New MDS Array Code Correcting Multiple Disk Failures

Hanxu Hou[†§], Kenneth W. Shum[‡], Minghua Chen[§] and Hui Li[†*]
[†] Shenzhen Eng. Lab of Converged Networks Technology, Shenzhen Key Lab of Cloud Computing Tech. and App.,
Peking University Shenzhen Graduate School
[§] Department of Information Engineering, the Chinese University of Hong Kong
[‡] Institute of Network Coding, the Chinese University of Hong Kong,
Shenzhen Key Lab. of Network Coding Key Tech. and Application, China.

***Abstract*— We present a new family of maximal-distance separable (MDS) array codes which can tolerate five disk failures. The encoding is based on bit-wise exclusive OR (XOR) and bit-wise cyclic shifts, and hence is amenable to practical implementation. Efficient repair method for correcting up to two disk failures is also given. The proposed coding scheme provides a larger spectrum of parameters, with comparable encoding and repairing complexities in compare with existing MDS array codes, such as the row-diagonal parity (RDP) code and the EVENODD code.**

***Index Terms*—Array code, RAID, efficient repairing.**

I. Introduction

Array codes are error-correcting codes with application to storage systems such as Redundant Arrays of Inexpensive Disks (RAID) architectures [1]. A binary array code consists of arrays of size $m \times n$, with each element of an array storing one bit. Among the n columns, k columns store the information bits and r columns store the parity-check bits ($n = r + k$). The m bits in a column are stored in the same data disk, or storage node. When a storage node fails, the corresponding column of the array code is considered to be an *erasure*. If the array code can tolerate any r disk erasures, then it is called a Maximum-Distance Separable (MDS) array code. In other words, in an MDS array code, the information bits can be recovered from any k columns.

There are quite a lot of existing constructions of MDS array codes [2]–[10]. Most of them are designed to correct two or three disk erasures. For example, the row-diagonal parity (RDP) code [2] and the EVENODD code [3] can tolerate any two disk erasures. MDS array codes such as the generalized EVENODD [6], STAR [7], Triple-Star [8] and generalized RDP [9] can tolerate any three erasures.

The Blaum-Roth (BR) code [11], the Blaum-Bruck-Vardy (BBV) code [4], and the Rabin-like code [10] are examples of array codes which can tolerate $r \geq 4$ disk erasures. Both BR and BBV codes are based on the arithmetic of the polynomials modulo a specific polynomial $M_p(x) := x^{p-1} + x^{p-2} + \cdots + x + 1$, which is irreducible over the binary field under certain technical condition. Encoding and decoding can be done by simple bit-wise cyclic shifts and XOR operations on the columns. It is proved in [11] that the BR is MDS for all admissible parameters, and in [4] that the BBV code is MDS for $r \leq 8$ for all but finitely many admissible parameters. The Rabin-like code in [10] is an MDS array code that is based on circular permutation matrices and can tolerate four or more concurrent failures.

In this paper we give a new class of array codes, and prove that it is MDS for $r \leq 5$, i.e., it can tolerate up to 5 disk failures. Like the BR and BBV codes, the encoding and decoding of the new class of MDS array codes require simple bit-wise cyclic shifts and XOR operations. But unlike the BR and BBV codes, arithmetic of polynomials modulo $x^p - 1$ is used instead. Since in practice it is more likely to have single or double disk failures instead of five disk failures, we give an efficient method for repairing one and two disk failures. We also compare the encoding and repairing complexity of RDP, EVENODD, BBV code, Rabin-like code and the proposed array code.

Another related work can be found in [12], which proposes a framework of designing codes employing XOR and bit-wise cyclic shifts. In this paper, we focus on a specific encoding matrix.

II. Preliminaries

A. Generalized Vandermonde Matrix

For $i = 1, 2, \ldots, k$ and variables $a_1, \ldots, a_k$, a *generalized Vandermonde matrix* is a matrix in the form

$$\mathbf{V}_i(a_1, \ldots, a_k) := \begin{bmatrix} 1 & a_1 & \cdots & a_1^{i-1} & a_1^{i+1} & \cdots & a_1^k \\ 1 & a_2 & \cdots & a_2^{i-1} & a_2^{i+1} & \cdots & a_2^k \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_k & \cdots & a_k^{i-1} & a_k^{i+1} & \cdots & a_k^k \end{bmatrix}.$$

When $i = k$, the generalized Vandermonde matrix reduces to the usual Vandermonde matrix.

This work was partially supported by the National Basic Research Program of China (No.2012CB315904), NSFC61179028, by a grant from University Grants Committee of Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08), by Peacok Scheme (KQCX20130628164008004), Shenzhen, China, and by the Shenzhen Key Laboratory of Network Coding Key Technology and Application, Shenzhen, China (ZSDY20120619151314964).

[*] Corresponding author.

Theorem 1. *For $i = 1, 2, \ldots, k$, the determinant of the generalized Vandermonde matrix $\mathbf{V}_i(a_1, \ldots, a_k)$ is given by*

$$\det \mathbf{V}_i(a_1, \ldots, a_k) = \sigma_{k-i}(a_1, \ldots, a_k) \det \mathbf{V}_k(a_1, \ldots, a_k),$$

where $\sigma_h(a_1, a_2, \cdots, a_k)$ denotes the h-th elementary symmetric polynomials

$$\sigma_h(a_1, a_2, \ldots, a_k) := \sum_{1 \le j_i < j_2 < \cdots < j_h \le k} a_{j_1} a_{j_2} \cdots a_{j_h}.$$

For example, we have

$$\begin{aligned} \det \mathbf{V}_2(a_1, a_2, a_3) &= \sigma_1(a_1, a_2, a_3) \cdot \det \mathbf{V}_3(a_1, a_2, a_3) \\ &= (a_1 + a_2 + a_3)(a_3 - a_2)(a_3 - a_1)(a_2 - a_1). \end{aligned}$$

A proof of Theorem 1 can be found in [13].

B. Erdős-Heilbronn Conjecture on Restricted Sum Sets

Let $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ denote the residues of integers modulo p. For any subset S of $\mathbb{Z}_p$, the 2-fold *restricted sum set* of S, denoted by $2^\wedge S$, is defined as the set

$$2^\wedge S := \{s_1 + s_2 \bmod p : s_1, s_2 \text{ are distinct elements in } S\}.$$

For example, for $p = 11$ and $S = \{1, 4, 5, 8\}$, the restricted sum set $2^\wedge S$ is equal to $\{1, 2, 5, 6, 9\}$. The Erdős-Heilbronn conjecture states that the size of a restricted sum set cannot be too small.

Theorem 2. *For any subset S of $\mathbb{Z}_p$, we have*

$$|2^\wedge S| \ge \min\{p, 2|S| - 3\},$$

where $|S|$ denotes the cardinality of set S and p is a prime.

The Erdős-Heilbronn conjecture was proved by Dias da Silva and Hamidoune [14]. An alternate algebraic proof can be found in [15].

III. Construction

In this section, we show how our proposed new MDS array code is constructed. Let k and r be positive integers and p be a prime larger than or equal to $\max\{k, r\}$. The array code to be constructed has $p - 1$ rows and $k + r$ columns. The first k columns store the information bits and are called the *information columns*. The remaining r columns store the parity-check bits and are called the *parity columns*. We label the rows from 0 to $p - 2$, the information columns from 0 to $k - 1$, and the parity columns from 0 to $r - 1$. For $i = 0, 1, \ldots, p-2$ and $j = 0, 1, \ldots, k-1$, we let $s_{i,j}$ denote the i-th bit in the j-th information column. For $i = 0, 1, \ldots, p-2$ and $j = 0, 1, \ldots, r-1$, we let $c_{i,j}$ denote the i-th bit in the j-th parity column.

We define the following short-hand notations:

$$s_{p-1,j} := s_{0,j} + s_{1,j} + \cdots + s_{p-2,j},$$

for $j = 0, 1, \ldots, k-1$. We call $s_{p-1,j}$ the *column parity-check bit* associated with the j-th column.

Remark: We do not store the column parity-check bits in the data disk. It is present only for notational convenience.

Definition. We define an array code $\mathcal{C}(k, r, p)$ by specifying the parity-check bits,

$$c_{i,j} = \sum_{\ell=0}^{k-1} s_{\langle i - j\ell \rangle_p, \ell} \tag{1}$$

for $i = 0, 1, \ldots, p-2$ and $j = 0, 1, \ldots, r-1$, where $\langle x \rangle_p$ denotes the remainder of x when we divide x by p, i.e., $\langle x \rangle_p$ is the unique integers in $\{0, 1, \ldots, p-1\}$ such that $x - \langle x \rangle_p$ is divisible by p. The addition is the XOR operation.

The first parity column is called the *row parity column*, as a parity-check bit in the row parity column is the XOR of the k information bits in the corresponding row, i.e.,

$$c_{i,0} = s_{i,0} + s_{i,1} + \cdots + s_{i,k-1},$$

for $i = 0, 1, \ldots, p-2$. The other $r-1$ parity columns are called the *diagonal parity columns*. The integer j can be interpreted as the "slope" of the associated parity column. Table I gives an example of the array code $\mathcal{C}(4, 3, 5)$.

In the following, we give an equivalent algebraic description, which will be useful in proving the MDS property. For $j = 0, 1, \ldots, r-1$, we define

$$c_{p-1,j} := c_{0,j} + c_{1,j} + \cdots + c_{p-2,j},$$

which is referred to as *column parity-check bit* associated to the j-th parity column. It is convenient to append an auxiliary row, consisting only of the parity-check bits $s_{p-1,0}, \ldots, s_{p-1,k-1}, c_{p-1,0}, \ldots, c_{p-1,r-1}$, at the bottom of the array,

$$\left[\begin{array}{cccccc} s_{0,0} & \cdots & s_{0,k-1} & c_{0,0} & \cdots & c_{0,r-1} \\ s_{1,0} & \cdots & s_{1,k-1} & c_{1,0} & \cdots & c_{1,r-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{p-2,0} & \cdots & s_{p-2,k-1} & c_{p-2,0} & \cdots & c_{p-2,r-1} \\ \hline s_{p-1,0} & \cdots & s_{p-1,k-1} & c_{p-1,0} & \cdots & c_{p-1,r-1} \end{array}\right].$$

Let R_p denote the quotient ring $\mathbb{F}_2[x]/(x^p + 1)$. An element in R_p can be represented by a polynomial of the form $a_0 + a_1 x + \cdots + a_{p-1} x^{p-1}$, with coefficients drawn from the binary field $\mathbb{F}_2$. Addition is the usual term-wise addition, and multiplication is performed modulo $x^p + 1$. In R_p, multiplication by x can be interpreted as a cyclic shift. For $j = 0, 1, \ldots, k-1$, we represents the $p-2$ information bit $s_{0,j}, \ldots, s_{p-2,j}$ stored in column j, together with the parity check bit $s_{p-1,j}$, by a polynomial

$$s_j(x) := s_{0,j} + s_{1,j} x + \cdots + s_{p-2,j} x^{p-2} + s_{p-1,j} x^{p-1} \tag{2}$$

in R_p. Also, we represent the bits in the j-th parity column by the polynomial

$$c_j(x) := c_{0,j} + c_{1,j} x + \cdots + c_{p-2,j} x^{p-2} + c_{p-1,j} x^{p-1}. \tag{3}$$

If we can represent the bits stored in the array code by a codeword of length $k + r$,

$$[s_0(x), \cdots, s_{k-1}(x), c_0(x), \cdots, c_{r-1}(x)], \tag{4}$$

TABLE I: Encoding of array code $\mathcal{C}(4,3,5)$.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5	Disk 6
$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$	$c_{0,0} = s_{0,0} + s_{0,1} + s_{0,2} + s_{0,3}$	$c_{0,1} = s_{0,0} + \sum_{i=0}^{3} s_{i,1} + s_{3,2} + s_{2,3}$	$c_{0,2} = s_{0,0} + s_{3,1} + s_{1,2} + \sum_{i=0}^{3} s_{i,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$c_{1,0} = s_{1,0} + s_{1,1} + s_{1,2} + s_{1,3}$	$c_{1,1} = s_{1,0} + s_{0,1} + \sum_{i=0}^{3} s_{i,2} + s_{3,3}$	$c_{1,2} = s_{1,0} + \sum_{i=0}^{3} s_{i,1} + s_{2,2} + s_{0,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$c_{2,0} = s_{2,0} + s_{2,1} + s_{2,2} + s_{2,3}$	$c_{2,1} = s_{2,0} + s_{1,1} + s_{0,2} + \sum_{i=0}^{3} s_{i,3}$	$c_{2,2} = s_{2,0} + s_{0,1} + s_{3,2} + s_{1,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$	$c_{3,0} = s_{3,0} + s_{3,1} + s_{3,2} + s_{3,3}$	$c_{3,1} = s_{3,0} + s_{2,1} + s_{1,2} + s_{0,3}$	$c_{3,2} = s_{3,0} + s_{1,1} + \sum_{i=0}^{3} s_{i,2} + s_{2,3}$

then the encoding can be performed by taking the product

$$[s_0(x), s_1(x), \cdots, s_{k-1}(x)] \cdot \mathbf{G},$$

with arithmetic performed in R_p, where $\mathbf{G}$ is the $k \times (k+r)$ generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 0 & 1 & x & \cdots & x^{r-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & x^{k-1} & \cdots & x^{(r-1)(k-1)} \end{bmatrix}.$$

The first k columns of $\mathbf{G}$ form the $k \times k$ identity matrix, and the last r columns form a rectangular Vandermonde matrix.

The encoding procedure can be described in terms of polynomials as follows. Given $k(p-1)$ information bits, we append k parity-check bits and form the message vector $[s_0(x), s_1(x), \cdots, s_{k-1}(x)]$, with each component belonging to the ring R_p. After obtaining the vector in (4), we store the coefficients of the terms in the polynomials of degrees 0 to $p-2$. The proposed array code can be considered as puncturing a systematic linear code over the ring R_p.

IV. Proof of the MDS Property

As in the construction of BR and BBV code, we will choose the prime p such that the multiplication order of 2 mod p is equal to $p-1$, i.e., $2^i \not\equiv 1 \bmod p$ for $i = 1, 2, \ldots, p-2$ and $2^{p-1} \equiv 1 \bmod p$. The reason for restricting to this class of prime p is that $x^p + 1$ can be factorized as a product of $x+1$ and

$$M_p(x) := x^{p-1} + x^{p-2} + \cdots + x + 1,$$

which is irreducible over $\mathbb{F}_2$. By the Chinese Remainder Theorem, the ring $R_p = \mathbb{F}_2[x]/(x^p+1)$ is isomorphic to the direct sum of $\mathbb{F}_2[x]/(x+1)$ and $\mathbb{F}_2[x]/(M_p(x))$. Indeed, we can set up an isomorphism

$$\theta : R_p \to \mathbb{F}_2[x]/(x+1) \oplus \mathbb{F}_2[x]/(M_p(x))$$

by defining

$$\theta(f) := (f \bmod x+1, f \bmod M_p(x)).$$

It is easy to check that θ is a ring homomorphism. The mapping θ is a bijection, because it has an inverse function $\phi(a,b)$ given by

$$\phi(a,b) := a \cdot (x+1) + b \cdot e(x) \bmod x^p + 1,$$

where $e(x) := 1 + M_p(x) = x + x^2 + \cdots + x^{p-1}$. It can be checked that the composition $\phi \circ \theta$ is the identity map of R_p.

As $x+1$ and $M_p(x)$ are irreducible polynomials in $\mathbb{F}_2[x]$, the rings R_p is isomorphic to the direct sum of two finite fields $\mathbb{F}_2$ and $\mathbb{F}_{2^{p-1}}$. By construction, the polynomial $s_j(x)$ satisfies $s_j(x) \equiv 0 \bmod x+1$ for all $j = 0, 1, \ldots, k-1$. Hence, the residue modulo $x+1$ of each component in (4) is equal to 0. The first components of $\theta(s_j(x))$'s and $\theta(c_j(x))$'s are all equal to zero. So, we are effectively working over the finite field $\mathbb{F}_{2^{p-1}}$.

Theorem 3. *If the determinant of any $k \times k$ sub-matrix of $\mathbf{G}$, after reduction modulo $M_p(x)$, is a nonzero polynomial in $\mathbb{F}_2[x]/(M_p(x))$, then $\mathcal{C}(k,r,p)$ satisfies the MDS property.*

Proof: Let $\mathbf{A}$ be any $k \times k$ sub-matrix of the generator matrix $\mathbf{G}$, and $\bar{\mathbf{A}}$ be the matrix obtained by reducing each entry of $\mathbf{A}$ mod $M_p(x)$. By the Chinese Remainder Theorem, $\bar{\mathbf{A}}$ can be regarded as a matrix over finite field $\mathbb{F}_{2^{p-1}}$. Since the determinant of $\bar{\mathbf{A}}$ is non-zero, we can find the inverse of $\bar{\mathbf{A}}$. Let $\phi(0, \bar{\mathbf{A}}^{-1})$ denote the matrix we obtain after applying the function $\phi(0,\cdot)$ to each entry of $\bar{\mathbf{A}}^{-1}$. Then the matrix product $\bar{\mathbf{A}}^{-1} \cdot \mathbf{A}$, with arithmetic carried out in R_p, is equal to the $k \times k$ identity matrix. The source information symbols can be recovered by multiplication by $\bar{\mathbf{A}}^{-1}$. ■

Theorem 3 can be re-formulated as follows.

Theorem 4. *Let $\mathbf{P}$ be the $k \times r$ sub-matrix consisting of the right-most r columns in the generator matrix $\mathbf{G}$, i.e.,*

$$\mathbf{P} := \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & x & \cdots & x^{r-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x^{k-1} & \cdots & x^{(r-1)(k-1)} \end{bmatrix}. \tag{5}$$

The array code $\mathcal{C}(k,r,p)$ is MDS if for all $m = 1, 2, \ldots, \min\{k, r\}$, the determinant of each $m \times m$ sub-matrix of $\mathbf{P}$, regarded as a polynomial in $\mathbb{F}_2[x]$, is not divisible by $x^p + 1$.

The next Theorem proves that the proposed array code $\mathcal{C}(k,r,p)$ satisfies the MDS property for $1 \leq r \leq 5$.

Theorem 5. *Let $p > 5$ be a prime such that the multiplicative order of 2 mod p is equal to $p-1$. For $1 \leq k \leq p$ and $1 \leq r \leq 5$, the array code $\mathcal{C}(k,r,p)$ satisfies the MDS property.*

Proof: It suffices to prove the theorem for $r = 5$. By Theorem 4, we need to prove that for $\ell = 1, 2, \ldots, 5$, the determinant of each sub-matrix of $\mathbf{P}$ in (5) of size $\ell \times \ell$ is not divisible by $x^p + 1$ in $\mathbb{F}_2[x]$. The determinant of an $\ell \times \ell$ sub-matrix of $\mathbf{P}$ can be written as

$$\begin{vmatrix} x^{i_1 j_1} & x^{i_1 j_2} & \cdots & x^{i_1 j_\ell} \\ x^{i_2 j_1} & x^{i_2 j_2} & \cdots & x^{i_1 j_\ell} \\ \vdots & \vdots & \ddots & \vdots \\ x^{i_\ell j_1} & x^{i_\ell j_2} & \cdots & x^{i_\ell j_\ell} \end{vmatrix}$$

with $0 \leq i_1 < \cdots < i_\ell \leq k-1$ and $0 \leq j_1 < \cdots < j_\ell \leq 4$. By factoring out powers of x, it is sufficient to show that the determinant

$$\begin{vmatrix} 1 & x^{i_1(j_2-j_1)} & \cdots & x^{i_1(j_\ell-j_1)} \\ 1 & x^{i_2(j_2-j_1)} & \cdots & x^{i_2(j_\ell-j_1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x^{i_\ell(j_2-j_1)} & \cdots & x^{i_\ell(j_\ell-j_1)} \end{vmatrix} \tag{6}$$

is not divisible by $x^p + 1$ in the polynomial ring $\mathbb{F}_2[x]$.

When $\ell = 1$, the determinant of size 1×1 in (6) is equal to 1, and hence cannot be divisible by $x^p + 1$. For the 2×2 case, the determinant in (6) is equal to $x^{i_2(j_2-j_1)} + x^{i_1(j_2-j_1)}$. It is reduced to 0 in $\mathbb{F}_2[z]/(x^p + 1)$ if and only if

$$i_2(j_2 - j_1) \equiv i_1(j_2 - j_1) \bmod p. \tag{7}$$

Since $j_2 - j_1$ is strictly smaller than p, the condition in (7) implies that $i_1 = i_2$, which contradicts the fact that $i_1 < i_2$.

For the 3×3 case, consider the determinant

$$\begin{vmatrix} 1 & x^{a\alpha} & x^{a\beta} \\ 1 & x^{b\alpha} & x^{b\beta} \\ 1 & x^{c\alpha} & x^{c\beta} \end{vmatrix}$$

as an element in $\mathbb{F}_2[x]/(x^p + 1)$, with $0 \leq a < b < c \leq k-1$ and $1 \leq \alpha < \beta \leq 4$. If the above 3×3 determinant is equal to zero in $\mathbb{F}_2[x]/(x^p + 1)$, then the following six terms

$$x^{a\alpha+b\beta},\ x^{b\alpha+a\beta},\ x^{a\alpha+c\beta},\ x^{c\alpha+a\beta},\ x^{b\alpha+c\beta} \text{ and } x^{c\alpha+b\beta}$$

can be divided into 3 pairs such that the exponents in each pair are congruent modulo p. Consider the exponent of the first term. If $a\alpha + b\beta$ is congruent to $b\alpha + a\beta$, then we have

$$(a - b)\alpha \equiv (a - b)\beta \bmod p.$$

Since $a < b < p$, this implies that $\alpha = \beta$, contradicting the fact that α and β are distinct. If $a\alpha + b\beta$ is congruent to $a\alpha + c\beta$, then we can deduce that $b = c$, contradicting the assumption that b and c are distinct. Similarly, if $a\alpha + b\beta$ is congruent to $c\alpha + b\beta$, then it contradicts the assumption that a and c are distinct.

By the same argument, $c\alpha + a\beta$ cannot be congruent to $a\alpha + c\beta$, $c\alpha + b\beta$ nor $b\alpha + a\beta$ mod p. Also, $b\alpha + c\beta$ cannot be congruent to $a\alpha + c\beta$, $c\alpha + b\beta$ nor $b\alpha + a\beta$ mod p. We can re-arrange the six terms in the determinant as

$$(x^{a\alpha+b\beta} + x^{c\alpha+a\beta} + x^{b\alpha+c\beta}) + (x^{a\alpha+c\beta} + x^{c\alpha+b\beta} + x^{b\alpha+a\beta})$$

None of the terms in the first parenthesis is equal to any term in the second parenthesis if the exponents are reduced mod p, and *vice versa*. Therefore the 3×3 determinant cannot be equal to zero in $\mathbb{F}_2[z]/(x^p + 1)$.

For the case of $\ell = 4$, we consider the following 4×4 determinant

$$\begin{vmatrix} 1 & x^{a\alpha} & x^{a\beta} & x^{a\gamma} \\ 1 & x^{b\alpha} & x^{b\beta} & x^{b\gamma} \\ 1 & x^{c\alpha} & x^{c\beta} & x^{c\gamma} \\ 1 & x^{d\alpha} & x^{d\beta} & x^{d\gamma} \end{vmatrix},$$

with $0 \leq a < b < c < d \leq k-1$ and $1 \leq \alpha < \beta < \gamma \leq 4$. This is the determinant of a generalized Vandermonde matrix, and by Theorem 1, it is equal to

$$\begin{aligned} \sigma_i(x^a, x^b, x^c, x^d)&(x^b - x^a)(x^c - x^a)(x^d - x^a) \\ &\cdot (x^c - x^b)(x^d - x^b)(x^d - x^a), \end{aligned}$$

with i equal to 0, 1, 2, 3, or 4. Since the values of a, b, c and d are distinct and less than p, the factors $(x^b - x^a)$, $(x^c - x^a)$, $(x^d - x^a)$ etc. may be divisible by $x + 1$ but not by $M_p(x)$. Hence, it is sufficient to prove that $\sigma_i(x^a, x^b, x^c, x^d)$ are not divisible by $x^p + 1$, for $i = 0, 1, \ldots, 4$. It is obvious that $\sigma_0(x^a, x^b, x^c, x^d) = 1$ is not divisible by $x^p + 1$ and $\sigma_1(x^a, x^b, x^c, x^d) = x^a + x^b + x^c + x^d$ is not divisible by $x^p + 1$ if $p > 5$. For $i = 2$, the exponents of the terms in

$$\sigma_2(x^a, x^b, x^c, x^d) = x^{a+b} + x^{a+c} + x^{a+d} + x^{b+c} + x^{b+d} + x^{c+d}$$

are precisely the elements in the restricted sum set $2^\wedge\{a, b, c, d\} \bmod p$. By the Erdős-Heilbronn conjecture, there are at least $2 \times 4 - 3 = 5$ distinct exponents among the six terms in $\sigma_2(x^a, x^b, x^c, x^d)$. If $\sigma_2(x^a, x^b, x^c, x^d)$ is equal to 0 in $\mathbb{F}_2[z]/(x^p + 1)$, then there are at most 3 distinct exponents in $\sigma_2(x^a, x^b, x^c, x^d)$, and it contradicts the Erdős-Heilbronn conjecture. For $i = 3$, we also have four terms in

$$\sigma_3(x^a, x^b, x^c, x^d) = x^{a+b+d} + x^{b+c+d} + x^{a+c+d} + x^{a+b+c}.$$

All four exponents are distinct because a, b, c and d are distinct mod p. Hence, there cannot be any cancelation. Finally, for $i = 4$, $\sigma_4(x^a, x^b, x^c, x^d)$ is a power of x and cannot be divisible by $x^p + 1$ in $\mathbb{F}_2[x]$.

When $\ell = 5$, the 5×5 determinant in (6) is the determinant of a Vandermonde matrix. It cannot be divisible by $M_p(x)$ as i_1, i_2, i_3, i_4 and i_5 are all less than p. This proves that the determinant in (6) is not divisible by $x^p + 1$ in $\mathbb{F}_2[x]$, and completes the proof of Theorem 5. ■

From the proof of Theorem 5, we have the following corollary.

Corollary 6. *If 2 is a primitive element in finite field $\mathbb{F}_p$ and $p \geq 5$, then $\mathcal{C}(k, r, p)$ satisfies the MDS property for $r \leq 4$.*

V. Efficient Repair for One or Two Information Columns

We have shown in Theorem 5 that we can decode the information bits in $\mathcal{C}(k, r, p)$ if there are r disk failures, for $r \leq 5$. If one of the information column is erased, we want to recover it as efficient as possible. This is called the repair problem. The straightforward method is to read all of the remaining information bits and the row parity bits to recover the erased bits. For example, if disk 0 of $\mathcal{C}(4,3,5)$ fails, one can read $s_{0,1}$, $s_{0,2}$, $s_{0,3}$ and $c_{0,0}$ and compute the sum to recover $s_{0,0}$, need to read 16 bits to recover disk 0. This method is not optimal in terms of the number of disk reads.

By utilizing the row parity bits and the diagonal parity bits, we can repair the first column by reading 12 bits. Recall from Table I that $c_{31} = s_{3,0} + s_{2,1} + s_{1,2} + s_{0,3}$, and $c_{22} = s_{2,0} + s_{0,1} + s_{3,2} + s_{1,3}$. We can recover the first column by reading $s_{0,1}$, $s_{1,1}$ and $s_{2,1}$ in disk 1, $s_{0,2}$, $s_{1,2}$ and $s_{3,2}$ in disk 2, $s_{0,3}$, $s_{1,3}$ in disk 3, and the parity bits $c_{0,0}$, $c_{1,0}$, $c_{3,1}$ and $c_{2,2}$. The lost bits can be decoded by

$$\begin{aligned}
s_{0,0} &= s_{0,1} + s_{0,2} + s_{0,3} + c_{0,0},\\
s_{1,0} &= s_{1,1} + s_{1,2} + s_{1,3} + c_{1,0},\\
s_{2,0} &= s_{0,1} + s_{3,2} + s_{1,3} + c_{2,2},\\
s_{3,0} &= s_{2,1} + s_{1,2} + s_{0,3} + c_{3,1}.
\end{aligned}$$

The repair of other information column can be done similarly.

In the remainder of this section, we give an efficient method which can recover two erased information columns, by access the remaining $k-2$ information columns and any 2 parity columns. We need the following lemma about the inverse of a binomial $1 + x^b$ in $\mathbb{F}_2[x]/(M_p(x))$.

Lemma 7. *For $0 < b \le p-1$, the multiplicative inverse of $1 + x^b$ in $\mathbb{F}_2[x]/(M_p(x))$ is*

$$\sum_{i=1}^{(p-1)/2} x^{(2i-1)b} = x^b + x^{3b} + \cdots + x^{(p-2)b}.$$

Proof: We can check that in the field the $\mathbb{F}_2[x]/(M_p(x))$,

$$\begin{aligned}
&(1 + x^b)(x^b + x^{3b} + \cdots + x^{(p-2)b})\\
&= x^b + x^{2b} + x^{3b} + x^{4b} + \cdots + x^{(p-2)b} + x^{(p-1)b}\\
&= x + x^2 + x^3 + x^4 + \cdots + x^{p-2} + x^{p-1}\\
&\equiv 1 \bmod M_p(x).
\end{aligned}$$

The second last equality follows from the fact that in the finite field $\mathbb{F}_p$, multiplication by b is an injective function mapping non-zero elements to non-zero elements. ■

Let $j < \ell$ be integers between 0 and $k-1$. Suppose that information columns j and ℓ are erased. We want to recover the lost data bits in columns j and ℓ by reading columns i, for $i \in \{0, 1, \ldots, k-1\} \setminus \{j, \ell\}$, and parity columns a and b, for $0 \le a < b \le r-1$.

The accessed bits are represented by polynomials $s_i(x)$ and

$$c_a(x) = \sum_{\nu=0}^{k-1} x^{a\nu} s_\nu(x) \text{ and } c_b(x) = \sum_{\nu=0}^{k-1} x^{b\nu} s_\nu(x).$$

Let $f_a(x)$ and $f_b(x)$ be the polynomials by subtracting the known values of $s_i(x)$, for $i \in \{0, 1, \ldots, k-1\} \setminus \{j, \ell\}$, from $c_a(x)$ and $c_b(x)$, respectively. We can repair the two erasures by solving the following system of linear equations

$$\begin{bmatrix} x^{aj} & x^{a\ell} \\ x^{bj} & x^{b\ell} \end{bmatrix} \begin{bmatrix} s_j(x) \\ s_\ell(x) \end{bmatrix} = \begin{bmatrix} f_a(x) \\ f_b(x) \end{bmatrix}.$$

The entries of the matrices are regarded as elements in the ring R_p. The determinant of the above matrix is

$$x^{aj+b\ell} + x^{a\ell+bj} = x^{a\ell+bj}(1 + x^{(b-a)(j-\ell)}).$$

If the polynomial $1 + x^{(b-a)(j-\ell)}$ is regarded as an element in $\mathbb{F}_2[x]/M_p(x)$, by Lemma 7, its multiplicative inverse is equal to

$$\sum_{\mu=1}^{(p-1)/2} x^{(2\mu-1)(b-a)(j-\ell)}.$$

Hence, we can solve for $\begin{bmatrix} s_j(x) \\ s_\ell(x) \end{bmatrix}$ by

$$x^{-(a\ell+bj)} \Big(\sum_{\mu=1}^{(p-1)/2} x^{(2\mu-1)(b-a)(j-\ell)} \Big) \begin{bmatrix} x^{b\ell} & x^{a\ell} \\ x^{bj} & x^{aj} \end{bmatrix} \begin{bmatrix} f_a(x) \\ f_b(x) \end{bmatrix}.$$

We note that $x^{b\ell} f_a(x) + x^{a\ell} f_b(x)$ can be computed by cyclically shifting $f_a(x)$ and $f_b(x)$ to the right by $b\ell$ and $a\ell$, respectively, and adding the resulting polynomials. Multiplication by $x^{-(a\ell+bj)}$ is simply cyclically shifting to the left by $a\ell + bj$. Adding the parity-check bit to formulate the polynomials $c_a(x)$ and $c_b(x)$ takes $2(p-2)$ XORs, and computing $f_a(x)$ and $f_b(x)$ involves $2(k-2)p$ XORs. The number of XORs of solving $s_j(x)$ and $s_\ell(x)$ is $((p-1)/2 - 1) + 2(p-1)$. Thus, we obtain the total number of XORs of repairing two erased information columns is $2(k-2)p + (9p-15)/2$.

VI. Performance Comparison

In this section, we evaluate the encoding/repairing complexity for the proposed $\mathcal{C}(k,r,p)$ as well as the other very important MDS array codes. We determine the *normalized encoding complexity* as the ratio of the average number of XORs needed to construct single parity column to the number of information bits, and *normalized repairing complexity* as the ratio of the average number of XORs needed to reconstruct the data file after two information columns failure to the number of information bits.

A. Encoding Complexity

In the $p-1 \times n$ array of $\mathcal{C}(k,r,p)$, there are k information columns, and $k-1$ XORs are required to reduce the k information bits per row to the row parity column. The row parity thus requires $(k-1)(p-1)$ XORs. Each diagonal column contains a total of $p-1$ diagonal parity bits, requiring $k-1+p-1$ XORs to reduce one diagonal parity bit and $k-1$ XORs to reduce each of the other $p-2$ diagonal parity bits. Therefore, one diagonal parity column requires $(k-1+p-1)+(k-1)(p-2)$ XORs. The total number of XORs required for construction r parities are $(k-1)(p-1)+((k+p-2)+(k-1)(p-2))(r-1)$, and the normalized encoding complexity is $\frac{rk-1}{rk}$. The normalized encoding complexity of RDP, EVENODD and BBV are $1 - \frac{1}{p-1}$, $1 - \frac{1}{2(p-2)}$ [2] and $2 - \frac{1}{r} - \frac{2(r-1)}{rp}$ respectively.

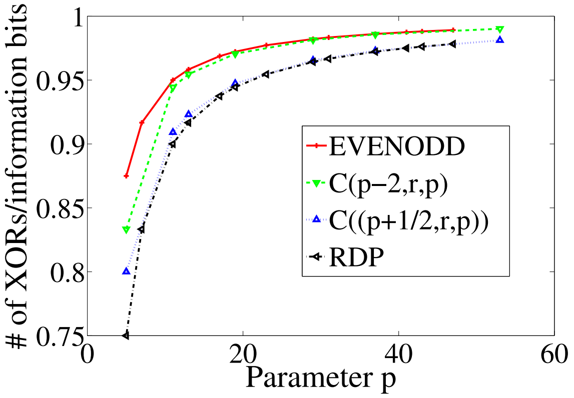

Fig. 1: The normalized encoding complexity ($r = 2$).

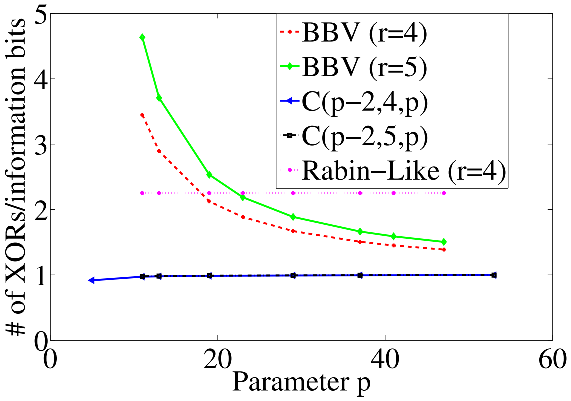

Fig. 2: The normalized encoding complexity ($r = 4$).

The normalized encoding complexity of $\mathcal{C}(k, r, p)$, RDP, EVENODD, BBV and Rabin-Like are summarized in Fig.1 ($r = 2$) and Fig.2 ($r = 4$), which show that the normalized encoding complexity of $\mathcal{C}(k, r, p)$, RDP and EVENODD are all asymptotically equal to one. While the encoding complexity of the BBV code and Rabin-Like code are 2 asymptotically and 2.5 respectively.

B. *Repairing Complexity*

In $\mathcal{C}(k, r, p)$, the total number of XORs of computing the lost two information columns is $2(k-2)p+\frac{9p}{2}-\frac{15}{2}$, while the number is $2(p-1)(p-2)$ in RDP [2] and the reconstruction algorithm described in the EVENODD paper [3] requires more XORs. The repairing algorithms for two information columns failure in STAR [7] and Triple-Star [8] are the same as that in EVENODD, the repairing algorithm of extension RDP [9] is the same as RDP. For fair comparison, we let $k = p - 1$ in $\mathcal{C}(k, r, p)$, we can thus have the normalized repairing complexity of $\mathcal{C}(k, r, p)$ and RDP are $\frac{2p^2-3.5p-1.5}{(p-1)^2}$ and $\frac{2(p-2)}{p-1}$ respectively, which are roughly the same when p is large.

VII. Discussions

The number of information column in some existing MDS array codes [2]–[4], [6], [7], [9] is usually restricted to $k = p-1$ or $k = p$, where p is a prime. The proposed MDS array code $\mathcal{C}(k, r, p)$ relaxes this restriction by allowing the parameter k to be any positive integer between 2 and p. The efficient encoding and repairing methods presented in this paper show that the encoding and repairing cost of $\mathcal{C}(k, r, p)$ are almost the same as in some existing MDS arry codes such as RDP and EVENODD. The new construction $\mathcal{C}(k, r, p)$ is more flexible as it provides more choices in code parameters.

In the proof of the MDS property, we use a powerful theorem in additive number theory, namely the Erdős-Heilbronn conjecture. To the best knowledge of the authors, this approach is novel. We believe that this approach is useful and can be further developed to the design of other MDS array codes.

References

[1] D. A. Patterson, P. Chen, G. Gibson, and R. H. Katz, "Introduction to Redundant Arrays of Inexpensive Disks (RAID)," in *Proc. IEEE COMPCON*, vol. 89, 1989, pp. 112–117.

[2] 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.

[3] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Trans. on Computers*, vol. 44, no. 2, pp. 192–202, 1995.

[4] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Trans. Information Theory*, vol. 42, no. 2, pp. 529–542, 1996.

[5] L. Xiang, Y. Xu, J. Lui, and Q. Chang, "Optimal recovery of single disk failure in RDP code storage systems," in *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 38, no. 1. ACM, 2010, pp. 119–130.

[6] M. Blaum, J. Brady, J. Bruck, J. Menon, and A. Vardy, "The EVENODD code and its generalization," *High Performance Mass Storage and Parallel I/O*, pp. 187–208, 2001.

[7] C. Huang and L. Xu, "STAR: An efficient coding scheme for correcting triple storage node failures," *IEEE Transactions on Computers*, vol. 57, no. 7, pp. 889–901, 2008.

[8] Y. Wang, G. Li, and X. Zhong, "Triple-Star: A coding scheme with optimal encoding complexity for tolerating triple disk failures in RAID," *International Journal of innovative Computing, Information and Control*, vol. 3, pp. 1731–1472, 2012.

[9] M. Blaum, "A family of MDS array codes with minimal number of encoding operations," in *IEEE Int. Symp. on Inf. Theory*, 2006, pp. 2784–2788.

[10] 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 (= 4) disk failures," *IEEE Trans. on Computers*, vol. 54, no. 12, pp. 1473–1483, 2005.

[11] 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, January 1993.

[12] K. W. Shum, H. Hou, M. Chen, H. Xu, and H. Li, "Regenerating codes over a binary cyclic code," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, July 2014, pp. 1046–1050.

[13] N. Kolokotronis, K. Limniotis, and N. Kaloupts idis, "Lower bounds on sequence complexity via generalised Vandermonde determinants," in *Sequences and Their Applications–SETA 2006*. Springer, 2006, pp. 271–284.

[14] J. A. Dias da Silva and Y. O. Hamidoune, "Cyclic spaces for Grassmann derivatives and additive theory," *Bulletin of the London Math. Society*, vol. 26, no. 2, pp. 140–146, 1994.

[15] N. Alon, M. B. Nathanson, and I. Ruzsa, "The polynomial method and restricted sums of congruence classes," *J. of Number Theory*, vol. 56, no. 2, pp. 404–417, 1996.