MDS Array Codes With (Near) Optimal Repair Bandwidth for All Admissible Repair Degree

Jie Li, Member, IEEE, Yi Liu, Member, IEEE,

Xiaohu Tang, Senior Member, IEEE, Yunghsiang S. Han, Fellow, IEEE,

Bo Bai, Senior Member, IEEE, and Gong Zhang

Abstract

Abundant high-rate (n, k) minimum storage regenerating (MSR) codes have been reported in the literature. However, most of them require contacting all the surviving nodes during a node repair process, resulting in a repair degree of d = n - 1. In practical systems, it may not always be feasible to connect and download data from all surviving nodes, as some nodes may be unavailable. Therefore, there is a need for MSR code constructions with a repair degree of d < n - 1. Up to now, only a few (n, k) MSR code constructions with repair degree d < n - 1 have been reported, some have a large sub-packetization level, a large finite field, or restrictions on the repair degree d. In this paper, we propose

J. Li, B. Bai, and G. Zhang are with the Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Shatin, New Territories, Hong Kong SAR, China (e-mails: li.jie9@huawei.com; baibo8@huawei.com; nicholas.zhang@huawei.com).

Y. Liu was with the Information Coding and Transmission Key Lab of Sichuan Province, CSNMT Int. Coop. Res. Centre (MoST), Southwest Jiaotong University, Chengdu, 610031, China (e-mail: yiliu.swjtu@outlook.com).

X. Tang is with the Information Coding and Transmission Key Lab of Sichuan Province, CSNMT Int. Coop. Res. Centre (MoST), Southwest Jiaotong University, Chengdu, 610031, China (e-mail: xhutang@swjtu.edu.cn).

Yunghsiang S. Han is with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, 518110, China (e-mail: yunghsiangh@gmail.com).

a new (n, k) MSR code construction that works for any repair degree d > k, and has a smaller subpacketization level or finite field than some existing constructions. Additionally, in conjunction with a previous generic transformation to reduce the sub-packetization level, we obtain an MDS array code with a small sub-packetization level and $(1 + \epsilon)$ -optimal repair bandwidth (i.e., $(1 + \epsilon)$ times the optimal repair bandwidth) for repair degree d = n - 1. This code outperforms some existing ones in terms of either the sub-packetization level or the field size.

Index Terms

Maximum distance separable, minimum storage regenerating codes, repair bandwidth, repair degree, sub-packetization.

I. INTRODUCTION

In distributed storage systems, data are stored across multiple unreliable storage nodes. Thus, redundancy needs to be introduced to provide fault tolerance. Classic Maximum Distance Separable (MDS) codes can provide an optimal tradeoff between fault tolerance and storage overhead and thus is an efficient redundancy mechanism deployed for many years. However, repairing a failed node requires an excessive *repair bandwidth*, defined as the amount of data downloaded to repair a failed node.

One way to reduce the repair bandwidth is to use MDS array codes, where the codeword is an array of size $N \times n$ instead of a vector. For a distributed storage system encoded by an (n,k) MDS array code, each node stores N symbols, where N is called the *sub-packetization level*. The cut-set bound in [1] shows that the repair bandwidth of (n,k) MDS array codes with sub-packetization level N is lower bounded by $\gamma_{\text{optimal}} = \frac{d}{d-k+1}N$. Here, d such that $k \leq d < n$ denotes the number of helper nodes contacted during the repair process and is named *repair degree*. MDS array codes with repair bandwidth attaining this lower bound are said to have the *optimal repair bandwidth* and are also referred to as MSR codes in [1].

2

During the past decade, various MSR codes have been proposed [2]–[24]. However, in the highrate (e.g., $\frac{k}{n} > \frac{1}{2}$) regime, existing constructions have two imperfections: i) most constructions have a repair degree of d = n - 1, meaning that repairing a failed node requires contacting all the remaining surviving nodes. However, it is not always feasible to connect and download data from all the surviving nodes in a practical system, as some nodes may be unavailable due to other assigned jobs or network congestion [25]; ii) all the known (n, k) MSR code constructions with repair degree d = n - 1 require a significantly large sub-packetization level N, i.e., $N \ge r^{\frac{n}{r+1}}$, where r = n - k. This can lead to reduced design space in various system parameters and make managing meta-data difficult, hindering implementation in practical systems [26].

A. Related work on (n,k) MSR codes with repair degree d < n-1

Up to now, only a few results on MSR codes with repair degree d < n-1 have been reported in the literature. In [27], [28], the authors showed the existence of MSR codes with repair degree d < n-1, and some explicit constructions were given in [18]–[22], [24]. In this paper, we focus only on explicit constructions.

In Sections IV and VIII of [18], Ye and Barg proposed two (n, k) MSR codes with subpacketization level $(d - k + 1)^n$ by using diagonal matrices and permutation matrices as the building blocks of the parity-check matrices. In [19], an (n, k) MSR code with a smaller subpacketization level of $(d - k + 1)^{\frac{n}{d-k+1}}$ was generated, however, d is restricted to be k + 1, k + 2, k + 3. In [20], Chen and Barg presented an (n, k) MSR codes with a sub-packetization level of $(d - k + 1)^n$. In [21], Liu *et al.* gave an (n, k) MSR code with a sub-packetization level of $(d - k + 1)^{\frac{n}{2}}$. Recently, in [22], an (n, k) MSR code was constructed with a sub-packetization level of $2^{\frac{n}{3}}$ and a repair degree of d = k + 1. This MSR code was generalized to support any repair degree d with $d \in [k + 1 : n - 1)$ and a sub-packetization level of $w^{\frac{n}{w+1}}$ in a follow-up work [23], where w = d - k + 1, but requires searching over a finite field with a size larger than $nw + \sum_{t=1}^{w+1} {wt(t-1) \choose t}$. Despite the additional effort required in searching (i.e., explicit constructions are unknown for general code parameters n, k, and d), the MSR codes in [22] and [23] have the smallest sub-packetization level among all existing MSR codes with the same n, k, d. Independent and parallel to this work, Zhang and Zhou proposed an (n, k) MSR code with a sub-packetization level of $2^{\frac{n}{2}}$ in a very recent work [32], which is similar to the one proposed in this paper but requires a larger finite field when d - k + 1 > 2. For convenience, in this paper, these eight codes are referred to as YB code 1, YB code 2, VBK code, CB code, LLT code, WLHY code, LWHY code, and ZZ code, respectively.

Overall, most of the aforementioned (n, k) MSR codes with repair degree d < n - 1 either have a large sub-packetization level (i.e., $N = (d - k + 1)^n$) [18], [20] or are limited to only a few values of the repair degree d [19], [22]. We want to point out that there are a few MSR codes with multiple repair degrees, e.g., [24] and MSR codes in Sections V and IX of [18], which are outside of the scope of this paper as we only focus on MSR codes with a single repair degree.

B. Related work on (n, k) MDS array codes with small sub-packetization level

Large sub-packetization levels in codes can hinder their implementation in practical systems [26], making it desirable to construct codes with small sub-packetization levels. Recent works have demonstrated that high-rate MDS array codes with small sub-packetization levels can be constructed by sacrificing the optimality of the repair bandwidth.

In [26], two high-rate MDS array codes with small sub-packetization levels and $(1+\epsilon)$ -optimal repair bandwidth were proposed. The first code has a sub-packetization level of $N = r^{\tau}$ and the repair bandwidth is no larger than $(1 + \frac{1}{\tau})$ times the optimal repair bandwidth, where τ is an integer and $1 \le \tau < \lceil \frac{n}{r} \rceil$. However, this code is constructed over a significantly large finite field \mathbf{F}_q , i.e., $q \ge n^{(r-1)N+1}$, which may hinder its deployment in practical systems. The second MDS array code is obtained by combining an MDS array code with optimal repair bandwidth and another error-correcting code with specific parameters. For convenience, we refer to these two codes as RTGE code 1 and RTGE code 2 in this paper.

Recently, a generic transformation was presented in [29] that can convert any MSR code into an MDS array code with a small sub-packetization level and $(1 + \epsilon)$ -optimal repair bandwidth, resulting in several explicit MDS array codes with small sub-packetization levels. Note that all the MDS array codes in [26], [29] have a repair degree of d = n - 1.

C. Main Contribution

The main contribution of this paper is the derivation of a new (n, k) MSR code construction with any given repair degree d such that $k < d \le n - 1$, where its sub-packetization level is $w^{\lceil \frac{n}{2} \rceil}$ with w = d - k + 1. The required field size q is a prime power such that $q > \lceil \frac{n}{2} \rceil (w + 2)$ if w = 2, $q > \lceil \frac{n}{2} \rceil (w + 1)$ if 2 < w < r, and $q > \lceil \frac{n}{2} \rceil w$ if w = r. Compared with existing constructions, the new MSR code C_1 has advantages in terms of either the sub-packetization level or the field size. Please refer to Tables II and III for more details.

Furthermore, the MSR code C_1 can also be used for d = n - 1. When combined with the generic transformation in [29], we obtain a new (n, k) MDS array code C_2 with a small sub-packetization level of $r^{\lceil \frac{n}{2s} \rceil}$ and repair degree d = n - 1, where r = n - k, s is any factor of n, and the require field size is $q > sr \lceil \frac{n}{2s} \rceil$. The sub-packetization level or finite field size of C_2 is smaller than that of existing ones.

The remainder of the paper is organized as follows. Section II reviews some necessary preliminaries of high-rate MDS array codes. The new (n, k) MSR code C_1 is presented in Section III. Section IV gives an MDS array code C_2 with a small sub-packetization level. Section V compares key parameters among the MDS array codes proposed in this paper and some existing ones. Finally, Section VI concludes the work.

II. PRELIMINARIES

In this section, we introduce some preliminaries on MDS array codes and a special partition for a given basis. Throughout this paper, we assume that q is a prime power and \mathbf{F}_q is the finite field with q elements. Let [a:b) be the set $\{a, a+1, \ldots, b-1\}$ for two integers a and b. For a matrix A, denote by A[a,b], the (a,b)-th entry and A[a,:] the a-th row, where $a, b \ge 0$.

A. (n, k) Array Codes

Let $\mathbf{f}_0, \mathbf{f}_1, \dots, \mathbf{f}_{n-1}$ be the data stored across a distributed storage system consisting of n nodes based on an (n, k) array code, where \mathbf{f}_i is a column vector of length N over \mathbf{F}_q . We consider (n, k) array codes defined by the following parity-check form:

$$\underbrace{\begin{pmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{r-1,0} & A_{r-1,1} & \cdots & A_{r-1,n-1} \end{pmatrix}}_{A} \begin{pmatrix} \mathbf{f}_{0} \\ \mathbf{f}_{1} \\ \vdots \\ \mathbf{f}_{n-1} \end{pmatrix} = \mathbf{0}_{rN}, \tag{1}$$

where r = n - k, $\mathbf{0}_{rN}$ denotes the zero column vector of length rN, and will be abbreviated as $\mathbf{0}$ in the sequel if its length is clear. The $rN \times nN$ block matrix A in (1) is called the *parity-check matrix* of the code, which can be written as $A = (A_{t,i})_{t \in [0:r), i \in [0:n)}$, to indicate the block entries, where $A_{t,i}$ is an $N \times N$ matrix.

Note that for each $t \in [0:r)$, $\sum_{i=0}^{n-1} A_{t,i} \mathbf{f}_i = 0$ contains N equations. For convenience, we say $\sum_{i=0}^{n-1} A_{t,i} \mathbf{f}_i = 0$ the t-th parity-check group.

B. The MDS property

An (n, k) array code defined by (1) is MDS if the source file can be reconstructed by connecting any k out of the n nodes. That is, any $r \times r$ sub-block matrix $(A_{t,i})_{t \in [0:r), i \in J}$ of the block matrix $(A_{t,i})_{t \in [0:r), i \in [0:n)}$ is non-singular [18], where J is any r-subset of [0:n). In the following, we introduce some lemmas that will be helpful when verifying the MDS property of the new codes in the later sections. **Lemma 1.** For $t, i \in [0:r)$, let $B_{t,i}$ be an $N \times N$ upper triangular matrix, i.e.,

$$B_{t,i}[a,b] = 0 \text{ for } 0 \le b < a < N,$$
(2)

then the block matrix $B = (B_{t,i})_{t \in [0:r), i \in [0:r)}$ is non-singular if

i)
$$B_{t,i}[a,a] = (B_{1,i}[a,a])^t$$
 for $i, t \in [0:r)$ and $a \in [0:N)$,

ii) $B_{1,i}[a, a] \neq B_{1,j}[a, a]$ for any $i, j \in [0:r)$ with $j \neq i$ and $a \in [0:N)$.

Proof: The proof is given in Appendix A.

C. Repair Mechanism

For an (n, k) array code, suppose that node i $(i \in [0 : n))$ fails. Let H_i be any given d-subset of $[0 : n) \setminus \{i\}$, which denotes the set of indices of the helper nodes, and let $L_i = [0 : n) \setminus (H_i \cup \{i\})$ be the set of indices of unconnected nodes. The data downloaded from helper node j can be represented by $R_{i,j}\mathbf{f}_j$, where $R_{i,j}$ is a $\beta_{i,j} \times N$ matrix of full rank with $\beta_{i,j} \leq N$. We refer to $R_{i,j}$ as the *repair matrix* of node i.

Note that the content of node *i* can be acquired from the parity-check equations. In this paper, similar to [29], for convenience, we only consider the symmetric situation where $\delta (N/r \le \delta \le N)$ linearly independent equations are acquired from each of the *r* parity-check groups, where these δ linear independent equations are linear combinations of the corresponding *N* parity-check equations in a parity-check group. Precisely, the δ linear independent equations from the *t*-th parity-check group can be obtained by multiplying it with a $\delta \times N$ matrix $S_{i,t}$ of full rank, where $S_{i,t}$ is called the *select matrix*. As a consequence, the following linear equations are available:

$$\begin{pmatrix}
S_{i,0}A_{0,i} \\
S_{i,1}A_{1,i} \\
\vdots \\
S_{i,r-1}A_{r-1,i}
\end{pmatrix} \mathbf{f}_{i} + \sum_{l \in L_{i}} \begin{pmatrix}
S_{i,0}A_{0,l} \\
S_{i,1}A_{1,l} \\
\vdots \\
S_{i,r-1}A_{r-1,l}
\end{pmatrix} \mathbf{f}_{l} + \sum_{j \in H_{i}} \begin{pmatrix}
S_{i,0}A_{0,j} \\
S_{i,1}A_{1,j} \\
\vdots \\
S_{i,r-1}A_{r-1,j}
\end{pmatrix} \mathbf{f}_{j} = 0. \quad (3)$$
useful data
$$\underbrace{ \begin{array}{c}
S_{i,0}A_{0,j} \\
S_{i,1}A_{1,j} \\
\vdots \\
S_{i,r-1}A_{r-1,j}
\end{bmatrix} \mathbf{f}_{j} = 0. \quad (3)$$

May 30, 2023

8

The repair of node *i* requires solving (3) from the downloaded data $R_{i,j}\mathbf{f}_j$, $j \in H_i$. Then the repair bandwidth of node *i* is $\gamma_i = \sum_{j \in H_i} \operatorname{rank}(R_{i,j})$. If $\gamma_i = \frac{d}{d-k+1}N$, then node *i* is said to have the *optimal repair bandwidth*, which can be accomplished if $\operatorname{rank}(R_{i,j}) = \frac{N}{d-k+1}$ for all $j \in H_i$. If the repair bandwidth of an MDS array code is $(1 + \epsilon)\frac{N}{d-k+1}$ where $\epsilon < 1$ is a small constant, the MDS array code is said to have $(1 + \epsilon)$ -optimal repair bandwidth in [31].

D. Partition of basis $\{e_0, \ldots, e_{N-1}\}$

Assuming that $N = w^m$ for two integers w and m with $w, m \ge 2$, let e_0, \ldots, e_{w^m-1} be a basis of $\mathbf{F}_q^{w^m}$. For simplicity, one can regard them as the standard basis, i.e.,

$$e_i = (0, \dots, 0, 1, 0, \dots, 0), \ i \in [0: w^m),$$

with only the *i*-th entry being nonzero.

Then for any $a, b \in [0:N)$, we have

$$e_a(e_b)^{\top} = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{otherwise,} \end{cases}$$
 (4)

where \top represents the transpose operator.

In [29], a class of special partition sets of $\{e_0, \ldots, e_{w^m-1}\}$ is given for $w \ge 2$. As these special partition sets play an important role in our proposed new construction, we revisit them for completeness in the following.

Given an integer $a \in [0 : w^m)$, denote by (a_0, \ldots, a_{m-1}) its w-ary expansion with a_0 being the most significant digit, i.e., $a = \sum_{j=0}^{m-1} w^{m-1-j}a_j$. For convenience, we also write $a = (a_0, \ldots, a_{m-1})$. For $i \in [0 : m)$ and $t \in [0 : w)$, define a subset of $\{e_0, \ldots, e_{w^m-1}\}$ as

$$V_{i,t} = \{e_a | a_i = t, 0 \le a < w^m\},\tag{5}$$

where a_i is the *i*-th element in the *w*-ary expansion of *a*.

Obviously, $|V_{i,t}| = w^{m-1}$, and $V_{i,0}, V_{i,1}, \dots, V_{i,w-1}$ is a partition of the set $\{e_0, \dots, e_{w^m-1}\}$ for any $i \in [0:m)$. Table I gives two examples of the set partitions defined in (5).

TABLE I

(a) and (b) denote the m partition sets of $\{e_0, \ldots, e_{w^m-1}\}$ defined by (5) for m=3, w=2, and m=2, w=3,

RESPECTIVELY.

i	0	1	2	i	0	1	2
$V_{i,0}$	e_0	e_0	e_0	$V_{i,1}$	e_4	e_2	e_1
	e_1	e_1	e_2		e_5	e_3	e_3
	e_2	e_4	e_4		e_6	e_6	e_5
	e_3	e_5	e_6		e_7	e_7	e_7
(A)							

-	i	0	1	i	0	1	i	0	1
		e_0	e_0		e_3	e_1		e_6	e_2
	$V_{i,0}$	e_1	e_3	$V_{i,1}$	e_4	e_4	$V_{i,2}$	e_7	e_5
		e_2	e_6		e_5	e_7		e_8	e_8
					(B)				

For convenience of notation, we also denote by $V_{i,t}$ the $w^{m-1} \times w^m$ matrix whose rows are formed by vectors e_a in their corresponding sets, and a is sorted in ascending order. For example, when m = 3 and w = 2, $V_{0,0}$ can be viewed as a 4×8 matrix as follows

$$V_{0,0} = \left(e_0^\top e_1^\top e_2^\top e_3^\top \right)^\top.$$

E. Basic Notations and Equalities

In this subsection, we introduce some useful notations and equalities that will facilitate the proof of the new codes. Let $N = w^m$, for $a = (a_0, \ldots, a_{m-1}) \in [0 : N)$, $i \in [0 : m)$ and $u \in [0 : w)$, define a(i, u) as

$$a(i, u) = (a_0, \dots, a_{i-1}, u, a_{i+1}, \dots, a_{m-1}),$$
(6)

i.e., replacing the i-th digit by u.

For $a = (a_0, a_1, \dots, a_{m-2}) \in [0 : N/w)$ and $i \in [0 : m)$, define

$$g_{i,u}(a) = (a_0, a_1, \dots, a_{i-1}, u, a_i, \dots, a_{m-2}),$$
(7)

i.e., inserting u to the i-th digit of $(a_0, a_1, \ldots, a_{m-2})$. Then for $i, j \in [0 : m)$ and $u, v \in [0 : w)$,

$$(g_{i,u}(a))_{j} = \begin{cases} a_{j}, & \text{if } j < i, \\ u, & \text{if } j = i, \\ a_{j-1}, & \text{if } j > i. \end{cases}$$
(8)

Replacing the *j*-th digit of $g_{i,u}(a)$ by v gives

$$(g_{i,u}(a))(j,v) = \begin{cases} g_{i,u}(a(j,v)), & \text{if } j < i, \\ g_{i,v}(a), & \text{if } j = i, \\ g_{i,u}(a(j-1,v)), & \text{if } j > i. \end{cases}$$
(9)

Let $e_0^{(N/w)}, e_1^{(N/w)}, \ldots, e_{N/w-1}^{(N/w)}$ be the standard basis vectors of $\mathbf{F}_q^{N/w}$ over \mathbf{F}_q , then by (7), $V_{i,u}$ in (5) can be rewritten as

$$V_{i,u} = \sum_{a=0}^{N/w-1} (e_a^{(N/w)})^\top e_{g_{i,u}(a)}, u \in [0:w),$$
(10)

i.e., the *a*-th row of the matrix $V_{i,u}$ is

$$V_{i,u}[a,:] = e_{g_{i,u}(a)}, 0 \le u < w, a \in [0:N/w).$$
(11)

III. A new (n,k) MSR code \mathcal{C}_1 with repair degree k < d < n

In this section, we propose an (n = 2m, k = n - r) MSR code construction C_1 with subpacketization level $N = w^m$ and repair degree d = k + w - 1 < n - 1 for some $w \in [2:r+1)$. The new MSR code can be viewed as a combination of the YB code 1 in [18] and CB code in [20], i.e., half of the parity-check matrix of C_1 is similar to the parity-check matrix of the YB code 1 while the other half is similar to that of CB code. This non-trivial combination leads to C_1 having a larger code length or, equivalently, a smaller sub-packetization level than that of the CB code and YB code 1. Throughout this section, let c be a primitive element of the finite field \mathbf{F}_q . **Construction 1.** For $N = w^m$ and $2 \le w \le r$, we define the parity-check matrix $(A_{t,i})_{t \in [0:r), i \in [0:n)}$ of the (n = 2m, k = n - r) array code C_1 over \mathbf{F}_q as

$$A_{t,i} = \begin{cases} \sum_{a=0}^{N-1} \lambda_{i,a_i}^t e_a^\top e_a + \sum_{a=0,a_i=0}^{N-1} \sum_{u=1}^{w-1} (\lambda_{i,0}^t - \lambda_{i,u}^t) e_a^\top e_{a(i,u)}, & \text{if } i \in [0:m), \\ \sum_{a=0}^{N-1} \lambda_{i,a_{i-m}}^t e_a^\top e_a, & \text{if } i \in [m:n), \end{cases}$$
(12)

where the repair degree is d = k + w - 1, $\lambda_{i,j} \in \mathbf{F}_q$, a_i denotes the *i*-th digit of the *w*-ary expansion of *a*, and $\sum_{a=0,a_i=0}^{N-1}$ denotes *a* runs through all [0:N) but with the restriction $a_i = 0$. We further define the repair matrix and select matrix of node *i* as

$$R_{i,j} = S_{i,t} = \begin{cases} V_{i,0}, & \text{if } i \in [0:m), \\ V_{i,0} + V_{i,1} + \dots + V_{i,w-1}, & \text{if } i \in [m:n), \end{cases}$$
(13)

for $t \in [0:r)$ and $j \in H_i$, where H_i is any d-subset of $[0:n) \setminus \{i\}$, $V_{i,0}, V_{i,1}, \ldots, V_{i,w-1}$ for $i \in [0:m)$ are defined in (5) and we further define

$$V_{i,u} = V_{i-m,u}, \text{ for } i \in [m,n), u \in [0:w)$$
 (14)

for convenience of notation.

In what follows, we first give an example to show the connection between the new code and the YB code 1, CB code, and then anther example to show the main idea of this construction.

Example 1. Consider the example where r = 3, w = 2, and m = 6. In this case, let $(A_{t,i})_{t \in [0:3), i \in [0:12)}$ be the parity-check matrix of the (12,9) code C_1 , then $(A_{t,i})_{t \in [0:3), i \in [0:6)}$ is exactly the parity-check matrix of the (6,3) YB code 1 in [18] while $(A_{t,i})_{t \in [0:3), i \in [6:12)}$ is exactly the parity-check matrix of the (6,3) CB code in [20].

Example 2. An example of the (6,3) MSR code C_1 with sub-packetization level 8 and repair degree 4 over \mathbf{F}_q , where q is any prime power larger than 12. The parity-check matrix

 $(A_{t,i})_{t\in[0:3),i\in[0:6)}$ is defined as

$$A_{t,0} = \begin{pmatrix} \lambda_{0,0}^{t}e_{0} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{4} \\ \lambda_{0,0}^{t}e_{1} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{5} \\ \lambda_{0,0}^{t}e_{2} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{6} \\ \lambda_{0,0}^{t}e_{3} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{7} \\ \lambda_{0,1}^{t}e_{4} \\ \lambda_{0,1}^{t}e_{5} \\ \lambda_{0,1}^{t}e_{6} \\ \lambda_{0,1}^{t}e_{7} \end{pmatrix}, A_{t,1} = \begin{pmatrix} \lambda_{1,0}^{t}e_{0} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{3} \\ \lambda_{1,0}^{t}e_{1} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{3} \\ \lambda_{1,1}^{t}e_{3} \\ \lambda_{1,0}^{t}e_{4} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{6} \\ \lambda_{1,0}^{t}e_{5} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{6} \\ \lambda_{1,0}^{t}e_{5} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{7} \\ \lambda_{1,1}^{t}e_{6} \\ \lambda_{1,1}^{t}e_{7} \end{pmatrix},$$

$$A_{t,2} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} + (\lambda_{2,0}^{t} - \lambda_{2,1}^{t})e_{1} \\ \lambda_{2,1}^{t}e_{1} \\ \lambda_{2,0}^{t}e_{2} + (\lambda_{2,0}^{t} - \lambda_{2,1}^{t})e_{3} \\ \lambda_{2,1}^{t}e_{3} \\ \lambda_{2,0}^{t}e_{4} + (\lambda_{2,0}^{t} - \lambda_{2,1}^{t})e_{5} \\ \lambda_{2,0}^{t}e_{5} \\ \lambda_{2,0}^{t}e_{6} + (\lambda_{2,0}^{t} - \lambda_{2,1}^{t})e_{7} \\ \lambda_{2,1}^{t}e_{7} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{3,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,0}^{t}e_{3} \\ \lambda_{3,1}^{t}e_{4} \\ \lambda_{3,1}^{t}e_{5} \\ \lambda_{3,1}^{t}e_{5} \\ \lambda_{4,1}^{t}e_{5} \\ \lambda_{4,1}^{t}e_{5} \\ \lambda_{4,1}^{t}e_{6} \\ \lambda_{4,1}^{t}e_{7} \end{pmatrix}, A_{t,5} = \begin{pmatrix} \lambda_{5,0}^{t}e_{0} \\ \lambda_{5,0}^{t}e_{1} \\ \lambda_{5,0}^{t}e_{2} \\ \lambda_{5,0}^{t}e_{2} \\ \lambda_{5,0}^{t}e_{3} \\ \lambda_{5,0}^{t}e_{4} \\ \lambda_{5,0}^{t}e_{5} \\ \lambda_{4,1}^{t}e_{5} \\ \lambda_{4,1}^{t}e_{6} \\ \lambda_{4,1}^{t}e_{7} \end{pmatrix}, A_{t,5} = \begin{pmatrix} \lambda_{5,0}^{t}e_{0} \\ \lambda_{5,0}^{t}e_{1} \\ \lambda_{5,0}^{t}e_{2} \\ \lambda_{5,0}^{t}e_{4} \\ \lambda_{5,0}^{t}e_{5} \\ \lambda_{5,0}^{t}e_{6} \\ \lambda_{5,0}^{t}e_{6} \\ \lambda_{5,1}^{t}e_{7} \end{pmatrix}$$

where

$$\lambda_{0,0} = 1, \lambda_{1,0} = c^4, \lambda_{2,0} = c^8, \lambda_{3,0} = c^2, \lambda_{4,0} = c^6, \lambda_{5,0} = c^{10},$$

$$\lambda_{0,1} = c, \lambda_{1,1} = c^5, \lambda_{2,1} = c^9, \lambda_{3,1} = c^3, \lambda_{4,1} = c^7, \lambda_{5,1} = c^{11},$$

(15)

with c being a primitive element in \mathbf{F}_q .

Suppose that Node 3 fails and Node 0 is not connected, we claim that Node 3 can be repaired by connecting Nodes 1, 2, 4, 5 and downloading $(V_{0,0} + V_{0,1})\mathbf{f}_j$ (i.e., $(e_0 + e_4)\mathbf{f}_j, (e_1 + e_5)\mathbf{f}_j, (e_2 + e_5)\mathbf{f}_j, (e_3 + e_5)\mathbf{f}_j, (e_4 + e_5)\mathbf{f}_j, (e_5 + e_5)\mathbf{f}_j, (e_5 + e_5)\mathbf{f}_j, (e_6 + e_5)\mathbf{f}_j, (e_$ $(e_6)\mathbf{f}_j, (e_3 + e_7)\mathbf{f}_j)$ for j = 1, 2, 4, 5, and choose $S_{3,t} = V_{0,0} + V_{0,1}$ for t = 0, 1, 2. Then, from (3), DRAFT

May 30, 2023

we have

$$= - \begin{pmatrix} e_{0} + e_{4} \\ e_{1} + e_{5} \\ e_{2} + e_{6} \\ e_{3} + e_{7} \\ \lambda_{3,0}e_{0} + \lambda_{3,1}e_{4} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{5} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{6} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{2} + \lambda_{3,1}e_{7} \\ \lambda_{3,0}e_{1} + e_{5} \\ e_{2} + e_{6} \\ e_{3} + e_{7} \\ \lambda_{2,0}(e_{1} + e_{4}) + (\lambda_{2,0} - \lambda_{2,1})(e_{1} + e_{5}) \\ \lambda_{2,1}(e_{1} + e_{5}) \\ \lambda_{2,0}(e_{2} + e_{6}) + (\lambda_{2,0} - \lambda_{2,1})(e_{1} + e_{5}) \\ \lambda_{2,0}(e_{2} + e_{6}) + (\lambda_{2,0} - \lambda_{2,1})(e_{1} + e_{5}) \\ \lambda_{3,0}(e_{2} + e_{6}) \\ \lambda_{4,1}(e_{2} + e_{6}) \\ \lambda_{4,1}(e_{2} + e_{6}) \\ \lambda_{4,1}(e_{2} + e_{6}) \\ \lambda_{4,1}(e_{2} + e_{6}) \\ \lambda_{4,0}(e_{1} + e_{5}) \\ \lambda_{4,0}(e_{1} + e_{5}) \\ \lambda_{4,0}(e_{1} + e_{5}) \\ \lambda_{2,0}(e_{2} + e_{6}) + (\lambda_{2,0}^{2} - \lambda_{2,1}^{2})(e_{3} + e_{7}) \\ \lambda_{2,0}^{2}(e_{1} + e_{5}) \\ \lambda_{3,0}^{2}(e_{1} + e_{5}) \\ \lambda_{3,0}^{2}(e_{1} + e_{5}) \\ \lambda_{3,0}^{2}(e_{1} + e_{5}) \\ \lambda_{4,0}^{2}(e_{1} + e_{5}) \\ \lambda_{5,0}^{2}(e_{2} + e_{6}) \\ \lambda_{5,0}^{2}(e_$$

DRAFT

which can be reformulated as

$$\underbrace{\begin{pmatrix} I_4 & I_4 & I_4 \\ \lambda_{3,0}I_4 & \lambda_{3,1}I_4 & \lambda_{0,0}I_4 \\ \lambda_{3,0}^2I_4 & \lambda_{3,1}^2I_4 & \lambda_{0,0}^2I_4 \end{pmatrix}}_{M} \begin{pmatrix} V_{0,0}\mathbf{f}_3 \\ V_{0,1}\mathbf{f}_3 \\ (V_{0,0}+V_{0,1})\mathbf{f}_0 \end{pmatrix} = \kappa_*,$$
(17)

where κ_* denotes the data related to $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_4, \mathbf{f}_5$ in (16) and can be determined from the downloaded data.

Using Lemma 1 and (15), we can see that the matrix M in (17) is non-singular. Therefore, we can solve (17) to obtain $V_{0,0}\mathbf{f}_3$ and $V_{0,1}\mathbf{f}_3$ (i.e., \mathbf{f}_3) and regenerate the lost data.

In Example 2, it is obvious to see that all the matrices $A_{t,i}$, $t \in [0:3)$, $i \in [0:6)$ are upper triangular. The situation also holds for the general case (cf. (12)). Therefore, the MDS property can be easily verified according to Lemma 1. In the following, we formally analyze the MDS property of the new code C_1 .

Theorem 1. The new code C_1 is an MDS array code if

- i) $\lambda_{i,u} \neq \lambda_{j,v}$ for $u, v \in [0:w)$ and $i, j \in [0:n)$ with $i \not\equiv j \mod m$,
- ii) $\lambda_{i,u} \neq \lambda_{i+m,u}$ for $u \in [0:w)$ and $i \in [0:m)$.

Proof: It suffices to prove that for any pairwise distinct $j_0, j_1, \ldots, j_{r-1} \in [0:n)$, the block matrix

$$\begin{pmatrix} A_{0,j_0} & A_{0,j_1} & \cdots & A_{0,j_{r-1}} \\ A_{1,j_0} & A_{1,j_1} & \cdots & A_{1,j_{r-1}} \\ \vdots & \vdots & \vdots & \vdots \\ A_{r-1,j_0} & A_{r-1,j_1} & \cdots & A_{r-1,j_{r-1}} \end{pmatrix}$$
(18)

is non-singular over \mathbf{F}_q .

May 30, 2023

DRAFT

For any $a, b \in [0:N)$, $i \in [0:n)$ and $t \in [0:r)$, according to (12), we have

$$\begin{aligned} A_{t,i}[a,b] &= e_a A_{t,i} e_b^\top \\ &= \begin{cases} e_a \left(\sum_{z=0}^{N-1} \lambda_{i,z_i}^t e_z^\top e_z + \sum_{z=0,z_i=0}^{N-1} \sum_{u=1}^{w-1} (\lambda_{i,0}^t - \lambda_{i,u}^t) e_z^\top e_{z(i,u)} \right) e_b^\top, & \text{if } i \in [0:m), \\ e_a \left(\sum_{z=0}^{N-1} \lambda_{i,z_{i-m}}^t e_z^\top e_z \right) e_b^\top, & \text{if } i \in [m:n), \end{cases} \\ &= \begin{cases} \lambda_{i,a_i}^t e_a e_b^\top + \left(e_a \sum_{z=0,z_i=0}^{N-1} \sum_{u=1}^{w-1} (\lambda_{i,0}^t - \lambda_{i,u}^t) e_z^\top e_{z(i,u)} \right) e_b^\top, & \text{if } i \in [0:m), \\ \lambda_{i,a_{i-m}}^t e_a e_b^\top, & \text{if } i \in [m:n), \end{cases} \\ &\lambda_{i,0}^t - \lambda_{i,u}^t, & \text{if } i \in [0:m), \text{ and } b = a, \\ \lambda_{i,0}^t - \lambda_{i,u}^t, & \text{if } i \in [m:n), a_i = 0, \text{ and } b = a(i,u) \text{ for } u = 1, 2, \dots, w - 1, \\ \lambda_{i,a_{i-m}}^t, & \text{if } i \in [m:n) \text{ and } b = a, \\ 0, & \text{otherwise,} \end{cases} \end{aligned}$$

which implies that $A_{t,i}[a, b] = 0$ for $0 \le b < a < N$ (i.e., $A_{t,i}$ is an upper triangular matrix) and

$$A_{t,i}[a,a] = \lambda_{i,a_{i\%m}}^t = (A_{1,i}[a,a])^t \text{ for } a \in [0:N),$$
(20)

where % denotes the modulo operation, $t \in [0 : r)$, and $i \in [0 : n)$. This implies that i) of Lemma 1 holds for the matrix in (18).

For any $t \in [0:r), a \in [0:N)$ and $0 \le i < j < n$, by (20), we have

$$A_{1,i}[a,a] - A_{1,j}[a,a] = \begin{cases} \lambda_{i,a_{i\%m}} - \lambda_{j,a_{j\%m}}, & \text{if } i \neq j \mod m, \\ \\ \lambda_{i,a_{i\%m}} - \lambda_{j,a_{i\%m}}, & \text{otherwise,} \end{cases}$$

which together with i) and ii) implies $A_{1,i}[a, a] - A_{1,j}[a, a] \neq 0$, i.e., ii) of Lemma 1 holds for the matrix in (18). Finally, applying Lemma 1, we claim that the matrix in (18) is non-singular, and then we reach the desired result.

Analyzing the repair property requires that (3) is solvable based on the downloaded data. Thus it is helpful to characterize the product of $S_{i,t}$ and $A_{t,j}$ beforehand.

16

 $\begin{aligned} \text{Lemma 2. For any } i, j \in [0:n), \text{ rewrite them as } i = g_0 m + i' \text{ and } j = g_1 m + j' \text{ for } g_0, g_1 \in \{0,1\} \\ \text{and } i', j' \in [0:m). \text{ Then for } t \in [0:r), \text{ we have} \\ i) \quad S_{i,t}A_{t,i} = \begin{cases} \lambda_{i,0}^t V_{i,0} + (\lambda_{i,0}^t - \lambda_{i,1}^t) V_{i,1} + \dots + (\lambda_{i,0}^t - \lambda_{i,w-1}^t) V_{i,w-1}, & \text{if } i \in [0:m), \\ \lambda_{i,0}^t V_{i,0} + \lambda_{i,1}^t V_{i,1} + \dots + \lambda_{i,w-1}^t V_{i,w-1}, & \text{if } i \in [m:2m), \end{cases} \\ ii) \quad S_{i,t}A_{t,j} = B_{t,j,i}R_{i,j} \text{ for } j \neq i, \text{ where } B_{t,j,i} \text{ is an } \frac{N}{w} \times \frac{N}{w} \text{ matrix define } by \end{cases} \\ \begin{cases} \sum_{a=0}^{N/w-1} \lambda_{j,a,j}^t (e_a^{(N/w)})^\top e_a^{(N/w)} \\ + \sum_{a=0,a_j=0}^{N/w-1} \sum_{u=1}^{w-1} (\lambda_{j,0}^t - \lambda_{j,u}^t) (e_a^{(N/w)})^\top e_{a(j,u)}^{(N/w)}, & \text{if } j \in [0:i'], \end{cases} \\ B_{t,j,i} = \begin{cases} \sum_{a=0}^{N/w-1} \lambda_{j,a,j-1}^t (e_a^{(N/w)})^\top e_a^{(N/w)} \\ + \sum_{a=0,a_j-1=0}^{N/w-1} \sum_{u=1}^{w-1} (\lambda_{j,0}^t - \lambda_{j,u}^t) (e_a^{(N/w)})^\top e_{a(j-1,u)}^{(N/w)}, & \text{if } j \in [i'+1:m), \end{cases} \\ \sum_{a=0}^{N/w-1} \lambda_{j,a,j-m}^t (e_a^{(N/w)})^\top e_a^{(N/w)}, & \text{if } j \in [m:m+i'], \end{cases} \\ \sum_{a=0}^{N/w-1} \lambda_{j,a,j-m}^t (e_a^{(N/w)})^\top e_a^{(N/w)}, & \text{if } j \in [m+i'+1:n), \end{cases} \\ \lambda_{j,0}^t I_{N/w}, & \text{if } j \equiv i \mod m, \end{cases} \end{aligned}$

where $e_0^{(N/w)}, e_1^{(N/w)}, \ldots, e_{N/w-1}^{(N/w)}$ are the standard basis of $\mathbf{F}_q^{N/w}$.

iii) For the matrix in (21), we have

$$B_{t,j,i}[a,b] = 0 \text{ for any } t \in [0,r), 0 \le b < a < N/w,$$
(22)

and

$$B_{t,j,i}[a,a] = \begin{cases} \lambda_{j,a_{j'}}^t, & \text{if } j' < i', \\ \lambda_{j,0}^t, & \text{if } j' = i', \\ \lambda_{j,a_{j'-1}}^t, & \text{if } j' > i', \end{cases}$$
(23)

for any $a \in [0: N/w)$.

Proof: The proof is given in Appendix B.

With this lemma, we can now analyze the repair property according to (3).

Theorem 2. The new code C_1 the optimal repair bandwidth with repair degree d = k + w - 1 if

- i) $\lambda_{i,u} \neq \lambda_{i,v}$ for $u, v \in [0:w)$ with $u \neq v$ and $i \in [0:n)$,
- ii) $\lambda_{i,u} \neq \lambda_{j,v}$ for $u, v \in [0:w)$ and $i, j \in [0:n)$ with $i \not\equiv j \mod m$,
- iii) If w < r, $\lambda_{i,0} \neq \lambda_{i+m,u}$ and $\lambda_{i,u} \neq \lambda_{i+m,0}$ for $u \in [0:w)$ and $i \in [0:m)$.

Proof: We consider the repair of node i when w < r, where we only check $i \in [0 : m)$ since the case $i \in [m : n)$ can be verified similarly. By Lemma 2, we can express (3) as

$$\begin{pmatrix} V_{i,0} \\ \lambda_{i,0}V_{i,0} + \sum_{t=1}^{w-1} (\lambda_{i,0} - \lambda_{i,t})V_{i,t} \\ \vdots \\ \lambda_{i,0}^{r-1}V_{i,0} + \sum_{t=1}^{w-1} (\lambda_{i,0}^{r-1} - \lambda_{i,t}^{r-1})V_{i,t} \end{pmatrix} \mathbf{f}_{i} + \sum_{l \in L_{i}} \begin{pmatrix} B_{0,l,i} \\ B_{1,l,i} \\ \vdots \\ B_{r-1,l,i} \end{pmatrix} R_{i,l}\mathbf{f}_{l} + \sum_{j \in H_{i}} \begin{pmatrix} B_{0,j,i} \\ B_{1,j,i} \\ \vdots \\ B_{r-1,j,i} \end{pmatrix} R_{i,j}\mathbf{f}_{j} = \mathbf{0},$$
(24)

Let $L_i = \{l_0, l_1, \dots, l_{r-w-1}\}$, substituting them into the above equations, we then have

$$\begin{pmatrix}
I_{N/w} & 0_{N/w} & \cdots & 0_{N/w} & B_{0,l_0,i} & \cdots & B_{0,l_{r-w-1},i} \\
\lambda_{i,0}I_{N/w} & (\lambda_{i,0} - \lambda_{i,1})I_{N/w} & \cdots & (\lambda_{i,0} - \lambda_{i,w-1})I_{N/w} & B_{1,l_0,i} & \cdots & B_{1,l_{r-w-1},i} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{i,0}^{r-1}I_{N/w} & (\lambda_{i,0}^{r-1} - \lambda_{i,1}^{r-1})I_{N/w} & \cdots & (\lambda_{i,0}^{r-1} - \lambda_{i,w-1}^{r-1})I_{N/w} & B_{r-1,l_0,i} & \cdots & B_{r-1,l_{r-w-1},i} \end{pmatrix}$$

$$\begin{pmatrix}
V_{i,0}\mathbf{f}_i \\
\vdots \\
V_{i,w-1}\mathbf{f}_i \\
R_{i,l_0}\mathbf{f}_{l_0} \\
\vdots \\
R_{i,l_r-w-1}\mathbf{f}_{l_{r-w-1}}
\end{pmatrix} = -\sum_{j\in H_i} \begin{pmatrix}
B_{0,j,i} \\
B_{1,j,i} \\
\vdots \\
B_{r-1,j,i}
\end{pmatrix} R_{i,j}\mathbf{f}_j.$$
(25)

May 30, 2023

DRAFT

It is easy to see that the matrix B can be converted to

$$B' = \begin{pmatrix} I_{N/w} & I_{N/w} & \cdots & I_{N/w} & B_{0,l_{0,i}} & \cdots & B_{0,l_{r-w-1,i}} \\ \lambda_{i,0}I_{N/w} & \lambda_{i,1}I_{N/w} & \cdots & \lambda_{i,w-1}I_{N/w} & B_{1,l_{0,i}} & \cdots & B_{1,l_{r-w-1,i}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \lambda_{i,0}^{r-1}I_{N/w} & \lambda_{i,1}^{r-1}I_{N/w} & \cdots & \lambda_{i,w-1}^{r-1}I_{N/w} & B_{r-1,l_{0,i}} & \cdots & B_{r-1,l_{r-w-1,i}} \end{pmatrix}$$
(26)

by elementary column operations.

By Lemma 2-iii), we have that the matrices $B_{1,l_0,i}, \ldots, B_{1,l_{r-w-1},i}$ are upper triangular and $B_{t,l_j,i}[a,a] = (B_{1,l_j,i}[a,a])^t$, for $t \in [0:r)$, $j \in [0:r-w)$, and $a \in [0:N/w)$. Similar to the proof of Theorem 1, by Lemma 1 and (23), we easily have that the block matrix B' in (26) is non-singular if

$$\lambda_{i,0}, \lambda_{i,1}, \dots, \lambda_{i,w-1}, B_{1,l_0,i}[a,a], \dots, B_{1,l_{r-w-1},i}[a,a],$$

are pairwise distinct for any $i \in [0:m)$, $l_0, \ldots, l_{r-w-1} \in L_i$, and $a \in [0:N)$, i.e.,

$$\lambda_{i,0}, \lambda_{i,1}, \dots, \lambda_{i,w-1}, B_{1,j,i}[a,a], j \in [0:n) \setminus \{i\},$$

are pairwise distinct for any $i \in [0:m)$ and $a \in [0:N)$ since L_i is an arbitrary (r-w)-subset of $[0:n) \setminus \{i\}$, which can be satisfied if i)-iii) hold according to (23). Therefore, if conditions i)-iii) hold, then B in (25) is non-singular. As a result, we can solve for $V_{i,0}\mathbf{f}_i, \cdots, V_{i,w-1}\mathbf{f}_i$ (i.e., \mathbf{f}_i) and $R_{i,l}\mathbf{f}_l, l \in L_i$, since the right side hand of (25) is known from the downloaded data.

When w = r, the proof is similar to the case w < r instead that the condition in iii) is not needed by noting $L_i = \emptyset$ in (24).

Theorem 3. The requirements in items i), ii) of Theorem 1 and i)–iii) of Theorem 2 can be fulfilled by setting

$$\lambda_{i,u} = \begin{cases} c^{i(w+2)+u}, & \text{if } w = 2, \\ c^{i(w+1)+u}, & \text{if } w \in [3:r), \\ c^{iw+u}, & \text{if } w = r, \end{cases} \quad \lambda_{i+m,u} = \begin{cases} c^{i(w+2)+w+u}, & \text{if } w = 2, \\ c^{i(w+1)+w}, & \text{if } w \in [3:r), u = 0, \\ c^{i(w+1)+u\%(w-1)+1}, & \text{if } w \in [3:r), u \ge 1, \\ c^{iw+(u+1)\%r}, & \text{if } w = r, \end{cases}$$
(27)

for $i \in [0:m)$ and $u \in [0:w)$, where c is a primitive element of \mathbf{F}_q with

$$q > \begin{cases} m(w+2), & \text{if } w = 2, \\ m(w+1), & \text{if } w \in [3:r), \\ mw, & \text{if } w = r. \end{cases}$$

Proof: We only verify the case $w \in [3:r)$, as the proofs for the remaining cases are similar. For any $i, j \in [0:n)$ and $u, v \in [0:w)$ with $(i, u) \neq (j, v)$, rewrite $i = g_0 m + i'$ and $j = g_1 m + j'$, where $g_0, g_1 \in \{0, 1\}$ and $i', j' \in [0:m)$.

i) When $i \neq j\%m$, by (27), we have $\lambda_{i,u} = c^{i'(w+1)+t}$ and $\lambda_{j,v} = c^{j'(w+1)+s}$ for some $t, s \in [0:w+1)$. Then $\lambda_{i,u} - \lambda_{j,v} = c^{i'(w+1)+t}(1 - c^{(j'-i')(w+1)+s-t}) \neq 0$ since

$$0 < |(j' - i')(w + 1) + s - t| \le (m - 1)(w + 1) + w = m(w + 1) - 1 < q - 1.$$

Therefore, i) of Theorem 1 and also ii) of Theorem 2 are satisfied.

ii) For $i \in [0:m)$, by (27), we have

$$\lambda_{i+m,0} = c^{i(w+1)+w} = c^w c^{i(w+1)} = c^w \lambda_{i,0} \neq \lambda_{i,0},$$
$$\lambda_{i+m,u} = c^{i(w+1)+u\%(w-1)+1} \neq c^{i(w+1)+u} = \lambda_{i,u} \text{ for } u \ge 1$$

since $u\%(w-1) + 1 \neq u$ for $u \in [1:w)$, which shows that ii) of Theorem 1 is satisfied.

- iii) From (27), it is obvious to see $\lambda_{i,u} \neq \lambda_{i,v}$ for $u \neq v$, i.e., i) of Theorem 2 is satisfied.
- iv) For $u \in [1:w)$ and $i \in [0:m)$, by (27), we have

$$\lambda_{i,0} - \lambda_{i+m,u} = c^{i(w+1)} - c^{i(w+1) + (u\%(w-1)) + 1} \neq 0,$$

since $c^{(u\%(w-1))+1} \neq 1$ and $\lambda_{i,u} - \lambda_{i+m,0} = c^{i(w+1)+u} - c^{i(w+1)+w} \neq 0$. Note that $\lambda_{i,0} \neq \lambda_{i+m,0}$ has been proved in ii), thus, iii) of Theorem 2 is satisfied.

This completes the proof.

Remark 1. In Construction 1, we assumed that $2 \mid n$ for the (n, k) MSR code C_1 . If $2 \nmid n$, through shortening, one can easily obtain an (n, k) MSR code with repair degree d from an (n + 1, k + 1) MSR code C_1 with repair degree d + 1 [2, Theorem 6].

May 30, 2023

Remark 2. When w = r, since Theorem 2-iii) is not needed to satisfy, then we can choose $\lambda_{i+m,0}, \lambda_{i+m,1}, \ldots, \lambda_{i+m,w-1}$ from the set $\{\lambda_{i,0}, \lambda_{i,1}, \ldots, \lambda_{i,w-1}\}$, which leads to a smaller finite field compared to the case w < r.

IV. A New MDS array code \mathcal{C}_2 with small sub-packetization level

In [29], a generic transformation that can transform any (n', k') MSR code into a new (n = sn', k) MDS array code was proposed for any $s \ge 2$, which can greatly reduce the subpacketization level by sacrificing a bit repair bandwidth. Note that the sub-packetization level of the base code determines that of the new array code. Thus, it is desirable to choose an MSR code with a small sub-packetization level as the base code. The MSR code C_1 in the previous section has a small sub-packetization and is suitable to serve as the base code. In this section, by applying the generic transformation in [29] to the (n', k') code C_1 with d' = n' - 1 in the previous section, we construct an (n = sn', k) MDS array code C_2 with small sub-packetization level and $(1 + \epsilon)$ -optimal repair bandwidth, where the repair degree is d = n - 1.

Construction 2. Based on the generic transformation in [29], the new (n, k) array code C_2 is constructed throught two steps as follows.

Step 1. Choosing the (n', k') MSR code C_1 with repair degree d' = n' - 1 in Section III as the base code. Let $(A_{t,i'})_{t \in [0:r), i' \in [0:n')}$, $S'_{i',t}$, and $R'_{i',j'}$ denote its parity-check matrix, select matrices, and repair matrices, where r = n' - k', $i', j' \in [0:n')$, $j' \neq i'$, and $t \in [0:r)$.

Step 2. Applying the generic transformation in [29] to the (n', k') MSR code C_1 , then an (n = sn', k) array code C_2 with repair degree d = n - 1 is obtained, where the parity-check matrix $(A_{t,i})_{t \in [0:r), i \in [0:n)}$, select matrices $S_{i,t}$, and repair matrices $R_{i,j}$ are given as

$$A_{t,i} = x_{t,i}A'_{t,i\%n'}, \ S_{i,t} = S'_{i\%n',t}, R_{i,j} = \begin{cases} R'_{i\%n',j\%n'}, & \text{if } j \not\equiv i \mod n', \\ I, & \text{otherwise}, \end{cases}$$
(28)

with $i, j \in [0:n)$, $j \neq i$, $t \in [0,r)$, $x_{t,i} \in \mathbf{F}_q \setminus \{0\}$, and again % denotes the modulo operation.

Lemma 3. ([29, Theorem 2]) Every failed node of the new (n, k) array code C_2 obtained by the generic transformation can be regenerated by the repair matrices defined in (28), the repair bandwidth is $(1 + \frac{(s-1)(r-1)}{n-1})\gamma_{\text{optimal}}$, where $\gamma_{\text{optimal}} = \frac{n-1}{r}N$ denotes the optimal repair bandwidth.

Example 3. From Construction 1, the parity-check matrix $(A_{t,i})_{t \in [0:2), i \in [0:4)}$ of the (n' = 4, k' = 2)MSR code C_1 with d' = 3 and $N = (d' - k' + 1)^{n'/2} = 2^2$ is given as

$$A_{t,0} = \begin{pmatrix} \lambda_{0,0}^{t}e_{0} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{2} \\ \lambda_{0,0}^{t}e_{1} + (\lambda_{0,0}^{t} - \lambda_{0,1}^{t})e_{3} \\ \lambda_{0,1}^{t}e_{2} \\ \lambda_{0,1}^{t}e_{3} \end{pmatrix}, A_{t,1} = \begin{pmatrix} \lambda_{1,0}^{t}e_{0} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{1} \\ \lambda_{1,0}^{t}e_{2} + (\lambda_{1,0}^{t} - \lambda_{1,1}^{t})e_{3} \\ \lambda_{1,1}^{t}e_{3} \end{pmatrix}, A_{t,2} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{2,0}^{t}e_{1} \\ \lambda_{2,1}^{t}e_{2} \\ \lambda_{2,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{3,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,0}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,0}^{t}e_{2} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{3} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{1} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{3} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{3} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_{t,3} = \begin{pmatrix} \lambda_{2,0}^{t}e_{0} \\ \lambda_{3,1}^{t}e_{3} \end{pmatrix}, A_$$

where $t \in [0:2)$, $\lambda_{t,i}$, $t \in [0:2)$, $i \in [0:4)$ are set according to (27). By setting s = 2 in Construction 2, we obtain an (n = 8, k = 6) MDS array code C_2 with $N = 2^2$ and d = 7, based on the (n' = 4, k' = 2) MSR code C_1 from Construction 1. The parity-check matrix $(A_{t,i})_{t \in [0:2), i \in [0:8)}$ of the new MDS array code C_2 is given as

$$A_{t,0} = \begin{pmatrix} \lambda_{0,0}^t e_0 + (\lambda_{0,0}^t - \lambda_{0,1}^t) e_2 \\ \lambda_{0,0}^t e_1 + (\lambda_{0,0}^t - \lambda_{0,1}^t) e_3 \\ \lambda_{0,1}^t e_2 \\ \lambda_{0,1}^t e_3 \end{pmatrix}, \quad A_{t,1} = \begin{pmatrix} \lambda_{1,0}^t e_0 + (\lambda_{1,0}^t - \lambda_{1,1}^t) e_1 \\ \lambda_{1,1}^t e_1 \\ \lambda_{1,0}^t e_2 + (\lambda_{1,0}^t - \lambda_{1,1}^t) e_3 \\ \lambda_{1,1}^t e_3 \end{pmatrix}, \quad A_{t,2} = \begin{pmatrix} \lambda_{2,0}^t e_0 \\ \lambda_{2,0}^t e_1 \\ \lambda_{2,1}^t e_2 \\ \lambda_{2,1}^t e_3 \end{pmatrix},$$

$$A_{t,3} = \begin{pmatrix} \lambda_{3,0}^t e_0 \\ \lambda_{3,1}^t e_1 \\ \lambda_{3,0}^t e_2 \\ \lambda_{3,1}^t e_3 \end{pmatrix}, \ A_{t,4} = c^{4t} \begin{pmatrix} \lambda_{0,0}^t e_0 + (\lambda_{0,0}^t - \lambda_{0,1}^t) e_2 \\ \lambda_{0,0}^t e_1 + (\lambda_{0,0}^t - \lambda_{0,1}^t) e_3 \\ \lambda_{0,1}^t e_2 \\ \lambda_{0,1}^t e_3 \end{pmatrix}, \ A_{t,5} = c^{4t} \begin{pmatrix} \lambda_{1,0}^t e_0 + (\lambda_{1,0}^t - \lambda_{1,1}^t) e_1 \\ \lambda_{1,1}^t e_1 \\ \lambda_{1,0}^t e_2 + (\lambda_{1,0}^t - \lambda_{1,1}^t) e_3 \\ \lambda_{1,1}^t e_3 \end{pmatrix},$$
$$A_{t,6} = c^{4t} \begin{pmatrix} \lambda_{2,0}^t e_0 \\ \lambda_{2,0}^t e_1 \\ \lambda_{2,1}^t e_2 \\ \lambda_{2,1}^t e_3 \end{pmatrix}, \ A_{t,7} = c^{4t} \begin{pmatrix} \lambda_{3,0}^t e_0 \\ \lambda_{3,1}^t e_1 \\ \lambda_{3,0}^t e_2 \\ \lambda_{3,1}^t e_3 \end{pmatrix},$$

May 30, 2023

where

$$\lambda_{0,0} = 1, \lambda_{1,0} = c^2, \lambda_{2,0} = c, \lambda_{3,0} = c^3, \lambda_{0,1} = c, \lambda_{1,1} = c^3, \lambda_{2,1} = 1, \lambda_{3,1} = c^2,$$
(29)

with c being a primitive element in \mathbf{F}_q where q > 8.

Theorem 4. Setting $x_{t,i}$ in (28) as

$$x_{t,i} = x_i^t, (30)$$

for some $x_i \in \mathbf{F}_q \setminus \{0\}$, where $i \in [0:n)$ and $t \in [0:r)$, the code C_2 in Construction 2 is an (n = sn', k) MDS array code with repair degree d = n - 1 over \mathbf{F}_q and repair bandwidth $(1 + \frac{(s-1)(r-1)}{n-1})\gamma_{\text{optimal}}$, where $\gamma_{\text{optimal}} = \frac{n-1}{r}N$, if the following conditions i)-iii) hold i) $x_i\lambda_{i',u} \neq x_j\lambda_{j',u'}$ for $u, u' \in [0:r)$ and $i, j \in [0:n)$ with $i \neq j \mod m$, ii) $x_i\lambda_{i',u} \neq x_j\lambda_{j',u}$ for $u \in [0:r)$ and $i, j \in [0:n)$ with $i \neq j$ and $i \equiv j \mod m$, iii) $\lambda_{i',u} \neq \lambda_{i',u'}$ for $u, u' \in [0:r)$ with $u \neq u'$ and $i' \in [0:n)$, where i' = i%n' and j' = j%n'.

Proof: The repair property follows from Lemma 3, and the proof of the MDS property is similar to that of Theorem 1. Therefore, we omit it here.

Theorem 5. The requirements in items i) - iii) of Theorem 4 can be fulfilled by setting $x_i = c^{\lfloor i/n' \rfloor mr}$ for $i \in [0:n)$, where c is a primitive element of \mathbf{F}_q with q > smr.

Proof: For $i, j \in [0:n)$, we rewrite them as $i = v_0 n' + i'$ and $j = v_1 n' + j'$ for $v_0, v_1 \in [0:s)$ and $i', j' \in [0:n)$, and further rewrite i' and j' as $i' = g_0 m + i''$ and $j' = g_1 m + j''$, where $g_0, g_1 \in \{0, 1\}$ and $i'', j'' \in [0:m)$. By (27), we have

$$x_i \lambda_{i'\,u} = c^{(v_0 m + i'')r + (u + g_0)\%r}.$$
(31)

Then, by (31), items i) - iv) of Theorem 4 can be verified according to the following three cases.

DRAFT

• For $u, u' \in [0:w)$ and $i, j \in [0:n)$ with $i \not\equiv j \mod m$, i.e., $i'' \neq j''$, we have

$$x_i \lambda_{i',u} - x_j \lambda_{j',u'} = c^{(v_0 m + i'')r + (u + g_0)\%r} (1 - c^{((v_1 - v_0)m + j'' - i'')r + (u' + g_1)\%r - (u + g_0)\%r}) \neq 0$$

since $0 < |((v_1 - v_0)m + j'' - i'')r + (u' + g_1)\%r - (u + g_0)\%r| \le smr - 1 < q - 1$. Then i) of Theorem 4 is satisfied.

• For $u \in [0:w)$ and $i, j \in [0:n)$ with $i \neq j$ and $i \equiv j \mod m$, i.e, i'' = j'', we have

$$x_i \lambda_{i',u} - x_j \lambda_{j',u} = c^{(v_0 m + i'')r + (u + g_0)\% r} (1 - c^{(v_1 - v_0)mr + (u + g_1)\% r - (u + g_0)\% r}) \neq 0$$

since $0 < |(v_1 - v_0)mr + (u + g_1)\%r - (u + g_0)\%r| \le (s - 1)mr + 1 < q - 1$, i.e., ii) of Theorem 4 is satisfied.

• It is obvious that iii) of Theorem 4 is satisfied according to (27).

This completes the proof.

V. COMPARISONS

In this section, we provide a detailed comparison of some key parameters among the proposed (n,k) MSR code C_1 with repair degree d < n - 1, (n,k) MDS array code C_2 with repair degree d = n - 1, and existing ones. Table II provides the details of the comparison between the proposed (n,k) MSR code C_1 with repair degree d < n - 1 and existing ones. Meanwhile, Figure 1 shows the sub-packetization levels and required field sizes of each code with a repair degree of d = k + 2 when the code length ranges from 10 to 100.

From Table II and Figure 1, we see that the new MSR code C_1 has the following advantages.

- i) The new MSR code C_1 has a significantly smaller sub-packetization level than the YB codes 1, 2 in [18], and the CB code in [20], and a smaller finite field than that of YB code 1.
- ii) The new MSR code C₁ works for any repair degree d ∈ [k + 1 : n), which is much more flexible than that of the VBK code in [19], which is restricted to d = k + 1, k + 2, k + 3. Additionally, C₁ requires a much smaller finite field than the VBK code when w ∈ {2,3,4}. Specifically, when w = 2, 3, and 4, C₁ requires a finite field F_q with size q > 4[ⁿ/₂], q > 3[ⁿ/₂],

A comparison of the key parameters of (n,k) MSR codes with sub-packetization level N.

	N	Field size	Repair degree	References
YB code 1	w^n	$q \ge wn$	$d = k + w - 1 \in [k+1:n)$	[18, Section IV]
YB code 2	w^n	q > n	$d = k + w - 1 \in [k+1:n)$	[18, Section VIII]
VBK code	$w^{\lceil \frac{n}{w} \rceil}$ $(w = 2, 3, 4)$	$q \ge \begin{cases} 6\lceil \frac{n}{2} \rceil + 2, w = 2\\ 18\lceil \frac{n}{w} \rceil + 2, w = 3, 4 \end{cases}$	d = k + 1, k + 2, k + 3	[19]
CB code	w^n	q > n + w	$q > n + w \qquad \qquad d = k + w - 1 \in [k + 1 : n)$	
LLT code	$w^{\lceil \frac{n}{2} \rceil}$	$q > n + \lceil \frac{n}{2} \rceil w$	$d = k + w - 1 \in [k + 1 : n - 1)$	[21]
ZZ code	$w^{\lceil \frac{n}{2}\rceil}$	$q \ge wn$	$d = k + w - 1 \in [k+1:n)$	[32]
New code C_1	$w^{\lceil \frac{n}{2} \rceil}$	$q > \begin{cases} 4\lceil \frac{n}{2} \rceil, w = 2\\ \lceil \frac{n}{2} \rceil(w+1), w \in [3:r)\\ \lceil \frac{n}{2} \rceil w, w = r \end{cases}$	$d = k + w - 1 \in [k+1:n)$	Theorem 3



Fig. 1. Comparision of the sub-packetization level and finite field size among the new (n, k) MSR code C_1 and some known ones with repair degree d = k + 2.

and $q > 4\lceil \frac{n}{2} \rceil$, respectively. In contrast, the VBK code requires a finite field \mathbf{F}_q with size $q > 6\lceil \frac{n}{2} \rceil + 1$, $18\lceil \frac{n}{3} \rceil + 1$, and $18\lceil \frac{n}{4} \rceil + 1$, respectively. However, it should be noted that when $d \in \{k + 2, k + 3\}$, the VBK code has a smaller sub-packetization level than that of the new code C_1 .

- iii) C_1 has the same sub-packetization level as that of the LLT code in [21] and ZZ code in [32]. However, the (n, k) LLT code does not work for d = n 1 and requires a larger finite field than C_1 , while the ZZ code requires a larger finite field than C_1 when d > k + 1.
- iv) The new MSR code C_1 subsumes the YB code 1 in [18] and the CB code in [20] as subcodes, i.e., YB code 1 and CB code can be obtained by shortening the new code C_1 .

Table III provides the details of the comparison between the proposed MDS array code C_2 and existing ones with $(1 + \epsilon)$ -optimal repair bandwidth and repair degree d = n - 1. Figure 2 provides an additional example of the comparison of sub-packetization levels among the codes listed in Table III, with the exception of RTGE code 2 in [26]. This code relies on the existence of an error-correcting code with specific parameters, which may not always be available.

TABLE III

A comparison of the key parameters among the NeW (n = sn', k) MDS array code C_2 and existing ones with $(1 + \epsilon)$ -optimal repair bandwidth and repair degree d = n - 1, where $\epsilon = \frac{(s-1)(r-1)}{n-1}$ and r = n - k.

	Sub-packatization N	Field size	Repair bandwidth
RTGE code 1 in [26]	$r^{\lceil \frac{n'}{r}\rceil}$	$q > n^{(r-1)N+1}$	$(1+\epsilon)\gamma_{\rm optimal}$
RTGE code 2 in [26]	$O(r^{r\tau} \log n)$	O(n)	$\leq (1 + \frac{1}{\tau})\gamma_{\text{optimal}}$
MDS code C_1 in [29]	$r^{n'}$	$q > rn' \lceil \frac{s}{r} \rceil, r \mid (q-1)$ (i.e., $O(n)$)	$(1+\epsilon)\gamma_{\rm optimal}$
MDS code C_2 in [29]	$r^{n'-1}$	$q > r \lceil \frac{n'}{r} \rceil (s-1) + n' \text{ (i.e., } O(n))$	$(1+\epsilon)\gamma_{\rm optimal}$
MDS code C_4 in [29]	$r^{\lceil \frac{n'}{r+1}\rceil}$	$q > \left\lceil \frac{2n}{3} \right\rceil, \qquad \text{if } r = 2$	$(1+\epsilon)\gamma_{ m optimal}$
		q > N(r-1) + 1, if r > 2	
MDS code C_5 in [29]	$r^{n'}$	$q > rn' \lceil \frac{s}{r} \rceil$ i.e., $(O(n))$	$(1+\epsilon)\gamma_{ m optimal}$
New MDS code C_2	$r^{\lceil \frac{n'}{2} \rceil}$	$q > sr \lceil \frac{n'}{2} \rceil$ (i.e., $O(sn/2)$)	$(1+\epsilon)\gamma_{\rm optimal}$



Fig. 2. Comparison of the sub-packetization level among the new (n, k) MSR code C_1 and some known ones with r = 3.

From Table III and Figure 2, we can see that the proposed MDS array code C_2 has the following advantages compared to existing ones:

- Under the same repair bandwidth, the new MDS code C_2 has a much smaller sub-packetization level when compared to the MDS array codes C_1 , C_2 , and C_5 in [29].
- By noting that the MDS array code C₄ in [29] is implicit when r > 2, we have that among all the explicit MDS array codes with (1 + ε)-optimal repair bandwidth and r > 2, the new MDS code C₂ has the smallest sub-packetization level under the same code parameters except for the RTGE code 1 in [26], which requires a super large finite field.

VI. CONCLUSION

In this paper, we proposed a new (n, k) MSR code construction that works for any repair degree d > k. The new MSR code has a smaller sub-packetization level or finite field than existing ones. Additionally, we obtained a new (n, k) MDS array code with a small sub-packetization level, $(1 + \epsilon)$ -optimal repair bandwidth, and repair degree d = n - 1, which outperforms existing ones in terms of the sub-packetization level or the field size. For (n, k) MDS array code with small sub-packetization level, $(1 + \epsilon)$ -optimal repair bandwidth, and repair bandwidth, and repair degree d < n - 1,

few results have been reported in the literature. To the best of our knowledge, the only one is the construction in [30], which only works for very large parameters n, k and requires a huge finite field, thus it is infeasible to be implemented in practical systems. Constructions of (n, k)MDS array code over small finite fields with small sub-packetization level, $(1+\epsilon)$ -optimal repair bandwidth, and repair degree d < n - 1 will be left for our future research.

APPENDIX A

PROOF OF LEMMA 1

For $i \in [0:rN)$, let e_i be row i of the identity matrix of order rN. Then define an $rN \times rN$ permutation matrix Ψ as

$$\Psi = (e_0^{\top}, e_N^{\top}, \dots, e_{(r-1)N}^{\top}, e_1^{\top}, e_{1+N}^{\top}, \dots, e_{1+(r-1)N}^{\top}, \dots, e_{N-1}^{\top}, e_{N-1+N}^{\top}, \dots, e_{N-1+(r-1)N}^{\top})^{\top},$$

where \top denotes the transpose operator.

Multiplying matrices Ψ and Ψ^{\top} on the left and right sides of matrix $B = (B_{t,i})_{t \in [0:r), i \in [0:r)}$, respectively, we then have

$$\Psi B \Psi^{\top} = \begin{pmatrix} B'_{0,0} & B'_{0,1} & \cdots & B'_{0,N-1} \\ B'_{1,0} & B'_{1,1} & \cdots & B'_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ B'_{N-1,0} & B'_{N-1,1} & \cdots & B'_{N-1,N-1} \end{pmatrix},$$

where

$$B'_{a,b} = \begin{pmatrix} B_{0,0}[a,b] & B_{0,1}[a,b] & \cdots & B_{0,r-1}[a,b] \\ B_{1,0}[a,b] & B_{1,1}[a,b] & \cdots & B_{1,r-1}[a,b] \\ \vdots & \vdots & \ddots & \vdots \\ B_{r-1,0}[a,b] & B_{r-1,1}[a,b] & \cdots & B_{r-1,r-1}[a,b] \end{pmatrix}, a, b \in [0:N).$$

May 30, 2023

By (2), we have $B'_{a,b} = \mathbf{0}$ for $0 \le b < a < N$. For $a \in [0:N)$, by i), we have

$$B_{a,a}' = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ B_{1,0}[a,a] & B_{1,1}[a,a] & \cdots & B_{1,r-1}[a,a] \\ \vdots & \vdots & \ddots & \vdots \\ B_{1,0}^{r-1}[a,a] & B_{1,1}^{r-1}[a,a] & \cdots & B_{1,r-1}^{r-1}[a,a] \end{pmatrix},$$
(32)

L

which is a Vandermonde matrix and is non-singular according to ii). Therefore,

I.

$$|\Psi||B||\Psi^{\top}| = |\Psi B \Psi^{\top}| = \begin{vmatrix} B'_{0,0} & B'_{0,1} & \cdots & B'_{0,N-1} \\ & B'_{1,1} & \cdots & B'_{1,N-1} \\ & & \ddots & \vdots \\ & & & B'_{N-1,N-1} \end{vmatrix} = \prod_{a=0}^{N-1} |B'_{a,a}| \neq 0,$$

which implies that B is non-singular.

APPENDIX B

PROOF OF LEMMA 2

Proof: Hereafter we only check the case of $i \in [0:m)$ since the other case can be proved similarly. For any given $a = (a_0, a_1, \ldots, a_{m-2}) \in [0:N/w)$, $u \in [0:w)$, and $j \in [0:n)$, according to (12), we have

$$V_{i,u}[a,:]A_{t,j} = \begin{cases} e_{g_{i,u}(a)} \left(\sum_{b=0}^{N-1} \lambda_{j,b_j}^t e_b^\top e_b + \sum_{b=0,b_j=0}^{N-1} \sum_{v=1}^{w-1} (\lambda_{j,0}^t - \lambda_{j,v}^t) e_b^\top e_{b(j,v)} \right), & \text{if } j \in [0:m), \\ e_{g_{i,u}(a)} \left(\sum_{b=0}^{N-1} \lambda_{j,b_{j-m}}^t e_b^\top e_b \right), & \text{if } j \in [m:n), \end{cases}$$

$$= \begin{cases} \lambda_{j,0}^t e_{g_{i,u}(a)} + \sum_{v=1}^{w-1} (\lambda_{j,0}^t - \lambda_{j,v}^t) e_{(g_{i,u}(a))(j,v)}, & \text{if } j \in [0:m) \text{ and } (g_{i,u}(a))_j = 0, \\ \lambda_{j,(g_{i,u}(a))_j}^t e_{g_{i,u}(a)}, & \text{if } j \in [0:m) \text{ and } (g_{i,u}(a))_j \neq 0, \\ \lambda_{j,(g_{i,u}(a))_{j-m}}^t e_{g_{i,u}(a)}, & \text{if } j \in [m:n). \end{cases}$$

$$(33)$$

where the two equalities follow from (11) and (4), respectively.

DRAFT

By (8) and (9), we have $(g_{i,u}(a))_i = u$ and $e_{g_{i,u}(a)(i,v)} = e_{g_{i,v}(a)}$. Then by (11) and (33), when $j \equiv i \mod m$, i.e., j = i or j = i + m, we have

$$V_{i,u}[a,:]A_{t,j} = \begin{cases} \lambda_{i,0}^t V_{i,0}[a,:] + \sum_{v=1}^{w-1} (\lambda_{i,0}^t - \lambda_{i,v}^t) V_{i,v}[a,:], & \text{if } j = i, u = 0, \\ \\ \lambda_{j,u}^t V_{i,u}[a,:], & \text{if } j = i, u \neq 0 \text{ or } j = i + m. \end{cases}$$

That is, for $j \equiv i \mod m$ and $u \in [0:w)$, we have

$$V_{i,u}A_{t,j} = \begin{cases} \lambda_{i,0}^t V_{i,0} + \sum_{v=1}^{w-1} (\lambda_{i,0}^t - \lambda_{i,v}^t) V_{i,v}, & \text{if } j = i, u = 0, \\ \lambda_{j,u}^t V_{i,u}, & \text{if } j = i, u \neq 0 \text{ or } j = i + m. \end{cases}$$

which together with (13) implies

$$S_{i,t}A_{t,i} = V_{i,0}A_{t,i} = \lambda_{i,0}^t V_{i,0} + (\lambda_{i,0}^t - \lambda_{i,1}^t) V_{i,1} + \dots + (\lambda_{i,0}^t - \lambda_{i,w-1}^t) V_{i,w-1}$$

and $S_{i,t}A_{t,j} = \lambda_{j,0}^t R_{i,j}$ for j = i + m, i.e., i) is true and ii) holds for $i, j \in [0:n)$ with $j \neq i$ and $j \equiv i \mod m$.

Next, we prove that ii) holds for $j \not\equiv i \mod m$. Recall that we only check the case of $i \in [0:m)$, which is discussed in the following four cases.

Case 1. If $j \in [0:i)$, by applying (8) and (9) to (33), we then have

$$\begin{aligned} V_{i,u}[a,:] \cdot A_{t,j} \\ &= \begin{cases} \lambda_{j,0}^{t} e_{g_{i,u}(a)} + \sum_{v=1}^{w-1} (\lambda_{j,0}^{t} - \lambda_{j,v}^{t}) e_{g_{i,u}(a(j,v))}, & \text{if } a_{j} = 0, \\ \lambda_{j,a,j}^{t} e_{g_{i,u}(a)}, & \text{otherwise}, \end{cases} \\ &= \begin{cases} \left(\lambda_{j,0}^{t} e_{a}^{(N/w)} + \sum_{v=1}^{w-1} (\lambda_{j,0}^{t} - \lambda_{j,v}^{t}) e_{a(j,v)}^{(N/w)}\right) \cdot \sum_{b=0}^{N/w-1} (e_{b}^{(N/w)})^{\top} e_{g_{i,u}(b)}, & \text{if } a_{j} = 0, \\ \lambda_{j,a,j}^{t} e_{a}^{(N/w)} \cdot \sum_{b=0}^{N/w-1} (e_{b}^{(N/w)})^{\top} e_{g_{i,u}(b)}, & \text{otherwise}, \end{cases} \\ &= e_{a}^{(N/w)} \left(\sum_{b=0}^{N/w-1} \lambda_{j,b_{j}}^{t} (e_{b}^{(N/w)})^{\top} e_{b}^{(N/w)} + \sum_{b=0,b_{j}=0}^{N/w-1} \sum_{v=1}^{w-1} (\lambda_{j,0}^{t} - \lambda_{j,v}^{t}) (e_{b}^{(N/w)})^{\top} e_{b(j,v)}^{(N/w)} \right) V_{i,u} \\ &= e_{a}^{(N/w)} B_{t,j,i} \cdot V_{i,u} \end{aligned}$$

$$(34)$$

May 30, 2023

30

where the second, third, and fourth equalities follow from (4), (10), and (21), respectively. Applying (13) to (34), we have $S_{i,t}A_{t,j} = B_{t,j,i}R_{i,j}$, which finishes the proof of this case. *Case 2.* If $j \in [i + 1 : m)$, similar to the proof of Case 1, we also have $S_{i,t}A_{t,j} = B_{t,j,i}R_{i,j}$.

Case 3. If $j \in [m:m+i)$, then $(g_{i,u}(a))_{j-m} = a_{j-m}$ by (8). By (33), we have

$$V_{i,u}[a,:] \cdot A_{t,j} = \lambda_{j,a_{j-m}}^{t} e_{g_{i,u}(a)}$$

$$= \lambda_{j,a_{j-m}}^{t} e_{a}^{(N/w)} \cdot \sum_{b=0}^{N/w-1} (e_{b}^{(N/w)})^{\top} e_{g_{i,u}(b)}$$

$$= e_{a}^{(N/w)} (\sum_{b=0}^{N/w-1} \lambda_{j,b_{j-m}}^{t} (e_{b}^{(N/w)})^{\top} e_{b}^{(N/w)}) V_{i,u}$$

$$= e_{a}^{(N/w)} B_{t,j,i} \cdot V_{i,u}$$

$$= B_{t,j,i}[a,:] \cdot V_{i,u}$$
(35)

where the second, third, and fourth equalities follow from (4), (10), and (21), respectively. Thus we have $S_{i,t}A_{t,j} = B_{t,j,i}R_{i,j}$ by combining (13) and (35).

Case 4. If $j \in [m+i+1:n)$, similar to the proof of Case 3, we also have $S_{i,t}A_{t,j} = B_{t,j,i}R_{i,j}$. Collecting the above four cases, we can derive that ii) holds for $0 \le i \ne j < n$ with $i \ne j \mod m$. While the proof of iii) is similar to the analysis in (19); thus, we omit it.

REFERENCES

- A. G. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.
- [2] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.
- [3] T. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inform. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.
- [4] D.S. Papailiopoulos, A.G. Dimakis, and V.R. Cadambe, "Repair optimal erasure codes through hadamard designs," *IEEE Trans. Inform. Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.
- [5] C. Tian, "Characterizing the rate region of the (4, 3, 3) exact-repair regenerating codes," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 967–975, Dec. 2014.

- [6] J. Li, X. Tang, and U. Parampalli, "A framework of constructions of minimal storage regenerating codes with the optimal access/update property," *IEEE Trans. Inform. Theory*, vol. 61, no. 4, pp. 1920–1932, Apr. 2015.
- [7] Y. S. Han, H.-T. Pai, R. Zheng, P. K. Varshney "Update-efficient error-correcting product-matrix codes," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 1925–1938, June 2015.
- [8] X. Tang, B. Yang, J. Li, and H.D.L. Hollmann, "A new repair strategy for the Hadamard minimum storage regenerating codes for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 61, no. 10, pp. 5271–5279, Oct. 2015.
- [9] Z. Wang, T. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Trans. Inform. Theory*, vol. 62, no. 8, pp. 4466–4480, Aug. 2016.
- [10] B. Sasidharan, M. Vajha, and P.V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair," arXiv: 1607.07335 [cs.IT]
- [11] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Trans. Inform. Theory*, vol. 63, no. 10, pp. 6318–6328, Oct. 2017.
- [12] M. Ye and A. Barg, "Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization," *IEEE Trans. Inform. Theory*, vol. 63, no. 10, pp. 6307–6317, Oct. 2017.
- [13] J. Li, X. Tang, and C. Tian, "A generic transformation for optimal repair bandwidth and rebuilding access in MDS codes," in Proc. IEEE Int. Symp. Inf. Theory (ISIT), June 2017, pp. 1623–1627.
- [14] J. Li, X. Tang, and C. Tian, "A generic transformation to enable optimal repair in MDS codes for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 64, no. 9, pp. 6257–6267, Sept. 2018.
- [15] H. Hou and P.P.C. Lee, "Binary MDS array codes with optimal repair," *IEEE Trans. Inform. Theory*, vol. 66, no. 3, pp. 1405–1422, Mar. 2020.
- [16] M. Elyasi and S. Mohajer, "Cascade codes for distributed storage systems," *IEEE Trans. Inform. Theory*, vol. 66, no. 12, pp. 7490–7527, Dec. 2020.
- [17] J. Li, X. Tang, and C. Hollanti, "A generic transformation for optimal node repair in MDS array codes over F2," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 727–738, Feb. 2022.
- [18] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Inform. Theory*, vol. 63, no. 4, pp. 2001–2014, Apr. 2017.
- [19] M. Vajha, S.B. Balaji, and P. V. Kumar, "Small-d MSR codes with optimal access optimal sub-packetization and linear field size," arXiv: 1804.00598 [cs.IT], 2021.
- [20] Z. Chen and A. Barg, "Explicit constructions of MSR codes for clustered distributed storage: The rack-aware storage model," *IEEE Trans. Inform. Theory*, vol. 66, no. 2, pp. 886–899, Feb. 2020.
- [21] Y. Liu, J. Li, and X. Tang, "A generic transformation to generate MDS array codes with δ -optimal access property," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 759–768, Feb. 2022.
- [22] N. Wang, G. Li, S. Hu, and M. Ye, "Constructing MSR codes with subpacketization $2^{n/3}$ for k + 1 helper nodes," *IEEE Trans. Inform. Theory*, doi: 10.1109/TIT.2023.3238759, 2023.

- [23] G. Li, N. Wang, S. Hu, and M. Ye, "MSR codes with linear field size and smallest sub-packetization for any number of helper nodes," arXiv: 2303.10467 [cs.IT], 2023.
- [24] Y. Liu, J. Li, and X. Tang, "Optimal repair/access MDS array codes with multiple repair degrees," *IEEE Trans. Inform. Theory*, doi: 10.1109/TIT.2023.3248288, 2023.
- [25] K. Mahdaviani, S. Mohajer, and A. Khisti, "Product matrix MSR codes with bandwidth adaptive exact repair," *IEEE Trans. Inform. Theory*, vol. 64, no. 4, pp. 3121–3135, Apr. 2018.
- [26] A. S. Rawat, I. Tamo, V. Guruswami, and K. Efremenko, "MDS code constructions with small sub-packetization and near-optimal repair bandwidth," *IEEE Trans. Inform. Theory*, vol. 64, no. 10, pp. 6506–6525, Oct. 2018.
- [27] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes," in Proc. IEEE Information Theory and Applications Workshop (ITA), Jan.–Feb. 2016, pp. 1–6.
- [28] S. Goparaju, A. Fazeli, and A. Vardy, "Minimum storage regenerating codes for all parameters," *IEEE Trans. Inform. Theory*, vol. 63, no. 10, pp. 6318–6328, Oct. 2017.
- [29] J. Li, Y. Liu, and X. Tang, "A systematic construction of MDS codes with small sub-packetization level and near-optimal repair bandwidth," *IEEE Trans. Inform. Theory*, vol. 67, no. 4, pp. 2162–2180, Apr. 2021.
- [30] V. Guruswami, S. V. Lokam, and S. V. M. Jayaraman, "€-MSR codes: Contacting fewer code blocks for exact repair," *IEEE Trans. Inform. Theory*, vol. 66, no. 11, pp. 6749–6761, Nov. 2020.
- [31] J. Li, X. Tang, H. Hou, Y.S. Han, B. Bai, and G. Zhang, "PMDS array codes with small sub-packetization, small repair bandwidth/rebuilding access," *IEEE Trans. Inform. Theory*, vol. 69, no. 3, pp. 1551–1566, Mar. 2023.
- [32] Z. Zhang and L. Zhou, "A vertical-horizontal framework for building rack-aware regenerating codes," *IEEE Trans. Inform. Theory*, vol. 69, no. 5, pp. 2874–2885, May 2023.