An Efficient MDS Array Code on Toleration Triple Node Failures in Storage System

Wunan Wan
Network Engineering Department, ChengDu University of Information Technology,Cheng'Du, P. R. China
Email: nan_wwn@cuit.edu.cn

Wang Suo
Network Engineering Department, ChengDu University of Information Technology,Cheng'Du, P. R. China
Email: suowang@cuit.edu.cn

***Abstract*—Proper data placement schemes based on erasure correcting code are one of the most important components for a highly available data storage system. A new class of Maximum Distance Separable (MDS) array codes is presented for correcting triple storage failures, which is an extension of the double-erasure-correcting EVENODD code and is called the HDD-EOD code. The encoding and decoding procedures are described by geometrical line graph, which are easily implemented by soft hardware. Our analysis shows that the HDD-EOD code provides better decoding performance and higher reliability compared to other popular codes. Thus the HDD-EOD code is practically very meaningful for storage systems.**

***Index Terms*—fault tolerance, high availability, MDS array codes, triple node failures**

I. Introduction

The recent great advances in the networking storage, and in information technologies have paved the road for the introduction of new applications that put online huge amount of multimedia information, particularly in a business environment. It is essential to have highly available and reliable multiple hard disks to store huge amounts of data. Redundant Arrays of Inexpensive Disks (RAID) is an efficient approach to supply high reliability and high performance storage services[1], such as in many companies, universities, and government organizations. However, the chance of the disks' failure in RAID increases because of random damage and other reasons. To protect the data in RAID, constructing erasure codes for tolerating multiple disk failures is very important.

There are many schemes based on various erasure coding technologies [2-3]. Theoretically, in order to tolerate triple disk failures, we need at least triple redundant disks (in coding theory, this is known as the capacity of erasure channel, called Maximum Distance Separable (MDS) code).The well known Reed-Solomon code [4] is MDS code, which have been proposed in RAID, but the encoding and decoding of Reed-Solomon code require finite field arithmetic. On the other hand, the computational complexity of using RS code poses a significant problem.

Array codes are perhaps the most desirable codes that only involve XOR operations for disks storage systems, which can be more easily and efficiently implemented in hardware and/or software. In particular, array codes tend to require less complexity than the codes based on finite fields[5-18].

A few classes of MDS array codes have been successfully designed to recover double (simultaneous) storage node failures[5-8]. The recent ones include the EVENODD code[5], the X-Code[6], the B-Code [7] and the HD code[8]. However, the array codes for triple disk failure are yet to be developed. HDD1 code and HDD2 code[9]can tolerate three disk failures, but MDS property of the two code is not proved completely, and the encoding and decoding algorithms is used by Gaussian elimination. The Grid code[10],the Hover code[11],the WEAVER code[12] can recover multi storage node failure, but three array codes are not MDS code and storage efficient is about 50%. Gui-Liang Feng et al. propose the two array codes[13-14] and the papers present the Blaum code[15] and T code[16]. These codes are MDS code and can tolerate three failures, but the array codes do not easily implement by software and hardware.

In this paper we develop a new class of binary MDS array codes for three storage node failures, which is called HDD-EOD code. The HDD-EOD code is an extension of the EVENODD code, which has Horizontal and Dual Diagonal Parity. The property of a simple geometrical construction for HDD-EOD code leads to faster and easier encoding and decoding procedures.

The rest of this paper is organized as follows. Section II first briefly describes encoding of the HDD-EOD code. Section III gives the detailed decoding procedure of our scheme. We then analyze and discuss storage efficiency, encoding and decoding complexity in the HDD-EOD code and make comparisons with two related codes in Section IV. Finally, Section V concludes the paper.

II. The HDD-EOD Code Encoding

doi:10.4304/jcp.8.11.2902-2907

A. EVNODD Code and Encoding

The HDD-EOD code is an extension of the double-erasure-correcting EVENODD code, we first briefly describe the EVENODD code, which was initially proposed to address disk failures in disk array systems. Each codeword of the EVENODD code represents a two-dimensional array. The parity symbols of the EVENODD code are algebraically constructed as follows:

$$a_{u,m} = \bigoplus_{t=0}^{m-1} a_{u,t} \tag{1}$$

$$a_{u,m+1} = S_1 \oplus \left(\bigoplus_{\substack{t=0 \\ t \neq u-1}}^{m-1} a_{\langle u-t \rangle_m, t} \right) \tag{2}$$

where $S_1 = \bigoplus_{t=1}^{m-1} a_{m-1-t,t}$, S_1 is the adjuster. Let $m=5$,the following example gives a construction of EVENODD Code of size 4×7(see TABLE Ⅰ).

TABLE I.
ENCODING OF THE EVENODD CODE

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	$a_{00} \oplus a_{01} \oplus a_{02} \oplus a_{03} \oplus a_{04}$	$a_{00} \oplus a_{14} \oplus a_{23} \oplus a_{32} \oplus S_1$
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	$a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{14}$	$a_{01} \oplus a_{24} \oplus a_{33} \oplus a_{01} \oplus S_1$
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	$a_{20} \oplus a_{21} \oplus a_{22} \oplus a_{23} \oplus a_{24}$	$a_{02} \oplus a_{34} \oplus a_{11} \oplus a_{02} \oplus S_1$
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	$a_{30} \oplus a_{31} \oplus a_{32} \oplus a_{33} \oplus a_{34}$	$a_{03} \oplus a_{12} \oplus a_{21} \oplus a_{30} \oplus S_1$

where $S_1 = a_{04} \oplus a_{13} \oplus a_{22} \oplus a_{31}$

B. Geometric Encoding Description of the HDD-EOD Code

Extending from the EVENODD code, The HDD-EOD code uses the exact same encoding rules of the EVENODD code for the first two parity columns. The extension lies in the last parity column, which is constructed along diagonal of slope 2 instead of slope 1. Algebraically, the encoding of the last parity column can be represented as:

$$a_{u,m+2} = S_2 \oplus \bigoplus_{\substack{t=0 \\ \langle u-2t \rangle_m \neq m-1}}^{m-1} a_{\langle u-2t \rangle_m, t} \tag{3}$$

where $S_2 = \bigoplus_{t=1}^{m-1} a_{\langle m-1-2t \rangle_m, t}$, called the adjusters.

From the (1)-(3) equations, three parity columns of HDD-EOD code are composed of the horizontal parity column m and the two diagonal parity columns $m+1$, $m+2$, thus the proposed code is called the HDD-EOD code.

The following example gives a construction of HDD-EOD Code, Let $m=5$, the codeword is a 4×8 array, (see TABLE II).

TABLE II
ENCODING OF THE HDD-EOD CODE

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	$a_{00} \oplus a_{01} \oplus a_{02} \oplus a_{03} \oplus a_{04}$	$a_{00} \oplus a_{14} \oplus a_{23} \oplus a_{32} \oplus S_1$	$a_{00} \oplus a_{31} \oplus a_{12} \oplus a_{24} \oplus S_2$
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	$a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{14}$	$a_{01} \oplus a_{24} \oplus a_{33} \oplus a_{01} \oplus S_1$	$a_{10} \oplus a_{22} \oplus a_{03} \oplus a_{34} \oplus S_2$
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	$a_{20} \oplus a_{21} \oplus a_{22} \oplus a_{23} \oplus a_{24}$	$a_{02} \oplus a_{34} \oplus a_{11} \oplus a_{02} \oplus S_1$	$a_{20} \oplus a_{01} \oplus a_{32} \oplus a_{13} \oplus S_2$
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	$a_{30} \oplus a_{31} \oplus a_{32} \oplus a_{33} \oplus a_{34}$	$a_{03} \oplus a_{12} \oplus a_{21} \oplus a_{30} \oplus S_1$	$a_{30} \oplus a_{11} \oplus a_{23} \oplus a_{04} \oplus S_2$

where $S_1 = a_{04} \oplus a_{13} \oplus a_{22} \oplus a_{31}, S_2 = a_{21} \oplus a_{02} \oplus a_{33} \oplus a_{14}$

For simplicity, the encoding procedure is depicted in Fig. 1. Geometrically speaking, the adjusters S_1 and S_2 are just the checksums along diagonals of the slope from the coordinate $(m-1, 0)$, each symbol of the parity column is just the checksums along diagonals of slopes $0, 1, 2$ and the adjuster, respectively.

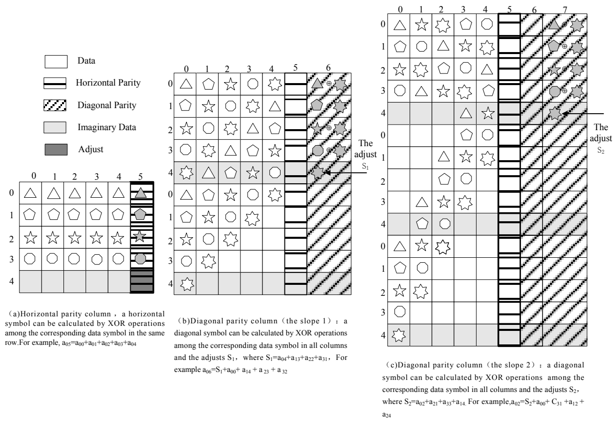

Figure 1. HDD-EOD code encoding procedure.

III. ERASURE DECODING ALGORITHM

We can give an efficient erasure decoding algorithm for the proposed code by using the geometric description. As the encoding algorithm of the code, decoding algorithms do not require any finite field operations. Instead, the only operations needed are just cyclic shifts and XORs, which can be implemented very efficiently with software and /or hardware.

The decoding process of the HDD-EOD code can be divided into cases based on different erasure patters: 1) decoding without parity erasures, where all erasures are information columns; and 2) decoding with parity erasures, where at least one erasure is a parity column. Since extending from the EVENODD code[5], the HDD-EOD code uses the exact same decoding rules for the recovery from arbitrary twice erasures. In the section the main case of decoding algorithm with three erased information columns is given.

A. *Decoding without Parity Erasures*

We consider the recovery of triple information column erasures at position i, j, k.

Firstly, we can recovery the two adjusters S_1 and S_2 in equations 1-3 by three parity columns. In general, this step of adjusters recovery can be summarized as:

$$S_1 = S' \oplus \left(\bigoplus_{u=0}^{m-2} a_{u,m+1} \right) \tag{4}$$

$$S_2 = S' \oplus \left(\bigoplus_{u=0}^{m-2} a_{u,m+2} \right) \tag{5}$$

Where S' is denoted by the XOR sums of all the symbols in first parity column, i.e., $S' = \left(\bigoplus_{u=0}^{m-2} a_{u,m} \right)$.

Secondly, Let to be $0 \le u \le m-1$,we define the syndrome only including triple information column erasures i , j , k . $\widetilde{S}^{(0)}, \widetilde{S}^{(1)}, \widetilde{S}^{(2)}$ be calculated by following as:

$$\widetilde{S_u}^{(0)} = \bigoplus_{\substack{t=0 \\ t \ne i,j,k}}^{m} a_{u,t} \tag{6}$$

$$\widetilde{S_u}^{(1)} = S_1 \oplus a_{u,m+1} \oplus \left(\bigoplus_{\substack{t=0 \\ t \ne i,j,k}}^{m} a_{\langle u-t \rangle_m, t} \right) \tag{7}$$

$$\widetilde{S_u}^{(2)} = S_2 \oplus a_{u,m+2} \oplus \left(\bigoplus_{\substack{t=0 \\ t \ne i,j,k}}^{m} a_{\langle u-2t \rangle_m, t} \right) \tag{8}$$

From the geometric description of encoding, we can also descript the syndrome equations.

For example, let $m=5$, the missed triple information columns $i=1$, $j=2$, $k=4$ as shown in TABLE Ⅲ. The erased information symbol is represented by "?".

TABLE Ⅲ
THE MISSED TRIPLE INFORMATION COLUMNS

a_{00}	?	?	a_{03}	?	$a_{00} \oplus a_{01} \oplus a_{02} \oplus a_{03} \oplus a_{04}$	$a_{00} \oplus a_{14} \oplus a_{23} \oplus a_{32} \oplus S_1$	$a_{00} \oplus a_{31} \oplus a_{12} \oplus a_{24} \oplus S_2$
a_{10}	?	?	a_{13}	?	$a_{10} \oplus a_{11} \oplus a_{12} \oplus a_{13} \oplus a_{14}$	$a_{01} \oplus a_{24} \oplus a_{33} \oplus a_{01} \oplus S_1$	$a_{10} \oplus a_{22} \oplus a_{03} \oplus a_{34} \oplus S_2$
a_{20}	?	?	a_{23}	?	$a_{20} \oplus a_{21} \oplus a_{22} \oplus a_{23} \oplus a_{24}$	$a_{02} \oplus a_{34} \oplus a_{11} \oplus a_{02} \oplus S_1$	$a_{20} \oplus a_{01} \oplus a_{32} \oplus a_{13} \oplus S_2$
a_{30}	?	?	a_{33}	?	$a_{30} \oplus a_{31} \oplus a_{32} \oplus a_{33} \oplus a_{34}$	$a_{03} \oplus a_{12} \oplus a_{21} \oplus a_{30} \oplus S_1$	$a_{30} \oplus a_{11} \oplus a_{23} \oplus a_{04} \oplus S_2$

where $S_1 = a_{04} \oplus a_{13} \oplus a_{22} \oplus a_{31}, S_2 = a_{21} \oplus a_{02} \oplus a_{33} \oplus a_{14}$

According the equations(4)-(8), the $\widetilde{S}^{(0)}$, $\widetilde{S}^{(1)}$ and $\widetilde{S}^{(2)}$ can be calculated. From the geometric description of encoding, we can also descript the syndrome equations by geometric graph as shown Fig. 2 .

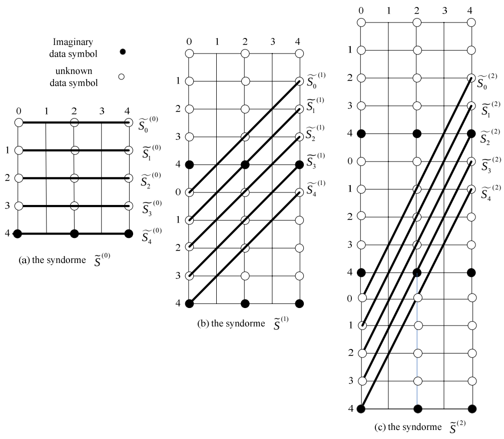

Figure 2. Decoding syndromes of the HDD-EOD code.

According to the Fig. 2, it is easy to show that every syndrome includes two unknown symbols at least, and at most the one unknown symbol of the same missed column. We cannot solve the unknown symbols directly. So, we can eliminate two column erasures by the elimination method of iterative syndromes. The middle erased column can be recovered by a group of cyclic equations with at most two unknown symbols in the middle column erasures. Before proving the correct of the algorithm, we give an example of decoding process.

Suppose that columns 1, 2 and 4 have been erased in Fig. 3.

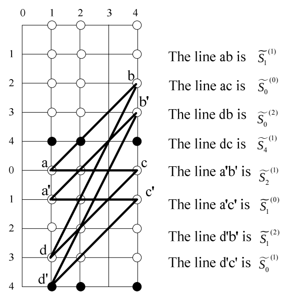

Figure 3. Decoding procedure of the HDD-EOD Code.

The recovery procedure of the middle column consists of the following steps.

Firstly, unknown symbols of the 1th and 4th column erasures can be eliminated by the syndromes, and the cyclic linear equations with the 2th column erasures can be given.

(1) The two graph circuits $a \to b \to c \to d$ and $a' \to b' \to c' \to d'$ in Fig. 3 are mapped by the overlapping of several syndrome linear equations. Thus, we can get the equation only includes the unknown information symbols of the middle column, as followed respectively:

$$a_{4,2} \oplus a_{1,2} \oplus a_{0,2} \oplus a_{2,2} = \widetilde{S_1}^{(1)} \oplus \widetilde{S_0}^{(2)} \oplus \widetilde{S_4}^{(1)} \oplus \widetilde{S_0}^{(0)} \tag{9}$$

$$a_{0,2} \oplus a_{2,2} \oplus a_{2,2} \oplus a_{3,2} = \widetilde{S_2}^{(1)} \oplus \widetilde{S_1}^{(2)} \oplus \widetilde{S_0}^{(1)} \oplus \widetilde{S_1}^{(0)} \tag{10}$$

(2) Next, the foregoing two graph circuits in Fig. 3 are overlapped and mapped the equation with two unknown symbols at most in the 2th column erasures. Thus, we can get:

$$\begin{aligned} a_{4,2} \oplus a_{3,2} &= \widetilde{S}_2^{(1)} \oplus \widetilde{S}_1^{(2)} \oplus \widetilde{S}_0^{(1)} \oplus \widetilde{S}_1^{(0)} \\ &\oplus \widetilde{S}_1^{(1)} \oplus \widetilde{S}_0^{(2)} \oplus \widetilde{S}_4^{(1)} \oplus \widetilde{S}_0^{(0)} \end{aligned} \tag{11}$$

(3) Repeating this process, the other cycle equations with two unknown symbols at most in the 2th column erasures are obtained as follows:

$$\begin{aligned} a_{3,2} \oplus a_{2,2} &= \widetilde{S}_0^{(1)} \oplus \widetilde{S}_4^{(2)} \oplus \widetilde{S}_3^{(1)} \\ &\oplus \widetilde{S}_4^{(0)} \oplus \widetilde{S}_1^{(1)} \oplus \widetilde{S}_0^{(2)} \oplus \widetilde{S}_4^{(1)} \oplus \widetilde{S}_0^{(0)} \end{aligned} \tag{12}$$

$$\begin{aligned} a_{2,2} \oplus a_{1,2} &= \widetilde{S}_4^{(1)} \oplus \widetilde{S}_3^{(2)} \oplus \widetilde{S}_2^{(1)} \\ &\oplus \widetilde{S}_3^{(0)} \oplus \widetilde{S}_0^{(1)} \oplus \widetilde{S}_4^{(2)} \oplus \widetilde{S}_3^{(1)} \oplus \widetilde{S}_4^{(0)} \end{aligned} \tag{13}$$

$$\begin{aligned} a_{1,2} \oplus a_{0,2} &= \widetilde{S}_3^{(1)} \oplus \widetilde{S}_2^{(2)} \oplus \widetilde{S}_1^{(1)} \\ &\oplus \widetilde{S}_2^{(0)} \oplus \widetilde{S}_4^{(1)} \oplus \widetilde{S}_3^{(2)} \oplus \widetilde{S}_2^{(1)} \oplus \widetilde{S}_3^{(0)} \end{aligned} \tag{14}$$

(4) According to the foregoing assumption, we can known that $a_{4,2}$ is imaginary symbol $a_{4,2} = 0$.

Thus we can the information symbol $a_{3,2}$ .We can get all the symbols of the 2th column erasures in turn: $a_{3,2} \to a_{2,2} \to a_{1,2} \to a_{0,2}$.

Theorem 1. If three columns i, j, k have been erased, where $0 \le i < j < k \le m-1$,Let $r = j - i$, $s = k - j$, an positive integer l_d, $1 \le l_d < m$ always exists , where l_d is determined by the equation $\langle s - l_d r \rangle_m = 0$, we have:

$$a_{u,j} \oplus a_{\langle u+2s \rangle_m, j} = \sum_{v=0}^{l_d-1} \begin{pmatrix} \widetilde{S}^{(1)}_{\langle u+vr+j \rangle_m} \oplus \widetilde{S}^{(2)}_{\langle u+vr+j+k \rangle_m} \\ \oplus \widetilde{S}^{(1)}_{\langle u+vr+j+r+s \rangle_m} \oplus \widetilde{S}^{(0)}_{\langle u+vr+r \rangle_m} \end{pmatrix} \tag{15}$$

Proof. From the equations(6)-(8), we have :

$$\widetilde{S}^{(1)}_{\langle u+j \rangle_m} = a_{\langle u+r \rangle_m, i} \oplus a_{u,j} \oplus a_{\langle u-s \rangle_m, k} \tag{16}$$

$$\widetilde{S}^{(2)}_{\langle u+j+k \rangle_m} = a_{\langle u+s+2r \rangle_m, i} \oplus a_{\langle u+s \rangle_m, j} \oplus a_{\langle u-s \rangle_m, k} \tag{17}$$

$$\widetilde{S}^{(1)}_{\langle u+j+r+s \rangle_m} = a_{\langle u+s+2r \rangle_m, i} \oplus a_{\langle u+s+r \rangle_m, j} \oplus a_{\langle u+r \rangle_m, k} \tag{18}$$

$$\widetilde{S}^{(0)}_{\langle u+r \rangle_m} = a_{\langle u+r \rangle_m, i} \oplus a_{\langle u+r \rangle_m, j} \oplus a_{\langle u+r \rangle_m, k} \tag{19}$$

therefore ,we have

$$\begin{aligned} & a_{u,j} \oplus a_{\langle u+r \rangle_m, j} \oplus a_{\langle u+s \rangle_m, j} \oplus a_{\langle u+s+r \rangle_m, j} \\ &= \widetilde{S}^{(1)}_{\langle u+j \rangle_m} \oplus \widetilde{S}^{(2)}_{\langle u+j+k \rangle_m} \oplus \widetilde{S}^{(1)}_{\langle u+j+r+s \rangle_m} \oplus \widetilde{S}^{(0)}_{\langle u+r \rangle_m} \end{aligned} \tag{20}$$

Thus $\sum_{v=0}^{l_d-1} \begin{pmatrix} \widetilde{S}^{(1)}_{\langle u+vr+j \rangle_m} \oplus \widetilde{S}^{(2)}_{\langle u+vr+j+k \rangle_m} \\ \oplus \widetilde{S}^{(1)}_{\langle u+vr+j+r+s \rangle_m} \oplus \widetilde{S}^{(0)}_{\langle u+vr+r \rangle_m} \end{pmatrix}$,we can get:

$$\begin{array}{lllllllll} & a_{u,j} & \oplus & \cancel{a}_{\langle u+r \rangle_m, j} & \oplus\cdots\oplus & \cancel{a}_{\langle u+(l_d-2)r \rangle_m, j} & \oplus & \cancel{a}_{\langle u+(l_d-1)r \rangle_m, j} \\ \oplus & \cancel{a}_{\langle u+r \rangle_m, j} & \oplus & \cancel{a}_{\langle u+2r \rangle_m, j} & \oplus\cdots\oplus & \cancel{a}_{\langle u+(l_d-1)r \rangle_m, j} & \oplus & a_{\langle u+l_d r \rangle_m, j} \\ \oplus & a_{\langle u+s \rangle_m, j} & \oplus & \cancel{a}_{\langle u+r+s \rangle_m, j} & \oplus\cdots\oplus & \cancel{a}_{\langle u+(l_d-2)r+s \rangle_m, j} & \oplus & \cancel{a}_{\langle u+(l_d-1)r+s \rangle_m, j} \\ \oplus & \cancel{a}_{\langle u+r+s \rangle_m, j} & \oplus & \cancel{a}_{\langle u+2r+s \rangle_m, j} & \oplus\cdots\oplus & \cancel{a}_{\langle u+(l_d-1)r+s \rangle_m, j} & \oplus & c_{\langle u+l_d r+s \rangle_m, j} \end{array}$$

since $\langle s - l_d r \rangle_m = 0$, therefore

$$a_{\langle u+s \rangle_m, j} = a_{\langle u+l_d r \rangle_m, j} \tag{21}$$

thus, the equation (15) is proved.

From theorem 1, we can first get the unknown information symbol $a_{\langle m-1+2s \rangle_m, j}$ because imaginary symbol $a_{m-1,j} = 0$ .Thus we can recovery all the information symbols of the middle erased column in turn $a_{\langle m-1+2s \rangle_m, j}$, $a_{\langle m-1+4s \rangle_m, j}$, $\cdots$, $a_{\langle m-1+2(m-2)s \rangle_m, j}$.

B. Decoding with Parity Erasures

We consider the recovery of two information column erasures and one parity column erasures at position i, j, k ,where $0 \le i < j \le m-1$, $m \le k \le m+2$ .We have three case:

Case1: $k = m+2$,i.e., the last parity column $m+2$ has failed. The HDD-EOD code uses the EVENODD decoding rules for the recovery from arbitrary erasures of the columns i, j by the horizontal parity column m and the one diagonal parity columns $m+1$.

Case 2: $k = m+1$, namely, the parity column of slope 1 are failed, we can reconstruct the failed columns by the horizontal parity column m and the one diagonal parity columns, $m+2$, the recovery algorithms is modification of the EVENODD decoding, we do not give in detail.

Case 3: $k = m$, the horizontal parity column m has failed. We can reconstruct the failed columns by the two diagonal parity columns $m+1$, $m+2$.

In this case, $0 \le i < j \le m-1$, $k = m$,we can get two adjusters S_1, S_2 by the following as:

(1) Firstly, we can get $S_1 \oplus S_2$ by the following equation:

$$S_1 \oplus S_2 = \left(\bigoplus_{u=0}^{m-2} a_{u,m+1} \right) \oplus \left(\bigoplus_{u=0}^{m-2} a_{u,m+2} \right) \tag{22}$$

(2) Next, we can divide into three cases: $i = 0$ or $i > 0, i + j = m$, or $i > 0, i + j \ne m$.

If $i = 0$, from the definition of S_2, we can get S_2 by the following equation:

$$S_2 = a_{j-1,m+1} \oplus a_{j-1,m+2} \quad (23)$$

If $i > 0, i + j = m$, from the definition of S_2, we can get S_2 by the following equation:

$$S_2 = a_{i-1,m+1} \oplus a_{i-1,m+1} \quad (24)$$

If $i > 0, i + j \neq m$, from the definition of S_2, we can get S_2 by the following equation:

$$S_2 = a_{i-1,m+1} \oplus a_{i-1,m+1} \oplus a_{i-1,m+2} \quad (25)$$

From the geometric description of encoding, we can also descript the recovery process of S_2.

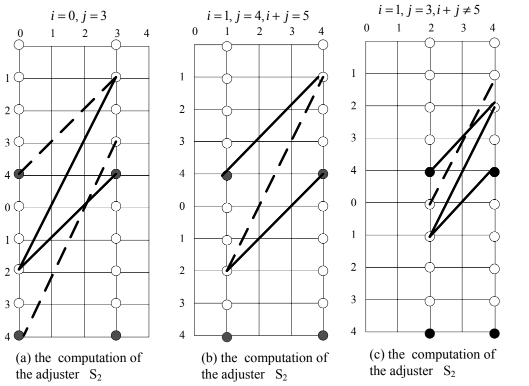

Figure 4. Recovery process of the adjuster S_2

For example, the missed one parity column $k = 5$, the two missed information column have three case. The first case is $i = 0$, $j = 3$ as shown in Fig. 4(a).The second case is $i \neq 0$ and $i + j = 5$ in Fig. 4(b).The third case is $i \neq 0$ and $i + j \neq 5$ in Fig. 4(c).

(3) From the equations (21)-(24), the adjuster S_1 can recover. The next step is similar to the decoding of the case without parity erasures. we leave to interested readers.

According all the above cases together, we can get the following theorem:

Theorem 2.The HDD-EOD code can recover any triple column erasures and it is a MDS code.

IV. The Property of HDD-EOD Code

In this section we explore three metrics of storage efficiency, encoding and decoding performance and update complexity .We then compare the HDD-EOD code to other codes with respect to these metrics.

The storage efficiency represents the fraction of the storage space that can be used for independent data.

According to the theory terminology, we know the HDD-EOD code gives the optimal storage efficiency because it is the MDS code.

Encoding and decoding complexity is particularly crucial parameter when the codes are used in storage systems. Due to special properties of array codes, the encoding and decoding procedures are performed with pure XOR and shift operations, we compare the complexity of HDD-EOD code with a Reed Solomon (RS)[17], Blaum Codes[15].

We assume that the every symbol is a bit. The encoding complexity is defined that the ratio of all XOR operations for encoding to all information blocks in the array. According to the encoding procedure in section II, the number of XORs operation is $(m-1)^2$ in the first parity column. The number of XORs in other two parity column is $2(m-1)^2 + 2(m-2)$. By adding the total, we conclude that the HDD-EOD Code needs a total of $3(m-1)^2 + 2(m-2)$ XOR operations. Thus, the encoding complexity of the HDD-EOD code is $3-(m+1)/(m-1)m$.

The decoding complexity is defined that the ratio of all XOR operations for decoding to all information blocks in the array. According to the decoding procedure in section III, In step 1, the syndrome calculations of any parity direction for a code block without erasures take $3(m-3)(m-1)+2(m-1)$. In step 2 the number of XORs is $(4l_d-1)(m-1)+(m-2)$. In summary, the total number of XORs required to decode triple information column erasures takes $(4l_d-2+3m)(m-1)-3$. Thus ,we can get the decoding complexity is $3+(4l_d(m-1)-2)/m(m-1)$.

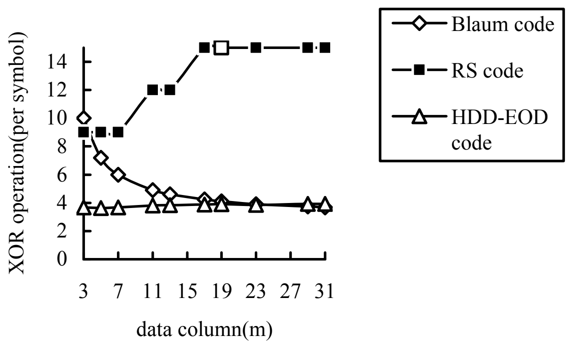

Figure 5. The comparison of decoding complexity

We compare the erasure decoding complexity of the HDD-EOD code to two other XOR-based codes in Fig 5.,one proposed by Blaum et al.,and the other code by Blomer et.al. The erasure decoding of the Blaum code require the total number of XORs is $(3m+21)(m-1)$ [15], The XOR-Based codes decoding algorithm in[17] involves krL^2 and r^2 operations in finite field $GF(2^L)$ .Thus, the RS codes normalized decoding complexity is kL.

Comparison results with HDD-EOD code are shown in Fig. 5,where can see that the complexity of the Blaum code is rather high for small k values, and the complexity of HDD-EOD code decoding fairly constant, and the complexity of The XOR-Based codes is the highest. The HDD-EOD code is thus probably more desirable than other two codes.

V. CONCLUSIONS

We have proposed a new MDS array code for triple storage nodes in this paper. The HDD-EOD code has lower complexity encoding, erasure decoding than RS code and the Blaum code have.

We will present an extension of the HDD-EOD code to a more general case in a sequel paper.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work was supported in part by National Natural Science Foundation 60873216, and Key Project of Sichuan Provincial Department of Education 12ZA223.

REFERENCES

[1] PATTERSON D. A., GIBSON G. A., KATZ R.H. A Case for Redundant Arrays of Inexpensive Disks(RAID). [C]// Proceedings of ACM SIGMOD,1988,109~116 ACM SIGMOD conference proceedings, Chicago, IL,USA: ACM Press, 1988: 109-116.
[2] XIANG H L, JI W S. Summary of Research for Erasure Code in Storage System[J].Journal of Compute Research and Development,2012,49（1）：1-11.
[3] LI M, SHU J, ZHENG J. GRID codes: Strip-based Erasure Code with High Fault Tolerance for Storage Systems[J]. ACM Transactions on Storage, 2009，4(4):1-22.
[4] PLANK J. S. A Tutorial on Reed-Solomon Coding for Fault Tolerance in RAID-Like Systems[J]. Software Practice and Experience (SPE) 1997,27(9): 995-1012.
[5] BLAUM M., BRADY J., BRUCK J., Menon J.. EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures[J]. IEEE Trans. Computer. 1995, 44(2): 192-202.
[6] XU L., BRUCK J.. X-Code: MDS Array Codes with Optimal Encoding[J]. IEEE Trans. on Information theory, 1999, 45(1): 272-276.
[7] XU L., BOHOSSIAN V., BRUCK J., WAGNER D. G.. Low Density MDS Codes and Factors of Complete Graphs[J]. IEEE Trans. on Information Theory, 1999, 45(6): 1817-1826.
[8] CHEN T W., SHEN G W., XU B H., et al.H-Code：A Hybird MDS Array Code to Optimize Partial Stripe Writes in RAID-6[C].// Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium, Anchorage, Alaska, USA, IEEE Press,2011:782-793.
[9] TAU Chih-shing, WANG Tzone-I. Efficient Parity Placement Schemes for Tolerating Triple Disk Failures in RAID Architectures[C]// Proceedings of the 17th International Conference on Advanced Information Networking and Applications(AINA’03),Washington DC,IEEE Press,2003:132-138.
[10] M. Li, J. Shu, and W. Zheng. GRID codes: Strip-based erasure code with high fault tolerance for storage systems[J]. ACM Transactions on Storage, 4(4):Article 15, January 2009.
[11] HAFNER J. L.. HoVer Erasure Codes for Disk Arrays[C]. // Proceedings of the 2006 Int Conf on Dependable Systems and Networks. Piscataway,NJ:IEEE Press,2010:217-226.
[12] HAFNER J. L.. WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems[C]. // Proceedings 4th Usenix Conference on File and Storage Technologies (FAST-2005). San Francisco, CA, USENIX Association,2005:211-214.
[13] FENG G.-L., DENG R., BAO F., SHEN J.-C.. New Efficient MDS Array Codes for RAID, Part I: Reed Solomon Like Codes for Tolerating Three Disk Failures[J]. IEEE Trans. Computers, 2005,54(9): 1071-1080.
[14] FENG G.-L., DENG R., BAO F., SHEN J.-C. New Efficient MDS Array Codes for RAID, Part II: Rabin-Like Codes for Tolerating Multiple (⩾4)Disk Failures[J]. IEEE Trans. Computers, 2005,54(12): 1473-148.
[15] BLAUM M., ROTH R.M., VARDY A. MDS Array Codes with Independent Parity Symbols[J]. IEEE Trans. Information Theory,1996,42(2):529-542.
[16] SHENG L, GANG W, STONES DS, et al. T-code: 3 Erasure Longest Lowest-Density MDS Codes [J]. IEEE Journal on Selected Areas in Communications, 2010, 28(2):289-296.
[17] BLOEMER J. M., KALFANE M., KARPINSKI, R.. An XOR-based Erasure-Resilient Coding Scheme[R]., Technical report at ICSI, ICSI TR-95-048,August 1995.

Wunan Wan, received the Ph.D. degree in 1996 from Chengdu Institute of Computer Applications, Chinese Academy of Sciences. She is currently a postdoctoral researcher at University of Electronic Science and Technology of China, ChengDu, China. She has been an associate professor at ChengDu University of Information Technology. Her research interests are information security and coding theory.

Wang Suo, is currently a Ph.D. candidate at Si’Chuan University, ChengDu. He has been a lecturer. His research interests are information security and coding for distributed storage.