GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322165

Generalized Expanded-Blaum-Roth Codes and Their Efficient Encoding/Decoding

You Wu[†‡], Hanxu Hou[‡⋆], Yunghsiang S. Han[‡], Patrick P. C. Lee[§] and Guojun Han[†]
[†] School of Information Engineering, Guangdong University of Technology
[‡] School of Electrical Engineering & Intelligentization, Dongguan University of Technology
[§] Department of Computer Science and Engineering, The Chinese University of Hong Kong

***Abstract*— Expanded-Blaum-Roth (EBR) code encodes a $(p-1)\times k$ information array into a $p\times p$ array such that any bit in a column can be recovered within the column and any k out of p columns can retrieve all $(p-1)\times k$ information bits, where p is a prime number. In this paper, we generalize the construction of EBR code with a more flexible parameter, i.e., the number of bits stored in a column in the proposed construction can be not only a prime number but also an even number. In addition, we present an efficient encoding/decoding method for the proposed generalized EBR codes based on the LU factorization of Vandermonde matrix. We show that the proposed encoding/decoding method has less computational complexity than the existing method. Moreover, we show that the minimum symbol distance of generalized EBR codes is the same as that of EBR code for some parameters.**

***Index Terms*—Array codes, Expanded-Blaum-Roth codes, efficient decoding.**

I. Introduction

Modern distributed storage systems employ erasure codes to maintain data availability and reliability. Redundancy is necessary to provide high data reliability, and two main methods of introducing redundancy are replication and erasure coding. Compared to replication technology, erasure coding can deliver higher data reliability with much lower storage overhead. With erasure coding, the data file is divided into k information symbols of the same size, which are encoded to obtain r parity symbols. Both k information symbols and r parity symbols are stored in the storage system to achieve high data reliability.

Array codes that consist of $m\times n$ arrays have been widely used in storage systems such as Redundant Array of Independent Disk (RAID) [1] to enhance data reliability. Consider a binary array code of size $m\times n$, in which each element stores one bit in the array code. Among the n columns, the first k columns store information bits to form k information columns, and the remaining $r=n-k$ columns store parity bits to form r parity columns.

Maximum distance separable (MDS) array code is a special class of array code, where any k out of the n columns can retrieve all $m\times k$ information bits stored in the k information columns. There are many existing MDS array codes, and most of them are designed to tolerate two or three failed columns. For example, EVENODD [2], [3] and RDP [4] are two important codes that can correct double disk failures. Star code [5] and triple-fault-tolerance code [6] can correct three disk failures. Generalized RDP codes [7], generalized EVENODD codes [8], Blaum-Roth (BR) codes [9], codes [10] and Rabin-like codes [11], [12] are array codes that can tolerate four or more column failures. In addition to a column failure, i.e., all m bits are failed in the failure column, another failure pattern is that one bit of a column is failed. Recently, Expanded-Blaum-Roth (EBR) [13] codes were proposed to efficiently repair both column failures and one bit failure within any column by adding a parity bit for each column of BR codes. The extensions of BR codes and extended EVENODD codes to be the corresponding codes that can recover one or more bits within a column are given in [14].

EBR(p,r) codes can be represented by a $p\times p$ array, where p is a prime number and $1\le r<p$. The $p\times p$ array is obtained by encoding the $(p-1)\times k$ information array. For $i=0,1,\ldots,p-1$ and $j=0,1,\ldots,p-1$, denote by $a_{i,j}$ the entry in row i and column j of the $p\times p$ array, where the $k(p-1)$ information bits are the bits in the entries with $i=0,1,\ldots,p-2$ and $j=0,1,\ldots,k-1$. Given the $(p-1)\times k$ information array, the parity bit $a_{p-1,j}$ in the last row of information column j, $0\le j\le k-1$, are computed by summing all $p-1$ information bits in the same column, i.e.,

$$a_{p-1,j}=\sum_{u=0}^{p-2}a_{u,j}. \tag{1}$$

Furthermore, the p bits $a_{0,j},a_{1,j},\ldots,a_{p-1,j}$ in column j are represented as an *information polynomial*

$$a_j(x)=a_{0,j}+a_{1,j}x+\ldots+a_{p-1,j}x^{p-1}$$

over $\mathbb{F}_2[x]/(1+x^p)$. Similarly, the p bits in column j with $j=k,k+1,\ldots,p-1$ are also represented as a *parity polynomial* $a_j(x)$ over $\mathbb{F}_2[x]/(1+x^p)$. The relationship between information polynomials and parity polynomials is given as

$$\mathbf{H}_{r\times p}\cdot\begin{bmatrix}a_0(x) & a_1(x) & \cdots & a_{p-1}(x)\end{bmatrix}^T=\mathbf{0}^T,$$

This work was partially supported by the National Natural Science Foundation of China (No. 61701115, 61671007, 61871136), Start Fund of Dongguan University of Technology (No. GB200902-19, KCYXM2017025), Research Grants Council of HKSAR (GRF 14216316 and AoE/P-404/18), Innovation and Technology Fund (ITS/315/18FX), the Quality Engineering Project of 2018 DGUT (No. 201802016), 2019 Guangdong Province Teaching Quality and Teaching Reform (No. 2063016020).

⋆ corresponding author.

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

where $\mathbf{0}^T$ is the all-zero column of length r and $\mathbf{H}_{r\times p}$ is the $r \times p$ parity-check matrix

$$\mathbf{H}_{r\times p} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & x & x^2 & \cdots & x^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{r-1} & x^{2(r-1)} & \dots & x^{(r-1)(p-1)} \end{bmatrix}.$$

In solving the above linear equations, the requirement in (1) should also be satisfied for all $j = k, k+1, \dots, p-1$.

In this paper, we give a more general construction of EBR codes, called Generalized Expanded-Blaum-Roth (GEBR) codes, that can support much more parameters compared to the construction of EBR codes in [13]. Particularly, the EBR codes can be viewed as special cases of the proposed GEBR codes. In addition, we present an efficient decoding method for the proposed GEBR code based on the LU factorization of Vandermonde matrix and show that the encoding complexity of GEBR codes with the LU decoding method is less than that of the existing methods [13], [14]. Moreover, we show that GERB codes have the same minimum symbol distance as that of EBR codes for some parameters.

II. Generalized Expanded-Blaum-Roth Codes

A. Construction

The proposed GEBR code is an array of size $p\tau \times (k+r)$ by encoding $k(p-1)\tau$ information bits, where τ is a positive integer and p is an odd prime number. Denote by $s_{i,j}$ the bit in row i and column j in this array, where $i = 0, 1, \dots, p\tau - 1$, $j = 0, 1, \dots, k+r-1$. Given $k(p-1)\tau$ information bits $s_{i,j}$ with $i = 0, 1, \dots, (p-1)\tau - 1$ and $j = 0, 1, \dots, k-1$, we compute τ parity bits $s_{(p-1)\tau,j}, s_{(p-1)\tau+1,j}, \dots, s_{p\tau-1,j}$ for column j as

$$s_{(p-1)\tau+\mu,j} = \sum_{\ell=0}^{p-2} s_{\ell\tau+\mu,j}, \qquad (2)$$

where $\mu = 0, 1, \dots, \tau - 1$. We represent $p\tau$ bits $s_{0,j}, s_{1,j}, \dots, s_{p\tau-1,j}$ in column j as the polynomial

$$s_j(x) = s_{0,j} + s_{1,j}x + s_{2,j}x^2 + \dots + s_{p\tau-1,j}x^{p\tau-1},$$

over $\mathbb{F}_2[x]/(1+x^{p\tau})$, where $j = 0, 1, \dots, k+r-1$. Thus, we obtain k information polynomials $s_0(x), s_1(x), \dots, s_{k-1}(x)$ and r parity polynomials $s_k(x), s_{k+1}(x), \dots, s_{k+r-1}(x)$. We can compute the r parity polynomials by solving the following linear equations

$$\mathbf{H}_{r\times(k+r)} \cdot \begin{bmatrix} s_0(x) & s_1(x) & \cdots & s_{k+r-1}(x) \end{bmatrix}^T = \mathbf{0}^T \qquad (3)$$

over the quotient ring $\mathbb{F}_2[x]/(1+x^{p\tau})$, where $\mathbf{H}_{r\times(k+r)}$ is the $r \times (k+r)$ parity-check matrix

$$\mathbf{H}_{r\times(k+r)} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & x & x^2 & \cdots & x^{k+r-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x^{r-1} & x^{2(r-1)} & \dots & x^{(r-1)(k+r-1)} \end{bmatrix}. \qquad (4)$$

Note that we have more than one solution of $s_k(x), s_{k+1}(x), \dots, s_{k+r-1}(x)$ in the above equations, we need to choose one solution such that the condition in (2) is satisfied for $j = k, k+1, \dots, k+r-1$. We denote the generalized Expanded-Blaum-Roth codes defined in (3) as $\text{GEBR}(p, k, r, \tau)$. Note that the EBR code proposed in [13] is a special case as $\text{GEBR}(p, k, r = p-k, \tau = 1)$.

Let $R_{p\tau}$ be the quotient ring $\mathbb{F}_2[x]/(1+x^{p\tau})$. A polynomial in $R_{p\tau}$ is a polynomial of degree less than $p\tau$ with coefficients in $\mathbb{F}_2$. The ring $R_{p\tau}$ has been discussed in [6], [15], [16] and has been used to design regenerating codes [17], [18] for computational complexity reduction.

Let $C_{p\tau}$ be a subset of $R_{p\tau}$ with polynomials being a multiple of $1+x^\tau$, and

$$C_{p\tau} = \{a(x)(1+x^\tau) \bmod (1+x^{p\tau}) | a(x) \in R_{p\tau}\}.$$

The next lemma shows the necessary and sufficient condition of $s_j(x) \in C_{p\tau}$.

Lemma 1. *[16, Theorem 1] The polynomial $s_j(x)$ is in $C_{p\tau}$ if and only if the coefficients of polynomial $s_j(x)$ satisfy* (2).

By Lemma 2 in [16], we have that the ring $R_{p\tau}$ is isomorphic to the direct sum of two rings $\mathbb{F}_2[x]/(1+x^\tau)$ and $\mathbb{F}_2[x]/M_{p\tau}(x)$, where

$$M_{p\tau}(x) = 1 + x^\tau + \dots + x^{(p-1)\tau}.$$

Therefore, the ring $C_{p\tau}$ is isomorphic to $\mathbb{F}_2[x]/M_{p\tau}(x)$, where the isomorphism $\theta : C_{p\tau} \to \mathbb{F}_2[x]/M_{p\tau}(x)$ is defined as $\theta(f(x)) = f(x) \bmod M_{p\tau}(x)$ and the inverse function is $\phi(f(x)) = f(x)(1 + M_{p\tau}) \bmod (1+x^{p\tau})$.

By (3) and (4), we have

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & x & \cdots & x^{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x^{r-1} & \cdots & x^{(r-1)(k-1)} \end{bmatrix} \begin{bmatrix} s_0(x) \\ s_1(x) \\ \vdots \\ s_{k-1}(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x^k & x^{k+1} & \dots & x^{k+r-1} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(r-1)k} & x^{(r-1)(k+1)} & \dots & x^{(r-1)(k+r-1)} \end{bmatrix} \begin{bmatrix} s_k(x) \\ s_{k+1}(x) \\ \vdots \\ s_{k+r-1}(x) \end{bmatrix} \qquad (5)$$

For $j = 0, 1, \dots, k-1$, the coefficients of the information polynomial $s_j(x)$ satisfy (2) and, according to Lemma 1, $s_j(x) \in C_{p\tau}$. We can compute the r parity polynomials $s_k(x), s_{k+1}(x), \dots, s_{k+r-1}(x)$ by solving (5) over $R_{p\tau}$ if the determinant of the $r \times r$ matrix on the right-hand side of (5) is invertible over $\mathbb{F}_2[x]/M_{p\tau}(x)$. More generally, we present the condition of solving the k information polynomials from any k out of $k+r$ polynomials in next theorem.

Theorem 2. *If two polynomials $1 + x^i$ and $M_{p\tau}(x)$ are relatively prime over $\mathbb{F}_2[x]$, where $i = 1, 2, \dots, k+r-1$, then we can compute the other r polynomials from any k out of $k+r$ polynomials $s_0(x), s_1(x), \dots, s_{k+r-1}(x)$.*

Proof. It is sufficient to show that the determinant of any $r \times r$ sub-matrix of $\mathbf{H}_{r\times(k+r)}$ in (4) is invertible over $\mathbb{F}_2[x]/M_{p\tau}(x)$. Since any $r \times r$ sub-matrix of $\mathbf{H}_{r\times(k+r)}$

is a Vandermonde matrix, the determinant can be written as the multiplication of $r(r-1)/2$ factors $1+x^i$, where $i \in \{1,2,\ldots,k+r-1\}$. In other words, the determinant can be viewed as a polynomial in $\mathbb{F}_2[x]/M_{p\tau}(x)$ after modulo $M_{p\tau}(x)$, and is invertible over the ring $\mathbb{F}_2[x]/M_{p\tau}(x)$. Therefore, we can compute the other r polynomials from any k out of $k+r$ polynomials, if $1+x^i$ is invertible over $\mathbb{F}_2[x]/M_{p\tau}(x)$ for all $i=1,2,\ldots,k+r-1$. □

If τ is a power of 2, we have

$$M_{p\tau}(x) = 1+x^\tau+\ldots+x^{(p-1)\tau} = (1+x+\ldots+x^{p-1})^\tau.$$

Note that the polynomial $1+x+\ldots+x^{p-1}$ is irreducible polynomial over $\mathbb{F}_2[x]$ if 2 is a primitive element in $\mathbb{Z}_p$ [17]. Therefore, $a(x)$ is invertible in the ring $\mathbb{F}_2[x]/M_{p\tau}(x)$ if and only if $a(x)$ and $1+x+\ldots+x^{p-1}$ are relatively prime in $\mathbb{F}_2[x]$, when τ is a power of 2. If 2 is a primitive element in $\mathbb{Z}_p$ and τ is a power of p, then $M_{p\tau}(x)$ is irreducible over $\mathbb{F}_2[x]$ [19]. Therefore, GEBR(p,k,r,τ) is MDS for $k+r \leq (p-1)\tau$, if 2 is a primitive element in $\mathbb{Z}_p$ and τ is a power of p.

By Theorem 2, we can compute the r parity polynomials that are in $C_{p\tau}$ if $1+x^i$ is relatively prime to $M_{p\tau}(x)$ over $\mathbb{F}_2[x]$ for all $i=1,2,\ldots,k+r-1$. For $j=0,1,\ldots,k+r-1$, the obtained polynomial $s_i(x) \in C_{p\tau}$ and the coefficients of $s_j(x)$ satisfy (2) according to Lemma 1. Therefore, the proposed GEBR(p,k,r,τ) can recover any one bit in a column by reading other $p-1$ bits in the same column according to (2). In addition, GEBR(p,k,r,τ) can recover up to τ consecutive bits in a column by only reading other bits in the same column. Moreover, GEBR(p,k,r,τ) can recover any r column failures.

B. Vandermonde Matrix over $R_{p\tau}$

Since the ring $C_{p\tau}$ is isomorphic to $\mathbb{F}_2[x]/M_{p\tau}(x)$, we can compute the r parity polynomials over $R_{p\tau}$ by first solving the $r\times r$ Vandermonde linear system over $\mathbb{F}_2[x]/M_{p\tau}(x)$ and then applying the inverse function ϕ.

Let $\mathbf{V}_{r\times r}(\mathbf{a})$ be an $r\times r$ Vandermonde matrix,

$$\mathbf{V}_{r\times r}(\mathbf{a}) = \begin{bmatrix} 1 & x^{a_1} & \cdots & x^{(r-1)a_1} \\ 1 & x^{a_2} & \cdots & x^{(r-1)a_2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x^{a_r} & \cdots & x^{(r-1)a_r} \end{bmatrix},$$

where $\mathbf{a}=[a_1,\ldots,a_r]$ and $a_1,\ldots,a_r$ are distinct integers that range from 1 to $k+r-1$. Let $\mathbf{u}=(u_1(x),\ldots,u_r(x)) \in R_{p\tau}^r$ and $\mathbf{v}=(v_1(x),\ldots,v_r(x)) \in C_{p\tau}^r$. Consider the linear equations

$$\mathbf{u}\mathbf{V}_{r\times r}(\mathbf{a}) = \mathbf{v} \bmod (1+x^{p\tau}). \tag{6}$$

In the next theorem, we show that there are many vectors $\mathbf{u}$ satisfying (6) given $\mathbf{V}_{r\times r}(\mathbf{a})$ and $\mathbf{v}$.

Theorem 3. *Let $a_1,a_2,\ldots,a_r$ be r integers such that the polynomial $x^{a_{i_1}}+x^{a_{i_2}}$ is invertible in the ring $\mathbb{F}_2[x]/M_{p\tau}(x)$ for all $1 \leq i_1 < i_2 \leq r$. All vectors $\mathbf{u}$ that satisfy* (6) *are congruent to each other modulo $M_{p\tau}(x)$.*

Proof. By Lemma 2 in [16], we have an isomorphic

$$\theta(f(x)) = (f(x) \bmod (1+x^\tau), f(x) \bmod M_{p\tau}(x)),$$

where $f(x) \in R_{p\tau}$. Because $1+x^\tau$ and $M_{p\tau}(x)$ are relatively prime polynomials over $\mathbb{F}_2[x]$, by the Chinese remainder theorem, the inverse function θ^{-1} of θ is

$$\begin{aligned}&\theta^{-1}(a(x),b(x))\\ =&a(x)\cdot M_{p\tau}(x)+b(x)\cdot(1+M_{p\tau}(x)) \bmod (1+x^{p\tau}).\end{aligned}$$

Therefore, it is sufficient to solve $\mathbf{u}$ by considering the following two equations

$$\mathbf{u}\mathbf{V}_{r\times r}(\mathbf{a}) = \mathbf{v} \bmod (1+x^\tau), \text{ and} \tag{7}$$

$$\mathbf{u}\mathbf{V}_{r\times r}(\mathbf{a}) = \mathbf{v} \bmod M_{p\tau}(x). \tag{8}$$

Note that (7) can be rewritten as

$$\mathbf{u} \bmod (1+x^\tau)\cdot(\mathbf{V}_{r\times r}(\mathbf{a}) \bmod (1+x^\tau)) = \mathbf{v} \bmod (1+x^\tau).$$

Recall that $v_i(x) \in C_{p\tau}$ for $i=1,2,\ldots,r$, we have $\mathbf{v}$ mod $(1+x^\tau)=\mathbf{0}$. Therefore, there are many solutions $\mathbf{u}'$ and $\mathbf{u}'=\mathbf{0}$ is one of the solutions.

For (8), the determinant of $\mathbf{V}_{r\times r}(\mathbf{a})$ is

$$\det(\mathbf{V}_{r\times r}(\mathbf{a})) = \prod_{i_1<i_2}(x^{a_{i_1}}+x^{a_{i_2}}),$$

which is invertible in the ring $\mathbb{F}_2[x]/M_{p\tau}(x)$ by the assumption. Therefore, we can solve (8) to obtain the unique solution $\mathbf{u}''$. After obtaining the solutions $u_i'(x) \in \mathbb{F}_2[x]/(1+x^\tau)$ and $u_i''(x) \in \mathbb{F}_2[x]/M_{p\tau}$ to (7) and (8), respectively, for all i, we can obtain the solution to (6) by

$$\begin{aligned}&\theta^{-1}(u_i'(x),u_i''(x))\\ =&M_{p\tau}(x)u_i'(x)+(1+M_{p\tau}(x))u_i''(x) \bmod (1+x^{p\tau}).\end{aligned}$$

Therefore, there are many solutions to (6) and are congruent to each other modulo $M_{p\tau}(x)$. □

From Theorem 3, there are many solutions of $\mathbf{u}$ in (6) and one of the solutions satisfies that $\mathbf{u} \in C_{p\tau}^r$. In the next section, we will present an efficient method to solve $\mathbf{u}$ in (6) with all the entries of $\mathbf{u}$ being in $C_{p\tau}$ based on the LU factorization of the Vandermonde matrix. Note that the result in Theorem 1 in [20] can be viewed as a special case of the result in Theorem 3 with $\tau=1$.

III. Efficient Decoding

In this section, we present an efficient decoding method for solving the Vandermonde linear system over $R_{p\tau}$ based on LU factorization of the Vandermonde matrix.

A. Efficient Division by $1+x^b$ over $R_{p\tau}$

We need to first review an efficient decoding algorithm in [16] for dividing by $1+x^b$ over $R_{p\tau}$ before showing the efficient LU decoding method, where b is a positive integer such that $1+x^b$ and $M_{p\tau}(x)$ are relatively prime. Given the integer b and the polynomial $f(x) \in C_{p\tau}$, we want to solve $g(x) \in C_{p\tau}$ from the equation

$$(1+x^b)g(x) = f(x) \bmod (1+x^{p\tau}). \tag{9}$$

The next lemma shows an efficient decoding algorithm for solving $g(x) \in C_{p\tau}$ from (9).

Lemma 4. *[16, Lemma 4] Let b be an integer with $1 \leq b < p\tau$ and the greatest common divisor (GCD) of b and p is $\gcd(b,p) = 1$, and let $\gcd(b,\tau) = a$. We can first compute the coefficients g_j of the polynomial $g(x)$ in (9) with $j = 0, 1, \ldots, a-1$ by*

$$g_j = \sum_{i=\frac{\tau}{a}}^{\frac{2\tau}{a}-1} f_{(j-ib) \bmod p\tau} + \sum_{i=\frac{3\tau}{a}}^{\frac{4\tau}{a}-1} f_{(j-ib) \bmod p\tau} + \cdots + \sum_{i=\frac{(p-2)\tau}{a}}^{\frac{(p-1)\tau}{a}-1} f_{(j-ib) \bmod p\tau} \tag{10}$$

and the other coefficients of $g(x)$ iteratively by

$$g_{b\ell+j} = f_{b\ell+j} + g_{b(\ell-1)+j} \tag{11}$$

with the index ℓ running from 1 to $p\tau/a - 1$ and $j = 0, 1, \ldots, a-1$.

By Lemma 4, there are

$$a\left(\frac{p-1}{\tau} \cdot \frac{\tau}{a} - 1\right) + (p\tau - a) = \frac{3p\tau - \tau - 4a}{2}$$

XORs involved in solving $g(x)$ from (9).

B. LU Decoding Method

We first review the LU factorization of a Vandermonde matrix in [21], and then show the LU decoding algorithm for solving $\mathbf{u}$ from the Vandermonde linear system in (6).

Theorem 5. *[21] For a positive integer r, the $r \times r$ Vandermonde matrix $\mathbf{V}_{r\times r}(\mathbf{a})$ can be factorized into*

$$\mathbf{V}_{r\times r}(\mathbf{a}) = \mathbf{L}_r^{(1)}\mathbf{L}_r^{(2)} \ldots \mathbf{L}_r^{(r-1)}\mathbf{U}_r^{(r-1)}\mathbf{U}_r^{(r-2)} \ldots \mathbf{U}_r^{(1)}$$

where $\mathbf{U}_r^{(\ell)}$ is the upper triangular matrix

$$\mathbf{U}_r^{(\ell)} = \left[\begin{array}{c|cccccc} \mathbf{I}_{r-l-1} & & & \mathbf{0} & & & \\ \hline & 1 & x^{a_1} & 0 & \cdots & 0 & 0 \\ & 0 & 1 & x^{a_2} & \cdots & 0 & 0 \\ \mathbf{0} & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ & 0 & 0 & 0 & \cdots & 1 & x^{a_l} \\ & 0 & 0 & 0 & \cdots & 0 & 1 \end{array}\right]$$

and $\mathbf{L}_r^{(\ell)}$ is the lower triangular matrix

$$\left[\begin{array}{c|ccccc} \mathbf{I}_{t-1} & & & \mathbf{0} & & \\ \hline & 1 & 0 & \cdots & 0 & 0 \\ & 1 & x^{a_{t+1}} + x^{a_t} & \cdots & 0 & 0 \\ \mathbf{0} & \vdots & \vdots & \ddots & \vdots & \vdots \\ & 0 & 0 & \cdots & x^{a_{r-1}} + x^{a_t} & 0 \\ & 0 & 0 & \cdots & 1 & x^{a_r} + x^{a_t} \end{array}\right]$$

for $\ell = 1, 2, \ldots, r-1$, where $t = r - \ell$.

When $r = 3$, the 3×3 Vandermonde matrix $\mathbf{V}_{r\times r}(x^{a_1}, x^{a_2}, x^{a_3})$ can be factorized as

$$\begin{aligned} &\mathbf{L}_3^{(1)}\mathbf{L}_3^{(2)}\mathbf{U}_3^{(2)}\mathbf{U}_3^{(1)} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & x^{a_3} + x^{a_2} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & x^{a_2} + x^{a_1} & 0 \\ 0 & 1 & x^{a_3} + x^{a_1} \end{bmatrix} \cdot \\ &\begin{bmatrix} 1 & x^{a_1} & 0 \\ 0 & 1 & x^{a_2} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & x^{a_1} \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

With the LU factorization of Vandermonde matrix in Theorem 5, we can solve the Vandermonde linear system in (6) by solving the following linear equations

$$\mathbf{u}\mathbf{L}_r^{(1)}\mathbf{L}_r^{(2)} \cdots \mathbf{L}_r^{(r-1)}\mathbf{U}_r^{(r-1)}\mathbf{U}_r^{(r-2)} \cdots \mathbf{U}_r^{(1)} = \mathbf{v}.$$

Algorithm 1 Solving a Vandermonde Linear System

```
Input: positive integer $r$, prime number $p$, integers
    $a_1, a_2, \ldots, a_r$, and $\mathbf{v} = (v_1(x), v_2(x), \ldots, v_r(x)) \in C_{p\tau}^r$.
Output: $\mathbf{u} = (u_1(x), u_2(x), \ldots, u_r(x)) \in C_{p\tau}$ that satisfies
    $\mathbf{u}\mathbf{V}_{r\times r}(\mathbf{a}) = \mathbf{v} \bmod (1 + x^{p\tau})$.
Require: $x^{a_{i_1}} + x^{a_{i_2}}$ is relatively prime to $M_{p\tau}(x)$ over
    $\mathbb{F}_2[x]$ for all $1 \leq i_1 \leq i_2 \leq r$.
 1: $\mathbf{u} \leftarrow \mathbf{v}$
 2: for $i$ from 1 to $r-1$ do
 3:     for $j$ from $r-i+1$ to $r$ do
 4:         $u_j(x) \leftarrow u_j(x) + u_{j-1}(x)x^{a_{i+j-r}}$
 5: for $i$ from $r-1$ down to 1 do
 6:     Solve $g(x)$ from $(x^{a_r} + x^{a_{i+j-r}})g(x) = u_r(x)$ by
        Lemma 4
 7:     $u_r(x) \leftarrow g(x)$
 8:     for $j$ from $r-1$ down to $r-i+1$ do
 9:         Solve $g(x)$ from $(x^{a_j} + x^{a_{r-i}})g(x) = (u_j(x) +$
            $u_{j+1}(x))$ by Lemma 4
10:         $u_j(x) \leftarrow g(x)$
11:     $u_{r-i}(x) \leftarrow u_{r-i}(x) + u_{r-i+1}(x)$
12: return $\mathbf{u} = (u_1(x), ..., u_r(x))$
```

The decoding algorithm of solving the Vandermonde linear system based on the LU factorization of Vandermonde matrix is given in Algorithm 1. In Algorithm 1, steps 2-4 require $r(r-1)/2$ additions and $r(r-1)/2$ multiplications and steps 5-9 require $r(r-1)/2$ additions and $r(r-1)/2$ divisions by factors of the form $x^{a_j} + x^{a_{r-i}}$.

Consider an example of GEBR$(p = 3, k = 3, r = 3, \tau = 2)$. We have three information polynomials

$$\begin{aligned} s_0(x) &= 1 + x + x^4 + x^5, \\ s_1(x) &= x + x^2 + x^3 + x^4, \\ s_2(x) &= x + x^5, \end{aligned}$$

where each polynomial is in $C_{3\cdot 2}$. By (4), the parity-check matrix $\mathbf{H}_{3\times 6}$ is

$$\mathbf{H}_{3\times 6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & x & x^2 & x^3 & x^4 & x^5 \\ 1 & x^2 & x^4 & x^6 & x^8 & x^{10} \end{bmatrix}.$$

Therefore, we can obtain

$$\begin{bmatrix} s_3(x) & s_4(x) & s_5(x) \end{bmatrix} \begin{bmatrix} 1 & x^3 & x^6 \\ 1 & x^4 & x^8 \\ 1 & x^5 & x^{10} \end{bmatrix}$$

$$= \begin{bmatrix} s_0(x)+s_1(x)+s_2(x) \\ s_0(x)+xs_1(x)+x^2s_2(x) \\ s_0(x)+x^2s_1(x)+x^4s_2(x) \end{bmatrix}^T = \begin{bmatrix} 1+x+x^2+x^3 \\ 1+x^2 \\ x+x^5 \end{bmatrix}^T$$

According to Theorem 5, the above Vandermonde matrix can be factorized as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & x^4+x^5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & x^3+x^4 & 0 \\ 0 & 1 & x^3+x^5 \end{bmatrix} \cdot$$

$$\begin{bmatrix} 1 & x^3 & 0 \\ 0 & 1 & x^4 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & x^3 \\ 0 & 0 & 1 \end{bmatrix}.$$

By Algorithm 1, we can solve the three parity polynomials as follows. First, we solve the following linear system

$$\begin{bmatrix} s_3'''(x) & s_4'''(x) & s_5'''(x) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & x^3 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1+x+x^2+x^3 \\ 1+x^2 \\ x+x^5 \end{bmatrix}^T$$

to obtain

$$(s_3'''(x), s_4'''(x), s_5'''(x)) = (1+x+x^2+x^3, 1+x^2, x+x^3).$$

Then, we solve the following linear system

$$\begin{bmatrix} s_3''(x) & s_4''(x) & s_5''(x) \end{bmatrix} \begin{bmatrix} 1 & x^3 & 0 \\ 0 & 1 & x^4 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1+x+x^2+x^3 \\ 1+x^2 \\ x+x^3 \end{bmatrix}^T$$

to obtain

$$\begin{aligned} &(s_3''(x), s_4''(x), s_5''(x)) \\ =&(1+x+x^2+x^3, x^2+x^3+x^4+x^5, 1+x^2). \end{aligned}$$

Next, we solve the following linear system

$$\begin{bmatrix} s_3'(x) & s_4'(x) & s_5'(x) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & x^3+x^4 & 0 \\ 0 & 1 & x^3+x^5 \end{bmatrix}$$

$$= \begin{bmatrix} 1+x+x^2+x^3 \\ x^2+x^3+x^4+x^5 \\ 1+x^2 \end{bmatrix}^T$$

to obtain

$$(s_3'(x), s_4'(x), s_5'(x)) = (x+x^5, 1+x^4, x+x^2+x^3+x^4).$$

Finally, we solve the following linear system

$$\begin{bmatrix} s_3(x) & s_4(x) & s_5(x) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & x^4+x^5 \end{bmatrix}$$

$$= \begin{bmatrix} x+x^5 \\ 1+x^4 \\ x+x^2+x^3+x^4 \end{bmatrix}^T$$

to obtain

$$\begin{aligned} &(s_3(x), s_4(x), s_5(x)) \\ =&(1+x^3+x^4+x^5, x^3+x^5, x+x^2+x^3+x^4). \end{aligned}$$

As the decoding method can be viewed as a special case of encoding method, we only evaluate the encoding complexity. We define the normalized encoding complexity as the ratio of the total number of XORs involved in the encoding procedure to the number of information bits. In the encoding procedure of GEBR codes, we first compute τ parity bits for the first k columns by (2) that takes $k\tau(p-2)$ bits. Then, we compute the multiplication of k polynomials and the $r \times k$ Vandermonde matrix that requires $(k-1)rp\tau$ XORs, and solve the Vandermonde linear system. In solving the $r \times r$ Vandermonde linear system, there are $r(r-1)$ additions that require $r(r-1)p\tau$ XORs, $r(r-1)/2$ divisions that require $(r(r-1)/2) \cdot ((3p\tau - \tau - 4a)/2)$ XORs. Therefore, the normalized encoding complexity of GEBR codes is

$$\frac{\frac{1}{4}r(r-1)(7p\tau - \tau - 4a) + (k-1)rp\tau + k\tau(p-2)}{k(p-1)\tau},$$

where $a = \gcd(b, \tau)$ which is defined in Lemma 4.

The encoding/decoding method of EBR is given in [13], [14], and the normalized encoding complexity is

$$\frac{(\frac{1}{2}r^2 - \frac{5}{2}r + 2^r + rk - 1)p + \frac{1}{4}r(r-1)(3p-5) + k(p-2)}{k(p-1)},$$

where $k = p - r$. We give the comparison of EBR and our proposed codes about the encoding complexity in Table I. The results of Table I show that the proposed LU decoding method has less encoding complexity compared with the decoding methods in [13], [14].

TABLE I
COMPARISON OF ENCODING ALGORITHMS.

p	τ	r	$k = p - r$	EBR	GEBR	Improvment%
5	1	2	3	3.67	3.67	0
5	1	3	2	8.88	8.25	7.0
7	1	4	3	13.22	11.28	14.7
11	1	5	6	14.42	11.48	20.4
17	1	7	10	25.63	15.11	41.0
19	1	8	11	38.69	17.67	54.3
23	1	10	13	100.72	22.88	77.3

IV. MINIMUM SYMBOL DISTANCE

In the following, we consider the symbol distance that is the number of symbols in which two codewords differ.

Theorem 6. *The minimum symbol distance of GEBR(p, k, r, τ) is larger than or equal to* $2(r+1)$.

Proof. Since the code is MDS, there are at least $r+1$ non-zero columns. Together with the result that each non-zero column has a weight of at least 2, we can obtain the result. □

Next, we show that the minimum symbol distance of GEBR(p, k, r, τ) is $2(r+1)$ when $r = 2, 3$ and τ is small enough.

Theorem 7. *When $r = 2$ and $\tau \leq \lfloor \frac{k+1}{2} \rfloor$, the minimum symbol distance of GEBR(p, k, r, τ) is $2(r+1) = 6$. When $r = 3$ and $\tau \leq \lfloor \frac{k+2}{3} \rfloor$, the minimum symbol distance of GEBR(p, k, r, τ) is $2(r+1) = 8$.*

Proof. By Theorem 6, if we can find a codeword composed of $r+1$ non-zero polynomials each with weight 2 and $k-1$ zero polynomials, then the minimum symbol distance is $2(r+1)$.

When $r = 2$, by Theorem 6, each non-zero codeword contains at least three non-zero polynomials. Without loss of generality, suppose that the three non-zero polynomials are $s_\alpha(x), s_\beta(x), s_\gamma(x)$ and the other $k-1$ polynomials are zero, where $0 \leq \alpha < \beta < \gamma \leq k+1$. Suppose that the weight of $s_\alpha(x)$ is 2. According to (4), we obtain that

$$\begin{bmatrix} s_\alpha(x) \\ x^\alpha s_\alpha(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ x^\beta & x^\gamma \end{bmatrix} \cdot \begin{bmatrix} s_\beta(x) \\ s_\gamma(x) \end{bmatrix}.$$

Therefore, we can compute that $s_\gamma(x) = s_\alpha(x) \frac{x^\alpha + x^\beta}{x^\beta + x^\gamma}$ and $s_\beta(x) = s_\alpha(x) \frac{x^\alpha + x^\gamma}{x^\beta + x^\gamma}$. Recall that $\lfloor \frac{k+1}{2} \rfloor \geq \tau$, we have $k+1 \geq 2\tau$. Let $(\alpha, \beta, \gamma) = (0, \tau, 2\tau)$ and $s_0(x) = 1 + x^\tau$, then $s_\tau(x) = x^\tau + x^{p\tau - \tau}$ and $s_{2\tau}(x) = 1 + x^{p\tau - \tau}$. When the weight of $s_\alpha(x)$ is larger than 2, the total weight of the codeword is at least 7 since every non-zero column has a weight of at least 2. Therefore, the minimum symbol distance of GEBR$(p, k, r = 2, \lfloor \frac{k+1}{2} \rfloor \geq \tau)$ is $2(r+1) = 6$.

When $r = 3$, by Theorem 6, we have at least four non-zero polynomials. Without loss of generality, suppose that the four non-zero polynomials are $s_\alpha(x), s_\beta(x), s_\gamma(x), s_\eta(x)$ and the other $k-1$ polynomials are zero, where $0 \leq \alpha < \beta < \gamma < \eta \leq k+2$. We assume that the weight of $s_\alpha(x)$ is 2. By (4), we have

$$\begin{bmatrix} s_\alpha(x) \\ x^\alpha s_\alpha(x) \\ x^{2\alpha} s_\alpha(x) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x^\beta & x^\gamma & x^\eta \\ x^{2\beta} & x^{2\gamma} & x^{2\eta} \end{bmatrix} \cdot \begin{bmatrix} s_\beta(x) \\ s_\gamma(x) \\ s_\eta(x) \end{bmatrix}.$$

Since $\lfloor \frac{k+2}{3} \rfloor \geq \tau$, we have $k+2 \geq 3\tau$. Let $(\alpha, \beta, \gamma, \eta) = (0, \tau, 2\tau, 3\tau)$ and $s_0(x) = 1 + x^\tau$, then we can solve $s_{3\tau}(x) = x^{p\tau-3} + x^{p\tau-2}$, $s_{2\tau}(x) = 1 + x^{p\tau-3}$ and $s_\tau(x) = x\tau + x^{p\tau-2}$ all with weights 2. When the weight of $s_\alpha(x)$ is larger than 2, the total weight of the codeword is at least 9 due to the fact that every non-zero column has a weight of at least 2. Therefore, the minimum symbol distance is $2(r+1) = 8$ when $\lfloor \frac{k+2}{3} \rfloor \geq \tau$ and $r = 3$. □

When $\tau = 1$ and $r = 1, 2, 3$, the minimum symbol distance is discussed in Lemma 30 in [14]. By Lemma 7, we have that the minimum symbol distance with τ is small enough is equal to that of EBR codes when $r = 2, 3$. The minimum symbol distance of $r = 2, \tau > \lfloor \frac{k+1}{2} \rfloor$ and $r = 3, \tau > \lfloor \frac{k+1}{3} \rfloor$ is an open problem. For $r \geq 4$, the minimum symbol distance is an open problem and is one of our future work.

V. Conclusion

In this paper, we propose GEBR codes that generalize the construction of EBR codes with more flexible parameters. We propose an efficient LU decoding method for GEBR codes based on the LU factorization of Vandermonde matrix. We also show that GEBR codes have the same minimum symbol distance as that of EBR codes for some parameters.

References

[1] D. A. Patterson, P. Chen, G. Gibson, and R. H. Katz, "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," in *Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage*, 1989, pp. 112–117.

[2] M. Blaum, J. Brady, J. Bruck, and Jai Menon, "EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures," *IEEE Trans. Computers*, vol. 44, no. 2, pp. 192–202, 1995.

[3] H. Hou and P. P. C. Lee, "A New Construction of EVENODD Codes With Lower Computational Complexity," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1120–1123, 2018.

[4] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-Diagonal Parity for Double Disk Failure Correction," in *Proceedings of the 3rd USENIX Conference on File and Storage Technologies*, 2004, pp. 1–14.

[5] C. Huang and L. Xu, "STAR: An Efficient Coding Scheme for Correcting Triple Storage Node Failures," *IEEE Trans. Computers*, vol. 57, no. 7, pp. 889–901, 2008.

[6] H. Hou, P. P. C. Lee, Y. S. Han, and Y. Hu, "Triple-Fault-Tolerant Binary MDS Array Codes with Asymptotically Optimal Repair," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 839–843.

[7] M. Blaum, "A Family of MDS Array Codes with Minimal Number of Encoding Operations," in *Proc. IEEE Int. Symp. Inf. Theory*, 2006.

[8] M. Blaum, J. Brady, J. Bruck, J. Jai Menon, and V. Alexander, "The EVENODD Code and its Generalization: An Effcient Scheme for Tolerating Multiple Disk Failures in RAID Architectures," in *High Performance Mass Storage and Parallel I/O*. Wiley-IEEE Press, 2002, ch. 8, pp. 187–208.

[9] M. Blaum and R. M. Roth, "New Array Codes for Multiple Phased Burst Correction," *IEEE Trans. Information Theory*, vol. 39, no. 1, pp. 66–77, 1993.

[10] H. Hou, K. W. Shum, M. Chen, and H. Li, "New MDS Array Code Correcting Multiple Disk Failures," in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 2369–2374.

[11] G. L. Feng, R. H. Deng, F. Bao, and J.-C. Shen, "New Efficient MDS Array Codes for RAID. Part II. Rabin-Like Codes for Tolerating Multiple ($\geq$ 4) Disk Failures," *IEEE Trans. Computers*, vol. 54, no. 12, pp. 1473–1483, 2005.

[12] H. Hou and Y. S. Han, "A New Construction and an Efficient Decoding Method for Rabin-Like Codes," *IEEE Trans. Communications*, vol. 66, no. 2, pp. 521–533, 2018.

[13] M. Blaum, V. Deenadhayalan, and S. Hetzler, "Expanded Blaum-Roth Codes With Efficient Encoding and Decoding Algorithms," *IEEE Communications Letters*, vol. 23, no. 6, pp. 954–957, 2019.

[14] M. Blaum and S. R. Hetzler, "Array Codes with Local Properties," *IEEE Trans. Information Theory*, vol. 66, no. 6, pp. 3675–3690, 2020.

[15] H. Hou and Y. S. Han, "A Class of Binary MDS Array Codes with Asymptotically Weak-Optimal Repair," *Science China(Information Sciences)*, vol. 61, no. 10, pp. 52–62, 2018.

[16] H. Hou, Y. S. Han, P. P. C. Lee, Y. Hu, and H. Li, "A New Design of Binary MDS Array Codes with Asymptotically Weak-Optimal Repair," *IEEE Trans. Information Theory*, vol. 65, no. 11, pp. 7095–7113, 2019.

[17] H. Hou, K. W. Shum., M. Chen, and H. Li, "BASIC Codes: Low-Complexity Regenerating Codes for Distributed Storage Systems," *IEEE Trans. Information Theory*, vol. 62, no. 6, pp. 3053–3069, 2016.

[18] H. Hou, Y. S. Han, P. P. C. Lee, and Q. Zhou, "New Regenerating Codes over Binary Cyclic Codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 216–220.

[19] T. Itoh, "Characterization for a Family of Infinitely Many Irreducible Equally Spaced Polynomials," *Information Processing Letters*, vol. 37, no. 5, pp. 273–277, 1991.

[20] H. Hou, Y. S. Han, K. W. Shum, and H. Li, "A Unified Form of EVENODD and RDP Codes and Their Efficient Decoding," *IEEE Trans. Communications*, vol. 66, no. 11, pp. 5053–5066, 2018.

[21] S.-L. Yang, "On The LU factorization of The Vandermonde Matrix," *Discrete Applied Mathematics*, vol. 146, no. 1, pp. 102–105, 2005.