Binary MDS Array Codes With Asymptotically Optimal Repair Correcting Triple Failures

***Abstract*—Binary maximal-distance separable (MDS) array code is one class of erasure codes with k information columns and r parity columns, of which each column storing L bits such that any k columns are sufficient to recover all the information bits. It is shown that if a single column fails, then we need to download at least $L/(d-k+1)$ bits in each of the d surviving columns. The code is said to have the optimal repair property if the lowest bits downloaded is achieved. In this paper, we present an explicit construction of binary MDS array codes with asymptotically optimal repair for $r = 3$ and $d = k + 1$.**

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 $L \times n$, with each element of an array storing one bit. Among the n columns, k *information columns* store the information bits and $r = n - k$ *parity columns* store the redundant bits. The L bits in a column are stored in the same disk, or storage node. We refer to a disk as a column or a storage node interchangeably, and an entry in the array as a bit. 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 MDS array code. Examples of existing MDS array codes include X-code [2] and RDP [3] with $r = 2$, STAR [4], generalized RDP [5] and TIP [6] can tolerate any three erasures.

When a disk fails in a distributed storage system, we will rebuild the erased bits in the failed disk by downloading some bits from the surviving disks. The amount of the downloaded bits in a repair process is called the *repair bandwidth*. As the repair bandwidth plays a crucial role in the overall recovery time [3] and influences the system service performance [7]. It is important to minimize repair bandwidth.

The repair problem was first formulated and studied by Dimakis *et al.* in [8] using the concept of information flow graph. In the framework in [8], a data file of size B symbols is encoded and distributed to n storage nodes, with each node storing L symbols such that the file can be decoded from any k of them. Furthermore, upon the failure of a storage node, a new node replaces the failed node by downloading β symbols from each of d surviving nodes, and the repair bandwidth is $d\beta$. It is shown in [8] that the optimal repair bandwidth is

$$\frac{(k+1)L}{2}, \tag{1}$$

when $d = k + 1$. Although the optimal repair bandwidth can be achieved in [9], [10] over a large enough finite field, how to design binary MDS array codes to obtain the optimal repair bandwidth is less clear.

It is shown in [11], [12] that the optimal repair bandwidth of X-code and RDP is 50% larger than the optimal value in (1). For the case of two parity columns, Gad *et al.* [13] proposed a class of binary MDS array codes with optimal repair. Wang *et al.* [14], [15] constructed the MDR codes over binary field and achieved the optimal repair. However, the optimal repair of binary MDS array codes with $r \geq 3$ parity columns is still an open problem.

In this paper, we give a new construction of binary MDS array codes with three parity columns. We show that the proposed binary MDS array code has comparable encoding complexity and decoding complexity (for two failures), compared to the existing binary MDS array codes. More importantly, the optimal repair bandwidth in (1) can be achieved asymptotically for any one information failure.

This paper is organized as follows. We first give the construction of binary MDS array codes and then present the MDS property condition in Section II. In Section III, we give the repair algorithm for one single information erasure. An efficient decoding method for any two information erasures is given in Section IV. We conclude in Section V.

II. Binary MDS Array Code

In this section, we show how our proposed new binary MDS array code is constructed.

A. Construction of Binary MDS Array Code

Let $k \geq 3$ and $L = (p-1)\tau$ be positive integers, where $\tau = 2^{k-2}$ and p is a prime number such that $2^i \not\equiv 1 \bmod p$ for $i = 1, 2, \ldots, p-2$. Assume that a file of size $k(p-1)\tau$ denoted by information bits $s_{0,i}, s_{1,i}, \ldots, s_{(p-1)\tau-1,i} \in \mathbb{F}_2^{(p-1)\tau}$ for $i = 1, 2, \ldots, k$, which are employed to generate $3(p-1)\tau$ redundant bits $s_{0,j}, s_{1,j}, \ldots, s_{(p-1)\tau-1,j} \in \mathbb{F}_2^{(p-1)\tau}$ for $j = k+1, k+2, k+3$. For $\ell = 1, 2, \ldots, k+3$ and $\mu = 0, 1, \ldots, \tau-1$, we define the following short-hand notations

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

We call $s_{(p-1)\tau+\mu,\ell}$ as the *parity-check bit* associated with $s_{\mu,\ell}, s_{\tau+\mu,\ell}, \ldots, s_{(p-2)\tau+\mu,\ell}$. For $\ell = 1, 2, \ldots, k+3$, we present the bits $s_{0,\ell}, s_{1,\ell}, \ldots, s_{(p-1)\tau-1,\ell}$ in column ℓ, together with τ parity-check bits $s_{(p-1)\tau,\ell}, s_{(p-1)\tau+1,\ell}, \ldots, s_{p\tau-1,\ell}$, by a polynomial $s_\ell(x)$ over the ring $\mathbb{F}_2[x]$,

$$s_\ell(x) = s_{0,\ell} + s_{1,\ell}x + s_{2,\ell}x^2 + \ldots + s_{p\tau-1,\ell}x^{p\tau-1}.$$

Information Columns				Redundant Columns		
1	2	3	4	1	2	3
$s_{0,1}$	$s_{0,2}$	$s_{0,3}$	$s_{0,4}$	$s_{0,1}+s_{0,2}+s_{0,3}+s_{0,4}$	$\boldsymbol{s_{11,1}+s_{10,2}+s_{8,3}}+s_{0,4}$	$s_{0,1}+\boldsymbol{s_{8,2}+s_{10,3}+s_{11,4}}$
$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$s_{1,4}$	$s_{1,1}+s_{1,2}+s_{1,3}+s_{1,4}$	$s_{0,1}+\boldsymbol{s_{11,2}+s_{9,3}}+s_{1,4}$	$s_{1,1}+\boldsymbol{s_{9,2}+s_{11,3}}+s_{0,4}$
$s_{2,1}$	$s_{2,2}$	$s_{2,3}$	$s_{2,4}$	$s_{2,1}+s_{2,2}+s_{2,3}+s_{2,4}$	$s_{1,1}+s_{0,2}+\boldsymbol{s_{10,3}}+s_{2,4}$	$s_{2,1}+\boldsymbol{s_{10,2}}+s_{0,3}+s_{1,4}$
$s_{3,1}$	$s_{3,2}$	$s_{3,3}$	$s_{3,4}$	$s_{3,1}+s_{3,2}+s_{3,3}+s_{3,4}$	$s_{2,1}+s_{1,2}+\boldsymbol{s_{11,3}}+s_{3,4}$	$s_{3,1}+\boldsymbol{s_{11,2}}+s_{1,3}+s_{2,4}$
$s_{4,1}$	$s_{4,2}$	$s_{4,3}$	$s_{4,4}$	$s_{4,1}+s_{4,2}+s_{4,3}+s_{4,4}$	$s_{3,1}+s_{2,2}+s_{0,3}+s_{4,4}$	$s_{4,1}+s_{0,2}+s_{2,3}+s_{3,4}$
$s_{5,1}$	$s_{5,2}$	$s_{5,3}$	$s_{5,4}$	$s_{5,1}+s_{5,2}+s_{5,3}+s_{5,4}$	$s_{4,1}+s_{3,2}+s_{1,3}+s_{5,4}$	$s_{5,1}+s_{1,2}+s_{3,3}+s_{4,4}$
$s_{6,1}$	$s_{6,2}$	$s_{6,3}$	$s_{6,4}$	$s_{6,1}+s_{6,2}+s_{6,3}+s_{6,4}$	$s_{5,1}+s_{4,2}+s_{2,3}+s_{6,4}$	$s_{6,1}+s_{2,2}+s_{4,3}+s_{5,4}$
$s_{7,1}$	$s_{7,2}$	$s_{7,3}$	$s_{7,4}$	$s_{7,1}+s_{7,2}+s_{7,3}+s_{7,4}$	$s_{6,1}+s_{5,2}+s_{3,3}+s_{7,4}$	$s_{7,1}+s_{3,2}+s_{5,3}+s_{6,4}$

Fig. 1: An example of s ge code for three redundant columns. When information column 1 fails, the bits in the solid line box are downloaded to repair the information bits $s_{0,1}, s_{2,1}, s_{4,1}, s_{6,1}$ and the bits in the dashed box are used to repair the information bits $s_{1,1}, s_{3,1}, s_{5,1}, s_{7,1}$.

The polynomial $s_i(x)$, corresponds to the i information column for $i = 1, 2, \ldots, k$, is called *data polynomial*. While the polynomial $s_j(x)$, corresponds to the $j-k$ parity column for $j = k+1, k+2, k+3$, is called *coded polynomial*.

Note that we do not store the parity-check bits in the disk. It is present only for notational convenience. We write the k data polynomials and 3 coded polynomials as the row vector

$$[s_1(x), s_2(x), \cdots, s_{k+3}(x)], \tag{3}$$

which can be computed by taking the product

$$[s_1(x), s_2(x), \cdots, s_{k+3}(x)] = [s_1(x), s_2(x), \cdots, s_k(x)] \cdot \mathbf{G}$$

with arithmetic performed in $\mathbb{F}_2[x]/(1+x^{p\tau})$, where the $k \times (k+3)$ *generator matrix* $\mathbf{G}$ is composed by the $k \times k$ identity matrix $\mathbf{I}$ and a $k \times 3$ *encoding matrix* $\mathbf{P}$,

$$\mathbf{P} := \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ x & x^2 & x^4 & \cdots & x^{2^{k-2}} & 1 \\ 1 & x^{2^{k-2}} & x^{2^{k-3}} & \cdots & x^2 & x \end{bmatrix}^T.$$

The proposed code is denoted as $\mathcal{C}(k, 3, p)$. Consider an example of $k = 4$ and $p = 3$, the 32 information bits are represented by $s_{0,i}, s_{1,i}, \ldots, s_{7,i}$, for $i = 1, 2, 3, 4$. The encoding matrix of this example is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ x & x^2 & x^4 & 1 \\ 1 & x^4 & x^2 & x \end{bmatrix}^T.$$

The example is illustrated in Fig. 1, where the bit with bold type is parity-check bit.

The encoding procedure can be described in terms of polynomials as follows. Given $k(p-1)\tau$ information bits, we append τ parity-check bits for each of $(p-1)\tau$ information bits and form the message vector $[s_1(x), s_2(x), \cdots, s_k(x)]$. After obtaining the vector in (3), we store the coefficients of the terms in the polynomials of degrees 0 to $(p-1)\tau - 1$. The proposed array code can be considered as puncturing a systematic linear code over $\mathbb{F}_2[x]/(1+x^{p\tau})$.

B. Proof of the MDS Property

As we choose the prime p such that the multiplication order of 2 mod p is equal to $p-1$, we have $x^{p\tau}+1$ can be factorized as a product of two co-prime factors $x^\tau + 1$ and

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

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

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

by defining

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

The mapping θ is a ring homomorphism and a bijection, because it has an inverse function $\phi(a, b)$ given by

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

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

By construction, $s_\ell(x) \equiv 0 \bmod x^\tau + 1$ for all $\ell = 1, 2, \ldots, k+3$. Hence, the first components of $\theta(s_\ell(x))$'s are all equal to zero. So, we are effectively working over the ring $\mathbb{F}_2[x]/(M_p^\tau(x))$. Recall that $\tau = 2^{k-2}$, we have

$$\begin{aligned} M_p^\tau(x) =& x^{(p-1)\tau} + x^{(p-2)\tau} + \cdots + x^\tau + 1 \\ =& (x^{p-1} + x^{p-2} + \cdots + x + 1)^\tau := (M_p(x))^\tau. \end{aligned}$$

As $M_p(x)$ is irreducible in $\mathbb{F}_2[x]$ [16], $\mathbb{F}_2[x]/(M_p^\tau(x))$ is isomorphic to the direct sum of τ finite fields $\mathbb{F}_2[x]/(M_p(x))$. With the same discussion in [16], we have the following result.

Theorem 1. *$\mathcal{C}(k, 3, p)$ is MDS if for $t = 1, 2, 3$, the determinant of each $t \times t$ sub-matrix of $\mathbf{P}$ is not divisible by $x^p + 1$.*

We need the following lemma in the proof of MDS property.

Lemma 2. *Let p be a prime such that the multiplicative order of 2 mod p is equal to $p-1$. For $i = 1, 2, \ldots, p-2$, the following equation holds*

$$2^i + 2^{i+\frac{p-1}{2}} \equiv 0 \bmod p.$$

Proof. By Fermat's little theorem, we have $2^{p-1} \equiv 1 \mod p$. As $2^{(p-1)/2}$ is a root of $x^2 - 1 \mod p$, so $2^{(p-1)/2}$ is equal to either 1 or $-1 \mod p$. Recall that the multiplicative order of 2 mod p is equal to $p-1$, we have $2^{(p-1)/2} \not\equiv 1 \mod p$ and $2^{(p-1)/2} + 1 \equiv 0 \mod p$. Multiply both sides of the above equation by 2^i and we get the equation in Lemma 2 holds. □

The next theorem gives a sufficient MDS property condition.

Theorem 3. *If $p \geq \max\{2k-8, k\}$ is a prime such that the multiplicative order of 2 mod p is equal to $p-1$, then the code $\mathcal{C}(k, 3, p)$ satisfies the MDS property.*

Proof. By Theorem 1, we need to prove that for $t = 1, 2, 3$, the determinant of each sub-matrix of $\mathbf{P}$ of size $t \times t$ is not divisible by $x^p + 1$ in $\mathbb{F}_2[x]$. When $t = 1$, the determinant is equal to a power of x, and hence cannot be divisible by $x^p + 1$. When $t = 2$, the determinant can be classified as $x^i + 1$ with $i = 1, 2$, $x^{2^i} + x^{2^j}$ with $0 \leq i < j \leq k-2$, and $x^{2^i+2^{k-j-1}} + x^{2^j+2^{k-i-1}}$ with $1 \leq i < j \leq k-2$.

It is easy to check that $x^i + 1$ cannot be divisible by $x^p + 1$ for $i = 1, 2$. If $x^{2^i} + x^{2^j} = x^{2^i}(1 + x^{2^j-2^i})$ is divisible by $x^p + 1$, then $2^j - 2^i \equiv 0 \bmod p$. We have $j \equiv i \bmod p$, which contradicts the fact that $0 \leq i < j < p$. Suppose $x^{2^i+2^{k-j-1}} + x^{2^j+2^{k-i-1}}$ is divisible by $x^p + 1$, we have

$$2^j - 2^i + 2^{k-i-1} - 2^{k-j-1} \equiv (2^{j-i} - 1)(2^i + 2^{k-j-1}) \equiv 0 \bmod p.$$

Since p is a prime and $2^{j-i} - 1 \not\equiv 0 \bmod p$, this implies that $2^i + 2^{k-j-1} \equiv 0 \bmod p$. By Lemma 2, the equation $p = 2k - 2i - 2j - 1$ holds. As $1 \leq i < j \leq k-2$, we have $p = 2k - 2i - 2j - 1 \leq 2k - 7$, which contradicts $p \geq 2k - 8$.

For $t = 3$, we need to consider the following 3 determinants

$$\begin{vmatrix} 1 & 1 & 1 \\ x & x^{2^i} & 1 \\ 1 & x^{2^{k-i-1}} & x \end{vmatrix} \tag{4}$$

with $1 \leq i \leq k-2$,

$$\begin{vmatrix} 1 & 1 & 1 \\ x & x^{2^i} & x^{2^j} \\ 1 & x^{2^{k-i-1}} & x^{2^{k-j-1}} \end{vmatrix} \tag{5}$$

with $1 \leq i < j \leq k-2$, and

$$\begin{vmatrix} 1 & 1 & 1 \\ x^{2^i} & x^{2^j} & x^{2^\ell} \\ x^{2^{k-i-1}} & x^{2^{k-j-1}} & x^{2^{k-\ell-1}} \end{vmatrix} \tag{6}$$

with $1 \leq i < j < \ell \leq k-2$.

If the determinant in (4) is equal to zero in $\mathbb{F}_2[x]/(x^p + 1)$, then the following six terms

$$x^{2^i+1},\ x^{2^{k-i-1}+1},\ x^{2^{k-i-1}},\ x^{2^i},\ x^2 \text{ and } 1$$

can be divided into 3 pairs such that the exponents in each pair are congruent modulo p. Consider the exponent of the last term. As 2^{k-i-1}, 2^i and 2 are not congruent to 0 modulo p, we only need to consider the case of 0 congruent to $2^i + 1$ or $2^{k-i-1} + 1$. If 0 is congruent to $2^i + 1$, then we have $i = \frac{p-1}{2}$ by Lemma 2 and the exponents of the remaining four terms $x^{2^{k-\frac{p-1}{2}-1}+1}$, $x^{2^{k-\frac{p-1}{2}-1}}$, x^{p-1}, x^2 are not congruent modulo p with each other. If 0 is congruent to $2^{k-i-1} + 1$, then we have $k - i - 1 = \frac{p-1}{2}$ by Lemma 2, it indicates that

$$i = k - \frac{p+1}{2} < \frac{p-7}{2} - \frac{p+1}{2} = -4,$$

which contradicts the fact that $i \geq 1$.

The determinant in (5) is

$$(x^{2^i+2^{k-j-1}} + x^{2^j} + x^{2^{k-i-1}+1}) + (x^{2^j+2^{k-i-1}} + x^{2^i} + x^{2^{k-j-1}+1}).$$

None of the terms in the first parenthesis is equal to any term in the second parenthesis if the exponents are reduced modulo p, and *vice versa*. Otherwise, we can deduce the contradiction of $1 \leq i < j \leq k-2$.

Likewise, the determinant in (6) can be re-arranged as

$$(x^{2^j+2^{k-\ell-1}} + x^{2^\ell+2^{k-i-1}} + x^{2^i+2^{k-j-1}}) + (x^{2^\ell+2^{k-j-1}} + x^{2^i+2^{k-\ell-1}} + x^{2^j+2^{k-i-1}}).$$

None of the terms in the first parenthesis is equal to any term in the second parenthesis if the exponents are reduced modulo p, and *vice versa*. This proves that the determinant in (6) is not divisible by $x^p + 1$, and completes the proof. □

III. Efficient Repair of One Information Failure

In this section, we always assume that information column f erases, f can be any value from 1 to k, we want to recover the bits $s_{0,f}, s_{1,f}, \ldots, s_{(p-1)\tau-1,f}$ stored in column f. Recall that we can compute the parity-check bits by (2). For notational convenience, we refer the *bits* of column i as the $p\tau$ bits $s_{0,i}$, $s_{1,i}, \ldots, s_{p\tau-1,i}$. Before giving the efficient repair algorithm, we formally define the *parity set* as follows.

Definition 1. *For $0 \leq \ell \leq p\tau - 1$, we define the ℓ-th parity set of the first, the second and the third parity column as*

$$P_{\ell,1} = \{s_{\ell,1}, s_{\ell,2}, \ldots, s_{\ell,k}\},$$

$$P_{\ell,2} = \{s_{\ell-2^0,1}, s_{\ell-2^1,2}, \ldots, s_{\ell-2^{k-2},k-1}, s_{\ell,k}\} \text{ and}$$

$$P_{\ell,3} = \{s_{\ell,1}, s_{\ell-2^{k-2},2}, s_{\ell-2^{k-3},3}, \ldots, s_{\ell-2^0,k}\}$$

respectively.

Note that all the indices in Definition 1 and throughout the paper are taken modulo $p\tau$. From definition 1, we see that the parity set $P_{\ell,j}$ consists of information bits which are used to generate the redundant bit $s_{\ell,k+j}$. When we say an information bit is repaired by a parity column, it means that we access the redundant bit of the parity column, and all the information bits in this parity set, except the erased bit. Consider the example in Fig. 1, suppose that the first column is erased. One can access the bits $s_{0,2}, s_{0,3}, s_{0,4}$ and the redundant bit $s_{0,1} + s_{0,2} + s_{0,3} + s_{0,4}$ to rebuild $s_{0,1}$ by

$$s_{0,2} + s_{0,3} + s_{0,4} + (s_{0,1} + s_{0,2} + s_{0,3} + s_{0,4}).$$

The repair algorithm is stated in Algorithm 1.

Theorem 4. *When $f \in \{1, 2, \ldots, \lceil k/2 \rceil\}$, the repair bandwidth of information column f by Algorithm 1 is*

$$(p-1)((k+2)2^{k-3} - 2^{k-f-2}).$$

Proof. By Algorithm 1, the bits $s_{\ell,f}$ are repaired by the parity sets $P_{\ell,1}$ of the first parity column for $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$ and $\ell < (p-1)\tau$. Therefore, we need to access $(p-1)\tau/2$ information bits $s_{\ell,i}$ from each of the remaining $k-1$ information columns for $i \in \{1, 2, \ldots, f -$

Algorithm 1 Efficient repair of one information failure

1: Suppose the information column f is failed.
2: **if** $f \in \{1, 2, \ldots, \lceil k/2 \rceil\}$. **then**
3: Repair the bit $s_{\ell,f}$ by the first parity, for $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$. Otherwise, repair the bit $s_{\ell,f}$ by the second parity, for $\ell \mod 2^f \in \{2^{f-1}, 2^{f-1} + 1, 2^{f-1} + 2, \ldots, 2^f - 1\}$.
4: **if** $f \in \{\lceil k/2 \rceil + 1, \lceil k/2 \rceil + 2, \ldots, k\}$. **then**
5: Repair the bit $s_{\ell,f}$ by the first parity, for $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$. Otherwise, repair the bit $s_{\ell,f}$ by the third parity, for $\ell \mod 2^f \in \{2^{f-1}, 2^{f-1} + 1, 2^{f-1} + 2, \ldots, 2^f - 1\}$.

$1, f+1, \ldots, k\}$ and $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$, and download $(p-1)\tau/2$ redundant bits $s_{\ell,k+1}$ for $\ell \mod 2^i \in \{0, 1, 2, \ldots, 2^{i-1} - 1\}$ from the first parity column. Thus, there are $k(p-1)\tau/2$ bits to be downloaded.

For $\ell \mod 2^f \in \{2^{f-1}, 2^{f-1} + 1, 2^{f-1} + 2, \ldots, 2^f - 1\}$, the bits $s_{\ell,f}$ are repaired by $P_{\ell+2^{f-1},2}$. Recall that

$$P_{\ell+2^{f-1},2} = \{s_{\ell+2^{f-1}-2^0,1}, \ldots, s_{\ell+2^{f-1}-2^{k-2},k-1}, s_{\ell+2^{f-1},k}\}.$$

So we need to access $(p-1)\tau/2$ redundant bits $s_{\ell+2^{f-1},k+2}$. For column i with $i \in \{1, 2, \ldots, f-1\}$, we need $(p-1)\tau/2$ bits $s_{\ell,i}$ for all the values of $\ell \mod 2^f$ in the set

$$\{0, 1, \ldots, 2^{f-1} - 2^{i-1} - 1, 2^f - 2^{i-1}, 2^f - 2^{i-1} + 1, \ldots, 2^f - 1\}.$$

While for column i with $i \in \{f+1, f+2, \ldots, k\}$, we need $(p-1)\tau/2$ bits $s_{\ell,i}$ for $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$.

Note that the bits $s_{\ell,i}$ for $\ell \mod 2^f \in \{0, 1, 2, \ldots, 2^{f-1} - 1\}$ and $\ell < (p-1)\tau$ have downloaded in the repair by the first parity column. Thus, we only need to download $(p-1)\tau/2$ redundant bits from the second parity column, and $(p-1)2^{k+i-f-3}$ bits from column i for $i = 1, 2, \ldots, f-1$.

We can count that the total number of bits downloaded from $k+2$ columns to repair the information column f is

$$\underbrace{k(p-1)2^{k-3}}_{\text{the first parity column}} + \underbrace{\sum_{i=1}^{f-1}(p-1)2^{k+i-f-3}}_{\text{the second parity column}}$$
$$= (p-1)((k+2)2^{k-3} - 2^{k-f-2}).$$

□

When $1 \le f \le \lceil k/2 \rceil$, the repair bandwidth of column $k+1-f$ is the same of that of column f according to Algorithm 1. Therefore, we only consider the cases of $1 \le f \le \lceil k/2 \rceil$. By Theorem 4, repair bandwidth increases with f increases. When $f = 1$, the repair bandwidth is $(k+1)(p-1)2^{k-3}$, which achieves the optimal value in (1). Even for the worst case of $f = \lceil k/2 \rceil$, the repair bandwidth is

$$(p-1)((k+2)2^{k-3} - 2^{k-\lceil k/2 \rceil - 2}) < (p-1)(k+2)2^{k-3},$$

which is strictly less than $\frac{k+2}{k+1}$ times of the value in (1).

It should be noted that the parity sets of the first parity column in our codes are the same of that of the first parity column in RDP and EVENODD. The key difference of our codes and the existing binary MDS array codes is the construction of the second and the third parity columns. First, the parity sets of the second and the third parity columns in our codes are not bits that correspond to straight lines in the array, but the bits that correspond to polygonal lines. Second, the row number of the array in our codes is divisible by 2^{k-2}. The two properties are essential for reducing the repair bandwidth.

IV. Decoding for Two Information Columns

Let a, b be integers between 1 and k. Suppose that columns a and b are erased. We want to recover the lost information bits in columns a and b by reading columns i, for $i \in \{1, 2, \ldots, k\} \setminus \{a, b\}$, and the first two parity columns.

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

$$s_{k+1}(x) = \sum_{i=1}^{k} s_i(x) \text{ and } s_{k+2}(x) = s_k(x) + \sum_{i=1}^{k-1} s_i(x)x^{2^{i-1}}.$$

Let $f_1(x)$ and $f_2(x)$ be the polynomials by subtracting the known values of $s_i(x)$, for $i \in \{1, 2, \ldots, k\} \setminus \{a, b\}$, from $s_{k+1}(x)$ and $s_{k+2}(x)$, respectively. Without loss of generality, we assume that $1 < a < b < k$. As the decoding method of the other cases is analogous. The two erasures can be repaired by solving the following system of linear equations

$$\begin{bmatrix} 1 & 1 \\ x^{2^{a-1}} & x^{2^{b-1}} \end{bmatrix} \begin{bmatrix} s_a(x) \\ s_b(x) \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}.$$

Therefore we can solve for $s_a(x)$ by $(x^{2^{a-1}} + x^{2^{b-1}})^{-1}(x^{2^{b-1}} f_1(x) + f_2(x))$, and $s_b(x)$ by $s_a(x) + f_1(x)$.

Note that $x^i f_1(x) + x^j f_2(x)$ can be computed by cyclically shifting $f_1(x)$ and $f_2(x)$ to the right by i and j, respectively, and adding the resulting polynomials. Before giving the decoding complexity, we need the following lemma about how to compute the division of the form $1/(1+x^b)$ in $R_{p\tau}$.

Lemma 5. *Given the equation $(1+x^b)s(x) = c(x)$, where b is a positive integer and $s(x), c(x) \in \mathbb{F}_2[x]/M_p^\tau(x)$. Let $(b, \tau) = a$, we have the following equation*

$$s_{(p-1)b\tau/a} = \sum_{i=1}^{\tau/a} c_{bi} + \sum_{i=2\tau/a+1}^{3\tau/a} c_{bi} + \cdots + \sum_{i=(p-3)\tau/a+1}^{(p-2)\tau/a} c_{bi},$$

where $s(x) = \sum_{i=0}^{p\tau-1} s_i x^i$ and $c(x) = \sum_{i=0}^{p\tau-1} c_i x^i$.

Proof. By the equation $(1+x^b)s(x) = c(x)$, we have,

$$c_{b(\ell\frac{\tau}{a}+1)} + c_{b(\ell\frac{\tau}{a}+2)} + \cdots + c_{b(\ell+1)\frac{\tau}{a}} = s_{b\ell\frac{\tau}{a}} + s_{b(\ell+1)\frac{\tau}{a}}.$$

We thus have that

$$\sum_{i=1}^{\tau/a} c_{bi} + \sum_{i=2\tau/a+1}^{3\tau/a} c_{bi} + \cdots + \sum_{i=(p-3)\tau/a+1}^{(p-2)\tau/a} c_{bi} + s_{(p-1)b\tau/a}$$
$$= s_0 + s_{b\tau/a} + s_{2b\tau/a} + \cdots + s_{(p-2)b\tau/a} + s_{(p-1)b\tau/a}$$
$$= s_0 + s_\tau + s_{2\tau} + \cdots + s_{(p-2)\tau} + s_{(p-1)\tau} = 0.$$

The second equality follows from the fact that $\ell b\tau/a \neq 0 \mod p\tau$ for $(b/a, p) = 1$ and $1 \le \ell \le p-1$. □

By the same argument of Lemma 5, we have that

$$s_{(p-1)b\frac{\tau}{a}+j} = \sum_{i=1}^{\tau/a} c_{bi+j} + \sum_{i=2\frac{\tau}{a}+1}^{3\tau/a} c_{bi+j} + \cdots + \sum_{i=(p-3)\frac{\tau}{a}+1}^{(p-2)\tau/a} c_{bi+j},$$

for $1 \leq j \leq a$. All the other coefficients of $s(x)$ can be computed recursively by

$$c_{(p-1)b\tau/a+\ell} = s_{(p-1)b\tau/a-b+\ell} + s_{(p-1)b\tau/a+\ell}$$

for $\ell = 1, 2, \ldots, p\tau - 1$. There are at most $\frac{3p\tau}{2}$ XORs involved in the solving $s(x)$ from $(1 + x^b)s(x) = c(x)$.

TABLE I: Comparison of binary MDS array codes.

	r	Repair	Encoding	Decoding
Gad's code	2	optimal	$k + k/2\lfloor k/2 \rfloor$	–
MDR-I [14]	2	optimal	$k - 1$	–
MDR-II [14]	2	optimal	k	–
Our code	3	optimal	$4k/3 - 1$	≈ 3
RDP	2	not optimal	$k - 1$	$2(k-1)/k$

Encoding complexity is defined as the average number of XORs needed to generate one redundant bit, and decoding complexity is defined as the ratio of the total number of XORs required to recover the erased information columns failures to the number of information bits.

Theorem 6. *The encoding complexity of $\mathcal{C}(k, 3, p)$ is at most $4k/3 - 1$, and the decoding complexity of two information failures with the above decoding method is at most*

$$3 + 1/(p-1) + 3.5/k. \tag{7}$$

Proof. By the construction, we should first compute $k\tau$ parity-check bits by (2), which involves $k\tau(p-2)$ XORs. Then generate the coefficients of degree from 0 to $(p-1)\tau - 1$ for three coded polynomials that take $3(p-1)\tau(k-1)$ XORs. Therefore, the encoding complexity is

$$\frac{k\tau(p-2) + 3(p-1)\tau(k-1)}{3(p-1)\tau} < 4k/3 - 1.$$

Consider the decoding process. Adding the parity-check bits to formulate $k-2$ data polynomials and 2 coded polynomials takes $k\tau(p-2)$ XORs. Computing the polynomials $f_1(x)$ and $f_2(x)$ involves $2(k-2)p\tau$ XORs. In the last step, recover $s_a(x)$ and $s_b(x)$ takes $3.5p\tau$ XORs at most by Lemma 5. Thus, the decoding complexity of two information erasures is

$$\frac{k\tau(p-2) + 2(k-2)p\tau + 3.5p\tau}{k(p-1)\tau} < 3 + 1/(p-1) + 3.5/k.$$

□

We summarize the comparison of binary MDS array codes in Table I. As there is no explicite decoding algorithm for Gad's code and MDR code, we don't give the decoding complexity in Table I. We can see that $\mathcal{C}(k, 3, p)$ has optimal repair bandwidth and comparable encoding and decoding complexities, compared with the existing binary MDS array codes.

V. Conclusion

In this paper, we present new binary MDS array codes with three parity columns such that the repair bandwidth of one information column is asymptotically optimal. The future work includes the extension of the construction with more parity columns and efficient repair algorithm for parity column.

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] L. Xu and J. Bruck, "X-code: MDS array codes with optimal encoding," *Information Theory, IEEE Transactions on*, vol. 45, no. 1, pp. 272–276, 1999.

[3] 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.

[4] 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.

[5] 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.

[6] Y. Zhang, C. Wu, J. Li, and M. Guo, "Tip-code: A three independent parity code to tolerate triple disk failures with optimal update complexity," in *Ieee/ifip International Conference on Dependable Systems and Networks*, 2015, pp. 136–147.

[7] M. Holland, G. Gibson, D. P. Siewiorek *et al.*, "Fast, on-line failure recovery in redundant disk arrays," in *Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Third International Symposium on*. IEEE, 1993, pp. 422–431.

[8] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Information Theory*, vol. 56, no. 9, pp. 4539–4551, September 2010.

[9] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: Mds array codes with optimal rebuilding," *Information Theory, IEEE Transactions on*, vol. 59, no. 3, pp. 1597–1616, 2013.

[10] G. K. Agarwal, B. Sasidharan, and P. Vijay Kumar, "An alternate construction of an access-optimal regenerating code with optimal sub-packetization level," in *Communications (NCC), 2015 Twenty First National Conference on*. IEEE, 2015, pp. 1–6.

[11] S. Xu, R. Li, P. P. Lee, Y. Zhu, L. Xiang, Y. Xu, and J. Lui, "Single disk failure recovery for X-code-based parallel storage systems," *Computers, IEEE Transactions on*, vol. 63, no. 4, pp. 995–1007, 2014.

[12] 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.

[13] E. En Gad, R. Mateescu, F. Blagojevic, and C. Guyot, "Repair-optimal mds array codes over gf(2)," *Mathematics*, pp. 887–891, 2013.

[14] Y. Wang, X. Yin, and X. Wang, "Mdr codes: A new class of raid-6 codes with optimal rebuilding and encoding," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 1008–1018, 2013.

[15] ——, "Two new classes of two-parity mds array codes with optimal repair," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1293–1296, 2016.

[16] H. Hou, K. W. Shum, M. Chen, and H. Li, "Basic codes: Low-complexity regenerating codes for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3053–3069, 2016.