# MDS Array Codes With (Near) Optimal Repair 

# Bandwidth for All Admissible Repair Degree 

Jie Li, Member, IEEE, Yi Liu, Member, IEEE,<br>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 subpacketization 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].

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., $N \geq r^{\frac{n}{r+1}}$, 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 $n w+\sum_{t=1}^{w+1}\binom{w+1}{t} \frac{w t(t-1)}{2}$. 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+ $1+$-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 $\left(1+\frac{1}{\tau}\right)$ times the optimal repair bandwidth, where $\tau$ is an integer and $1 \leq \tau<\left\lceil\frac{n}{r}\right\rceil$. However, this code is constructed over a significantly large finite field $\mathbf{F}_{q}$, i.e., $q \geq n^{(r-1) N+1}$, 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 $k<d \leq n-1$, where its sub-packetization level is $w^{\left\lceil\frac{n}{2}\right\rceil}$ with $w=d-k+1$. The required field size $q$ is a prime power such that $q>\left\lceil\frac{n}{2}\right\rceil(w+2)$ if $w=2, q>\left\lceil\frac{n}{2}\right\rceil(w+1)$ if $2<w<r$, and $q>\left\lceil\frac{n}{2}\right\rceil w$ if $w=r$. Compared with existing constructions, the new MSR code $\mathcal{C}_{1}$ has advantages in terms of either the sub-packetization level or the field size. Please refer to Tables III and III for more details.

Furthermore, the MSR code $\mathcal{C}_{1}$ 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 $\mathcal{C}_{2}$ with a small subpacketization level of $r^{\left\lceil\frac{n}{2 s}\right\rceil}$ and repair degree $d=n-1$, where $r=n-k$, $s$ is any factor of $n$, and the require field size is $q>s r\left\lceil\frac{n}{2 s}\right\rceil$. The sub-packetization level or finite field size of $\mathcal{C}_{2}$ 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 $\mathcal{C}_{1}$ is presented in Section III. Section IV gives an MDS array code $\mathcal{C}_{2}$ with a small sub-packetization level. Section $\nabla$ 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 $a, b \geq 0$.

## A. $(n, k)$ Array Codes

Let $\mathbf{f}_{0}, \mathbf{f}_{1}, \ldots, \mathbf{f}_{n-1}$ 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:

$$
\underbrace{\left(\begin{array}{cccc}
A_{0,0} & A_{0,1} & \cdots & A_{0, n-1}  \tag{1}\\
A_{1,0} & A_{1,1} & \cdots & A_{1, n-1} \\
\vdots & \vdots & \ddots & \vdots \\
A_{r-1,0} & A_{r-1,1} & \cdots & A_{r-1, n-1}
\end{array}\right)}_{A}\left(\begin{array}{c}
\mathbf{f}_{0} \\
\mathbf{f}_{1} \\
\vdots \\
\mathbf{f}_{n-1}
\end{array}\right)=\mathbf{0}_{r N},
$$

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

$$
\begin{equation*}
B_{t, i}[a, b]=0 \text { for } 0 \leq b<a<N \tag{2}
\end{equation*}
$$

then the block matrix $B=\left(B_{t, i}\right)_{t \in[0: r), i \in[0: r)}$ is non-singular if
i) $B_{t, i}[a, a]=\left(B_{1, i}[a, a]\right)^{t}$ 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 $[0: n) \backslash\{i\}$, which denotes the set of indices of the helper nodes, and let $L_{i}=[0: n) \backslash\left(H_{i} \cup\{i\}\right)$ 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 $\delta$ ( $N / r \leq \delta \leq$ $N$ ) linearly independent equations are acquired from each of the $r$ parity-check groups, where these $\delta$ linear independent equations are linear combinations of the corresponding $N$ parity-check equations in a parity-check group. Precisely, the $\delta$ 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:

$$
\underbrace{\left(\begin{array}{c}
S_{i, 0} A_{0, i}  \tag{3}\\
S_{i, 1} A_{1, i} \\
\vdots \\
S_{i, r-1} A_{r-1, i}
\end{array}\right)}_{\text {useful data }} \mathbf{f}_{i}+\sum_{l \in L_{i}} \underbrace{\left(\begin{array}{c}
S_{i, 0} A_{0, l} \\
S_{i, 1} A_{1, l} \\
\vdots \\
S_{i, r-1} A_{r-1, l}
\end{array}\right)}_{\text {unknown data by } \mathbf{f}_{l}} \mathbf{f}_{l}+\sum_{j \in H_{i}} \underbrace{\left(\begin{array}{c}
S_{i, 0} A_{0, j} \\
S_{i, 1} A_{1, j} \\
\vdots \\
S_{i, r-1} A_{r-1, j}
\end{array}\right)}_{\text {interference by } \mathbf{f}_{j}} \mathbf{f}_{j}=0
$$

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 $\gamma_{i}=\sum_{j \in H_{i}} \operatorname{rank}\left(R_{i, j}\right)$. 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 $\operatorname{rank}\left(R_{i, j}\right)=\frac{N}{d-k+1}$ 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 $\left\{e_{0}, \ldots, e_{N-1}\right\}$

Assuming that $N=w^{m}$ for two integers $w$ and $m$ with $w, m \geq 2$, 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.,

$$
e_{i}=(0, \ldots, 0,1,0, \ldots, 0), i \in\left[0: w^{m}\right),
$$

with only the $i$-th entry being nonzero.
Then for any $a, b \in[0: N)$, we have

$$
e_{a}\left(e_{b}\right)^{\top}= \begin{cases}1, & \text { if } a=b  \tag{4}\\ 0, & \text { otherwise }\end{cases}
$$

where $T$ represents the transpose operator.
In [29], a class of special partition sets of $\left\{e_{0}, \ldots, e_{w^{m}-1}\right\}$ is given for $w \geq 2$. 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 $a \in\left[0: w^{m}\right)$, denote by $\left(a_{0}, \ldots, a_{m-1}\right)$ 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 $a=$ $\left(a_{0}, \ldots, a_{m-1}\right)$. For $i \in[0: m)$ and $t \in[0: w)$, define a subset of $\left\{e_{0}, \ldots, e_{w^{m}-1}\right\}$ as

$$
\begin{equation*}
V_{i, t}=\left\{e_{a} \mid a_{i}=t, 0 \leq a<w^{m}\right\}, \tag{5}
\end{equation*}
$$

where $a_{i}$ is the $i$-th element in the $w$-ary expansion of $a$.
Obviously, $\left|V_{i, t}\right|=w^{m-1}$, and $V_{i, 0}, V_{i, 1}, \ldots, V_{i, w-1}$ is a partition of the set $\left\{e_{0}, \ldots, e_{w^{m}-1}\right\}$ 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 $\left\{e_{0}, \ldots, e_{w^{m}-1}\right\}$ DEFINED BY 5 FOR $m=3, w=2$, AND $m=2$, $w=3$, RESPECTIVELY.

| $i$ | 0 | 1 | 2 | $i$ | 0 | 1 | 2 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $e_{0}$ | $e_{0}$ | $e_{0}$ |  | $e_{4}$ | $e_{2}$ | $e_{1}$ |
| $V_{i, 0}$ | $e_{1}$ | $e_{1}$ | $e_{2}$ |  | $e_{5}$ | $e_{3}$ | $e_{3}$ |
|  | $e_{2}$ | $e_{4}$ | $e_{4}$ |  | $e_{6}$ | $e_{6}$ | $e_{5}$ |
|  | $e_{3}$ | $e_{5}$ | $e_{6}$ |  | $e_{7}$ | $e_{7}$ | $e_{7}$ |

(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 $4 \times 8$ 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 $a=\left(a_{0}, \ldots, a_{m-1}\right) \in[0: N), i \in[0: m)$ and $u \in[0: w)$, define $a(i, u)$ as

$$
\begin{equation*}
a(i, u)=\left(a_{0}, \ldots, a_{i-1}, u, a_{i+1}, \ldots, a_{m-1}\right) \tag{6}
\end{equation*}
$$

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

For $a=\left(a_{0}, a_{1}, \ldots, a_{m-2}\right) \in[0: N / w)$ and $i \in[0: m)$, define

$$
\begin{equation*}
g_{i, u}(a)=\left(a_{0}, a_{1}, \ldots, a_{i-1}, u, a_{i}, \ldots, a_{m-2}\right), \tag{7}
\end{equation*}
$$

i.e., inserting $u$ to the $i$-th digit of $\left(a_{0}, a_{1}, \ldots, a_{m-2}\right)$. Then for $i, j \in[0: m)$ and $u, v \in[0: w)$,
we have that the $j$-th digit of $g_{i, u}(a)$ is

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

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

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

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

$$
\begin{equation*}
V_{i, u}=\sum_{a=0}^{N / w-1}\left(e_{a}^{(N / w)}\right)^{\top} e_{g_{i, u}(a)}, u \in[0: w), \tag{10}
\end{equation*}
$$

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

$$
\begin{equation*}
V_{i, u}[a,:]=e_{g_{i, u}(a)}, 0 \leq u<w, a \in[0: N / w) . \tag{11}
\end{equation*}
$$

III. A NEW $(n, k)$ MSR $\operatorname{code} \mathcal{C}_{1}$ WITH REPAIR DEGREE $k<d<n$

In this section, we propose an $(n=2 m, k=n-r)$ MSR code construction $\mathcal{C}_{1}$ 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 $\mathcal{C}_{1}$ 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 $\mathcal{C}_{1}$ 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 $2 \leq w \leq r$, we define the parity-check matrix $\left(A_{t, i}\right)_{t \in[0: r), i \in[0: n)}$ of the $(n=2 m, k=n-r)$ array code $\mathcal{C}_{1}$ over $\mathbf{F}_{q}$ as

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

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

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

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

$$
\begin{equation*}
V_{i, u}=V_{i-m, u}, \text { for } i \in[m, n), u \in[0: w) \tag{14}
\end{equation*}
$$

for convenience of notation.

In what follows, we first give an example to show the connection between the new code and the YB code $1, \mathrm{CB}$ 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 $\left(A_{t, i}\right)_{t \in[0: 3), i \in[0: 12)}$ be the parity-check matrix of the $(12,9)$ code $\mathcal{C}_{1}$, then $\left(A_{t, i}\right)_{t \in[0: 3), i \in[0: 6)}$ is exactly the parity-check matrix of the $(6,3) Y B$ code 1 in [18] while $\left(A_{t, i}\right)_{t \in[0: 3), i \in[6: 12)}$ is exactly the parity-check matrix of the $(6,3)$ CB code in [20].

Example 2. An example of the $(6,3) \mathrm{MSR}$ code $\mathcal{C}_{1}$ 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
$\left(A_{t, i}\right)_{t \in[0: 3), i \in[0: 6)}$ is defined as

$$
\begin{aligned}
& A_{t, 0}=\left(\begin{array}{c}
\lambda_{0,0}^{t} e_{0}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{4} \\
\lambda_{0,0}^{t} e_{1}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{5} \\
\lambda_{0,0}^{t} e_{2}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{6} \\
\lambda_{0,0}^{t} e_{3}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) 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{array}\right), A_{t, 1}=\left(\begin{array}{c}
\lambda_{1,0}^{t} e_{0}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{2} \\
\lambda_{1,0}^{t} e_{1}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{3} \\
\lambda_{1,1}^{t} e_{2} \\
\lambda_{1,1}^{t} e_{3} \\
\lambda_{1,0}^{t} e_{4}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{6} \\
\lambda_{1,0}^{t} e_{5}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{7} \\
\lambda_{1,1}^{t} e_{6} \\
\lambda_{1,1}^{t} e_{7}
\end{array}\right), \\
& A_{t, 2}=\left(\begin{array}{c}
\lambda_{2,0}^{t} e_{0}+\left(\lambda_{2,0}^{t}-\lambda_{2,1}^{t}\right) e_{1} \\
\lambda_{2,1}^{t} e_{1} \\
\lambda_{2,0}^{t} e_{2}+\left(\lambda_{2,0}^{t}-\lambda_{2,1}^{t}\right) e_{3} \\
\lambda_{2,1}^{t} e_{3} \\
\lambda_{2,0}^{t} e_{4}+\left(\lambda_{2,0}^{t}-\lambda_{2,1}^{t}\right) e_{5} \\
\lambda_{2,1}^{t} e_{5} \\
\lambda_{2,0}^{t} e_{6}+\left(\lambda_{2,0}^{t}-\lambda_{2,1}^{t}\right) e_{7} \\
\lambda_{2,1}^{t} e_{7}
\end{array}\right),\left(\begin{array}{c}
\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_{6} \\
\lambda_{3,1}^{t} e_{7}
\end{array}\right), A_{t, 4}=\left(\begin{array}{c}
\lambda_{4,0}^{t} e_{0} \\
\lambda_{4,0}^{t} e_{1} \\
\lambda_{4,1}^{t} e_{2} \\
\lambda_{4,1}^{t} e_{3} \\
\lambda_{4,0}^{t} e_{4} \\
\lambda_{4,0}^{t} e_{5} \\
\lambda_{4,1}^{t} e_{6} \\
\lambda_{4,1}^{t} e_{7}
\end{array}\right), A_{t, 5}=\left(\begin{array}{c}
\lambda_{5,0}^{t} e_{0} \\
\lambda_{5,1}^{t} e_{1} \\
\lambda_{5,0}^{t} e_{2} \\
\lambda_{5,1}^{t} e_{3} \\
\lambda_{5,0}^{t} e_{4} \\
\lambda_{5,1}^{t} e_{5} \\
\lambda_{5,0}^{t} e_{6} \\
\lambda_{5,1}^{t} e_{7}
\end{array}\right),
\end{aligned}
$$

where

$$
\begin{align*}
& \lambda_{0,0}=1, \lambda_{1,0}=c^{4}, \lambda_{2,0}=c^{8}, \lambda_{3,0}=c^{2}, \lambda_{4,0}=c^{6}, \lambda_{5,0}=c^{10}, \\
& \lambda_{0,1}=c, \lambda_{1,1}=c^{5}, \lambda_{2,1}=c^{9}, \lambda_{3,1}=c^{3}, \lambda_{4,1}=c^{7}, \lambda_{5,1}=c^{11} \tag{15}
\end{align*}
$$

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 $\left(V_{0,0}+V_{0,1}\right) \mathbf{f}_{j}\left(\right.$ i.e., $\left(e_{0}+e_{4}\right) \mathbf{f}_{j},\left(e_{1}+e_{5}\right) \mathbf{f}_{j},\left(e_{2}+\right.$ $\left.\left.e_{6}\right) \mathbf{f}_{j},\left(e_{3}+e_{7}\right) \mathbf{f}_{j}\right)$ 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),
we have
$\left(\begin{array}{c}e_{0}+e_{4} \\ e_{1}+e_{5} \\ e_{2}+e_{6} \\ e_{3}+e_{7} \\ \lambda_{3,0} e_{0}+\lambda_{3,1} e_{4} \\ \lambda_{3,0} e_{1}+\lambda_{3,1} e_{5} \\ \lambda_{3,0} e_{2}+\lambda_{3,1} e_{6} \\ \lambda_{3,0} e_{3}+\lambda_{3,1} e_{7} \\ \lambda_{3,0}^{2} e_{0}+\lambda_{3,1}^{2} e_{4} \\ \lambda_{3,0}^{2} e_{1}+\lambda_{3,1}^{2} e_{5} \\ \lambda_{3,0}^{2} e_{2}+\lambda_{3,1}^{2} e_{6} \\ e_{1}^{2}+e_{5} e_{3}+\lambda_{3,1}^{2} e_{7} \\ e_{2}+e_{6} \\ e_{3}+e_{7} \\ \lambda_{0,0}\left(e_{0}+e_{4}\right) \\ \lambda_{0,0}\left(e_{1}+e_{5}\right) \\ \lambda_{0,0}\left(e_{2}+e_{6}\right) \\ \lambda_{0,0}\left(e_{3}+e_{7}\right) \\ \lambda_{0,0}^{2}\left(e_{0}+e_{4}\right) \\ \lambda_{0,0}^{2}\left(e_{1}+e_{5}\right) \\ \lambda_{0,0}^{2}\left(e_{2}+e_{6}\right) \\ \lambda_{0,0}^{2}\left(e_{3}+e_{7}\right)\end{array}\right)\left(\begin{array}{c}e_{0}+e_{4} \\ e_{1}+e_{5} \\ e_{2}+e_{6} \\ e_{3}+e_{7} \\ \mathbf{f}_{0} \\ \mathbf{f}_{3} \\ \lambda_{1,0}\left(e_{0}+e_{4}\right)+\left(\lambda_{1,0}-\lambda_{1,1}\right)\left(e_{2}+e_{6}\right) \\ \lambda_{1,0}\left(e_{1}+e_{5}\right)+\left(\lambda_{1,0}-\lambda_{1,1}\right)\left(e_{3}+e_{7}\right) \\ \lambda_{1,1}\left(e_{2}+e_{6}\right) \\ \lambda_{1,1}\left(e_{3}+e_{7}\right) \\ \lambda_{1,0}^{2}\left(e_{0}+e_{4}\right)+\left(\lambda_{1,0}^{2}-\lambda_{1,1}^{2}\right)\left(e_{2}+e_{6}\right) \\ \lambda_{1,0}^{2}\left(e_{1}+e_{5}\right)+\left(\lambda_{1,0}^{2}-\lambda_{1,1}^{2}\right)\left(e_{3}+e_{7}\right) \\ \lambda_{1,1}^{2}\left(e_{2}+e_{6}\right) \\ \lambda_{1,1}^{2}\left(e_{3}+e_{7}\right) \\ \mathbf{f}_{1} \\ ( \end{array}\right)$

$$
=-\left(\begin{array}{c}
e_{0}+e_{4}  \tag{16}\\
e_{1}+e_{5} \\
e_{2}+e_{6} \\
e_{3}+e_{7} \\
\lambda_{2,0}\left(e_{0}+e_{4}\right)+\left(\lambda_{2,0}-\lambda_{2,1}\right)\left(e_{1}+e_{5}\right) \\
\lambda_{2,1}\left(e_{1}+e_{5}\right) \\
\lambda_{2,0}\left(e_{2}+e_{6}\right)+\left(\lambda_{2,0}-\lambda_{2,1}\right)\left(e_{3}+e_{7}\right) \\
\lambda_{2,1}\left(e_{3}+e_{7}\right) \\
\lambda_{2,0}^{2}\left(e_{0}+e_{4}\right)+\left(\lambda_{2,0}^{2}-\lambda_{2,1}^{2}\right)\left(e_{1}+e_{5}\right) \\
\lambda_{2,1}^{2}\left(e_{1}+e_{5}\right) \\
\lambda_{2,0}^{2}\left(e_{2}+e_{6}\right)+\left(\lambda_{2,0}^{2}-\lambda_{2,1}^{2}\right)\left(e_{3}+e_{7}\right) \\
\lambda_{2,1}^{2}\left(e_{3}+e_{7}\right) \\
e_{2}+e_{6} \\
e_{3}+e_{7} \\
\lambda_{4,0}\left(e_{0}+e_{4}\right) \\
\lambda_{4,0}\left(e_{1}+e_{5}\right) \\
\lambda_{4,1}\left(e_{2}+e_{6}\right) \\
\lambda_{4,1}\left(e_{3}+e_{7}\right) \\
\lambda_{4,0}^{2}\left(e_{0}+e_{4}\right) \\
\lambda_{4,0}^{2}\left(e_{1}+e_{5}\right) \\
\lambda_{4,1}^{2}\left(e_{2}+e_{6}\right) \\
e_{4,1}^{2}\left(e_{3}+e_{7}\right)
\end{array}\right) \mathbf{f}_{4}-\left(\begin{array}{c}
e_{0}-e_{4} \\
e_{2}+e_{6} \\
e_{3}+e_{7} \\
\lambda_{5,0}\left(e_{0}+e_{4}\right) \\
\lambda_{5,1}\left(e_{1}+e_{5}\right) \\
\lambda_{5,0}\left(e_{2}+e_{6}\right) \\
\lambda_{5,1}\left(e_{3}+e_{7}\right) \\
\lambda_{5,0}^{2}\left(e_{0}+e_{4}\right) \\
\lambda_{5,1}^{2}\left(e_{1}+e_{5}\right) \\
\lambda_{5,0}^{2}\left(e_{2}+e_{6}\right) \\
\lambda_{5,1}^{2}\left(e_{3}+e_{7}\right)
\end{array}\right) \mathbf{f}_{5},
$$

which can be reformulated as

$$
\underbrace{\left(\begin{array}{ccc}
I_{4} & I_{4} & I_{4}  \tag{17}\\
\lambda_{3,0} I_{4} & \lambda_{3,1} I_{4} & \lambda_{0,0} I_{4} \\
\lambda_{3,0}^{2} I_{4} & \lambda_{3,1}^{2} I_{4} & \lambda_{0,0}^{2} I_{4}
\end{array}\right)}_{M}\left(\begin{array}{c}
V_{0,0} \mathbf{f}_{3} \\
V_{0,1} \mathbf{f}_{3} \\
\left(V_{0,0}+V_{0,1}\right) \mathbf{f}_{0}
\end{array}\right)=\kappa_{*},
$$

where $\kappa_{*}$ 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 $\mathcal{C}_{1}$.

Theorem 1. The new code $\mathcal{C}_{1}$ 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 $i \not \equiv j \bmod m$,
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

$$
\left(\begin{array}{cccc}
A_{0, j_{0}} & A_{0, j_{1}} & \cdots & A_{0, j_{r-1}}  \tag{18}\\
A_{1, j_{0}} & A_{1, j_{1}} & \cdots & A_{1, j_{r-1}} \\
\vdots & \vdots & \vdots & \vdots \\
A_{r-1, j_{0}} & A_{r-1, j_{1}} & \cdots & A_{r-1, j_{r-1}}
\end{array}\right)
$$

is non-singular over $\mathbf{F}_{q}$.

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

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

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

$$
\begin{equation*}
A_{t, i}[a, a]=\lambda_{i, a_{i} \%_{m}}^{t}=\left(A_{1, i}[a, a]\right)^{t} \text { for } a \in[0: N), \tag{20}
\end{equation*}
$$

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 $0 \leq i<j<n$, by 20), we have

$$
A_{1, i}[a, a]-A_{1, j}[a, a]= \begin{cases}\lambda_{i, a_{i} \vartheta_{m} m}-\lambda_{j, a_{j \% m}}, & \text { if } i \not \equiv j \bmod m, \\ \lambda_{i, a_{i} \vartheta_{m} m}-\lambda_{j, a_{i \% m}}, & \text { otherwise },\end{cases}
$$

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.

Lemma 2. For any $i, j \in[0: n)$, rewrite them as $i=g_{0} m+i^{\prime}$ and $j=g_{1} m+j^{\prime}$ for $g_{0}, g_{1} \in\{0,1\}$ and $i^{\prime}, j^{\prime} \in[0: m)$. Then for $t \in[0: r)$, we have
i) $S_{i, t} A_{t, i}= \begin{cases}\lambda_{i, 0}^{t} V_{i, 0}+\left(\lambda_{i, 0}^{t}-\lambda_{i, 1}^{t}\right) V_{i, 1}+\cdots+\left(\lambda_{i, 0}^{t}-\lambda_{i, w-1}^{t}\right) V_{i, w-1}, & \text { if } i \in[0: m), \\ \lambda_{i, 0}^{t} V_{i, 0}+\lambda_{i, 1}^{t} V_{i, 1}+\cdots+\lambda_{i, w-1}^{t} V_{i, w-1}, & \text { if } i \in[m: 2 m),\end{cases}$
ii) $S_{i, t} A_{t, j}=B_{t, j, i} R_{i, j}$ for $j \neq i$, where $B_{t, j, i}$ is an $\frac{N}{w} \times \frac{N}{w}$ matrix define by

$$
B_{t, j, i}= \begin{cases}\sum_{a=0}^{N / w-1} \lambda_{j, a_{j}}^{t}\left(e_{a}^{(N / w)}\right)^{\top} e_{a}^{(N / w)}  \tag{21}\\ \quad+\sum_{a=0, a_{j}=0}^{N / w-1} \sum_{u=1}^{w-1}\left(\lambda_{j, 0}^{t}-\lambda_{j, u}^{t}\right)\left(e_{a}^{(N / w)}\right)^{\top} e_{a(j, u)}^{(N / w)}, & \text { if } j \in\left[0: i^{\prime}\right), \\ \sum_{a=0}^{N / w-1} \lambda_{j, a_{j-1}}^{t}\left(e_{a}^{(N / w)}\right)^{\top} e_{a}^{(N / w)} \\ +\sum_{a=0, a_{j-1}=0}^{N / w-1} \sum_{u=1}^{w-1}\left(\lambda_{j, 0}^{t}-\lambda_{j, u}^{t}\right)\left(e_{a}^{(N / w)}\right)^{\top} e_{a(j-1, u)}^{(N / w)}, & \text { if } j \in\left[i^{\prime}+1: m\right), \\ \sum_{a=0}^{N / w-1} \lambda_{j, a_{j-m}}^{t}\left(e_{a}^{(N / w)}\right)^{\top} e_{a}^{(N / w)}, & \text { if } j \in\left[m: m+i^{\prime}\right), \\ \sum_{a=0}^{N / w-1} \lambda_{j, a_{j-m-1}}^{t}\left(e_{a}^{(N / w)}\right)^{\top} e_{a}^{(N / w)}, & \text { if } j \in\left[m+i^{\prime}+1: n\right), \\ \lambda_{j, 0}^{t} I_{N / w}, & \text { if } j \equiv i \bmod m,\end{cases}
$$

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

$$
\begin{equation*}
B_{t, j, i}[a, b]=0 \text { for any } t \in[0, r), 0 \leq b<a<N / w \tag{22}
\end{equation*}
$$

and

$$
B_{t, j, i}[a, a]= \begin{cases}\lambda_{j, a_{j^{\prime}}}^{t}, & \text { if } j^{\prime}<i^{\prime}  \tag{23}\\ \lambda_{j, 0}^{t}, & \text { if } j^{\prime}=i^{\prime} \\ \lambda_{j, a_{j^{\prime}-1}}^{t}, & \text { if } j^{\prime}>i^{\prime}\end{cases}
$$

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 $\mathcal{C}_{1}$ 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 $i \not \equiv j \bmod m$,
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

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

Let $L_{i}=\left\{l_{0}, l_{1}, \ldots, l_{r-w-1}\right\}$, substituting them into the above equations, we then have

$$
\begin{align*}
& \underbrace{\text { B }}_{\left.\begin{array}{ccccccc}
I_{N / w} & 0_{N / w} & \cdots & 0_{N / w} & B_{0, l_{0}, i} & \cdots & B_{0, l_{r-w-1, i}} \\
\lambda_{i, 0} I_{N / w} & \left(\lambda_{i, 0}-\lambda_{i, 1}\right) I_{N / w} & \cdots & \left(\lambda_{i, 0}-\lambda_{i, w-1}\right) I_{N / w} & B_{1, l_{0}, i} & \cdots & B_{1, l_{r-w-1}, i} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{i, 0}^{r-1} I_{N / w} & \left(\lambda_{i, 0}^{r-1}-\lambda_{i, 1}^{r-1}\right) I_{N / w} & \cdots & \left(\lambda_{i, 0}^{r-1}-\lambda_{i, w-1}^{r-1}\right) I_{N / w} & B_{r-1, l_{0}, i} & \cdots & B_{r-1, l_{r-w-1, i}}
\end{array}\right)} \\
& \left(\begin{array}{c}
V_{i, 0} \mathbf{f}_{i} \\
\vdots \\
V_{i, w-1} \mathbf{f}_{i} \\
R_{i, l_{0}} \mathbf{f}_{l_{0}} \\
\vdots \\
R_{i, l_{r-w-1}} \mathbf{f}_{l_{r-w-1}}
\end{array}\right)=-\sum_{j \in H_{i}}\left(\begin{array}{c} 
\\
B_{0, j, i} \\
B_{1, j, i} \\
\vdots \\
B_{r-1, j, i}
\end{array}\right) R_{i, j} \mathbf{f}_{j} . \tag{25}
\end{align*}
$$

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

$$
B^{\prime}=\left(\begin{array}{ccccccc}
I_{N / w} & I_{N / w} & \cdots & I_{N / w} & B_{0, l_{0}, i} & \cdots & B_{0, l_{r-w-1}, i}  \tag{26}\\
\lambda_{i, 0} I_{N / w} & \lambda_{i, 1} I_{N / w} & \cdots & \lambda_{i, w-1} I_{N / w} & B_{1, l_{0}, i} & \cdots & B_{1, l_{r-w-1, i}} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\lambda_{i, 0}^{r-1} I_{N / w} & \lambda_{i, 1}^{r-1} I_{N / w} & \cdots & \lambda_{i, w-1}^{r-1} I_{N / w} & B_{r-1, l_{0}, i} & \cdots & B_{r-1, l_{r-w-1}, i}
\end{array}\right)
$$

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 $B_{t, l_{j}, i}[a, a]=\left(B_{1, l_{j}, i}[a, a]\right)^{t}$, 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 $B^{\prime}$ in (26) is non-singular if

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

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.,

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

are pairwise distinct for any $i \in[0: m)$ and $a \in[0: N)$ since $L_{i}$ is an arbitrary $(r-w)$-subset of $[0: n) \backslash\{i\}$, 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
$\lambda_{i, u}= \begin{cases}c^{i(w+2)+u}, & \text { if } w=2, \\ c^{i(w+1)+u}, & \text { if } w \in[3: r), \quad \lambda_{i+m, u}=\left\{\begin{array}{ll}c^{i(w+2)+w+u}, & \text { if } w=2, \\ c^{i(w+1)+w}, & \text { if } w \in[3: r), u=0, \\ c^{i w+u}, & \text { if } w=r, \\ c^{i(w+1)+u \%(w-1)+1}, & \text { if } w \in[3: r), u \geq 1, \\ c^{i w+(u+1) \% r}, & \text { if } w=r,\end{array}, l\right.\end{cases}$
for $i \in[0: m)$ and $u \in[0: w)$, where $c$ is a primitive element of $\mathbf{F}_{q}$ with

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

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

$$
0<\left|\left(j^{\prime}-i^{\prime}\right)(w+1)+s-t\right| \leq(m-1)(w+1)+w=m(w+1)-1<q-1
$$

Therefore, i) of Theorem 1 and also ii) of Theorem 2 are satisfied.
ii) For $i \in[0: m)$, by (27), we have

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

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 $(n, k) M S R$ code $\mathcal{C}_{1}$. If $2 \nmid n$, through shortening, one can easily obtain an $(n, k) M S R$ code with repair degree $d$ from an $(n+1, k+1)$ MSR code $\mathcal{C}_{1}$ with repair degree $d+1$ [2] Theorem 6].

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 $\left\{\lambda_{i, 0}, \lambda_{i, 1}, \ldots, \lambda_{i, w-1}\right\}$, which leads to a smaller finite field compared to the case $w<r$.

## IV. A NEW MDS ARRAY $\operatorname{Code} \mathcal{C}_{2}$ WITH SMALL SUb-PACKETIZATION LEVEL

In [29], a generic transformation that can transform any $\left(n^{\prime}, k^{\prime}\right)$ MSR code into a new ( $n=$ $\left.s n^{\prime}, k\right)$ MDS array code was proposed for any $s \geq 2$, 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 $\mathcal{C}_{1}$ 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 $\left(n^{\prime}, k^{\prime}\right)$ code $\mathcal{C}_{1}$ with $d^{\prime}=n^{\prime}-1$ in the previous section, we construct an $\left(n=s n^{\prime}, k\right)$ MDS array code $\mathcal{C}_{2}$ 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 $\mathcal{C}_{2}$ is constructed throught two steps as follows.

Step 1. Choosing the $\left(n^{\prime}, k^{\prime}\right) M S R$ code $\mathcal{C}_{1}$ with repair degree $d^{\prime}=n^{\prime}-1$ in Section III as the base code. Let $\left(A_{t, i^{\prime}}\right)_{t \in[0: r), i^{\prime} \in\left[0: n^{\prime}\right),}, S_{i^{\prime}, t}^{\prime}$, and $R_{i^{\prime}, j^{\prime}}^{\prime}$ denote its parity-check matrix, select matrices, and repair matrices, where $r=n^{\prime}-k^{\prime}, i^{\prime}, j^{\prime} \in\left[0: n^{\prime}\right), j^{\prime} \neq i^{\prime}$, and $t \in[0: r)$.

Step 2. Applying the generic transformation in [29] to the $\left(n^{\prime}, k^{\prime}\right) M S R$ code $\mathcal{C}_{1}$, then an $\left(n=s n^{\prime}, k\right)$ array code $\mathcal{C}_{2}$ with repair degree $d=n-1$ is obtained, where the parity-check matrix $\left(A_{t, i}\right)_{t \in[0: r), i \in[0: n)}$, select matrices $S_{i, t}$, and repair matrices $R_{i, j}$ are given as

$$
A_{t, i}=x_{t, i} A_{t, i \% n^{\prime}}^{\prime}, \quad S_{i, t}=S_{i \% n^{\prime}, t}^{\prime}, R_{i, j}= \begin{cases}R_{i \% n^{\prime}, j \% n^{\prime}}^{\prime}, & \text { if } j \not \equiv i \bmod n^{\prime}  \tag{28}\\ I, & \text { otherwise }\end{cases}
$$

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

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

Example 3. From Construction [] the parity-check matrix $\left(A_{t, i}\right)_{t \in[0: 2), i \in[0: 4)}$ of the $\left(n^{\prime}=4, k^{\prime}=2\right)$
MSR code $\mathcal{C}_{1}$ with $d^{\prime}=3$ and $N=\left(d^{\prime}-k^{\prime}+1\right)^{n^{\prime} / 2}=2^{2}$ is given as

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

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 $\mathcal{C}_{2}$ with $N=2^{2}$ and $d=7$, based on the $\left(n^{\prime}=4, k^{\prime}=2\right) M S R$ code $\mathcal{C}_{1}$ from Construction 1 The parity-check matrix $\left(A_{t, i}\right)_{t \in[0: 2), i \in[0: 8)}$ of the new MDS array code $\mathcal{C}_{2}$ is given as

$$
\begin{gathered}
A_{t, 0}=\left(\begin{array}{c}
\lambda_{0,0}^{t} e_{0}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{2} \\
\lambda_{0,0}^{t} e_{1}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{3} \\
\lambda_{0,1}^{t} e_{2} \\
\lambda_{0,1}^{t} e_{3}
\end{array}\right), A_{t, 1}=\left(\begin{array}{c}
\lambda_{1,0}^{t} e_{0}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{1} \\
\lambda_{1,1}^{t} e_{1} \\
\lambda_{1,0}^{t} e_{2}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{3} \\
\lambda_{1,1}^{t} e_{3}
\end{array}\right), A_{t, 2}=\left(\begin{array}{c}
\lambda_{2,0}^{t} e_{0} \\
\lambda_{2,0}^{t} e_{1} \\
\lambda_{2,1}^{t} e_{2} \\
\lambda_{2,1}^{t} e_{3}
\end{array}\right), \\
\left.A_{t, 3}=\left(\begin{array}{c}
\lambda_{3,0}^{t} e_{0} \\
\lambda_{3,1}^{t} e_{1} \\
\lambda_{3,0}^{t} e_{2} \\
\lambda_{3,1}^{t} e_{3}
\end{array}\right), A_{t, 4}=c^{4 t}\left(\begin{array}{c}
\lambda_{0,0}^{t} e_{0}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{2} \\
\lambda_{0,0}^{t} e_{1}+\left(\lambda_{0,0}^{t}-\lambda_{0,1}^{t}\right) e_{3} \\
\lambda_{0,1}^{t} e_{2} \\
\lambda_{0,1}^{t} e_{3}
\end{array}\right), A_{t, 5}^{\lambda_{1,0}^{t} e_{0}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{1}} \begin{array}{l}
\lambda_{1,1}^{t} e_{1} \\
\lambda_{1,0}^{t} e_{2}+\left(\lambda_{1,0}^{t}-\lambda_{1,1}^{t}\right) e_{3} \\
\lambda_{1,1}^{t} e_{3}
\end{array}\right), \\
A_{t, 6}=c^{4 t}\left(\begin{array}{c}
\lambda_{2,0}^{t} e_{0} \\
\lambda_{2,0}^{t} e_{1} \\
\lambda_{2,1}^{t} e_{2} \\
\lambda_{2,1}^{t} e_{3}
\end{array}\right), A_{t, 7}=c^{4 t}\left(\begin{array}{l}
\lambda_{3,0}^{t} e_{0} \\
\lambda_{3,1}^{t} e_{1} \\
\lambda_{3,0}^{t} e_{2} \\
\lambda_{3,1}^{t} e_{3}
\end{array}\right),
\end{gathered}
$$

where

$$
\begin{equation*}
\lambda_{0,0}=1, \lambda_{1,0}=c^{2}, \lambda_{2,0}=c, \lambda_{3,0}=c^{3}, \lambda_{0,1}=c, \lambda_{1,1}=c^{3}, \lambda_{2,1}=1, \lambda_{3,1}=c^{2}, \tag{29}
\end{equation*}
$$

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

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

$$
\begin{equation*}
x_{t, i}=x_{i}^{t}, \tag{30}
\end{equation*}
$$

for some $x_{i} \in \mathbf{F}_{q} \backslash\{0\}$, where $i \in[0: n)$ and $t \in[0: r)$, the code $\mathcal{C}_{2}$ in Construction 2 is an $\left(n=s n^{\prime}, k\right)$ MDS array code with repair degree $d=n-1$ over $\mathbf{F}_{q}$ and repair bandwidth $\left(1+\frac{(s-1)(r-1)}{n-1}\right) \gamma_{\text {optimal }}$, where $\gamma_{\text {optimal }}=\frac{n-1}{r} N$, if the following conditions $\left.i\right)-$ iii $)$ hold
i) $x_{i} \lambda_{i^{\prime}, u} \neq x_{j} \lambda_{j^{\prime}, u^{\prime}}$ for $u, u^{\prime} \in[0: r)$ and $i, j \in[0: n)$ with $i \not \equiv j \bmod m$,
ii) $x_{i} \lambda_{i^{\prime}, u} \neq x_{j} \lambda_{j^{\prime}, u}$ for $u \in[0: r)$ and $i, j \in[0: n)$ with $i \neq j$ and $i \equiv j \bmod m$,
iii) $\lambda_{i^{\prime}, u} \neq \lambda_{i^{\prime}, u^{\prime}}$ for $u, u^{\prime} \in[0: r)$ with $u \neq u^{\prime}$ and $i^{\prime} \in[0: n)$,
where $i^{\prime}=i \% n^{\prime}$ and $j^{\prime}=j \% n^{\prime}$.

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 $x_{i}=$ $c^{\left\lfloor i / n^{\prime}\right\rfloor m r}$ for $i \in[0: n)$, where $c$ is a primitive element of $\mathbf{F}_{q}$ with $q>s m r$.

Proof: For $i, j \in[0: n)$, we rewrite them as $i=v_{0} n^{\prime}+i^{\prime}$ and $j=v_{1} n^{\prime}+j^{\prime}$ for $v_{0}, v_{1} \in[0: s)$ and $i^{\prime}, j^{\prime} \in[0: n)$, and further rewrite $i^{\prime}$ and $j^{\prime}$ as $i^{\prime}=g_{0} m+i^{\prime \prime}$ and $j^{\prime}=g_{1} m+j^{\prime \prime}$, where $g_{0}, g_{1} \in\{0,1\}$ and $i^{\prime \prime}, j^{\prime \prime} \in[0: m)$. By (27), we have

$$
\begin{equation*}
x_{i} \lambda_{i^{\prime}, u}=c^{\left(v_{0} m+i^{\prime \prime}\right) r+\left(u+g_{0}\right) \% r} . \tag{31}
\end{equation*}
$$

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

- For $u, u^{\prime} \in[0: w)$ and $i, j \in[0: n)$ with $i \not \equiv j \bmod m$, i.e., $i^{\prime \prime} \neq j^{\prime \prime}$, we have

$$
x_{i} \lambda_{i^{\prime}, u}-x_{j} \lambda_{j^{\prime}, u^{\prime}}=c^{\left(v_{0} m+i^{\prime \prime}\right) r+\left(u+g_{0}\right) \% r}\left(1-c^{\left(\left(v_{1}-v_{0}\right) m+j^{\prime \prime}-i^{\prime \prime}\right) r+\left(u^{\prime}+g_{1}\right) \% r-\left(u+g_{0}\right) \% r}\right) \neq 0
$$

since $0<\left|\left(\left(v_{1}-v_{0}\right) m+j^{\prime \prime}-i^{\prime \prime}\right) r+\left(u^{\prime}+g_{1}\right) \% r-\left(u+g_{0}\right) \% r\right| \leq s m r-1<q-1$. Then i) of Theorem 4 is satisfied.

- For $u \in[0: w)$ and $i, j \in[0: n)$ with $i \neq j$ and $i \equiv j \bmod m$, i.e, $i^{\prime \prime}=j^{\prime \prime}$, we have

$$
x_{i} \lambda_{i^{\prime}, u}-x_{j} \lambda_{j^{\prime}, u}=c^{\left(v_{0} m+i^{\prime \prime}\right) r+\left(u+g_{0}\right) \% r}\left(1-c^{\left(v_{1}-v_{0}\right) m r+\left(u+g_{1}\right) \% r-\left(u+g_{0}\right) \% r}\right) \neq 0
$$

since $0<\left|\left(v_{1}-v_{0}\right) m r+\left(u+g_{1}\right) \% r-\left(u+g_{0}\right) \% r\right| \leq(s-1) m r+1<q-1$, 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 $\mathcal{C}_{1}$ with repair degree $d<n-1,(n, k)$ MDS array code $\mathcal{C}_{2}$ with repair degree $d=n-1$, and existing ones. Table $\Pi$ provides the details of the comparison between the proposed $(n, k)$ MSR code $\mathcal{C}_{1}$ 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 $\Pi$ and Figure 1, we see that the new MSR code $\mathcal{C}_{1}$ has the following advantages.
i) The new MSR code $\mathcal{C}_{1}$ 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 $\mathcal{C}_{1}$ works for any repair degree $d \in[k+1: n$ ), 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, $\mathcal{C}_{1}$ requires a much smaller finite field than the VBK code when $w \in\{2,3,4\}$. Specifically, when $w=2,3$, and $4, \mathcal{C}_{1}$ requires a finite field $\mathbf{F}_{q}$ with size $q>4\left\lceil\frac{n}{2}\right\rceil, q>3\left\lceil\frac{n}{2}\right\rceil$,

## TABLE II

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 | $w^{n}$ | $q \geq w n$ | $d=k+w-1 \in[k+1: n)$ | \|18, | Section | n IV |
| YB code 2 | $w^{n}$ | $q>n$ | $d=k+w-1 \in[k+1: n)$ | [18. | Section | V |
| VBK code | $\begin{gathered} w^{\left\lceil\frac{n}{w}\right\rceil} \\ (w=2,3,4) \end{gathered}$ | $q \geq\left\{\begin{array}{l}6\left\lceil\frac{n}{2}\right\rceil+2, w=2 \\ 18\left\lceil\frac{n}{w}\right\rceil+2, w=3,4\end{array}\right.$ | $d=k+1, k+2, k+3$ |  | 19] |  |
| CB code | $w^{n}$ | $q>n+w$ | $d=k+w-1 \in[k+1: n)$ |  | \|20] |  |
| LLT code | $w^{\left\lceil\frac{n}{2}\right\rceil}$ | $q>n+\left\lceil\frac{n}{2}\right\rceil w$ | $d=k+w-1 \in[k+1: n-1)$ |  | 21] |  |
| ZZ code | $w^{\left\lceil\frac{n}{2}\right\rceil}$ | $q \geq w n$ | $d=k+w-1 \in[k+1: n)$ |  | \|32| |  |
| New code $\mathcal{C}_{1}$ | $w^{\left\lceil\frac{n}{2}\right\rceil}$ | $q>\left\{\begin{array}{l} 4\left\lceil\frac{n}{2}\right\rceil, w=2 \\ \left\lceil\frac{n}{2}\right\rceil(w+1), w \in[3: r) \\ \left\lceil\frac{n}{2}\right\rceil w, w=r \end{array}\right.$ | $d=k+w-1 \in[k+1: n)$ |  | heorem | 3 |



Fig. 1. Comparision of the sub-packetization level and finite field size among the new $(n, k)$ MSR code $\mathcal{C}_{1}$ and some known ones with repair degree $d=k+2$.
and $q>4\left\lceil\frac{n}{2}\right\rceil$, respectively. In contrast, the VBK code requires a finite field $\mathbf{F}_{q}$ with size $q>6\left\lceil\frac{n}{2}\right\rceil+1,18\left\lceil\frac{n}{3}\right\rceil+1$, and $18\left\lceil\frac{n}{4}\right\rceil+1$, 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 $\mathcal{C}_{1}$.
iii) $\mathcal{C}_{1}$ 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 $\mathcal{C}_{1}$, while the ZZ code requires a larger finite field than $\mathcal{C}_{1}$ when $d>k+1$.
iv) The new MSR code $\mathcal{C}_{1}$ 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 $\mathcal{C}_{1}$.

Table III provides the details of the comparison between the proposed MDS array code $\mathcal{C}_{2}$ 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 ( $n=s n^{\prime}, k$ ) MDS array code $\mathcal{C}_{2}$ 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| | $r^{\left\lceil\frac{n^{\prime}}{r}\right\rceil}$ | $q>n^{(r-1) N+1}$ | $(1+\epsilon) \gamma_{\text {optimal }}$ |
| RTGE code 2 in \|26| | $O\left(r^{r \tau} \log n\right)$ | $O(n)$ | $\leq\left(1+\frac{1}{\tau}\right) \gamma_{\text {optimal }}$ |
| MDS code $\mathcal{C}_{1}$ in \|29| | $r^{n^{\prime}}$ | $q>r n^{\prime}\left\lceil\frac{s}{r}\right\rceil, r \mid(q-1)$ (i.e., $O(n)$ ) | $(1+\epsilon) \gamma_{\text {optimal }}$ |
| MDS code $\mathcal{C}_{2}$ in \|29| | $r^{n^{\prime}-1}$ | $q>r\left\lceil\frac{n^{\prime}}{r}\right\rceil(s-1)+n^{\prime}$ (i.e., $O(n)$ ) | $(1+\epsilon) \gamma_{\text {optimal }}$ |
| MDS code $\mathcal{C}_{4}$ in \|29| | $r^{\left\lceil\frac{n^{\prime}}{r+1}\right\rceil}$ | $\begin{array}{ll} q>\left\lceil\frac{2 n}{3}\right\rceil, & \text { if } r=2 \\ q>N\binom{n-1}{r-1}+1, & \text { if } r>2 \end{array}$ | $(1+\epsilon) \gamma_{\text {optimal }}$ |
| MDS code $\mathcal{C}_{5}$ in \|29| | $r^{n^{\prime}}$ | $q>r n^{\prime}\left\lceil\frac{s}{r}\right\rceil$ i.e., $(O(n))$ | $(1+\epsilon) \gamma_{\text {optimal }}$ |
| New MDS code $\mathcal{C}_{2}$ | $r^{\left\lceil\frac{n^{\prime}}{2}\right\rceil}$ | $q>s r\left\lceil\frac{n^{\prime}}{2}\right\rceil$ (i.e., $O(s n / 2)$ ) | $(1+\epsilon) \gamma_{\text {optimal }}$ |



Fig. 2. Comparision of the sub-packetization level among the new $(n, k) \operatorname{MSR}$ code $\mathcal{C}_{1}$ and some known ones with $r=3$.

From Table $[I I]$ and Figure 2, we can see that the proposed MDS array code $\mathcal{C}_{2}$ has the following advantages compared to existing ones:

- Under the same repair bandwidth, the new MDS code $\mathcal{C}_{2}$ has a much smaller sub-packetization level when compared to the MDS array codes $\mathcal{C}_{1}, \mathcal{C}_{2}$, and $\mathcal{C}_{5}$ in [29].
- By noting that the MDS array code $\mathcal{C}_{4}$ in [29] is implicit when $r>2$, we have that among all the explicit MDS array codes with $(1+\epsilon)$-optimal repair bandwidth and $r>2$, the new MDS code $\mathcal{C}_{2}$ 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 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+ $1+$-optimal repair bandwidth, and repair degree $d<n-1$ will be left for our future research.

## Appendix A

## Proof of Lemma 1

For $i \in[0: r N)$, let $e_{i}$ be row $i$ of the identity matrix of order $r N$. Then define an $r N \times r N$ permutation matrix $\Psi$ as

$$
\Psi=\left(e_{0}^{\top}, e_{N}^{\top}, \ldots, e_{(r-1) N}^{\top}, e_{1}^{\top}, e_{1+N}^{\top}, \ldots, e_{1+(r-1) N}^{\top}, \ldots, e_{N-1}^{\top}, e_{N-1+N}^{\top}, \ldots, e_{N-1+(r-1) N}^{\top}\right)^{\top},
$$

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

$$
\Psi B \Psi^{\top}=\left(\begin{array}{cccc}
B_{0,0}^{\prime} & B_{0,1}^{\prime} & \cdots & B_{0, N-1}^{\prime} \\
B_{1,0}^{\prime} & B_{1,1}^{\prime} & \cdots & B_{1, N-1}^{\prime} \\
\vdots & \vdots & \vdots & \vdots \\
B_{N-1,0}^{\prime} & B_{N-1,1}^{\prime} & \cdots & B_{N-1, N-1}^{\prime}
\end{array}\right)
$$

where

$$
B_{a, b}^{\prime}=\left(\begin{array}{cccc}
B_{0,0}[a, b] & B_{0,1}[a, b] & \cdots & B_{0, r-1}[a, b] \\
B_{1,0}[a, b] & B_{1,1}[a, b] & \cdots & B_{1, r-1}[a, b] \\
\vdots & \vdots & \ddots & \vdots \\
B_{r-1,0}[a, b] & B_{r-1,1}[a, b] & \cdots & B_{r-1, r-1}[a, b]
\end{array}\right), a, b \in[0: N) .
$$

By (2), we have $B_{a, b}^{\prime}=\mathbf{0}$ for $0 \leq b<a<N$. For $a \in[0: N)$, by i), we have

$$
B_{a, a}^{\prime}=\left(\begin{array}{cccc}
1 & 1 & \cdots & 1  \tag{32}\\
B_{1,0}[a, a] & B_{1,1}[a, a] & \cdots & B_{1, r-1}[a, a] \\
\vdots & \vdots & \ddots & \vdots \\
B_{1,0}^{r-1}[a, a] & B_{1,1}^{r-1}[a, a] & \cdots & B_{1, r-1}^{r-1}[a, a]
\end{array}\right)
$$

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

$$
|\Psi||B|\left|\Psi^{\top}\right|=\left|\Psi B \Psi^{\top}\right|=\left|\begin{array}{cccc}
B_{0,0}^{\prime} & B_{0,1}^{\prime} & \cdots & B_{0, N-1}^{\prime} \\
& B_{1,1}^{\prime} & \cdots & B_{1, N-1}^{\prime} \\
& & \ddots & \vdots \\
& & & B_{N-1, N-1}^{\prime}
\end{array}\right|=\prod_{a=0}^{N-1}\left|B_{a, a}^{\prime}\right| \neq 0
$$

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 $a=\left(a_{0}, a_{1}, \ldots, a_{m-2}\right) \in[0: N / w), u \in[0: w)$, and $j \in[0: n)$, according to (12), we have

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

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

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

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

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

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

which together with (13) implies

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

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 $j \equiv i \bmod m$.

Next, we prove that ii) holds for $j \not \equiv i \bmod m$. 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

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

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 $\left(g_{i, u}(a)\right)_{j-m}=a_{j-m}$ by (8). By (33), we have

$$
\begin{align*}
V_{i, u}[a,:] \cdot A_{t, j} & =\lambda_{j, a_{j-m}}^{t} e_{g_{i, u}(a)} \\
& =\lambda_{j, a_{j-m}}^{t} e_{a}^{(N / w)} \cdot \sum_{b=0}^{N / w-1}\left(e_{b}^{(N / w)}\right)^{\top} e_{g_{i, u}(b)} \\
& =e_{a}^{(N / w)}\left(\sum_{b=0}^{N / w-1} \lambda_{j, b_{j-m}}^{t}\left(e_{b}^{(N / w)}\right)^{\top} e_{b}^{(N / w)}\right) V_{i, u} \\
& =e_{a}^{(N / w)} B_{t, j, i} \cdot V_{i, u} \\
& =B_{t, j, i}[a,:] \cdot V_{i, u} \tag{35}
\end{align*}
$$

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 $0 \leq i \neq j<n$ with $i \not \equiv j \bmod m$. While the proof of iii) is similar to the analysis in (19); thus, we omit it.

## REFERENCES

[1] 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 $\delta$-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, " $\epsilon$-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.

