An Improved Decoding Algorithm for Generalized RDP Codes

Zhijie Huang, Hong Jiang, *Fellow, IEEE*, and Ke Zhou, *Member, IEEE*

***Abstract*—The generalized RDP codes are identified as the most practical and efficient triple-erasure correcting MDS (maximum-distance separable) codes, since they can encode optimally and support arbitrary code length. However, none of the existing decoding algorithms for these codes is adequate in that their decoding complexity leaves a relatively large room for improvement. In this letter, we present an improved decoding algorithm for the generalized RDP codes. Compared with the existing decoding algorithms, the proposed algorithm has a much lower decoding complexity, which is very close to the theoretical lower bound.**

***Index Terms*—Array codes, maximum-distance separable (MDS) codes, generalized RDP codes, storage systems.**

I. Introduction

MDS (maximum-distance separable) codes have been widely used in data communication and storage systems since they can correct a prescribed number of erasures (errors) with a minimum amount of redundancy. A particular class of such codes is *MDS array codes*, in which symbols are column vectors and hence each codeword is a two-dimensional array. In these codes, a column (symbol) is said to be erroneous if at least one element in that column is erroneous. MDS array codes over GF(2) are especially practical in that the encoding and decoding procedures of such codes use only simple XOR and cyclic shift operations, thus are much more efficient than the well-known Reed–Solomon codes in terms of computational complexity [1].

A typical application scenario of MDS array codes is the Redundant Arrays of Inexpensive Disks (RAID) architectures [2], which are deployed widely in modern computing systems as a storage building block. Many MDS array codes have been specially designed for RAID, including double-erasure correcting codes such as EVENODD [3], RDP [4], Liberation [5] and MDR codes [6], and triple-erasure correcting codes such as generalized EVENODD [7], STAR [8], and generalized RDP [9], [10] codes. Among these codes, RDP and generalized RDP represent the current state of the art of double-erasure and triple-erasure correcting codes respectively, since they have the lowest encoding and decoding complexities.

In RDP codes, each codeword contains k data columns and two parity columns, and their encoding/decoding procedure requires only $k-1$ XORs for each parity/missing bit when either $k+1$ or $k+2$ is a prime number. This is shown in [4] to be the lower bound of encoding/decoding complexity for MDS codes with the same parameters. The generalized RDP codes generalize the original RDP codes by adding a third parity column to the array so that the resultant codes can correct up to three column erasures. The encoding complexity of the generalized RDP codes remains optimal. However, for the decoding complexity, this is not the case — the decoding complexity is about $9 \sim 38$ percent and about $10 \sim 20$ percent higher than the lower bound using the decoding algorithms proposed in [9] and [10] respectively when $2 \leq k \leq 30$. Notice that in order to provide sufficient reliability, k is usually quite small (in most cases $k \leq 20$) in practice [4], [10].

In this paper, we present an improved decoding algorithm for generalized RDP codes. This new algorithm is motivated by our observation that both existing decoding algorithms are not optimized in that they involve some unnecessary computations during decoding. Our new decoding algorithm eliminates these unnecessary computations by altering the reconstruction order of the erased columns and by *reusing* certain *intermediate* results during decoding. It is shown in Section IV that the decoding complexity of the proposed algorithm is improved to be $3 \sim 12$ percent higher than the lower bound when $2 \leq k \leq 30$. For the storage systems employing generalized RDP codes, the proposed decoding algorithm can provide the following benefits:

- accelerating the triple-erasure recovery, and hence enhancing the system availability and shortening the response time of degraded reads during recovery.
- reducing the number of XORs required for triple-erasure decoding, and hence saving the CPUs' energy consumption to a certain extent.

II. The Generalized RDP Codes

A codeword of generalized RDP is an $(p-1) \times (p+2)$ array consisting of $p-1$ data columns and three parity columns, where p is an odd prime [9]. In what follows, we use $b_{i,j}$ to denote the ith bit in the jth column, where $0 \leq i \leq p-2$ and $0 \leq j \leq p+1$. Moreover, we let $\langle x \rangle = x \bmod p$, and $\langle x/2 \rangle = \langle x \cdot \frac{p+1}{2} \rangle$. To facilitate the description, we also assume that $b_{p-1,j} = 0$ for $0 \leq j \leq p-1$. Then, the three parity columns are obtained as follows:

$$b_{i,p-1} = \bigoplus_{j=0}^{p-2} b_{i,j} \tag{1}$$

$$b_{i,p} = \bigoplus_{j=0}^{p-1} b_{\langle i-j \rangle, j} \tag{2}$$

Manuscript received August 12, 2015; revised January 18, 2016; accepted January 21, 2016. Date of publication January 27, 2016; date of current version April 7, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61232004 and Grant 61502189, and in part by the U.S. National Science Foundation under Grant CNS-1016609 and Grant CNS-1116606. The associate editor coordinating the review of this paper and approving it for publication was Z. Ma. *(Corresponding author: Ke Zhou.)*

Z. Huang and K. Zhou are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: jayzy_huang@hust.edu.cn; k.zhou@hust.edu.cn).

H. Jiang is with the Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76010 USA (e-mail: hong.jiang@uta.edu).

Digital Object Identifier 10.1109/LCOMM.2016.2522414

1558-2558 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

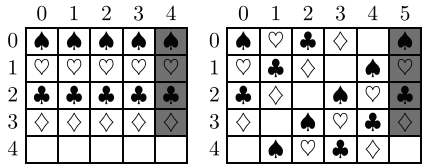

Figure 1. Row and diagonal parity sets of RDP codes with $p = 5$

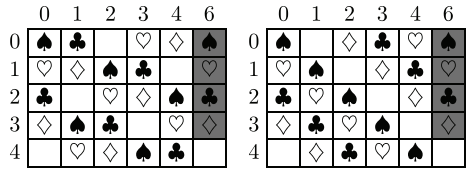

Figure 2. The third types of parity sets of MB-GRDP and RTP codes

$$b_{i,p+1} = \bigoplus_{j=0}^{p-1} b_{\langle i-s\cdot j\rangle,j} \tag{3}$$

where $0 \le i \le p-2$ and s depends on the specific variant of generalized RDP codes. For the generalized RDP codes presented by Mario Blaum in [9], $s = 2$, i.e., the parity sets corresponding to the last parity column are along lines of slope 2. For the generalized RDP codes presented in [10], $s = -1$, i.e., the parity sets corresponding to the last parity column are along lines of slope -1. Hereinafter we refer to the former as MB-GRDP and the latter as RTP.

As an example, Figure 1 shows the parity sets of the original RDP with $p = 5$, while Figure 2 shows the parity sets corresponding to the third parity column of MB-GRDP and RTP. Notice that the row parity column has to be computed first since it is involved in the computation of the other parity columns, and that the $(p-1)$th row is fictitious.

Both MB-GRDP and RTP achieve the optimal encoding in that they require exactly $k-1$ XOR operations for computing each parity bit, where k is the number of data columns. They can also decode optimally if the three erased columns contain at least one of the last two parity columns. However, if three of the first p columns are erased, then their decoding performances are *not so well*. Although MB-GRDP and RTP have their respective decoding algorithms, the basic ideas behind them are somewhat similar. In general, if three of the first p columns are erased, the decoding procedures of the two algorithms can be summarized as follows:

1. Compute the *latent* parity bits corresponding to the last parity sets that are ignored during encoding. Specifically, compute $b_{p-1,p}$ and $b_{p-1,p+1}$ as $b_{p-1,p} = \bigoplus_{i=0}^{p-2} b_{i,p}$ and $b_{p-1,p+1} = \bigoplus_{i=0}^{p-2} b_{i,p+1}$.
2. Compute the $3p-1$ syndromes corresponding to the $3p-1$ parity sets.
3. Obtain a set of equations containing only the missing bits in the second erased column from (1)–(3) by elimination.
4. Reconstruct the second erased column using the equations obtained in Step 3.
5. Reconstruct the other two erased columns using the two-erasure correcting algorithm presented in [4].

In the above procedures, the two *latent* parity bits and their corresponding syndromes are essential to Step 3, which makes the two existing decoding algorithms somewhat inefficient.

III. An Improved Decoding Algorithm

In this section we present an improved decoding algorithm that can notably reduce the computational complexity. In general, we differentiate between equidistant erasures and non-equidistant erasures. For non-equidistant erasures, we first use elimination to obtain $2(p-1)$ equations containing only the missing bits in two of the erased columns, then use these equations to reconstruct the two erased columns, and finally reconstruct the remaining erased column using row parities. In this way, we do not need to compute the *latent* parity bits and their *corresponding syndromes*. For equidistant erasures, we farthest reduce the decoding complexity by *reusing* certain *intermediate* results.

As explained in the last section, we only deal with the cases where *three of the first p columns of the array are erased*. In what follows, we assume that the ℓth, mth and rth columns are erased, where $0 \le \ell < m < r \le p-1$. Moreover, for simplicity, we will use RTP as a use case to demonstrate how the new algorithm works. Nevertheless, the interested readers can adopt it to MB-GRDP easily.

A. Correcting Any Three Erasures

First, we compute the row, diagonal, and anti-diagonal syndromes as follows:

$$S_i^{(0)} = \bigoplus_{\substack{j=0\\ j\ne \ell,m,r}}^{p-1} b_{i,j} \tag{4}$$

$$S_i^{(1)} = b_{i,p} \oplus \left(\bigoplus_{\substack{j=0\\ j\ne \ell,m,r}}^{p-1} b_{\langle i-j\rangle,j} \right) \tag{5}$$

$$S_i^{(-1)} = b_{i,p+1} \oplus \left(\bigoplus_{\substack{j=0\\ j\ne \ell,m,r}}^{p-1} b_{\langle i+j\rangle,j} \right) \tag{6}$$

where $0 \le i \le p-2$. Let $d_1 = m-\ell$, $d_2 = r-m$, and $\delta = d_1 + d_2 = r-\ell$, then from (1)–(6) we have

$$b_{i,\ell} \oplus b_{i,m} \oplus b_{i,r} = S_i^{(0)} \tag{7}$$

$$b_{\langle i+d_1\rangle,\ell} \oplus b_{i,m} \oplus b_{\langle i-d_2\rangle,r} = S_{\langle i+m\rangle}^{(1)}, i \ne p-1-m \tag{8}$$

$$b_{\langle i-d_1\rangle,\ell} \oplus b_{i,m} \oplus b_{\langle i+d_2\rangle,r} = S_{\langle i-m\rangle}^{(-1)}, i \ne m-1 \tag{9}$$

where $0 \le i \le p-1$. Notice that for $0 \le j \le p-1$, $b_{p-1,j} = 0$, thus $S_{p-1}^{(0)} = 0$. Adding (7) to (8) and (9) respectively we get

$$\mathcal{L}_i \oplus \mathcal{R}_i = S_i^{(0)} \oplus S_{\langle i+m\rangle}^{(1)}, i \ne p-1-m \tag{10}$$

$$\mathcal{L}_{\langle i-d_1\rangle} \oplus \mathcal{R}_{\langle i+d_2\rangle} = S_i^{(0)} \oplus S_{\langle i-m\rangle}^{(-1)}, i \ne m-1 \tag{11}$$

where $\mathcal{L}_i = b_{i,\ell} \oplus b_{\langle i+d_1\rangle,\ell}$ and $\mathcal{R}_i = b_{i,r} \oplus b_{\langle i-d_2\rangle,r}$. From (10) and (11) we can evaluate all the $\mathcal{L}_i$ and $\mathcal{R}_i$, and hence further retrieve all the missing bits in the ℓth and rth columns.

To simplify the notations, in what follows we let $\rho(x, y) = \langle p - 1 + x \cdot y\rangle$. Then, the specific decoding procedure can be described as follows:

Step 1. For $0 \le i \le p-1$ and $i \ne p-1-m$, let $\alpha_i \leftarrow S_i^{(0)} \oplus S_{\langle i+m\rangle}^{(1)}$. For $0 \le i \le p-1$ and $i \ne m-1$, let $\beta_i \leftarrow S_i^{(0)} \oplus S_{\langle i-m\rangle}^{(-1)}$

Step 2. Let $\xi_i = \mathcal{L}_i \oplus \mathcal{L}_{\langle p-2-d_1-i\rangle}$ and $\eta_i = \mathcal{R}_i \oplus \mathcal{R}_{\langle p-2+d_2-i\rangle}$, then from (10), (11) and Step 1, we have

$$\xi_i \oplus \eta_i = \alpha_i \oplus \beta_{\langle p-2-i\rangle}, i \ne p-1-m \tag{12}$$

$$\xi_{\langle i-d_1\rangle} \oplus \eta_{\langle i+d_2\rangle} = \beta_i \oplus \alpha_{\langle p-2-i\rangle}, i \ne m-1 \tag{13}$$

Observe that $\xi_{\langle (p-2-d_1)/2\rangle} = \eta_{\langle (p-2+d_2)/2\rangle} = 0$, thus every ξ_i and η_i can be evaluated by using (12) and (13) alternately.

Step 3. Let $\mathcal{P}_i = b_{\rho(i,d_1),\ell} \oplus b_{\rho(p-i,d_1),\ell}$ and $\mathcal{Q}_i = b_{\rho(i,d_2),r} \oplus b_{\rho(p-i,d_2),r}$, then we have $\mathcal{P}_i \oplus \mathcal{P}_{i+1} = \mathcal{L}_{\rho(i,d_1)} \oplus \mathcal{L}_{\rho(p-i-1,d_1)} = \xi_{\rho(i,d_1)}$ and $\mathcal{Q}_i \oplus \mathcal{Q}_{i+1} = \mathcal{R}_{\rho(i+1,d_2)} \oplus \mathcal{R}_{\rho(p-i,d_2)} = \eta_{\rho(i+1,d_2)}$. Since $p-1+\frac{p-1}{2}d_1 \equiv \frac{2p-2+pd_1-d_1}{2} \equiv (p-2-d_1)/2 (\mathrm{mod}\, p)$ and $\mathcal{P}_0 = 0$, we have $\mathcal{L}_{\langle (p-2-d_1)/2\rangle} = \mathcal{P}_{(p-1)/2} = \bigoplus_{i=0}^{(p-3)/2} \xi_{\rho(i,d_1)}$. Similarly, $\mathcal{R}_{\langle (p-2+d_2)/2\rangle} = \mathcal{Q}_{(p-1)/2} = \bigoplus_{i=0}^{(p-3)/2} \eta_{\rho(i+1,d_2)}$.

Step 4. Once $\mathcal{L}_{\langle (p-2-d_1)/2\rangle}$ and $\mathcal{R}_{\langle (p-2+d_2)/2\rangle}$ are obtained, we can easily evaluate other $\mathcal{L}_i$ and $\mathcal{R}_i$ using (10) and (11).

Step 5. From $\mathcal{L}_i$ and $\mathcal{R}_i$, $0 \le i \le p-1$, we can easily reconstruct the ℓth and rth columns.

Step 6.Reconstruct the mth column using (7).

The following lemma and theorems can validate the correctness of the above algorithm, where Theorem 1 is required by Step 2 and Step 4, and Theorem 2 is required by Step 5.

Lemma 1. In the sequence of numbers $\{(p-1+k\delta)(\mathrm{mod}\, p), k = 0, 1, \ldots, p\}$, with p being prime and $0 < \delta < p$, the endpoints are both equal to $p-1$, and all numbers $0, 1, \ldots, p-2$ occur exactly once in the sequence [4].

Theorem 1: If both $x_{\langle (p-2-d_1)/2\rangle}$ and $y_{\langle (p-2+d_2)/2\rangle}$ are known, then other x_i and y_i can be obtained from

$$x_i \oplus y_i = a_i, i \ne p-1-m$$
$$x_{\langle i-d_1\rangle} \oplus y_{\langle i+d_2\rangle} = c_i, i \ne m-1$$

where $a_i, c_i \in GF(2), 0 \le i \le p-1, 0 < d_1, d_2, d_1 + d_2 < p$.

Proof: Let $\delta = d_1 + d_2$, then the second equation is equivalent to $x_i \oplus y_{\langle i+\delta\rangle} = c_{\langle i+d_1\rangle}$, where $i \ne \langle m-1-d_1\rangle$. According to Lemma 1, $x_0, x_1, \ldots, x_{p-1}$ and $y_0, y_1, \ldots, y_{p-1}$ occur exactly once in the sequences

$$x_{p-1}, x_{\rho(1,\delta)}, x_{\rho(2,\delta)}, \ldots, x_{\rho(p-1,\delta)} \quad \text{(Seq .1)}$$
$$y_{p-1}, y_{\rho(1,\delta)}, y_{\rho(2,\delta)}, \ldots, y_{\rho(p-1,\delta)} \quad \text{(Seq .2)}$$

Thus, there must be $0 \le u, v \le p-1$ such that $\rho(u, \delta) = p-1-m$ and $\rho(v, \delta) = \langle m-1-d_1\rangle$. Recall that $\rho(p, \delta) = p-1$, so all the elements in Seq. 1 and Seq. 2 can also form the following two sequences

$$x_{\rho(u,\delta)}, y_{\rho(u+1,\delta)}, x_{\rho(u+1,\delta)}, \ldots, y_{\rho(v,\delta)}, x_{\rho(v,\delta)} \quad \text{(Seq .3)}$$
$$y_{\rho(v+1,\delta)}, x_{\rho(v+1,\delta)}, y_{\rho(v+2,\delta)}, \ldots, x_{\rho(u-1,\delta)}, y_{\rho(u,\delta)} \quad \text{(Seq .4)}$$

Clearly, if *any* element of Seq. 3 (Seq. 4) is known, then we can iteratively evaluate all the other elements of the sequence by using $x_i \oplus y_i = a_i$ and $x_i \oplus y_{\langle i+\delta\rangle} = c_{\langle i+d_1\rangle}$ alternately.

Next, we demonstrate that $x_{\langle (p-2-d_1)/2\rangle}$ and $y_{\langle (p-2+d_2)/2\rangle}$ necessarily fall into different sequences. From $\rho(u, \delta) = p-1-m$ and $\rho(v, \delta) = \langle m-1-d_1\rangle$, we have $\rho\left(\frac{u+v}{2}, \delta\right) = \langle (p-2-d_1)/2\rangle$ and $\rho\left(\frac{v+1+u}{2}, \delta\right) = \langle (p-2+d_2)/2\rangle$. If $\langle v-u\rangle$ is even, then $\langle u-v\rangle = p - \langle v-u\rangle$ is odd and hence $\langle u-(v+1)\rangle$ is even. In this case, we have $\rho\left(u + \frac{\langle v-u\rangle}{2}, \delta\right) = \rho\left(\frac{u+v}{2}, \delta\right)$ and $\rho\left(v+1+\frac{\langle u-(v+1)\rangle}{2}, \delta\right) = \rho\left(\frac{v+1+u}{2}, \delta\right)$, i.e., $x_{\langle (p-2-d_1)/2\rangle}$ appears in Seq. 3 while $y_{\langle (p-2+d_2)/2\rangle}$ appears in Seq. 4. If $\langle v-u\rangle$ is odd, then $\langle u-v\rangle = p - \langle v-u\rangle$ and $\langle v-(u+1)\rangle$ are both even. In this case, we have $\rho\left(v + \frac{\langle u-v\rangle}{2}, \delta\right) = \rho\left(\frac{u+v}{2}, \delta\right)$ and $\rho\left(u+1+\frac{\langle v-(u+1)\rangle}{2}, \delta\right) = \rho\left(\frac{v+1+u}{2}, \delta\right)$, i.e., $x_{\langle (p-2-d_1)/2\rangle}$ appears in Seq. 4 while $y_{\langle (p-2+d_2)/2\rangle}$ appears in Seq. 3. ■

Theorem 2: From any $p-1$ of the p equations $x_i \oplus x_{\langle i+\delta\rangle} = a_i (i = 0, 1, \ldots, p-1)$, where $a_i \in GF(2)$ and $0 < \delta < p$, we can obtain $x_0, x_1, \ldots, x_{p-2}$ if x_{p-1} is known.

Proof: According to Lemma 1, $x_0, x_1, \ldots, x_{p-2}$ occur exactly once in the sequence $x_{p-1}, x_{\rho(1,\delta)}, x_{\rho(2,\delta)}, \ldots, x_{\rho(p-1,\delta)}, x_{p-1}$. Moreover, equations $x_i \oplus x_{\langle i+\delta\rangle} = a_i (i = 0, 1, \ldots, p-1)$ are equivalent to $x_{\rho(i,\delta)} \oplus x_{\rho(i+1,\delta)} = a_{\rho(i,\delta)} (i = 0, 1, \ldots, p-1)$. It is quite clear that, if x_{p-1} is known, we can iteratively evaluate $x_{\rho(1,\delta)}, x_{\rho(2,\delta)}, \ldots, x_{\rho(p-1,\delta)}$ from any $p-1$ of the latter p equations. According to Lemma 1, we actually have obtained $x_0, x_1, \ldots, x_{p-2}$. ■

B. Correcting Equidistant Erasures

If $m - \ell = r - m$, or $r - m = \langle \ell - r\rangle$, or $\langle \ell - r\rangle = m - \ell$, then we have an alternative for decoding, which is more efficient than the universal one. Without loss of generality, in what follows we assume that $m - \ell = r - m$. Moreover, we let $d = m - \ell = r - m$ and $\delta = r - \ell = 2d$.

First, we obtain the *latent* diagonal parity bit $b_{p-1,p}$ by $b_{p-1,p} = \bigoplus_{i=0}^{p-2} b_{i,p}$, and compute the corresponding diagonal syndrome $S_{p-1}^{(1)}$ according to (5). Then, (8) and (9) have the following new form:

$$b_{\langle i+d\rangle,\ell} \oplus b_{i,m} \oplus b_{\langle i-d\rangle,r} = S_{\langle i+m\rangle}^{(1)} \tag{14}$$

$$b_{\langle i-d\rangle,\ell} \oplus b_{i,m} \oplus b_{\langle i+d\rangle,r} = S_{\langle i-m\rangle}^{(-1)}, i \ne m-1 \tag{15}$$

where $0 \le i \le p-1$. The difference of (14) and (15) is

$$\mathcal{X}_{\langle i-d\rangle} \oplus \mathcal{X}_{\langle i+d\rangle} = S_{\langle i+m\rangle}^{(1)} \oplus S_{\langle i-m\rangle}^{(-1)} \tag{16}$$

where $0 \le i \le p-1$, $i \ne m-1$, and $\mathcal{X}_i = b_{i,\ell} \oplus b_{i,r}$. Recall that $\delta = 2d$, thus (16) is equivalent to

$$\mathcal{X}_i \oplus \mathcal{X}_{\langle i+\delta\rangle} = S_{\langle i+m+d\rangle}^{(1)} \oplus S_{\langle i-m+d\rangle}^{(-1)} \tag{17}$$

where $0 \le i \le p-1$ and $i \ne m-1-d$. Notice that $\mathcal{X}_{p-1} = b_{p-1,\ell} \oplus b_{p-1,r} = 0$, thus according to Theorem 2, we can iteratively evaluate all the other $\mathcal{X}_i$ using (17).

Once every $\mathcal{X}_i$ is determined, according to (7) we have $b_{i,m} = S_i^{(0)} \oplus \mathcal{X}_i$. Then from (14) we have $b_{\langle i-d\rangle,\ell} \oplus b_{\langle i+d\rangle,r} = S_{\langle i+m\rangle}^{(1)} \oplus b_{i,m}$, i.e., $b_{i,\ell} \oplus b_{\langle i+\delta\rangle,r} = S_{\langle i+m+d\rangle}^{(1)} \oplus b_{\langle i+d\rangle,m}$. From this and $b_{i,\ell} \oplus b_{i,r} = \mathcal{X}_i$ we can

iteratively retrieve all the missing bits in the ℓth and rth columns. Since the procedure is identical to the two-erasure decoding procedure of RDP, we omit the unnecessary details here.

Remark 1: Note that every intermediate result $\mathcal{X}_i$ is used twice in the decoding.

IV. Complexity Analysis

In this section we first analyze the performance of the proposed decoding algorithm, from the perspective of computational complexity measured in terms of the number of XORs required. Then, we compare the new decoding algorithm with the existing ones in terms of decoding complexity. As in Section 3, we only consider the case of three of the first p columns of the array being erased, since for other erasure patterns the existing decoding algorithms are already optimal.

For a given generalized RDP code, the number of data columns k can be smaller than $p-1$ by assuming that the first $p-1-k$ columns hold nothing but zeros. Notice that all the *fictitious* 0-bits *do not* really participate in the computation. Thus, computing row syndromes requires $(k-3)(p-1)$ XORs, and computing diagonal and anti-diagonal syndromes requires at most $2(k-3)(p-1)+6$ XORs if $k \in \{p-1, p-2\}$ and $2(k-3)(p-1)+2(p+1-k)$ XORs if $3 \le k \le p-3$. For non-equidistant erasures, we calculate the number of XORs required for each step as follows:

- Step 1 requires $2(p-2)$ XORs, noting that $S_{p-1}^{(0)}=0$ is a *fictitious* row syndrome
- Step 2 requires $2(p-2)$ XORs, noting that $\xi_i = \xi_{\langle p-2-d_1-i\rangle}$ and $\eta_i = \eta_{\langle p-2+d_2-i\rangle}$, i.e., there are only $(p-1)/2$ unknown ξ_i and $(p-1)/2$ unknown η_i
- Step 3 requires $p-3$ XORs
- Step 4 requires $2(p-1)$ XORs
- Step 5 requires $2(p-3)$ XORs
- Step 6 requires $2(p-1)$ XORs

Therefore, in addition to the syndrome computations, correcting non-equidistant erasures requires $11p-21$ XORs. For equidistant erasures, we give the number of XORs required for each step below:

- obtaining $S_{p-1}^{(1)}$ requires at most $p-2+k-2$ XORs
- obtaining (16) requires $p-1$ XORs
- evaluating every $\mathcal{X}_i$ requires $p-3$ XORs
- evaluating every $b_{i,m}$ requires $p-1$ XORs
- updating (14) requires $p-2$ XORs
- retrieving all the missing bits in the ℓth and rth columns requires $2(p-2)$ XORs

Thus, in addition to the syndrome computations, correcting equidistant erasures requires $7p+k-15$ XORs.

Next, we compare the proposed algorithm with the existing ones in terms of decoding complexity, i.e., the number of XORs required for decoding. In order to better visualize the improvement, we normalize the decoding complexity by dividing it by the lower bound $3(p-1)(k-1)$, and take every possible erasure pattern into account. Specifically, we compute the number of XORs required for each of the $\binom{k+3}{3}$ possible erasure patterns, and take the average value as the decoding complexity of a given code. Figure 3 shows the normalized decoding complexities of different decoding algorithms. Notice that for each k, p is selected to be the first prime number that is greater than k.

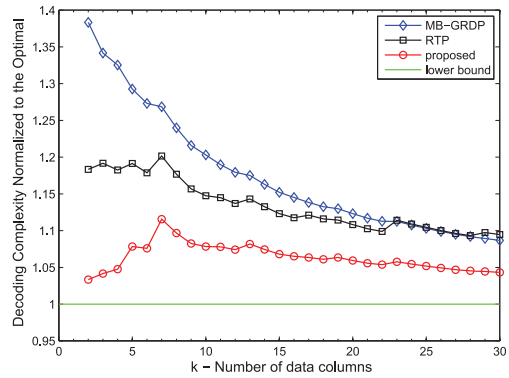

Figure 3 Normalized decoding complexities of different decoding algorithms

It can be seen that, compared with the decoding algorithms presented in MB-GRDP and RTP, our decoding algorithm has a much lower decoding complexity. Moreover, except for $k \in \{7, 8\}$, the decoding complexity of the proposed algorithm is at most 8 percent higher than the lower bound, and asymptotically achieves the lower bound as $k \to \infty$.

V. Conclusion

We have presented an improved decoding algorithm for the generalized RDP codes — the state-of-the-art MDS codes of distance 4. Our detailed complexity analysis shows that the decoding complexity of the proposed algorithm is notably lower than that of the best existing decoding algorithms, and is no more than 8 percent higher than the theoretical lower bound when the code length is neither 10 nor 11. We believe that the proposed decoding algorithm has a potential to be adopted by the industry.

References

[1] M. Blaum, P. G. Farrell, and H. C. van Tilborg, "Chapter on array codes," in *Handbook of Coding Theory*. Amsterdam, The Netherlands: Elsevier, North-Holland, 1998, vol. 2, pp. 1855–1909.

[2] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD'88)*, 1988, pp. 109–116.

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

[4] P. Corbett *et al.*, "Row-diagonal parity for double disk failure correction," in *Proc. 3rd USENIX Conf. File Storage Technol.*, 2004, pp. 1–14.

[5] J. S. Plank, "The RAID-6 liberation codes," in *Proc. 6th USENIX Conf. File Storage Technol.*, 2008, p. 7.

[6] Y. Wang, X. Yin, and X. Wang, "MDR codes: A new class of RAID-6 codes with optimal rebuilding and encoding," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 1008–1018, May 2014.

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

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

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

[10] A. Goel and P. Corbett, "Raid triple parity," *ACM SIGOPS Oper. Syst. Rev.*, vol. 46, pp. 41–49, 2012.