theory, this is known as the capacity of erasure channel [2]). The well-known Reed-Solomon code [3] can achieve this capacity. However, the encoding and decoding of Reed-Solomon code involve operations over finite fields and are thus very slow. It would be desirable to have binary linear codes that only involve exclusive-OR (XOR) operations. For $r = 2$, i.e., for tolerating two disk failures, many good codes have been developed [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. These codes are called MDS array codes. Array codes are a class of binary linear codes, where information and parity bits are placed in a two-dimensional (or multidimensional) array instead of a one-dimensional vector. The information and parity bits are defined over an Abelian group $G(q)$ with an addition operation. Usually, $q = 2$. The bits are just binary bits and addition is an XOR operation [14].The best results are EvenOdd codes [5], [14], X-codes [12], and B-codes [13]. However, these codes all have distance 3, meaning that they can be used for tolerating two disk failures. Recently, a generalization of EvenOdd codes has been developed [14]. However, for $r \geq 3$ the encoding and decoding are yet to be developed. In practical applications of RAID, the size of each individual symbol (i.e., m) can be as big as a whole vector: During update operations, we will want to update a minimal number of redundant symbols when a single information symbol is updated. That means the parity-check matrix

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & h_{1,1} & h_{1,2} & \dots & h_{1,n} \\ 0 & 1 & \dots & 0 & h_{2,1} & h_{2,2} & \dots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & h_{r,1} & h_{r,2} & \dots & h_{r,n} \end{bmatrix}. \qquad (1.1)$$

should be of the form
In this paper, we develop a new class of binary MDS array codes, which can be efficiently used in tolerating three disk failures in RAID. The codes are similar to Reed-Solomon codes.

The binary MDS array codes are a class of binary linear codes, where information bits form an $m \times n$ array and parity bits form an $m \times 3$ array. In applications of these new codes in RAID, m indicates the number of "data," which can be bytes or computer words, and are stored on a disk, $(m+1)$ is a very large prime, and n denotes the number of information disks on which information "data" will be stored. In RAID, n should be $20 \sim 50$. The code rate is $\frac{n}{n+3}$, i.e., it achieves the capacity of erasure channel [1]. These codes are low-density parity-check codes. Therefore, the encoding and decoding are very fast.

This paper is organized as follows: In Section 2, we introduce circular permutation matrices (CPM) and their algebra, which are very useful in the subsequent sections. In Section 3, we introduce the Reed-Solomon-like MDS array codes based on the Vender-monde matrices and circular permutation matrices, where the parity check matrices satisfy (1.1). For these codes, both encoding and decoding are very fast. When $r = 2$, it is reduced to the codes in [5]. In Section 4, a very fast decoding procedure is presented. Finally, conclusions are presented in Section 5.

SECTION II
NOTATIONS AND MAIN LEMMAS

In this section, we introduce and briefly review some mathematical results, which are very important in understanding the new codes and their fast encoding and decoding algorithms.

2.1 Circular Permutation Matrices and Their Algebra

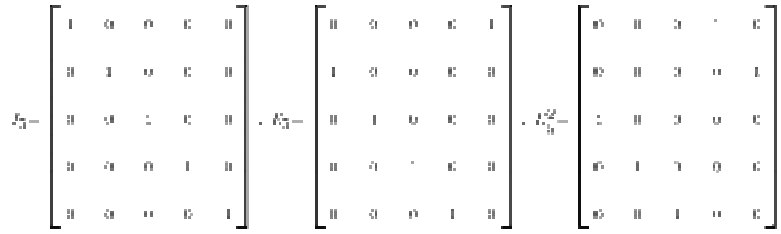
In this paper, for a matrix $M = (m_{i,j})_{l \times l}$, we always assume that $0 \le i, j \le l-1$, i.e., the order of rows (columns) is from 0 to $l-1$.

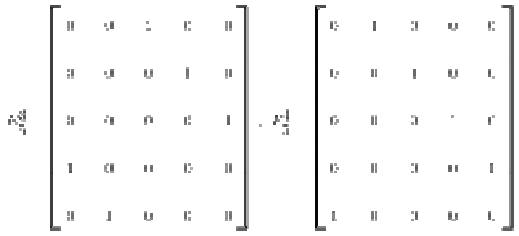
Let $p = m+1$ be an odd prime. Let I_m be an $m \times m$ *identity matrix* and O_m be an $m \times m$ zero matrix. Now, we define the *elemental cyclic matrix* E_{m+1} as

$$E_{m+1} \quad \begin{bmatrix} \vec{0} & 1 \\ I_m & \vec{0}^T \end{bmatrix}, \qquad (2.1)$$

where $\vec{1}$ is a $1 \times m$ vector of 1s and $\vec{0}^T$ is an $m \times 1$ vector of 0s. It can be easily checked that $\{I_{m-1}, E_{m-1}, E_{m-1}^2, \ldots, E_{m-1}^m\}$ form a group with matrix multiplication over $GF(2)$. We have $E_{m-1}^{m+1} = I_{m-1}$, $E_{m-1}^{-1} = E_{m+1}^m$.

For example, let $p = m + 1 = 5$, we have the group as follows:





Let $$\tilde{I}_{m+1} = \begin{bmatrix} I_m \\ 00\ldots0 \end{bmatrix}_{(m-1)\times m}. \qquad (2.2)$$

In the following, if no confusion arises, I, E, and $\tilde{I}$ are used in place of I_{m+1}, E_{m+1}, and $\tilde{I}_{m+1}$, respectively.

We also define $\langle a \rangle = a \bmod p$.
Thus, $0 \le \langle a \rangle \le p - 1$.

From (2.1), we have

Lemma 2.1

Let $E^\mu = (e_{i,j})_{p\times p}$, then $$e_{i,j} = \begin{cases} 1 & for\ i = \langle j + \mu \rangle \\ 0 & otherwise. \end{cases} \qquad (2.1')$$

Clearly, these matrices form an Abelian group with the traditional multiplication over $GF(2)$. The unity element is I, i.e., identity matrix. We have

$$E^i \times E^j = E^j \times E^i = E^{\langle i+j \rangle} \text{ and } E^0 = I.$$

It can be easily checked that $(I + E)$ has rank m. For any $1 \le a \le m$, there is $1 \le b \le m$ such that $\langle ab \rangle = 1$. Thus, we have

$$(I + E^a)\left(I + \sum_{j=1}^{b-1} E^{aj}\right) = I + E^{ab} = I + E.$$

Thus, the rank of $(I + E^a)$ is at least m. On the other hand, each column and each row has exactly two 1s. Therefore, the rank of $(I + E^a)$ is m, i.e., it is a singular $p \times p$ matrix.

We define a quasi-left-inverse matrix of $(I + E^{\mu})$, denoted by $(I + E^{\mu})^{-1}$, as follows:

$$(I \cdot E^{\mu})^{-1}(I \mid E^{\mu}) = \begin{bmatrix} I_m & \overrightarrow{1}^T \\ \overrightarrow{0}^t & 0 \end{bmatrix} \triangleq Q. \qquad (2.3)$$

Let

$$S \triangleq I - Q \quad \begin{bmatrix} O_m & \overrightarrow{1}^T \\ \overrightarrow{0} & 1 \end{bmatrix}, \qquad (2.4)$$

where O_m denotes $m \times m$ zero matrix.

To derive the explicit form of $(I + E^{\mu})^{-1}$, we introduce a function: For $\mu \neq 0$,

$$\pi_{\mu}(x) = \langle (x \mid 1)\mu^{-1} \quad 1 \rangle. \qquad (2.5)$$

It can be easily checked that $\{\pi_{\mu}(x) \mid 0 \leq x \leq m\} = \{0, 1, \ldots, m\}$ (2.6)

and $\pi_{\mu}(m) = m.$ (2.6′)

Definition 2.1

Let $\mu \neq 0$. A quasi-left-inverse matrix $(I + E^{\mu})^{-1} = (a_{i,j})$ is defined by

$$a_{i,j} - \begin{cases} 1 & for\ \pi_{\mu}(j) \leq \pi_{\mu}(i) \\ 0 & otherwise. \end{cases} \qquad (2.7)$$

Example 2.1

Let us consider the case of $p = 5$, $\mu = 1, 2$. Thus, $\pi_1(x) = \langle (x+1) - 1 \rangle = x$

and $\pi_2(x) = \langle (x - 1)3 - 1 \rangle = \langle 3x + 2 \rangle$,

because $\langle 2^{-1} \rangle = 3$.

$$(I \cdot E)^{-} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (I \cdot E^2)^{-1} \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

From the definition, we have

We have the following result:

Lemma 2.2

$$(I + E^{\mu})^{-1}(I + E^{\nu}) = \begin{bmatrix} I_m & 1^{\overrightarrow{1}^T} \\ \overrightarrow{0} & 0 \end{bmatrix} \triangleq Q. \qquad (2.8)$$

Proof

See Appendix.

It can be easily checked that the rank of Q is m. Thus, the rank of $(I + E^{\mu})$ is at least m. On the other hand, each row of $(I + E^{\mu})$ has exactly two 1s.

Therefore,

Lemma 2.3
$(I + E^{\mu})$ and $(I + E^{\mu})\tilde{I}$ have rank m for $\mu \neq 0$.

Remark
$(I + E^{\mu})\tilde{I}$ is the matrix formed after the deletion of the last column of $(I + E^{\mu})$.

Now, we are going to derive the relation between $(I + E)^{-1}$ and $(I + E^{\mu})^{-1}$.

We define a special permutation matrix $\Pi_{\mu} = (p_{i,j})_{p \times p}$:

$$p_{i,j} = \begin{cases} 1 & for\ j = \pi_{\mu}(i) \\ 0 & otherwise. \end{cases} \qquad (2.9)$$

We have the following lemma:

Lemma 2.4

$$\Pi_{\mu}^{t} = \Pi_{\mu}^{-1}. \qquad (2.10)$$

Proof
Let $\Pi_{\mu}\Pi_{\mu}^{t} = (d_{i,j})_{p \times p}$. Then, from (2.9), we have $d_{i,j} = \sum_{k=0}^{m} p_{i,k}p_{j,k} = p_{j,\pi_{\mu}(i)}$ and $d_{i,j} = 1$ if and only if $\pi_{\mu}(i) = \pi_{\mu}(j)$, i.e., $i = j$. Thus, $(d_{i,j})_{p \times p} = I$.

From (2.8) in Lemma 2.2, we have

Lemma 2.5

$$\Pi_{\mu}^{t}(I + E^{\mu})^{-1}\Pi_{\mu} = (I + E)^{-1} \qquad (2.11)$$

and

$$\Pi_{\mu}(I + E)^{-1}\Pi_{\mu}^{t} = (I - E^{\mu})^{-1}, \qquad (2.11')$$

where

$$(I + E)^{-1} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 & 0 \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}.$$

Furthermore, $(I + E^{\mu})^{-1}$ is a nonsingular matrix.

Proof

See Appendix.

Example 2.2

Let $p = m + 1 = 5$, $\mu = 2$. Then, we have $\mu^{-1} = 3$ because $\langle \mu \times \mu^{-1} \rangle = 1$. Furthermore, $\pi_\mu(x) = \pi_2(x) = \langle 3x + 2 \rangle$.

Thus, we have $\pi_2(0) = 2, \quad \pi_2(1) = 0, \quad \pi_2(2) = 3, \quad \pi_2(3) = 1, \quad \pi_2(4) = 4.$

From (2.9), we have

$$\Pi_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From (2.7), we have $a_{0,0} = a_{0,1} = a_{0,3} = 1$, because, $\pi_\mu(1) < \pi_\mu(3) < \pi_\mu(0) = \pi_\mu(0)$. For the same reason, we can determine all the values of $a_{i,j}$. Thus, we have

$$(I \cdot E^2)^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

It can be easily checked that $\Pi_2^t (I + E^2)^{-1} \Pi_2 = (I + E)^{-1}$, i.e.,

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Definition 2.2

A modified quasi-left-inverse matrix of $(I + E^\mu)$ for $\mu \neq 0$, denoted by $\overrightarrow{(I + E^\mu)^{-1}}$, is defined by

$$\overrightarrow{(I + E^\mu)^{-1}} Q (I + E^\nu) = Q, \qquad (2.12)$$

where

$$Q \triangleq \begin{bmatrix} I & 1^{\times I} \\ \vec{0} & 0 \end{bmatrix}.$$

This modified quasi-inverse matrix is very important in our decoding algorithm.

Example 2.3

Let $p = 5, \mu = 2$, we have

$$(I-E^2) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \text{ and } (I+E^2)^{-1} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

It can be easily checked that $(I + E^2)^{-1} \times (I + E^2) = Q$, i.e.,

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times$$

It can also be checked that

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

i.e., $\overrightarrow{(I + E^2)^{-1}} Q(I + E^2) = Q$ and

$$\overrightarrow{(I + E^2)^{-1}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From this example, it can be easily found that

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

We have the relation between $\overrightarrow{(I + E^{\mu})^{-1}}$ and $(I + E^{\mu})^{-1}$:

Lemma 2.6

Let $\overrightarrow{v}^T = (v_0, v_1, \ldots, v_m)^T$ be the sum of all columns of $(I + E^\mu)^{-1}$, i.e., $v_i = 1$ if the weight of row i of $(I + E^\mu)^{-1}$ is odd and otherwise 0. We have

$$\overrightarrow{(I - E^\mu)^{-1}} \quad (I + E^\mu)^{-1} + V, \qquad (2.13)$$

where $V = [\overrightarrow{v}^T \overrightarrow{v}^T \ldots \overrightarrow{v}^T \overrightarrow{0}^T]$.

Furthermore, $\overrightarrow{(I + E^\mu)^{-1}}$ is a nonsingular matrix.

Proof

See Appendix.

From the above lemma, we know that there is a modified quasi-left-inverse matrix such that (2.12) is true. Let us consider the matrix $M_t = (I + E^{\mu_1})(\prod_{j=2}^{t}(I + E^{\mu_j}))$, where $\mu_j \neq 0$. Since for each μ_j, there are nonsingular $\overrightarrow{(I - E^{\mu_j})^{-1}}$ and $(I + E^{\mu_j})^{-1}$. Then, from Definitions 2.1 and 2.2, we have

$$\left(\prod_{j=2}^{t} \overrightarrow{(I + E^{\mu_j})^{-1}}\right)(I + E^{\mu_1})^{-1} M_t = Q.$$

On the other hand, the rank of Q is m. Then, the rank of M_t is also m. Therefore, we have

Lemma 2.7

The matrix

$$\widetilde{M}_t = \left(\prod_{j=1}^{t}(I + E^{\mu_j})\right)\widetilde{I}, \qquad (2.14)$$

has rank m, for $\mu_j \neq 0$.

Now, we analyze some operations which are very important in the encoding process.

Let $\overrightarrow{v} = (v_0, v_1, \ldots, v_m)$.

Lemma 2.8

To implement $(I + E^\mu) \times \overrightarrow{v}^T$, we need μ XOR operations, and to implement $E^\mu \times \overrightarrow{v}^T$ and $\Pi_\mu \times \overrightarrow{v}^T$, we do not need any XOR operations.

Proof

See Appendix.

Lemma 2.9

There need to be m and $(m + p)$ XOR operations to implement $(I + E^\mu)^{-1} \times \overrightarrow{v}^T$ and $(I + E^\mu)^{-1} \times \overrightarrow{v}^T + \overrightarrow{w}$, respectively.

Proof

See Appendix.

2.2 The Reed-Solomon Codes

In the theory of error-correcting codes, the Reed-Solomon codes are very important. They are defined by the Vender-monde matrix and are Maximum Distance Separable (MDS) codes. MDS codes are also called *optimal* codes (see [15], p. 316).

First, we briefly review the Vender-monde matrix:

$$H_V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{r-1} & x_2^{r-1} & x_3^{r-1} & \dots & x_n^{r-1} \end{bmatrix}, \qquad (2.15)$$

where x_i s, for $1 \le i \le n$, are distinct from each other. A submatrix consisting of any r columns of H_V is a full rank matrix. A linear code C_{RS} defined by H_V as a parity check matrix is called a Reed-Solomon code. The code length is n, the code dimension is $k = n - r$, and the minimum distance is $d = r + 1$ (or $d = n - k + 1$). It is also an MDS code.
When $r = 3$, adding three columns, we have the following matrix

$$H_{EV} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 0 & x_1 & x_2 & x_3 & \dots & x_n \\ 0 & 0 & 1 & x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \end{bmatrix}. \qquad (2.16)$$

The linear code C_{ERS} defined by H_{EV} as a parity check matrix is called an extended Reed-Solomon code. This code is also an MDS code.

SECTION III
THE EXTENDED REED-SOLOMON-LIKECODES BASED ON CPM

In this section, we introduce the Reed-Solomon-like codes based on CPM and the extended Reed-Solomon-like codes on CPM.

We first define the following binary matrix:

$$\widetilde{H} \quad \begin{bmatrix} \widetilde{I} & \widetilde{I} & \widetilde{I} & \widetilde{I} & \dots & \widetilde{I} \\ \widetilde{I} & E\widetilde{I} & E^2\widetilde{I} & E^3\widetilde{I} & \dots & E^n\widetilde{I} \\ \widetilde{I} & E^2\widetilde{I} & E^4\widetilde{I} & E^6\widetilde{I} & \dots & E^{2n}\widetilde{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \widetilde{I} & E^{r-1}\widetilde{I} & E^{2(r-1)}\widetilde{I} & E^{3(r-1)}\widetilde{I} & \dots & E^{n(r-1)}\widetilde{I} \end{bmatrix}, \quad (3.1)$$

where $r < n < m$. This is an $r(m+1) \times (n+1)m$ binary matrix. It can be regarded as an $r \times (n+1)$ block matrix, where each block-column contains m columns, and each block-row contains $(m+1)$ rows.

Example 3.1

Let $m = 4$, i.e., $p = 5$, $n = 4$, and $r = 3$, we have

$$\widetilde{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E\widetilde{I} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad E^2\widetilde{I} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$E^3\widetilde{I} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad E^4\widetilde{I} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

and

$$\widetilde{H} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0 & 1&0&0&0 & 1&0&0&0 & 1&0&0&0 & 1&0&0&0 \\ 0&1&0&0 & 0&1&0&0 & 0&1&0&0 & 0&1&0&0 & 0&1&0&0 \\ 0&0&1&0 & 0&0&1&0 & 0&0&1&0 & 0&0&1&0 & 0&0&1&0 \\ 0&0&0&1 & 0&0&0&1 & 0&0&0&1 & 0&0&0&1 & 0&0&0&1 \\ 0&0&0&0 & 0&0&0&0 & 0&0&0&0 & 0&0&0&0 & 0&0&0&0 \\ \hline 1&0&0&0 & 0&0&0&0 & 0&0&0&1 & 0&0&1&0 & 0&1&0&0 \\ 0&1&0&0 & 1&0&0&0 & 0&0&0&0 & 0&0&0&1 & 0&0&1&0 \\ 0&0&1&0 & 0&1&0&0 & 1&0&0&0 & 0&0&0&0 & 0&0&0&1 \\ 0&0&0&1 & 0&0&1&0 & 0&1&0&0 & 1&0&0&0 & 0&0&0&0 \\ 0&0&0&0 & 0&0&0&1 & 0&0&1&0 & 0&1&0&0 & 1&0&0&0 \\ \hline 1&0&0&0 & 0&0&0&1 & 0&1&0&0 & 0&0&0&0 & 0&0&1&0 \\ 0&1&0&0 & 0&0&0&0 & 0&0&1&0 & 1&0&0&0 & 0&0&0&1 \\ 0&0&1&0 & 1&0&0&0 & 0&0&0&1 & 0&1&0&0 & 0&0&0&0 \\ 0&0&0&1 & 0&1&0&0 & 0&0&0&0 & 0&0&1&0 & 1&0&0&0 \\ 0&0&0&0 & 0&0&1&0 & 1&0&0&0 & 0&0&0&1 & 0&1&0&0 \end{array}\right].$$

We have the following theorem:

Theorem 3.1

Any r block-columns form a full rank submatrix, i.e., the columns of any r block-columns are linearly independent.

Proof

Let us consider the submatrix $\widetilde{H}_r$ consisting of block-columns $x_1, x_2, \ldots, x_r$:

$$\tilde{H}_r = \begin{bmatrix} \tilde{I} & \tilde{I} & \tilde{I} & \dots & \tilde{I} \\ E^{x_1}\tilde{I} & E^{x_2}\tilde{I} & E^{x_3}\tilde{I} & \dots & E^{x_r}\tilde{I} \\ E^{2x_1}\tilde{I} & E^{2x_2}\tilde{I} & E^{2x_3}\tilde{I} & \dots & E^{2x_r}\tilde{I} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ E^{(r-1)x_1}\tilde{I} & E^{(r-1)x_2}\tilde{I} & E^{(r-1)x_3}\tilde{I} & \dots & E^{(r-1)x_r}\tilde{I} \end{bmatrix}. \quad (3.2)$$

Multiplying $\tilde{H}_r$ on the left side by the sequence matrices:

$$\begin{bmatrix} I & O & O & \dots & O & O \\ O & I & O & & O & O \\ O & O & I & \dots & O & O \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ O & O & O & & I & O \\ O & O & O & \dots & E^{r-1} & I \end{bmatrix} \times \dots \times \begin{bmatrix} I & O & O & \dots & O & O \\ O & I & O & & O & O \\ O & E^{r-2} & I & \dots & O & O \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ O & O & O & \dots & I & O \\ O & O & O & \dots & E^{r-2} & I \end{bmatrix} \times$$

$$\begin{bmatrix} I & O & O & \dots & O & O \\ E^{r-1} & I & O & \dots & O & O \\ O & E^{r-2} & I & \dots & O & O \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ O & O & O & \dots & I & O \\ O & O & O & \dots & E^{1} & I \end{bmatrix},$$

where O represents an $(m+1) \times (m+1)$ 0-matrix, $\tilde{H}_r$ is reduced to the following matrix:

$$\begin{bmatrix} \tilde{I} & \tilde{I} & \tilde{I} & \dots & \tilde{I} \\ \tilde{O} & E^{x_2}(I - E^{x_2 - x_1})\tilde{I} & E^{x_3}(I + E^{x_3 - x_1})\tilde{I} & \dots & E^{x_r}(I + E^{x_r - x_1})\tilde{I} \\ \tilde{O} & \tilde{O} & E^{x_3 - x_2}(I + E^{x_3 - x_1})(I - E^{x_3 - x_2})\tilde{I} & \dots & E^{x_r - x_2}(I - E^{x_r - x_1})(I - E^{x_r - x_2})\tilde{I} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{O} & \tilde{O} & \tilde{O} & \dots & E^{x_r - x_{r-1}}\left(\prod_{j=1}^{r-1}(I - E^{x_r - x_j})\right)\tilde{I} \end{bmatrix},$$

where $\tilde{O}$ is an $(m+1) \times m$ 0-matrix. From Lemma 2.3, the submatrices in the diagonal block columns are full rank, i.e., the ranks are all m. Thus, the reduced matrix has rank rm, i.e., $\tilde{H}_r$ is a full rank matrix.

Since the summations of all rows of $E^i\tilde{I}$ and all rows of $\tilde{I}$ are rows of 1s, respectively (i.e., $(111\dots1)$), and the bottom row of $\tilde{I}$ is a row of 0s (i.e., $(000\dots0)$), the bottom row in each block-row in $\tilde{H}$ can be deleted. The reduced matrix is an $mr \times m(n+1)$ binary parity-check matrix. When the bottom row in each block-row in $\tilde{H}$ is deleted, it can be regarded as

multiplied by $\widetilde{I}^T$. Hence, we have the following matrix H, which is equivalent to $\widetilde{H}$ as a parity-check matrix:

$$H = \begin{bmatrix} I_m & I_m & I_m & I_m & \cdots & I_m \\ I_m & \widetilde{I}^1 E \widetilde{I} & \widetilde{I}^1 E^2 \widetilde{I} & \widetilde{I}^1 E^3 \widetilde{I} & \cdots & \widetilde{I}^1 E^n \widetilde{I} \\ I_m & \widetilde{I}^2 E^2 \widetilde{I} & \widetilde{I}^2 E^4 \widetilde{I} & \widetilde{I}^2 E^6 \widetilde{I} & \cdots & \widetilde{I}^2 E^{2n} \widetilde{I} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ I_m & \widetilde{I}^{r-1} E^{r-1} \widetilde{I} & \widetilde{I}^{r-1} E^{2(r-1)} \widetilde{I} & \widetilde{I}^{r-1} E^{3(r-1)} \widetilde{I} & \cdots & \widetilde{I}^{r-1} E^{n(r-1)} \widetilde{I} \end{bmatrix}, \tag{3.3}$$

This is an $rm \times (n+1)m$ binary matrix.

Example 3.2

Let $m = 4$, i.e., $p = 5$, $n = 4$, and $r = 3$, we have

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \widetilde{I}^1 E \widetilde{I} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \widetilde{I}^1 E^2 \widetilde{I} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\widetilde{I}^1 E^3 \widetilde{I} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \widetilde{I}^1 E^4 \widetilde{I} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and}$$

$$H = \left[\begin{array}{cccccccccccccccc} 1&0&0&0 & 1&0&0&0 & 1&0&0&0 & 1&0&0&0 & 1&0&0&0 \\ 0&1&0&0 & 0&1&0&0 & 0&1&0&0 & 0&1&0&0 & 0&1&0&0 \\ 0&0&1&0 & 0&0&1&0 & 0&0&1&0 & 0&0&1&0 & 0&0&1&0 \\ 0&0&0&1 & 0&0&0&1 & 0&0&0&1 & 0&0&0&1 & 0&0&0&1 \\ \hline 1&0&0&0 & 0&0&0&0 & 0&0&0&1 & 0&0&1&0 & 0&1&0&0 \\ 0&1&0&0 & 1&0&0&0 & 0&0&0&0 & 0&0&0&1 & 0&0&1&0 \\ 0&0&1&0 & 0&1&0&0 & 1&0&0&0 & 0&0&0&0 & 0&0&0&1 \\ 0&0&0&1 & 0&0&1&0 & 0&1&0&0 & 1&0&0&0 & 0&0&0&0 \\ \hline 1&0&0&0 & 0&0&0&1 & 0&1&0&0 & 0&0&0&0 & 0&0&1&0 \\ 0&1&0&0 & 0&0&0&0 & 0&0&1&0 & 1&0&0&0 & 0&0&0&1 \\ 0&0&1&0 & 1&0&0&0 & 0&0&0&1 & 0&1&0&0 & 0&0&0&0 \\ 0&0&0&1 & 0&1&0&0 & 0&0&0&0 & 0&0&1&0 & 1&0&0&0 \end{array}\right].$$

Now, we are going to introduce the extended Reed-Solomon-like codes based on CPM.

Let $(G, \oplus)$ be an Abelian group and let 0 be the identity element. Let $b \in \{0,1\}$, $g \in G$. We define

$$b \times g = g \times b = \begin{cases} 0 & for\ b = 0 \\ g & for\ b = 1. \end{cases} \tag{3.4}$$

Let $\overrightarrow{v} = (v_0, v_1, \ldots, v_{n-1})$ be a vector over G and $\overrightarrow{b} = (b_0, b_1, \ldots, b_{n-1})$ be a vector over $GF(2)$, we define

$$\begin{aligned} \overrightarrow{b} \times \overrightarrow{v} &= \overrightarrow{v} \times \overrightarrow{b} \\ &= (b_0 \times v_0) \oplus (b_1 \times v_1) \oplus \ldots \ominus (b_{n-1} \times v_{n-1}). \end{aligned} \tag{3.5}$$

Definition

Let C be the following linear code over an Abelian group defined by

$$C = \{\mathbf{c} = (\overrightarrow{c}_0, \overrightarrow{c}_1, \overrightarrow{c}_2, \ldots, \overrightarrow{c}_{n-3}) | H^* \mathbf{c}^T = \overrightarrow{0}^T\},$$

where $\overrightarrow{c}_i = (c_{i1}, c_{i2}, \ldots, c_{im})$, $c_{ij} \in G$, and

$$H^* \tag{3.6}$$

$$\begin{bmatrix} I_m & O_m & O_m & I_m & I_m & I_m & I_m & \ldots & I_m \\ O_m & I_m & O_m & I_m & \widetilde{I}^1 & \widetilde{I}^2 & \widetilde{I}^3 & \ldots & \widetilde{I}^{n-1} \\ O_m & O_m & I_m & I_m & \widetilde{I}^2 & \widetilde{I}^4 & \widetilde{I}^6 & \ldots & \widetilde{I}^{2n-2} \end{bmatrix} \triangleq [I^* | \widetilde{H}^*],$$

i.e., H^* is obtained by adding $r = 3$ parity-check block columns. For these codes, the first $r = 3$ vectors, $\overrightarrow{c}_0$, $\overrightarrow{c}_1$, and $\overrightarrow{c}_2$, are parity-check vectors, and the other $n + 1$ vectors, i.e., $\overrightarrow{c}_j$ for $3 \le j \le n + 3$, are information vectors.

In many applications, the Abelian group G can be computer words with bit-XOR operations.

When $r = 2$, this code is reduced to the EvenOdd code [5], [14]. These codes are MDS Abelian group array codes. At most r erasure error vectors $\overrightarrow{c}_i$ can be corrected. Thus, these codes can be efficiently used for tolerating multiple disk failures in $RAID$ architectures. In this paper, we are interested in the case of $r = 3$, i.e., the extended Reed-Solomon-like codes defined by (3.6) as a parity-check matrix.

The encoding process is to find $\overrightarrow{c_0}, \overrightarrow{c_1}, \overrightarrow{c_2}$, given $\overrightarrow{c_3}, \overrightarrow{c_4}, \cdots, \overrightarrow{c_{n+3}}$, by

$$\begin{bmatrix} \overrightarrow{c_0} \\ \overrightarrow{c_1} \\ \overrightarrow{c_2} \end{bmatrix} = \widetilde{H}^* \begin{bmatrix} \overrightarrow{c_3} \\ \overrightarrow{c_4} \\ \vdots \\ \overrightarrow{c_{n+3}} \end{bmatrix}. \tag{3.7}$$

From Lemmas 2.8 and 2.9, there needs to be $3mn$ XOR operations.

SECTION IV
DECODING OF THE EXTENDED REED-SOLOMON-LIKE CODES BASED ON CPM

Assume that a codeword $\mathbf{c} \triangleq (\overrightarrow{c}_0, \overrightarrow{c}_1, \ldots, \overrightarrow{c}_{n+3})$ is transmitted and that t packets, say $\overrightarrow{c}_{\mu_i}$ for $i = 1, 2, \ldots, t \le r = 3$, are lost. Then, the received codeword is given by

$$\mathbf{y} = (\overrightarrow{y}_0, \overrightarrow{y}_1, \ldots, \overrightarrow{y}_{n+3}),$$

where

$$\overrightarrow{y}_i = \begin{cases} \overrightarrow{c}_i & i \notin \{\mu_1, \mu_2, \ldots, \mu_t\} \\ \overrightarrow{0} & i \in \{\mu_1, \mu_2, \ldots, \mu_t\}. \end{cases}$$

We define the syndromes of the received codeword $\mathbf{y}$ as

$$H^* \mathbf{y}^T = \mathbf{s}^T, \qquad (4.1)$$

where $\mathbf{s} = (\vec{s}_0, \vec{s}_1, \vec{s}_2)$, $\vec{s}_i = (s_{i,0}, s_{i,2}, \ldots, s_{i,m-1}, s_{i,m})$, where $s_{i,j} \in G$ and $s_{i,m} = 0$.

Let $\mathbf{z} = (\vec{z}_0, \vec{z}_1, \ldots, \vec{z}_{n-3})$ and $\vec{z}_i = \begin{cases} \vec{0} & i \notin \{\mu_1, \mu_2, \ldots, \mu_t\} \\ \vec{c}_i & i \in \{\mu_1, \mu_2, \ldots, \mu_t\}. \end{cases}$

It follows that $\mathbf{c} \quad \mathbf{y} + \mathbf{z}$.

Since $H^* \mathbf{c}^T = \vec{0}^T$ and (4.1), we have $H^* \mathbf{y}^T = \mathbf{s}^T, \qquad (4.1')$

$$\begin{bmatrix} \widetilde{I} & \widetilde{I} & \cdots & \widetilde{I} \\ E^{\mu_1}\widetilde{I} & E^{\mu_2}\widetilde{I} & \cdots & E^{\mu_t}\widetilde{I} \\ \vdots & \vdots & \ddots & \vdots \\ E^{(t-1)\mu_1}\widetilde{I} & E^{(t-1)\mu_2}\widetilde{I} & \cdots & E^{(t-1)\mu_t}\widetilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \\ \vdots \\ \vec{c}_{\mu_t} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(0)} \\ \vdots \\ \vec{s}_{t-1}^{(0)} \end{bmatrix}. \qquad (4.2)$$

i.e.,

The decoding process can be briefly summarized as follows: Given a received codeword $\mathbf{y}$ and the locations of lost packets $\mu_1, \mu_2, \ldots, \mu_t$, we first compute syndromes from (4.1) and then determine the values of the lost packets $\vec{c}_{\mu_i}$ for $1 \le i \le t$ by solving (4.2).

In order to solve this problem, we now are going to derive a recursive algorithm for solving the linear equations with the Vender-monde matrix. First, we begin our derivation with the following simple example.

Assuming that there are three errors, for decoding we consider the following cases:

Case 1. The three errors are on the information disks, i.e., $0 \le \mu_1 < \mu_2 < \mu_3 \le n$.

In this case, recovering the information bits is reduced to solving the following equations:

$$\begin{bmatrix} \widetilde{I} & \widetilde{I} & \widetilde{I} \\ E^{\mu_1}\widetilde{I} & E^{\mu_2}\widetilde{I} & E^{\mu_3}\widetilde{I} \\ E^{2\mu_1}\widetilde{I} & E^{2\mu_2}\widetilde{I} & E^{2\mu_3}\widetilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \\ \vec{c}_{\mu_3} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(0)} \\ \vec{s}_2^{(0)} \end{bmatrix}.$$

First, consider the forward steps. Left multiply both sides of the above equation by the matrix

$$\begin{bmatrix} I & O & O \\ E^{\mu_1} & I & O \\ O & E^{\mu_2} & I \end{bmatrix},$$

where O is a $p \times p$ zero matrix. But, sometimes O denotes

a $p \times m$ zero matrix, if no confusion arises. We have

$$\begin{bmatrix} \widetilde{I} & \widetilde{I} & \widetilde{I} \\ O & (E^{\mu_1}+E^{\mu_2})\widetilde{I} & (E^{\mu_1}+E^{\mu_3})\widetilde{I} \\ O & E^{\mu_2}(E^{\mu_1}+E^{\mu_2})\widetilde{I} & E^{\mu_3}(E^{\mu_1}-E^{\mu_3})\widetilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \\ \vec{c}_{\mu_3} \end{bmatrix} \tag{4.11}$$
$$= \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(0)} + E^{\mu_1}\vec{s}_0^{(0)} \\ \vec{s}_2^{(0)} + E^{\mu_1}\vec{s}_1^{(0)} \end{bmatrix} \triangleq \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \\ \vec{s}_2^{(1)} \end{bmatrix}.$$

Left multiplying both sides of (4.11) by $\begin{bmatrix} I & O & O \\ O & I & O \\ O & E^{\mu_2} & I \end{bmatrix}$, we obtain

$$\begin{bmatrix} \widetilde{I} & \widetilde{I} & \widetilde{I} \\ O & (E^{\mu_1}+E^{\mu_2})\widetilde{I} & (E^{\mu_1}+E^{\mu_3})\widetilde{I} \\ O & O & (E^{\mu_2}+E^{\mu_3})(E^{\mu_1}+E^{\mu_3})\widetilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \\ \vec{c}_{\mu_3} \end{bmatrix}$$
$$\begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \\ \vec{s}_2^{(1)} + E^{\mu_2}\vec{s}_1^{(1)} \end{bmatrix} \triangleq \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \\ \vec{s}_2^{(2)} \end{bmatrix}. \tag{4.12}$$

It is equivalent to

$$\begin{bmatrix} \widetilde{I} & \widetilde{I} & \widetilde{I} \\ O & E^a(I+E^{d_{2,1}})\widetilde{I} & E^a(I+E^{d_{3,1}})\widetilde{I} \\ O & O & E^b(I+E^{d_{3,2}})(I+E^{d_{3,1}})\widetilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \\ \vec{c}_{\mu_3} \end{bmatrix}$$
$$= \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \\ \vec{s}_2^{(2)} \end{bmatrix}, \tag{4.12'}$$

where a, b, c, and $d_{i,j}$ denote μ_1, $\mu_2+\mu_1$, $\mu_3+\mu_2+\mu_1$, and $\mu_i-\mu_j$, respectively.

Now, we have rearranged the original coefficient matrix into an upper triangular matrix. Observe the right sides of (4.11)-(4.12), where we carried out $(2+1)=3(\vec{s}_a^{(b)}+E_{\mu_c}\vec{s}_{a-1}^{(b)})$-*type* operations. From Lemma 2.9, each such operation needs p additions in G. Thus, in these two forward steps $3p$ additions are needed.

Next, we perform the two backward steps. Left multiplying both sides of (4.12) by

$$\begin{bmatrix} I & O & \overrightarrow{(I+E^{\mu_3-\mu_1})^{-1}(I+E^{\mu_3-\mu_2})^{-1}E^{\mu_2-\mu_1}} \\ O & I & (I+E^{\mu_3-\mu_2})^{-1}E^{\mu_2} \\ O & O & I \end{bmatrix},$$

considering $Q \times \vec{I} = I_m$, where $(I+E^{\mu})^{-1}(I+E^{\mu}) = Q$ (See Section 2), we have

$$\begin{bmatrix} \overrightarrow{c}_{\mu_1} + \overrightarrow{c}_{\mu_2} \\ E^{\mu_2}(I+E^{\mu_2-\mu_1})\vec{I}\overrightarrow{c}_{\mu_2} \\ E^{\mu_1+\mu_2}(I+E^{\mu_3-\mu_2})(I+E^{\mu_3-\mu_1})\vec{I}\overrightarrow{c}_{\mu_3} \end{bmatrix} =$$

$$\begin{bmatrix} \overrightarrow{s}_0^{(0)} + \overrightarrow{(I+E^{\mu_3-\mu_1})^{-1}(I+E^{\mu_3-\mu_2})^{-1}E^{-\mu_2-\mu_1}\overrightarrow{s}_2^{(2)}} \\ \overrightarrow{s}_1^{(1)} + (I+E^{\mu_3-\mu_2})^{-1}E^{\mu_2-\mu_1}\overrightarrow{s}_2^{(2)} \\ \overrightarrow{s}_2^{(2)} \end{bmatrix} \triangleq \begin{bmatrix} \overrightarrow{s}_0^{(3)} \\ \overrightarrow{s}_1^{(3)} \\ \overrightarrow{s}_2^{(3)} \end{bmatrix}, \tag{4.13}$$

$$\begin{bmatrix} \overrightarrow{c}_{\mu_1} + \overrightarrow{c}_{\mu_2} \\ E^{\mu_1}(I+E^{\mu_2-\mu_1})\vec{I}\overrightarrow{c}_{\mu_2} \\ \overrightarrow{c}_{\mu_3} \end{bmatrix} = \begin{bmatrix} \overrightarrow{s}_0^{(3)} \\ \overrightarrow{s}_1^{(3)} \\ \overrightarrow{s}_0^{(3)} + \overrightarrow{s}_0^{(0)} \end{bmatrix} \triangleq \begin{bmatrix} \overrightarrow{s}_0^{(3)} \\ \overrightarrow{s}_1^{(3)} \\ \overrightarrow{s}_2^{(3)} \end{bmatrix}, \tag{4.14}$$

which can be rewritten as
Following the same approach, we have

$$\begin{bmatrix} \overrightarrow{c}_{\mu_1} \\ \overrightarrow{c}_{\mu_2} \\ \overrightarrow{c}_{\mu_3} \end{bmatrix} = \begin{bmatrix} \overrightarrow{s}_0^{(3)} + (I+E^{\mu_2})^{-1}E^{-\mu_1}\overrightarrow{s}_1^{(3)} \\ \overrightarrow{s}_0^{(3)} + \overrightarrow{s}_0^{(4)} \\ \overrightarrow{s}_2^{(3)} \end{bmatrix} \triangleq \begin{bmatrix} \overrightarrow{s}_0^{(4)} \\ \overrightarrow{s}_1^{(4)} \\ \overrightarrow{s}_2^{(4)} \end{bmatrix}. \tag{4.15}$$

Observe the right sides of (4.13)-(4.15), where we carried out $(2+1) = 3(I+E^{\mu_n})^{-1}E^b \times \overrightarrow{v})$-$type$ operations and $3(\overrightarrow{u}+\overrightarrow{w})-type$ operations. From Lemma 2.9 and Lemma 2.10, in the backward step, $3(m+p)$ additions in G are needed.

Thus, to solve this set of linear equations, a total of $(6p+3m)$ additions in G are needed.

Case 2. There is an error on the parity-check disk and the other two errors are on the information disks μ_1, and μ_2.

When the error is on the last parity-check disk, it is reduced to solving the following set of linear equations:

$$\begin{bmatrix} \tilde{I} & \tilde{I} \\ E^{\mu_1}\tilde{I} & E^{\mu_2}\tilde{I} \\ E^{2\mu_1}\tilde{I} & E^{2\mu_2}\tilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(0)} \\ \vec{s}_1^{(1)} \end{bmatrix}. \qquad (4.16)$$

It is equivalent to

$$\begin{bmatrix} \tilde{I} & \tilde{I} \\ O & (E^{\mu_1} - E^{\mu_2})\tilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(0)} - E^{\mu_1}\vec{s}_0^{(0)} \end{bmatrix} \triangleq \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \end{bmatrix},$$

i.e.,

$$\begin{bmatrix} \tilde{I} & \tilde{I} \\ O & E^{u}(I + E^{d_{2,1}})\tilde{I} \end{bmatrix} \begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} \\ \vec{s}_1^{(1)} \end{bmatrix},$$

where $u = \mu_1$ and $d_{2,1} = \mu_2 - \mu_1$. Thus, we have

$$\begin{bmatrix} \vec{c}_{\mu_1} \\ \vec{c}_{\mu_2} \end{bmatrix} = \begin{bmatrix} \vec{s}_0^{(0)} - E^{u}(I + E^{d_{2,1}})^{-1}\tilde{I}\vec{s}_1^{(1)} \\ E^{u}(I + E^{d_{2,1}})^{-1}\tilde{I}\vec{s}_1^{(1)} \end{bmatrix}$$

When the error is on the first parity-check disk, it is reduced to (4.16). But, $\vec{c}_{\mu_i}$ and $\vec{s}_i^{(0)}$ should be changed by $E^{\mu_i}\vec{c}_{\mu_i}$ and $\vec{s}_{i+1}^{(0)}$ for $i = 1, 2$, respectively.

When the error is on the middle parity-check disk, it is reduced to (4.16). But, E^{μ_i} and $\vec{s}_1^{(0)}$ should be changed by $E^{2\mu_i}$ and $\vec{s}_2^{(0)}$ for $i = 1, 2$, respectively.

The decoding process consists of two steps: calculating the syndromes from (4.1') and recovering the lost data from (4.2). There needs to be $3mn$ XOR operations for the first step, and $9(m+1)$ XOR operations for the second step. Thus, a total of $(3mn + 9(m+1))$ XOR operations are needed for decoding.

SECTION V
CONCLUSIONS

In this paper, we have presented a class of MDS array codes, which are based on circular permutation matrices in the Vender-monde matrix. The parity-check matrix is a low-density parity-check matrix. These codes are very efficient for tolerating up to three disk failures in RAID. For tolerating two disk failures in RAID, these codes are equivalent to EvenOdd codes [5]. There need to be $3mn$ XOR operations for encoding, and $(3mn + 9(m+1))$ XOR operations for decoding. When 32 codewords are encoded/decoded simultaneously, a 32-fold improvement can be achieved in efficiency.

However, this approach cannot be generalized for tolerating four or more disk failures in RAID architectures.

APPENDIX
PROOF OF LEMMA 2.2.

Let $(b_{i,j})_{p\times p} = (a_{i,j})_{p\times p}(c_{i,j})_{p\times p}$, where $(a_{i,j})_{p\times p} = (I + E^{\mu})^{-1}$ and $(c_{i,j})_{p\times p} = (I + E^{\mu})$.

$$(b_{i,j})_{p\times p} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 1 \\ 0 & 0 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

We need to prove that

$$b_{i,j} = \sum_{k=0}^{p-1} a_{i,k}c_{k,j}. \qquad (A.1)$$

From the multiplication of matrices, we have

where $(c_{i,j})_{p\times p} = (I + E^{\mu})$, i.e., $c_{\langle j+\mu\rangle,j} = c_{j,j} = 1.$ (A.2)

From (2.8), we have $b_{i,j} = \sum_{k=0}^{m} a_{i,k}c_{k,j}$. There are two 1s in $c_{k,j}$: $c_{j,j}$ and $c_{\langle j+\mu\rangle,j}$ from

(A.1). Thus, we have $b_{i,j} = a_{i,j} + a_{i,\langle j+\mu\rangle}.$ (A.3)

From (2.7), we have $\pi_{\mu}(\langle j+\mu\rangle) - \langle (j+\mu+1)\mu^{-1} - 1\rangle - \langle (j-1)\mu^{-1}\rangle.$ (A.4)

$$\pi_{\mu}(j) = \langle (j-1)\mu^{-1} - 1\rangle = \langle (j+1)\mu^{-1}\rangle - 1 \qquad (A.5)$$
$$= \pi_{\mu}(\langle j+\mu\rangle) - 1.$$

When $j \neq m$, we have

Now, let us consider the following cases:

Case 1. $0 \leq i \leq m$ and $j < m$: If $i = j$, then $b_{j,j} = a_{j,j} + a_{j,\langle j+\mu\rangle}$. From (2.7) and (A.5), we have $a_{j,j} = 1$ and $a_{j,\langle j+\mu\rangle} = 0$. Hence, $b_{j,j} = 1$.

If $i \neq j$ and $\pi_{\mu}(i) < \pi_{\mu}(j)$, then from (2.7) and (A.5) we have

$\pi_{\mu}(i) < \pi_{\mu}(j) < \pi_{\mu}(\langle j+\mu\rangle)$, and $a_{i,j} = a_{i,\langle j+\mu\rangle} = 0$, i.e., $b_{i,j} = 0$.

If $i \neq j$ and $\pi_{\mu}(i) > \pi_{\mu}(j)$, then from (2.7) and (A.5), we have

$\pi_{\mu}(i) > \pi_{\mu}(\langle j+\mu\rangle) > \pi_{\mu}(j)$, and $a_{i,j} = a_{i,\langle j+\mu\rangle} = 1$, i.e., $b_{i,j} = 0$.

Case 2. $i = j = m$: We have $b_{m,m} = a_{m,m} + a_{m,\mu-1}$. Since $\pi_{\mu}(m) = m \geq \pi_{\mu}(\mu - 1)$, from (2.7) we have $a_{m,m} = a_{m,\mu-1} = 1$, i.e., $b_{m,m} = 0$.

Case 3. $i < m$ and $j = m$: We have $b_{i,m} = a_{i,m} + a_{i,\mu-1}$. Since $\pi_{\mu}(m) = m > \pi_{\mu}(i)$ for $i < m$, and $\pi_{\mu}(\mu - 1) = 0 \leq \pi_{\mu}(i)$, we have $a_{i,m} = 0$ and $a_{i,\mu-1} = 1$, i.e., $b_{i,m} = 1$.

The proof is completed.

Proof of Lemma 2.5

Let $(I + E)^{-1} = (a_{i,j})_{p\times p}$. From (2.8), we have $a_{i,j} = 1$ if and only if $j \leq i$.

Let $(I + E)^{-1}\Pi_{\mu}^{T} = (b_{i,j})_{p\times p}$, then we have $b_{i,j} = \sum_{k=0}^{m} a_{i,k}p_{j,k}$. On the other hand, $p_{j,k} = 1$ if and only if $k = \pi_{\mu}(j)$. Hence, $b_{i,j} = a_{i,\pi_{\mu}(j)}$.

Let $\Pi_\mu (b_{i,j})_{p\times p} = (c_{i,j})_{p\times p}$. Then, we have $c_{i,j} = \sum_{k=0}^{m} p_{i,k} b_{k,j}$. From $p_{i,k} = 1$ if and only if $k = \pi_\mu(i)$, we have $c_{i,j} = b_{\pi_\mu(i),j}$. Combining the above two equations, we have

$$c_{i,j} = b_{\pi_\mu(i),j} = a_{\pi_\mu(i),\pi_\mu(j)}.$$

Thus, $c_{i,j} = 1$, i.e., $a_{\pi_\mu(i),\pi_\mu(j)} = 1$, if and only if $\pi_\mu(j) \le \pi_\mu(i)$. From (2.8), we know that $(c_{i,j})_{p\times p} = (I + E^\mu)^{-1}$.

On other hand, it can be easily checked that $(I + E)^{-1}$ and Π_μ are nonsingular matrices. Thus, $(I + E^\mu)^{-1}$ is also a nonsingular matrix.

Proof of Lemma 2.6

From (2.6), we have

$$Q = \begin{bmatrix} I_m & \vec{1}^T \\ \vec{0} & 0 \end{bmatrix} = I_{(m+1)} + \begin{bmatrix} O_m & \vec{1}^T \\ \vec{0} & 1 \end{bmatrix} = I + U,$$

where $U = [O_{p\times m}\ \vec{1}^T_{p\times 1}]$.

From (2.1'), we have

$$\begin{aligned} Q(I + E^\mu) &= (I + U)(I + E^\mu) \\ &= (I + E^\mu) + U(I + E^\mu) = (I + E^\mu) + W, \end{aligned} \tag{A.6}$$

where $W = [\vec{0}^T \dots \vec{0}^T \vec{1}^T \vec{0}^T \dots \vec{0}^T \vec{1}^T]$, in which column $(m - \mu)$ and the last column are $\vec{1}^T$ s.

From the above equations, we have

$$\begin{aligned} &((I + E^\mu)^{-1} + V)Q(I + E^\mu) \\ &= ((I + E^\mu)^{-1} + V)((I - E^\mu) + W) = (I + E^\mu)^{-1}(I + E^\mu) \\ &\quad + (I + E^\mu)^{-1}W + V(I + E^\mu) + VW. \end{aligned} \tag{A.7}$$

From (2.6), we have $(I + E^\mu)^{-1}(I + E^\mu) = Q.$ (A.8)

From the definitions of V and W, we have

$$VW = [\vec{v}^{1T}\vec{v}^{1} \dots \vec{v}^{1T}\vec{0}^{T}][\vec{0}^{T} \dots \vec{0}^{T}\vec{1}^{T}\vec{0}^{T} \dots \vec{0}^{T}\vec{1}^{T}].$$

The columns of VW are columns of 0s, except column $(m - \mu)$ and the last column (i.e., column m), where the column order is $0, 1, \dots, m$ and the total number of columns is $(m + 1)$. On the other hand, each row of V has no 1 or has an even number (m) of 1s and nonzero columns of W are columns of 1s. Thus, columns $(m - \mu)$ and m are columns of 0s. Therefore,

$$VW = O_{p\times p}. \tag{A.9}$$

From (A.6), we have $(I - E^\mu)^{-1}W = (I + E^\mu)^{-1}[\vec{0}^T \dots \vec{0}^T \vec{1}^T \vec{0}^T \dots \vec{0}^T \vec{1}^T].$

The columns of $(I + E^{\mu})^{-1}W$ are columns of 0s, except column $(m - \mu)$ and the last column. From the definition of $\vec{v}$, these two columns should be $\vec{v}$ s. Thus, we have

$$(I - E^{\mu})^{-1}W = [\vec{0}^T \ldots \vec{0}^T \vec{v}^T \vec{0}^T \ldots \vec{0}^T \vec{v}^T].$$

Now, we consider $V(I + E^{\mu}) = [\vec{v}^T \vec{v}^T \ldots \vec{v}^T \vec{0}^T](I + E^{\mu})$. It is known that each column and each row of $(I + E^{\mu})$ has two 1s. On the other hand, the bottom row (i.e., row m) of $(I + E^{\mu})$ has two 1s and they are in column $m - \mu$ and the last column. This means that in the above m rows, all columns except column $m - \mu$ and the last column have two 1s. Therefore,

$$V(I + E^{\mu}) = [\vec{0}^T \ldots \vec{0}^T \vec{v}^T \vec{0}^T \ldots \vec{0}^T \vec{v}^T]. \qquad \text{(A.11)}$$

From (A.7)-(A.11), we have $((I + E^{\mu})^{-1} + V)Q(I + E^{\mu}) = Q$ that means $\overrightarrow{(I + E^{\mu})^{-1}} = (I + E^{\mu})^{-1} + V$.

It can be easily checked that $\overrightarrow{(I + E^{\mu})^{-1}}$ is a nonsingular matrix.

Proof of Lemma 2.8

Let $(I + E^{\mu}) \times \vec{v}^T = \vec{u}^T = (u_0, u_1, \ldots, u_m)^T$. From (2.1'), we have

$$u_i \quad v_i + v_{(i-\mu)} \qquad for \ \ 0 \le i \le m. \qquad \text{(A.12)}$$

Thus, μ additions are needed. Let $E^{\mu} \times \vec{v}^T = \vec{u}^T$. From (2.1'), we have

$$u_i = v_{(i \ \mu)} \qquad for \ \ 0 \le i \le m. \qquad \text{(A.13)}$$

Thus, there is no need for addition.

Proof of Lemma 2.9

Let $(I + E)^{-1} \times \vec{v}^T = \vec{x}^T$. From Lemma 2.5, we have

$$x_i = \sum_{k=0}^{i} v_k \qquad for \ \ 0 \le i \le m. \qquad \text{(A.14)}$$

It can be implemented as $for \quad i = 0 \ to \ m \quad do \quad x_i = x_{i-1} + v_i,$ where $x_{-1} = 0$.

Thus, m additions are needed to obtain $\vec{x}$.

Let $(I + E^{\mu})^{-1} \times \vec{v}^T = \vec{u}^T$. From (2.1'), we have

$$(I + E^{\mu})^{-1} \times \vec{v}^T = \Pi_{\mu}^T (I + E)^{-1} \Pi_{\mu} \vec{v}^T.$$

From (A.14) and Lemma 2.6, to implement the above operation there needs to be $m$additions. Thus, it is clear that to implement $((I + E^{\mu})^{-1} \times \vec{v}^T + \vec{w})$, $(m + p)$ additions are needed.

Acknowledgment

The authors would like to thank Mr. Hua Qian and Ms. Anna Robin for helpful discussions.

REFERENCES

1. D. Patterson, G. Gibson and R. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID),"*Proc. ACM SIGMOD '88,* pp. 109-116, June 1988.

2. P. Elias, "Coding for Two Noisy Channels," *Proc. Third London Symp. Information Theory,* pp. 61-76, Sept. 1955.

3. M. Blahut, *Algebraic Codes for Data Transmission.* Cambridge Univ. Press, 2003.

4. M. Blaum, "A Class of Byte-Correcting Array Code," IBM Research Report, RJ, 5652, 57151, May 1987.

5. M. Blaum, J. Bradt, J. Bruck and J. Menon, "EVEN-ODD: An Efficient Scheme for Toleranting Double Disk Failures in RAID Architectures," *IEEE Trans. Computers,* vol. 44, no. 2, pp. 192-202, Feb. 1995.

6. M. Blaum, J. Bruck and A. Vardy, "MDS Array Codes with Independent Parity Symbols," *IEEE Trans. Information Theory,* pp. 529-542, Mar. 1996.

7. M. Blaum, H. Hao, R. Mattson and J. Menon, "A Coding Technique for Double Disk Failures in Disk Arrays," US Patent 5,271,012, Dec. 1993.

8. M. Blaum and R. Roth, "New Array Codes for Multiple Phased Burst Correction," *IEEE Trans. Information Theory,* pp. 66-77, Jan. 1993.

9. M. Blaum and R. Roth, "On Lowest-Density MDS Codes," *IEEE Trans. Information Theory,* pp. 46-59, Jan. 1999.

10. T. Fuja, C. Heegard and M. Blaum, "Cross Parity Check Convolutional Code," *IEEE Trans. Information Theory,* pp. 1264-1276, July 1989.

11. R. Goodman, R.J. McEliece and M. Sayano, "Phased Burst Correcting Array Codes," *IEEE Trans. Information Theory,* pp. 684-693, Mar. 1993.

12. L. Xu and J. Bruck, "X-Code: MDS Array Codes with Optimal Encoding," *IEEE Trans. Information Theory,* pp. 272-276, Jan. 1999.

13. L. Xu, V. Bohossian, J. Bruck and D.G. Wagner, "Low-Density MDS Codes and Factors of Complete Graphs," *IEEE Trans. Information Theory,* pp. 1817-1826, Sept. 1999.

14. M. Blaum, J. Bradt, J. Bruck, J. Menon and A. Vardy, "The EVENODD Code and Its Generalization: An Efficient Scheme for Tolerating Multiple Disk Failures in RAID Architectures," *High Performance Mass Storage and Parallel I/O,* chapter 14, 2002.

15. F.J. MacWilliams and N.J. A. Sloane, *The Theory of Error-Correcting Codes.* Elsevier Science Publishers B.V., 1977.