Efficient MDS Array Codes for Correcting Multiple Column Erasures

Zhijie Huang, Hong Jiang, Hao Che
Dept. of Computer Sci. & Engr.
University of Texas at Arlington
TX 76010, USA
{zhijie.huang, hong.jiang, hche}@uta.edu

Nong Xiao
Sch. of Data and Computer Sci.
Sun Yat-Sen University
Guangzhou, 510006, P. R. China
xiaon6@mail.sysu.edu.cn

Ning Li
Dept. of Computer Sci. & Engr.
University of Texas at Arlington
TX 76010, USA
ning.li@uta.edu

***Abstract*—The RΛ-Code is an efficient family of maximum distance separable (MDS) array codes of column distance 4, which involves two types of parity constraints: the row parity and the Λ parity formed by diagonal lines of slopes 1 and -1. Benefitting from the common expressions between the two parity constraints, the encoding and decoding complexities are distinctly lower than most (if not all) of other triple-erasure-correcting codes. It was left as an open problem generalizing the RΛ-Code to arbitrary column distances. In this paper, we present such a generalization, namely, we construct a family of MDS array codes being capable of correcting any prescribed number of erasures/errors by introducing multiple Λ parity constraints. Essentially, the generalized RΛ-Code is derived from a certain variant of the Blaum-Roth codes, and hence retains the error/erasure correcting capability of the latter. Compared with the Blaum-Roth codes, the generalized RΛ-Code has two advantages: a) by exploiting common expressions between row parity and different Λ parity constraints, and reusing the intermidate results during the syndrome calculations, it can encode and decode faster; and b) the memory footprint during encoding/decoding, and the I/O cost caused by degraded reads, are both reduced by 50%.**

I. Introduction

Array codes [1] have been widely employed in storage systems/devices to prevent data loss caused by disk/node failures and latent sector errors in the past decades. In these codes, each codeword is a two-dimensional array, where each column is regarded as a symbol, i.e., a column is considered to be erroneous if at least one of its components has been erased/altered. As with conventional erasure codes (e.g., Reed-Solomon codes[2]), maximum distance separable (MDS) array codes are those array codes that satisfy $n - k + 1 = d$, where n, k, and d represent the code length, the dimension, and the column-wise distance respectively. The main advantage of array codes is that only simple XOR and cyclic shift operations are involved in their encoding and decoding procedures, which makes them much more efficient than the well-known Reed-Solomon codes in terms of computational complexity, and makes them easily implementable in both softwares and hardwares.

Although there are many array codes in the literature, most of them are of distance 3 to 5 [11][12][13][10][8][7], which may be insufficient for certain communication and storage applications. This work is concerned with MDS array codes of large column distance, namely, those codes that are capable of correcting a large number of erasures/errors. There are only several classes of such codes[4][3][5], of which the Blaum-Roth codes [3] may be the most representative in that they can be interpreted as Reed-Solomon codes over certain polynomial ring, and hence are as powerful as Reed-Solomon codes in terms of error/erasure correcting capability. To the best of our knowledge, the Blaum-Roth codes remain the most efficient MDS codes being capable of correcting any prescribed number of errors/erasures.

The Blaum-Roth codes are MDS array codes over $\mathrm{GF}(q)$, in which every codeword is an $(p-1) \times p$ array of elements in $\mathrm{GF}(q)$, where p is a prime number that is not the characteristic of $\mathrm{GF}(q)$. They can be characterized in two ways: geometric presentation and algebraic presentation. Geometrically, the codes with r parity columns are constructed with r types of parity constraints along diagonal lines of slopes $0, 1, 2, \cdots, r-1$ in the $(p-1) \times p$ array. Algebraically, these codes can be interpreted as Reed-Solomon codes over the ring of polynomials over $\mathrm{GF}(q)$ modulo $1 + x + x^2 + \cdots + x^{p-1}$. The authors presented an efficient decoding algorithm (the encoding procedure can be regarded as a special case of the erasure decoding procedure) for error patterns of all erasures and single error combined with multiple erasures. Moreover, when q is primitive in $\mathrm{GF}(p)$, the Blaum-Roth codes become (conventional) Reed-Solomon codes of length p over $\mathrm{GF}(q^{p-1})$, in which case the above-mentioned decoding algorithm can be incorporated into the Berlekamp-Massey algorithm to correct any prescribed number of errors/erasures in a faster way.

In this paper, we present a new family of MDS array codes based on the Blaum-Roth codes, which retains the error/erasure correcting capability of the latter but has lower encoding/decoding complexity. In what follows we refer to the new codes as "the generalized RΛ-Code", since they can be regarded as a generalization of our recent work RΛ-Code [9], though they lose the systematic property of the latter. The RΛ-Code involves two types of parity constraints, i.e., the row parity and the Λ parity formed by diagonal lines of slopes 1 and -1, and can correct up to three column erasures. And just as its name implies, the generalized RΛ-Code introduces multiple Λ parities formed by lines of certain slopes s and $-s$

to provide higher error/erasure correcting capability, where the s's are distinct. Actually, the construction of the generalized RΛ-Code is inspired by the relation between our previous works XI-Code[6] and RΛ-Code[9]. For instance, if r is odd, we first derive a *variant* of the Blaum-Roth codes using parity constraints along lines of slopes $\frac{1-r}{2}, \cdots, -1, 0, 1, \cdots, \frac{r-1}{2}$, and then apply the transformation we used in [9] to the intermediate codes to obtain the generalized RΛ-Code of distance $r+1$. Compared with the Blaum-Roth codes, the generalized RΛ-Code has the following advantages while retaining the same error/erasure correcting capability:

- Common expressions between different parity constraints can be exploited and reused during the syndrome calculation procedure, which results in a lower encoding/decoding complexity;
- The column size is $(p-1)/2$ rather than $(p-1)$, meaning that the memory footprint during encoding/decoding, and the I/O cost caused by degraded reads, are both reduced by 50%.

It is worth mentioning that, there are *no* common expressions between any different parity constraints in the Blaum-Roth codes, since every two different lines have *at most one* point of intersection. Thus, it is impossible to reduce the encoding/decoding complexity of the Blaum-Roth codes to the same extent achieved by the generalized RΛ-Code, by only improving/optimizing the encoding/decoding algorithms proposed in [3] without altering the construction of the codes.

II. Blaum-Roth Codes Review

Since the generalized RΛ-Code is constructed based on a certain variant of the Blaum-Roth codes, to facilitate the presentation of the former, we first briefly review the construction and characteristics of the latter in this section. For simplicity, we only focus on the Blaum-Roth codes over $\mathrm{GF}(2)$, which are most widely applied in real applications.

A. Geometric Presentation

Let p be an odd prime, then there exists an associated Blaum-Roth code whose codewords are $(p-1) \times n$ arrays, where $n \leqslant p$. To facilitate the following description, we use $b_{i,j}$ to denote the i-th bit in the j-th column, where i and j both count from 0. For an integer x, let $\langle x \rangle$ stand for the integer in $\{0, 1, \cdots, p-1\}$ such that $\langle x \rangle \equiv x \pmod{p}$. To simplify notations in the following, we also assume that each array has an extra all-zero row $[b_{p-1,0}, b_{p-1,1}, \cdots, b_{p-1,n-1}]$. Let $\mathcal{M}(p-1, n)$ denote the space of all $(p-1) \times n$ binary matrices (arrays), then the Blaum-Roth code of distance $r+1$ is defined as a subspace of $\mathcal{M}(p-1, n)$, which consists of all arrays that satisfy the following $p \cdot r$ linear constraints:

$$\bigoplus_{j=0}^{n-1} b_{\langle i-js \rangle, j} = 0, 0 \leqslant i \leqslant p-1, 0 \leqslant s \leqslant r-1 \qquad (1)$$

In other words, the bits along each line of slope s ($0 \leqslant s \leqslant r-1$) must sum to zero.

These codes can be regarded as $[n, n-r]$ *non-systematic* MDS codes over the column alphabet of size 2^{p-1}, hence any r columns of the array are uniquely determined by the remaining $n-r$ columns, i.e., any $n-r$ columns may serve as information columns while the remaining r columns serve as parity columns. Usually the locations of the information columns are preset to the first $n-r$ columns, and the parity columns are obtained using the erasure decoding algorithm by assuming that the last r columns are erased. In other words, the encoding procedure is essentially a special case of the erasure decoding procedure.

B. Algebraic Presentation

For any odd prime p, let $M_p(x) = 1 + x + x^2 + \cdots + x^{p-1}$, and let R_p denote the ring of polynomials of degree $< p-1$ over $\mathrm{GF}(2)$ with multiplication taken modulo $M_p(x)$. Let $C(x)$ be a polynomial over $\mathrm{GF}(2)$, and use the notation

$$C(\alpha) = c_0 + c_1\alpha + \cdots + c_{p-2}\alpha^{p-2}$$

to denote a polynomial modulo $M_p(x)$. Correspondingly, $B(\alpha)C(\alpha)$ denotes polynomial multiplication modulo $M_p(x)$, while $B(x)C(x)$ denotes usual polynomial multiplication. It was proven in [3] that

Lemma 1: Elements of the form α^i and $\alpha^i + \alpha^j$ are invertible modulo $M_p(x)$. In other words, if $i \not\equiv j \pmod{p}$ then

$$\gcd(x^i, M_p(x)) = \gcd(x^i + x^j, M_p(x)) = 1.$$

With the above notations, a linear $[n, n-r]$ code over R_p ($r \leqslant n \leqslant p$) can be defined by the following parity-check matrix:

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{r-1} & \alpha^{2(r-1)} & \cdots & \alpha^{(n-1)(r-1)} \end{bmatrix}.$$

Observe that any r columns of H form a Vandermonde matrix over R_p, which is nonsingular according to Lemma 1. In other words, the above defined linear $[n, n-r]$ code is MDS.

III. The Generalized RΛ-Code

In this section, we first give the formal definition of the generalized RΛ-Code, then prove its MDS property by revealing the relation between the generalized RΛ-Code and the Blaum-Roth codes.

A. The Construction

For any odd prime p, there exists a standard generalized RΛ-code whose codewords are $\frac{p-1}{2} \times p$ arrays. Codes of length n ($n \leqslant p$) can be easily obtained from the standard code by assuming that $p-n$ columns of the array are always zero. For simplicity, in what follows we only focus on the standard codes. Let $\mathcal{M}\left(\frac{p-1}{2}, p\right)$ denote the space of all $\frac{p-1}{2} \times p$ binary matrices (arrays), then the generalized RΛ-code of distance $r+1$ is defined as a subspace of $\mathcal{M}\left(\frac{p-1}{2}, p\right)$, which consists

of all $\frac{p-1}{2} \times p$ arrays that satisfy the following $\frac{p-1}{2} \times r$ linear constraints:

$$\bigoplus_{t=0}^{p-1} b_{i,t} = 0, 0 \leqslant i \leqslant \frac{p-3}{2} \quad (2)$$

$$\bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{t,\langle j+\frac{t+1}{s}\rangle}) = 0, 1 \leqslant j \leqslant p-1 \quad (3)$$

where $1 \leqslant s \leqslant (r-1)/2$ when r is an odd number, and $(p+1-r)/2 \leqslant s \leqslant (p-1)/2$ when r is an even number. Note that $\left\langle \frac{1}{s} \right\rangle$ denotes the multiplicative inverse of s in $\mathrm{GF}(p)$, e.g., if $p = 7$, then $\left\langle \frac{1}{2} \right\rangle = 4$ and $\left\langle \frac{1}{3} \right\rangle = 5$. In addition, (2) is required *only when* r is an odd number.

Now let us look into an working example, namely, the generalized RΛ-Code with $p = 7$ and $r = 4$. Figure 1 shows the two groups of Λ parity sets in a codeword, which are labeled with integers and letters respectively. Each Λ parity set consists of all the bits appearing in a certain linear constraint defined above. Let $\Lambda(s, j)$ denote the specific Λ parity set associated with the linear constraint with certain s and j, then from Figure 1 we have $\Lambda(3, 2) = \{b_{0,0}, b_{0,4}, b_{1,5}, b_{1,6}, b_{2,1}, b_{2,3}\}$ and $\Lambda(2, 3) = \{b_{0,0}, b_{0,6}, b_{1,2}, b_{1,4}, b_{2,1}, b_{2,5}\}$.

	0	1	**2**	3	4	5	6
0	**2**5	36	4	15	**2**6	3	14
1	34	45	56	6	1	1**2**	**23**
2	16	**2**	13	**2**4	35	46	5

	0	1	2	**3**	4	5	6
0	**c**d	de	ef	f	a	ab	b**c**
1	af	b	a**c**	bd	**c**e	df	e
2	be	**c**f	d	ae	bf	**c**	ad

Figure 1. The Λ parity sets in the generalized RΛ-Code with $p = 7$ and $r = 4$

B. The MDS Property

Next, we show the MDS property of the generalized RΛ-Code by proving the following theorem.

Theorem 1: The generalized RΛ-code defined above has a column distance of $r + 1$, i.e., it is MDS.

Proof: Observe that both $\langle j - \frac{t+1}{s}\rangle$ and $\langle j + (t+1)/s\rangle$ traverse all the values of $\{0, 1, \cdots, p-1\}$ exactly once as j varies from 0 to $p-1$, thus we have

$$\bigoplus_{j=0}^{p-1} \bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{t,\langle j+\frac{t+1}{s}\rangle}) = 0.$$

From this equation and (3) we can easily get

$$\bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle p-\frac{t+1}{s}\rangle} \oplus b_{t,\langle \frac{t+1}{s}\rangle}) = 0,$$

thus (3) is equivalent to

$$\bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{t,\langle j+\frac{t+1}{s}\rangle}) = 0, 0 \leqslant j \leqslant p-1 \quad (4)$$

Next, extend the $\frac{p-1}{2} \times p$ array to a $p \times p$ array, and let $b_{p-1,j} \leftarrow 0$, $b_{i,j} \leftarrow b_{p-2-i,j}$ for $0 \leqslant j \leqslant p-1$ and $(p-1)/2 \leqslant i \leqslant p-2$. Now let us look into this extended $p \times p$ array to see what linear constraints it satisfies. Since $b_{i,j} = b_{p-2-i,j}$ holds for $0 \leqslant j \leqslant p-1$, from (4) we have

$$\bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{p-2-t,\langle j+\frac{t+1}{s}\rangle}) = 0, \quad (5)$$

$$\bigoplus_{t=0}^{(p-3)/2} (b_{p-2-t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{t,\langle j+\frac{t+1}{s}\rangle}) = 0, \quad (6)$$

where $0 \leqslant j \leqslant p-1$. Let $u = p-2-t$, then

$$\bigoplus_{t=0}^{(p-3)/2} b_{p-2-t,\langle j+\frac{t+1}{s}\rangle} = \bigoplus_{u=(p-1)/2}^{p-2} b_{u,\langle j-(u+1)/s\rangle} = 0,$$

$$\bigoplus_{t=0}^{(p-3)/2} b_{p-2-t,\langle j-\frac{t+1}{s}\rangle} = \bigoplus_{u=(p-1)/2}^{p-2} b_{u,\langle j+(u+1)/s\rangle} = 0,$$

where $0 \leqslant j \leqslant p-1$. Thus, (5) and (6) are equivalent to

$$\bigoplus_{t=0}^{p-2} b_{t,\langle j-\frac{t+1}{s}\rangle} = 0 = b_{p-1,j}, \quad 0 \leqslant j \leqslant p-1 \quad (7)$$

$$\bigoplus_{t=0}^{p-2} b_{t,\langle j+\frac{t+1}{s}\rangle} = 0 = b_{p-1,j}, \quad 0 \leqslant j \leqslant p-1 \quad (8)$$

Observe that both $\langle j - \frac{t+1}{s}\rangle$ and $\langle j + (t+1)/s\rangle$ traverse all the values of $\{0, 1, \cdots, p-1\}$ exactly once as t varies from 0 to $p-1$, thus (7) and (8) are equivalent to

$$\bigoplus_{u=0}^{p-1} b_{\langle sj-1-su\rangle,u} = 0, \quad 0 \leqslant j \leqslant p-1 \quad (9)$$

$$\bigoplus_{u=0}^{p-1} b_{\langle p-1-sj+su\rangle,u} = 0, \quad 0 \leqslant j \leqslant p-1 \quad (10)$$

Similarly, since both $\langle sj-1\rangle$ and $\langle p-1-sj\rangle$ traverse all the values of $\{0, 1, \cdots, p-1\}$ exactly once as j varies from 0 to $p-1$, (9) and (10) are further equivalent to

$$\bigoplus_{u=0}^{p-1} b_{\langle i-su\rangle,u} = 0, \quad 0 \leqslant i \leqslant p-1 \quad (11)$$

$$\bigoplus_{u=0}^{p-1} b_{\langle i+su\rangle,u} = 0, \quad 0 \leqslant i \leqslant p-1 \quad (12)$$

Then we distinguish between the following two cases:

(a) *r is an odd number*

In this case, from $b_{i,j} = b_{p-2-i,j}$, $b_{p-1,j} = 0$ and (2)

$$\bigoplus_{t=0}^{p-1} b_{i,t} = 0, \quad 0 \leqslant i \leqslant p-1 \quad (13)$$

Recall that $1 \leqslant s \leqslant (r-1)/2$, thus (11), (12) and (13) can be combined into

$$\bigoplus_{u=0}^{p-1} b_{\langle i-su\rangle,u} = 0, \quad 0 \leqslant i \leqslant p-1, \frac{1-r}{2} \leqslant s \leqslant \frac{r-1}{2} \quad (14)$$

Geometrically, (14) indicates that all the bits along each of the p lines of slope s sum to zero, where $\frac{1-r}{2} \leqslant s \leqslant \frac{r-1}{2}$. In other words, (14) defines a variant of the Blaum-Roth codes, in which the parity sets are placed along lines of

slopes $\frac{1-r}{2}, \cdots, -1, 0, 1, \cdots, \frac{r-1}{2}$ rather than $0, 1, \cdots, r-1$. As discussed in Section II, each column can be regarded algebraically as an element of the polynomial ring R_p. For example, the column j can be represented as $C_j(\alpha) = b_{0,j} + b_{1,j}\alpha + \cdots + b_{p-2,j}\alpha^{p-2}$, where $b_{i,j}$ $(0 \leqslant i \leqslant p-2)$ is the i-th bit in the column. For the ring element α, it was shown in [3] that its multiplicative order is p, and it has multiplicative inverse, which can be denoted as α^{-1}. Denote the extended array as $A = (C_0(\alpha), C_1(\alpha), \cdots, C_{p-1}(\alpha))$, and let

$$H_{\text{odd}} = \begin{bmatrix} 1 & \alpha^{(1-r)/2} & \alpha^{1-r} & \cdots & \alpha^{(p-1)(1-r)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(p-1)} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^{2} & \cdots & \alpha^{p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{(r-1)/2} & \alpha^{r-1} & \cdots & \alpha^{(p-1)(r-1)} \end{bmatrix},$$

then from (14) we can deduce that

$$A \times H_{\text{odd}}^T = \mathbf{0} \tag{15}$$

Observe that any r columns of H_{odd} form a Vandermonde matrix over R_p, which is nonsingular according to Lemma 1, thus any r erased columns can be recovered using (15).

(b) *r is an even number*

In this case, there are not row parity constraints. Recall that $(p+1-r)/2 \leqslant s \leqslant (p-1)/2$, thus (11) and (12) can be combined into

$$\bigoplus_{u=0}^{p-1} b_{\langle i-su\rangle,u} = 0, 0 \leqslant i \leqslant p-1, \frac{p+1-r}{2} \leqslant s \leqslant \frac{p-1+r}{2} \tag{16}$$

Similarly, (16) defines a variant of the Blaum-Roth codes, in which the parity sets are placed along lines of slopes $\frac{p+1-r}{2}, \cdots, \frac{p-1}{2}, \frac{p+1}{2}, \cdots, \frac{p-1+r}{2}$ rather than $0, 1, \cdots, r-1$. Use the same algebraic notations as the last case, and let

$$H_{\text{even}} = \begin{bmatrix} 1 & \alpha^{(p+1-r)/2} & \alpha^{p+1-r} & \cdots & \alpha^{(p-1)(p+1-r)/2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{(p-1)/2} & \alpha^{p-1} & \cdots & \alpha^{(p-1)(p-1)/2} \\ 1 & \alpha^{(p+1)/2} & \alpha^{p+1} & \cdots & \alpha^{(p-1)(p+1)/2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{(p-1+r)/2} & \alpha^{p-1+r} & \cdots & \alpha^{(p-1)(p-1+r)/2} \end{bmatrix}$$

then from (16) we can deduce that

$$A \times H_{\text{even}}^T = \mathbf{0} \tag{17}$$

Again, since any r columns of H_{even} form a Vandermonde matrix over R_p, which is nonsingular according to Lemma 1, any r erased columns can be recovered using (17).

From the above, the generalized RΛ-code can correct any r column erasures, i.e., it is MDS. □

Note that when r is even, we can deduce (17) from (16), but not the opposite. For further details regarding the relation between the geometric presentation and the algebraic presentation of the Blaum-Roth codes, please refer to [3].

IV. Decoding/Encoding

As with the Blaum-Roth codes, the generalized RΛ-Code is also non-systematic, meaning that any r columns can serve as parity columns, while the remaining $p-r$ columns can be regarded as information columns. Thus, the encoding procedure is essentially a special case of the erasure decoding procedure.

As mentioned in the last section, the generalized RΛ-Code can be transformed into certain variants of the Blaum-Roth codes, which may simplify our discussion substantially. For a codeword with r columns erased, we first set the erased columns to be zeros, then calculate the row (if r is odd) and Λ syndromes as follows:

$$S_i^0 = \bigoplus_{t=0}^{p-1} b_{i,t} \qquad 0 \leqslant i \leqslant (p-3)/2 \tag{18}$$

$$S_j^{\Lambda(s)} = \bigoplus_{t=0}^{(p-3)/2} (b_{t,\langle j-\frac{t+1}{s}\rangle} \oplus b_{t,\langle j+\frac{t+1}{s}\rangle})$$
$$0 \leqslant j \leqslant p-1, s = \begin{cases} 1, 2, \cdots, \frac{r-1}{2} (2 \nmid r) \\ \frac{p+1-r}{2}, \cdots, \frac{p-1}{2} (2|r) \end{cases} \tag{19}$$

Note that all the 0-bits (including the erased bits) do not really participate in the calculations. Now consider the extended $p \times p$ array defineded in the last subsection, where the r erased columns are also set to be zeros. Let S_i^s denote the syndrome associated with the i-th linear constraint of slope s, i.e.,

$$S_i^s = \bigoplus_{u=0}^{p-1} b_{\langle i-su\rangle,u}, \qquad 0 \leqslant i \leqslant p-1$$

Then, according to the symmetry of the extended array, these syndromes can be obtained by $S_{j-1}^s \leftarrow S_j^{\Lambda(s)}$ and $S_{p-1-j}^{-s} \leftarrow S_j^{\Lambda(s)}$ for $0 \leqslant j \leqslant p-1$, where $\leftarrow$ is an assignment operator. If r is odd, then let $S_i^0 \leftarrow S_{p-2-i}^0$ for $(p-1)/2 \leqslant i \leqslant p-2$. After obtaining all the syndromes associated with the extended array, the r erased columns can be recovered using the decoding algorithm described in [3].

From the above, the key to decode the generalized RΛ-Code lies in the calculation of the syndromes. As in [9], this procedure can be optimized by exploiting certain common expressions and reusing the intermediate results. For example, from Figure 1 we can find that the intersection of $\Lambda(3,2)$ and $\Lambda(2,3)$ contains two elements, i.e., $b_{0,0}$ and $b_{2,1}$, thus the value of $b_{0,0} \oplus b_{2,1}$ can be reused in calculating $S_2^{\Lambda(3)}$ and $S_3^{\Lambda(2)}$. To facilitate the following discussion, we give a useful proposition:

Proposition 1: The intersection of parity sets $\Lambda(s_1, j_1)$ and $\Lambda(s_2, j_2)$ contains exactly two elements, iff $s_1 \neq s_2$ and $j_1 \neq j_2$.

Therefore, there are totally 42 common expressions in this example, i.e., $b_{0,j} \oplus b_{2,\langle j+1\rangle}$, $b_{0,j} \oplus b_{2,\langle j-1\rangle}$, $b_{0,j} \oplus b_{1,\langle j+2\rangle}$, $b_{0,j} \oplus b_{1,\langle j-2\rangle}$, $b_{1,j} \oplus b_{2,\langle j+3\rangle}$, and $b_{1,j} \oplus b_{2,\langle j-3\rangle}$, where $0 \leqslant j \leqslant 6$. However, only 14 of them can be used simultaneously, since the common expressions in the same Λ parity constraint should not intersect with each other (otherwise the common bit will be canceled while calculating the corresponding Λ

syndrome), e.g., $b_{0,0} \oplus b_{2,1}$ and $b_{0,0} \oplus b_{1,5}$ should not be used simultaneously.

In what follows, we give an algorithm for calculating two groups of Λ syndromes cooperatively, in which $e(j_1, j_2)$ denotes the common expression determined by the Λ parity sets $\Lambda(s_1, j_1)$ and $\Lambda(s_2, j_2)$.

Algorithm 1 (Λ *Syndrome Calculation Algorithm*):

1. For $0 \leqslant j_1 \leqslant p-1$
2. For $0 \leqslant j_2 \leqslant p-1$ and $j_2 \neq j_1$, add $e(j_1, j_2)$ to set $E(j_1)$
3. Let $m \leftarrow (p-1)/2$.
4. Pick m common expressions from $E(j_1)$, store them in $E'(j_1)$, such that every two common expressions in $E'(j_1)$ have no common bit.
5. If Step 4 failed, let $m \leftarrow m-1$ and goto Step 4.
6. For each $e(j_1, j_2) \in E'(j_1)$, evaluate it, and add the result to both $S_{j_1}^{\Lambda(s_1)}$ and $S_{j_2}^{\Lambda(s_2)}$.
7. Add the bits in $\Lambda(s_1, j_1)$ that are not involved in any common expression in $E'(j_1)$ to $S_{j_1}^{\Lambda(s_1)}$
8. For $0 \leqslant j_2 \leqslant p-1$
9. Add the bits in $\Lambda(s_2, j_2)$ that are not involved in any common expression in $E'(j_1)$ to $S_{j_2}^{\Lambda(s_2)}$

We did a lot of experiments and found that $m = (p-1)/2 - \varepsilon$, where ε is a very small number compared to p and in most cases $\varepsilon \in \{0, 1\}$. In other words, each syndrome has $(p-1)/2 - \varepsilon$ common expressions with the other group of Λ syndromes. Therefore, Algorithm 1 saves about 25 percent of computational overhead.

According to the specific number of parity columns, we can calculate the syndromes by using the above algorithm and/or the algorithm described in [9] to fully utilize the common expressions. For example, if $r = 7$, we can calculate S^0 and $S^{\Lambda(1)}$ using the encoding algorithm in [9], and calculate $S^{\Lambda(2)}$ and $S^{\Lambda(3)}$ using the above algorithm. Recall that the algorithm in [9] can save about 16.7 percent of computational overhead, thus about $\frac{16.7\times3+25\times4}{7} = 21.4$ percent of the overall syndrome computation overhead can be saved. To summarize, Table I shows how much computational overhead can be saved for different values of r. It is clear that the average savings of the computational overhead during the syndrome calculation procedure is around 16.7 to 25 percent as r varies from 3 to ∞.

Table I
COMPUTATIONAL OVERHEAD SAVINGS FOR DIFFERENT VALUES OF r

		Percentage Saved	Comment
$2 \mid r$	$4 \mid r$	25	the ideal case
	$4 \nmid r$	$\frac{25(r-2)}{r}$	one group of Λ syndromes are calculated alone
$2 \nmid r$	$4 \mid (r-1)$	$\frac{25(r-1)}{r}$	the row syndromes are calculated alone
	$4 \nmid (r-1)$	$\frac{16.7\times3+25(r-3)}{r}$	Algorithm 1 and the algorithm in [9] are both involved in this case

V. CONCLUSION

We presented a new family of MDS array codes, called the generalized RΛ-Code, which is essentially derived from a certain variant of the Blaum-Roth codes, and hence can also correct any prescribed number of column erasures/errors. The main novelty of the proposed codes is that the parity constraints are formed along polygonal lines rather than diagonal lines, making it possible to reuse the values of certain common expressions between different parity constraints during the encoding/decoding procedure. This feature makes the generalized RΛ-Code outperform the Blaum-Roth codes and most (if not all) of other array codes with the same erasure/error correcting capability in terms of encoding/decoding complexity. In addition, due to the smaller column size of the generalized RΛ-Code, the memory footprint during encoding/decoding, and the I/O cost caused by degraded reads, are both reduced by 50%. Therefore, we believe the new codes will be very competitive in certain communication and storage applications.

ACKNOWLEDGMENT

This research is partially supported by a research grant from NetApp, grants from the U.S. National Science Foundation (NSF) under Grant Nos. CCF-1704504 and CCF-1629625, the National Key R&D Program of China under Grant No. 2017YFB1001600, and the National Science Foundation of China (NSFC) under Grant Nos. 61832020, 61702569, and 61872392. We would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] Mario Blaum, Patrick G Farrell, van Tilborg, and C A Henk. Chapter on array codes. *Handbook of Coding Theory*, 2:18551909, 1998.
[2] I. S. Reed and G. Solomon, Polynomial Codes over Certain Finite Fields, *J. SIAM*, 8(10), 300-304, 1960.
[3] M. Blaum and R. M. Roth, New array codes for multiple phased burst correction, *IEEE Trans. Inform. Theory*, vol. 39, pp. 66-77, 1993
[4] M. Blaum, J. Bruck, and A. Vardy, MDS array codes with independent parity symbols, *IEEE Trans. Information Theory*, vol. 42, no. 2, pp. 529542, Mar. 1996.
[5] G. Feng, R. Deng, F. Bao, and J. Shen. New efficient MDS array codes for RAID Part II: Rabin-like codes for tolerating multiple (4) disk failures. *IEEE Transactions on Computers*, 54(12):14731483, 2005.
[6] Z. Huang, H. Jiang, K. Zhou, C. Wang, and Y. Zhao. XI-Code: A family of practical lowest density MDS array codes of distance 4. *IEEE Transactions on Communications*, 64(7):27072718, July 2016.
[7] Zhijie Huang, Hong Jiang, Ke Zhou, Yuhong Zhao, and Chong Wang. Lowest density MDS array codes of distance 3. *IEEE Communications Letters*, 19(10):16701673, Oct 2015.
[8] Chao Jin, Hong Jiang, Dan Feng, and Lei Tian. P-Code: A new RAID-6 code with optimal properties. In the 23rd *ACM International Conference on Supercomputing (ICS09)*, pages 360369. ACM, 2009.
[9] Zhijie Huang, Hong Jiang, and Nong Xiao. Efficient lowest density MDS array codes of column distance 4. *2017 IEEE International Symposium on Information Theory (ISIT)*, Pages: 834 - 838. IEEE, 2017.
[10] Y. Cassuto and J. Bruck. Cyclic lowest density MDS array codes. *IEEE Transactions on Information Theory*, 55(4):17211729, 2009.
[11] M. Blaum, J. Brady, J. Bmck, and J. Menon, EVENODD: An optimal scheme for tolerating double disk failures in RAID architectures, *IEEE Trans. Comput.*, vol. 44, pp. 192-202, 1995.
[12] P. Corbett, B. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong and S. Sankar, Row-Diagonal Parity for Double Disk Failure Correction, *Proc. of USENIX FAST 2004*, Mar. 31 to Apr. 2, San Francisco,CA, USA.
[13] L. Xu and J. Bruck, X-Code: MDS Array Codes with Optimal Encoding, *IEEE Trans. on Information Theory*, 45(1), 272-276, Jan., 1999.