Explicit Constructions of Optimal-Access MDS Codes With Nearly Optimal Sub-Packetization

Min Ye and Alexander Barg, *Fellow, IEEE*

***Abstract*— An (n, k, l) maximum distance separable (MDS) array code of length n, dimension $k = n - r$, and sub-packetization l is formed of $l \times n$ matrices over a finite field F, with every column of the matrix stored on a separate node in the distributed storage system and viewed as a coordinate of the codeword. Repair of a failed node (recovery of one erased column) can be performed by accessing a set of $d \leq n - 1$ surviving (helper) nodes. The code is said to have the optimal access property if the amount of data accessed at each of the helper nodes meets a lower bound on this quantity. For optimal-access MDS codes with $d = n-1$, the sub-packetization l satisfies the bound $l \geq r^{(k-1)/r}$. In our previous work (IEEE Trans. Inf. Theory, vol. 63, no. 4, 2017), for any n and r, we presented an explicit construction of optimal-access MDS codes with sub-packetization $l = r^{n-1}$. In this paper, we take up the question of reducing the sub-packetization value l to make it to approach the lower bound. We construct an explicit family of optimal-access codes with $l = r^{\lceil n/r \rceil}$, which differs from the optimal value by at most a factor of r^2. These codes can be constructed over any finite field F as long as $|F| \geq r\lceil n/r \rceil$, and afford low-complexity encoding and decoding procedures. We also define a version of the repair problem that bridges the context of regenerating codes and codes with locality constraints (LRC codes), which we call *group repair with optimal access*. In this variation, we assume that the set of $n = sm$ nodes is partitioned into m repair groups of size s, and require that the amount of accessed data for repair is the smallest possible whenever the $d = s + k - 1$ helper nodes include all the other $s-1$ nodes from the same group as the failed node. For this problem, we construct a family of codes with the group optimal access property. These codes can be constructed over any field F of size $|F| \geq n$, and also afford low-complexity encoding and decoding procedures.**

***Index Terms*— Distributed storage, MDS array codes, minimum storage regenerating codes, optimal access, optimal sub-packetization.**

I. Introduction

THE repair problem of array codes is motivated by applications of codes in distributed storage systems (e.g., the Google File System (GFS) and Hadoop Distributed File System (HDFS)) which assume that the data is spread across a large number of drives (nodes). A popular solution of the task of protecting the data from node failures relies on maximum distance separable (MDS) array codes which provide a universal mechanism of node recovery regardless of the location of the failed nodes. By distributing the codeword across different nodes, we ensure that in the event of node failure it is possible to recover the missing data using the information stored in functional nodes. Among the parameters of the code that are important for storage applications are the amount of data transferred in the system during node repair (the repair bandwidth), which characterizes the network usage, and the volume of accessed data which corresponds to the number of disk I/O operations. Therefore, recent research on MDS codes for distributed storage has focused on codes that can minimize these two quantities; see in particular the paper by Dimakis *et al.* [1] which motivated most of the recent research in coding for storage and derived lower bounds on the repair bandwidth of MDS codes.

Manuscript received June 5, 2016; revised April 24, 2017; accepted June 21, 2017. Date of publication July 24, 2017; date of current version September 13, 2017. This work was supported by NSF under Grant CCF1422955 and Grant CCF1618603.

The authors are with the Department of ECE and ISR, University of Maryland, College Park, MD 20742 USA (e-mail: yeemmi@gmail.com; abarg@umd.edu).

Communicated by C.-C. Wang, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2017.2730863

A. Exact-Repair Regenerating Codes

Most studies of codes with optimal repair bandwidth in the literature are concerned with a particular class of codes known as *array codes* [2]. An (n, k, l) array code $\mathcal{C}$ is formed of $l \times n$ matrices $(C_1, \ldots, C_n) \in (F^l)^n$, where F is a finite field. Each column C_i of the matrix is a codeword coordinate, and the parameter l that determines the dimension of the column vector C_i is called *sub-packetization*. The code $\mathcal{C}$ is said to have the MDS property if every k out of n columns of the matrix suffice to recover the remaining r columns. In this paper, we consider only MDS array codes. As usual in distributed storage applications, we assume that an MDS array code of length n formed of k information coordinates and $r = n - k$ parity coordinates is spread across n different nodes of the storage cluster. Each node of the cluster stores a coordinate of the code. At the same time, the basic repair task studied for MDS array codes consists in recovering one erased column (a failed node) by accessing information stored in the other nodes of the same codeword.

Suppose that a node becomes unavailable, and the system attempts to repair its content by connecting to d surviving (helper) nodes, $k \leq d \leq n - 1$. From the perspective of system architecture, efficient repair requires that the amount of information accessed and downloaded from the helper nodes be as small as possible, and this introduces the notion of access and repair bandwidth. To formalize this concept, let us give the following definition.

Definition 1: Consider an (n, k, l) MDS array code $\mathcal{C}$ over F. We write a codeword of $\mathcal{C}$ as $(C_1, \ldots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,l-1})^T \in F^l, i = 1, \ldots, n$. We say that a node $i \in [n]$ can be repaired from a subset of helper nodes $\mathcal{R}_i \subset [n]\backslash\{i\}, |\mathcal{R}_i| \geq k$ by accessing $\omega_i(\mathcal{R}_i)$ symbols of F and downloading $\beta_i(\mathcal{R}_i)$ symbols of F if there are numbers $\beta_{i,j}, j \in \mathcal{R}_i$ and $\omega_{i,j}, j \in \mathcal{R}_i$, and $|\mathcal{R}_i| + 1$ functions

$f_{i,j} : F^l \to F^{\beta_{i,j}}$, $j \in \mathcal{R}_i$ and $g_i : F^{\sum_j \beta_{i,j}} \to F^l$ such that

$$C_i = g_i(f_{i,j}(C_j), j \in \mathcal{R}_i),$$
$$\sum_{j\in\mathcal{R}_i} \beta_{i,j} = \beta_i(\mathcal{R}_i), \quad \text{and} \quad \sum_{j\in\mathcal{R}_i} \omega_{i,j} = \omega_i(\mathcal{R}_i),$$

where the function $f_{i,j}$ depends on $\omega_{i,j}$ coordinates of the node $C_j = (c_{j,0}, c_{j,1}, \ldots, c_{j,l-1})^T$.

Informally, this definition says that we download some functions $f_{i,j}$ of the information stored in the helper nodes and perform repair using the function g_i that takes the values of $f_{i,j}$ as its arguments. The quantities $\omega_i(\mathcal{R}_i)$ and $\beta_i(\mathcal{R}_i)$ control the number of the accessed and downloaded symbols of the field F, respectively.

The smallest volume of the downloaded data $\beta_i^*(\mathcal{R}_i) = \min \beta_i(\mathcal{R}_i)$ over the choice of the functions $\{f_{i,j}\}_{j\in\mathcal{R}_i}$ and g_i is called the $(i, \mathcal{R}_i)$-*repair bandwidth* of the code $\mathcal{C}$. The quantity $\beta(\mathcal{C}) := \max_{i\in[n]} \beta_i^*([n]\backslash\{i\})$ is called the *repair bandwidth* of the code $\mathcal{C}$.

The smallest amount of the accessed data $\omega_i^*(\mathcal{R}_i) = \min \omega_i(\mathcal{R}_i)$ over the choice of the functions $\{f_{i,j}\}_{j\in\mathcal{R}_i}$ and g_i is called the $(i, \mathcal{R}_i)$-*access* of the code $\mathcal{C}$. The quantity $\omega(\mathcal{C}) := \max_{i\in[n]} \omega_i^*([n]\backslash\{i\})$ is called the *access* of the code $\mathcal{C}$. Clearly, $\omega_i^*(\mathcal{R}_i) \geq \beta_i^*(\mathcal{R}_i)$ for any $i \in [n]$ and any $\mathcal{R}_i \subset [n]\backslash\{i\}$, $|\mathcal{R}_i| \geq k$. Consequently, $\omega(\mathcal{C}) \geq \beta(\mathcal{C})$.

The lower bounds on the repair bandwidth and access were established in the recent literature on regenerating codes. As shown in [1], for an (n, k, l) MDS array code $\mathcal{C}$, the recovery of a single failed node from d helper nodes requires to download at least a $1/(d+1-k)$ fraction of the data stored in each of the helper nodes. Using our notation,

$$\omega_i^*(\mathcal{R}_i) \geq \beta_i^*(\mathcal{R}_i) \geq \frac{|\mathcal{R}_i| l}{|\mathcal{R}_i| + 1 - k} \tag{1}$$

for any $i \in [n]$ and any $\mathcal{R}_i \subset [n]\backslash\{i\}$, $|\mathcal{R}_i| \geq k$. Note that the right-hand side of (1) is a decreasing function of $|\mathcal{R}_i|$. Since (1) is an achievable lower bound (as discussed below in the introduction), the repair bandwidth is minimized when $\mathcal{R}_i = [n]\backslash\{i\}$. In this case, (1) becomes

$$\omega_i^*([n]\backslash\{i\}) \geq \beta_i^*([n]\backslash\{i\}) \geq \frac{(n-1)l}{n-k}. \tag{2}$$

We say that an (n, k, l) MDS array code $\mathcal{C}$ has the *optimal repair* property (and call it an optimal-repair (n, k, l) code) if $\beta(\mathcal{C}) = \frac{(n-1)l}{n-k}$. A coding-theoretic perspective of the bound (1) and of other related results was recently developed in [3].

According to Definition 1, even for codes with the optimal repair property, we might still need to access a larger amount of data than the lower bounds in (1) during the recovery of a failed node. We say that an (n, k, l) MDS array code $\mathcal{C}$ has the *optimal access* property (and call it an optimal-access (n, k, l) code) if $\omega(\mathcal{C}) = \frac{(n-1)l}{n-k}$. We call $\mathcal{C}$ a *systematic optimal-access* MDS array code if $\omega_i^*([n]\backslash\{i\}) = \frac{(n-1)l}{n-k}$ for any systematic node C_i. Clearly, optimal-access MDS array codes are a subclass of optimal-repair MDS array codes.

The construction problem of optimal-repair MDS array codes has been extensively studied over the last several years. More specifically, for $k \leq (n+1)/2$ (the low rate regime), MDS array codes with d-optimal repair property were constructed in [4]–[6]. For $k > (n+1)/2$ (the high-rate regime) papers [7]–[11] showed that for large enough base field F there exist MDS array codes that can optimally repair any single systematic node failure using all the surviving nodes, and [12] showed the same for all rather than only systematic nodes. However, explicit constructions of such codes with rate larger than $1/2$ and $r > 3$ were found only recently [13], where we proposed two families of MDS array codes that can repair any number of failed nodes from any number of helper nodes by downloading the minimum possible amount of data from the helper nodes.

The subclass of optimal-access codes is of particular interest among optimal-repair codes. At the same time, the sub-packetization parameter of optimal-access codes is constrained from below. Namely, according to a result of [14], if an (n, k, l) MDS array code has the optimal access property, then $l \geq r^{(k-1)/r}$. Existence and constructions of optimal-access codes were studied in several recent works. We mention the results of [12] which established existence of optimal-access MDS array codes with $l = r^k$ and [15], [16] which proved existence of such codes with $l = r^{\lceil n/r \rceil}$, although both results require a large-size finite field F. As for explicit constructions, until recently they were known only for $r = 2, 3$. Namely, [12] constructed codes with $l = r^k$ over the field $\mathbb{F}_3$ if $r = 2$ and $\mathbb{F}_4$ if $r = 3$ (independent of k). Reference [17] constructed systematic optimal-access MDS array codes[1] with $l = r^{k/r}$, where k is a multiple of r. The size of the underlying field for [17] is at least $k/2 + 1$ for $r = 2$ and $2k + 1$ for $r = 3$. In [13], we proposed a family of optimal-access MDS array codes with sub-packetization $l = r^{n-1}$, and this is the only explicit family of optimal-access MDS array codes for $r > 3$ known in the literature.

In this work we present an explicit construction of optimal-access MDS array codes for any r and n with sub-packetization $l = r^{\lceil n/r \rceil}$, which differs from the lower bound by a factor of at most r^2. These codes can be constructed over any finite field F as long as $|F| \geq r\lceil n/r \rceil$, and the encoding and decoding procedures of these codes have low complexity.

Remark 2: The repair problem has been studied for a relatively short time, so the related terminology is still somewhat unsettled. For instance, [15] and [16] refer to codes with $l = r^{\lceil n/r \rceil}$ as codes with polynomial sub-packetization. This implicitly assumes the asymptotic regime of $k = Rn$, i.e., of codes with a fixed rate R bounded away from 1. At the same time, arguably the case of $r = o(n)$, for instance constant r, is more important for the repair problem because the encoded data in storage are likely to include only a small number of parity checks. In this regime the above value of l is an exponential function of the block length. To cover all the possible cases, we prefer not to use the terms polynomial or exponential to describe the growth rate of the parameter l.

Remark 3: Recently, the repair problem was extended from array codes to classes of scalar codes, such as Reed-Solomon (RS) codes. This line of research was initiated by Guruswami and Wootters [18] who gave a characterization of linear repair schemes of general scalar linear MDS codes. They also presented a specific repair scheme for a family of RS codes and proved that (in some cases) the repair bandwidth of RS codes under this scheme is the smallest possible among

[1]Reference [17] also considered relaxing systematic optimal-access to systematic optimal-repair, and gave an explicit construction of codes for $r = 3$ and k a multiple of 4 with sub-packetization $l = r^{k/(r+1)}$ over the field of size linear in k.

all linear repair schemes and all scalar linear MDS codes with the same parameters. At the same time, the repair bandwidth of RS codes attained in [18] is rather far from the bound (2).

In a subsequent work [19], the authors used the approach of [18] to construct an explicit family of RS codes whose repair bandwidth asymptotically achieves the bound (2). Very recently, Tamo *et al.* [20] presented an explicit construction of RS codes that achieve the bound (2) for any given code parameters n and k. While the construction in [20] has superexponential sub-packetization (specifically, the sub-packetization scales as $e^{n \log n}$), the same paper shows this growth order is also necessary for scalar linear codes to achieve the bound (2) using linear repair schemes. In contrast to this, for vector (array) codes to attain the bound (2), it is sufficient to use exponentially growing sub-packetization.

Remark 4: ADDED ON JUNE 30, 2017: Shortly after the release of this paper, in an independent work, Sasidharan *et al.* [28] presented a construction of codes that is very similar to our Construction 1.

B. Repair Groups and Node Regeneration: Group Optimal Access Property

In this paper, we also introduce a coding problem for distributed storage that bridges codes with locality, in particular, LRC codes (e.g., [21]–[23]) and regenerating codes. To motivate it, consider an architecture of distributed storage under which $n = sm$ storage nodes are partitioned into m local groups (we assume throughout that $s \le r$). Nodes in the same group are logically better connected (for instance, they are geographically close to each other and thus have stable links between them), while the connectivity between nodes from different groups fluctuates smoothly over time (for instance, relying on a slowly fading channel). When a node fails, the system seeks to perform repair by accessing the minimum possible amount of data on a set of helper nodes. Efficient repair also suggests that the helper nodes are chosen from a subset of nodes most easily reachable from the failed node. Since the failed node can always connect to all the nodes in the same group as itself, an MDS array code with the $(s, d = s+k-1)$-group optimal access property can minimize the disk I/O and network traffic during the repair of any single failed node for such systems.

This motivates the following definition.

Definition 5: Let $\mathcal{C}$ be an $(n = sm, k, l)$ MDS array code whose nodes are partitioned into m groups of size s each. Referring to Def. 1, we say that $\mathcal{C}$ has the $(s, d = s+k-1)$-*group optimal access property* if $\omega_i^*(\mathcal{R}_i) = dl/s$ for any i and any set of helper nodes $\mathcal{R}_i$ of size d that contains all the $s-1$ nodes from the same group as i.

By definition, repair of the failed node in the group repair mode can be performed by accessing the volume of data that attains the lower bound (1), justifying the optimality qualifier.

In this paper, we construct an explicit family of $(n = sm, k = n-r, l = s^m)$ MDS array codes with the $(s, d = s+k-1)$-group optimal access property for any s, m and r such that $r \ge s$. These codes can be constructed over any finite field F as long as $|F| \ge n$, and are equipped with low-complexity encoding and decoding. Our construction is flexible in the sense that it allows any number m of local groups and any number $s \le r$ of storage nodes in a local group.

The code construction is presented in the next section. Section III contains a proof of the MDS property of the constructed codes, and Section IV gives a proof of the group optimal access property.

We note that coding designs that address data regeneration based on different conditions at different helper nodes, based on access conditions or transient unavailability (degraded reads or hard errors) have been considered in a number of earlier works. For instance, in [13] we constructed codes that support repair of one or more failed nodes by accessing any set of d helper nodes in the same encoding block. A different line of research that establishes conditions under which helper node selection improves the storage/bandwidth tradeoff was recently developed in [24], [25]. Yet another link between regenerating codes and LRC codes is the "local regeneration" problem [26], [27], where the local repair of the code is also required to have small bandwidth.

II. CODE CONSTRUCTION

Let $\mathcal{C} \in F^{ln}$ be an (n, k, l) array code with nodes $C_i \in F^l, i = 1, \ldots, n$, where each C_i is a column vector with coordinates $C_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,l-1})^T$. Throughout this paper we consider codes defined by the following r parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \ldots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0,\ t = 0, 1, \ldots, r-1\}, \tag{3}$$

where $A_{t,i}, 0 \le t \le r-1, 1 \le i \le n$ are $l \times l$ matrices over F. Let $A(a, b)$ be the entry in the a-th row and b-th column of matrix A, $0 \le a, b \le l-1$. Throughout we assume that $0^0 = 1$.

Construction 1: Let s, r and m be positive integers such that $s \le r \le sm$, let $n = sm, l = s^m$. Let F be a finite field of size $|F| \ge n$, let $\{\lambda_i\}_{i \in [n]}$ be n distinct elements in F, and let $\gamma \in F \backslash \{0, 1\}$. Given an integer $a, 0 \le a \le l-1$, we write its s-ary expansion as $a = (a_m, a_{m-1}, \ldots, a_1)$.

For $v \in [m]$ and $0 \le u \le s-1$, define $a(v, u) := (a_m, a_{m-1}, \ldots, a_{v+1}, u, a_{v-1}, a_{v-2} \ldots, a_1)$. For $v \in [m], u = 0, 1, \ldots, s-1$, and $t = 0, 1, \ldots, r-1$, define an $l \times l$ matrix $A_{t,(v-1)s+u+1}$ as follows: for $0 \le a, b \le l-1$, let

$$A_{t,(v-1)s+u+1}(a, b) = \begin{cases} \lambda^t_{(v-1)s+u+1} & \text{if } a_v < u,\ b = a, \\ \gamma\, \lambda^t_{(v-1)s+u+1} & \text{if } a_v > u,\ b = a, \\ \lambda^t_{(v-1)s+w+1} & \text{if } a_v = u,\ b = a(v, w) \\ & \text{for some } w \in \{0, 1, \ldots, s-1\}, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

We construct an $(n, k = n-r, l)$ array code defined by (3)*, where the matrices $A_{t,i}$ are defined in* (4)*.*

We will show that the code $\mathcal{C}$ defined by Construction 1 has the MDS property. In the case of $s = r$ it also has the optimal access property, while if $s < r$ it has the group optimal access property.

In Section III we give an example of the above matrices for $s = r = 3$ and $m = 2$ and show that the obtained codes have the MDS property.

In Construction 1 we assumed that the code length is a product of two numbers, s and m. While this assumption leads

to a simple uniform formulation of the code construction, it can be easily lifted at the expense of a more detailed notation. Namely, the following construction extends the case of $s = r$ in Construction 1 to cover all possible code length n and has essentially the same properties as the codes defined above.

Construction 2: Let $n = rm + r'$ and $l = r^{m+1}$, where $r > 0, m \geq 0$ are integers and $1 \leq r' \leq r-1$. Let F be a finite field of size $|F| \geq r(m+1)$, let $\{\lambda_i, i = 1, \ldots, r(m+1)\}$ be distinct elements of F, and let $\gamma \in F \backslash \{0, 1\}$. Given an integer a between 0 and $l-1$, we write its r-ary expansion as $a = (a_{m+1}, a_m, \ldots, a_1)$. For $v \in \{1, \ldots, m+1\}$ and $0 \leq u \leq r-1$, define $a(v, u) := (a_{m+1}, a_m, \ldots, a_{v+1}, u, a_{v-1}, a_{v-2} \ldots, a_1)$.

For $v \in [m]$, $u = 0, 1, \ldots, r-1$, and $t = 0, 1, \ldots, r-1$, define an $l \times l$ matrix $A_{t,(v-1)r+u+1}$ as follows: for $0 \leq a, b \leq l-1$, let

$$A_{t,(v-1)r+u+1}(a,b) = \begin{cases} \lambda^t_{(v-1)r+u+1} & \text{if } a_v < u,\ b = a, \\ \gamma\,\lambda^t_{(v-1)r+u+1} & \text{if } a_v > u,\ b = a, \\ \lambda^t_{(v-1)r+w+1} & \text{if } a_v = u,\ b = a(v,w) \\ & \text{for some } w \in \{0, 1, \ldots, r-1\}, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

For $u = 0, 1, \ldots, r' - 1$, and $t = 0, 1, \ldots, r-1$, define an $l \times l$ matrix $A_{t,mr+u+1}$ as follows: for $0 \leq a, b \leq l-1$, let

$$A_{t,mr+u+1}(a,b) = \begin{cases} \lambda^t_{mr+u+1} & \text{if } a_{m+1} < u,\ b = a, \\ \gamma\,\lambda^t_{mr+u+1} & \text{if } a_{m+1} > u,\ b = a, \\ \lambda^t_{mr+w+1} & \text{if } a_{m+1} = u,\ b = a(m+1,w) \\ & \text{for some } w \in \{0, 1, \ldots, r-1\}, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

We construct an $(n = rm+r', k = n-r, l = r^{m+1})$ array code defined by (3), where the matrices $A_{t,i}, 0 \leq t \leq r-1, 1 \leq i \leq n$ are defined in (5)-(6).

III. The MDS Property

In this section we show that the code family given by Construction 1 has the MDS property. We start with an example that shows the working of the definition (3)-(4) as well as provides intuition for the proof of the MDS property given below in this section. While the notation in the proof makes it difficult to glean an intuitive picture, this example serves to visualize the ideas behind the construction and the proof.

A. Example

Take $s = r = 3$ and $m = 2$ in Construction 1, so $n = 6$ and $l = 9$. Let us write out the 9×9 matrices $A_{t,i}, i = 1, \ldots, 6$. The code presented below can be realized over any field of size $|F| \geq n = 6$, so the smallest field is $\mathbb{F}_7$.

$$A_{t,1} = \begin{bmatrix} \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma\lambda_1^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma\lambda_1^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma\lambda_1^t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^t \end{bmatrix}$$

$$A_{t,2} = \begin{bmatrix} \lambda_2^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma\lambda_2^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2^t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2^t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^t \end{bmatrix}$$

$$A_{t,3} = \begin{bmatrix} \lambda_3^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_3^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_3^t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_3^t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3^t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3^t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^t & \lambda_2^t & \lambda_3^t \end{bmatrix}$$

$$A_{t,4} = \begin{bmatrix} \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 & 0 \\ 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 \\ 0 & 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t \\ 0 & 0 & 0 & \gamma\lambda_4^t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma\lambda_4^t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma\lambda_4^t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_4^t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_4^t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_4^t \end{bmatrix}$$

$$A_{t,5} = \begin{bmatrix} \lambda_5^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_5^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_5^t & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 & 0 \\ 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 \\ 0 & 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^t \end{bmatrix}$$

$$A_{t,6} = \begin{bmatrix} \lambda_6^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_6^t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_6^t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_6^t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_6^t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_6^t & 0 & 0 & 0 \\ \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 & 0 \\ 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t & 0 \\ 0 & 0 & \lambda_4^t & 0 & 0 & \lambda_5^t & 0 & 0 & \lambda_6^t \end{bmatrix}$$

$$\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_6 & 0 & 0 \\
0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_4^2 & 0 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 & 0 & 0 \\
0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & \lambda_4 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_6 & 0 \\
0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & \lambda_4^2 & 0 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 & 0 \\
0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_4 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_6 \\
0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^2 & 0 & 0 & 0 & 0 & 0 & \lambda_4^2 & 0 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^2
\end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17} \\ x_{18} \\ x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \end{bmatrix} = 0. \tag{8}$$

$$\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & 0 & 0 & 0 \\
0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & \lambda_4 & 0 & \lambda_5 & 0 & 0 & \lambda_6 & 0 \\
0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & \lambda_2^2 & 0 & 0 & 0 & 0 & \lambda_4^2 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 & 0 \\
0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & \lambda_4 & 0 & \lambda_5 & 0 & 0 & \lambda_6 \\
0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & \lambda_4^2 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 \\
0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2 & 0 & 0 & 0 & 0 & 0 & \lambda_5 & 0 & 0 & \lambda_6 \\
0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_2^2 & 0 & 0 & 0 & 0 & 0 & \lambda_5^2 & 0 & 0 & \lambda_6^2 \\
0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \gamma & 0 \\
0 & 0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & \lambda_3 & 0 & 0 & 0 & 0 & \gamma\lambda_5 & 0 \\
0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & \lambda_3^2 & 0 & 0 & 0 & 0 & \gamma\lambda_5^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma & 0 & 0 & 0 & 0 & 0 & \gamma \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5 \\
0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_1^2 & 0 & 0 & 0 & 0 & 0 & \lambda_1^2 & \lambda_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & \gamma\lambda_5^2
\end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_9 \\ x_{10} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{18} \\ x_{19} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \end{bmatrix} = 0. \tag{9}$$

The MDS property states that any 3×3 block submatrix of the 3×6 block matrix formed of the matrices $A_{t,i}$ is invertible (here the blocks are $l \times l$ matrices). Below we show this for the matrix

$$B = \begin{bmatrix} A_{0,1} & A_{0,2} & A_{0,5} \\ A_{1,1} & A_{1,2} & A_{1,5} \\ A_{2,1} & A_{2,2} & A_{2,5} \end{bmatrix}.$$

Let X is a column vector in F^{27} with coordinates $X = (x_0, x_1, \ldots, x_{26})^T$. Our claim will follow if we prove that $BX = 0$ implies that $X = 0$.

We proceed as follows. For convenience of presentation, let us permute the rows of B to obtain a matrix $D = PB$, where the permutation matrix $(P_{ij})_{0 \le i,j \le 26}$ is given by

$$P_{ij} = 1 \text{ iff } i = (j - j \bmod 9)/9 + 3(j \bmod 9). \tag{7}$$

It is clear that P has exactly one 1 in each row, so it is indeed a permutation on $\{0, 1, \ldots, 26\}$ (note that multiplication by a full-rank matrix does not change the rank). We will prove that the matrix D has a trivial null space, i.e., $DX = 0$ implies that $X = 0$. Writing out the condition $DX = 0$ explicitly, we obtain a system of equations given in (8), as shown at the top of this page.

Since the coefficients λ_i are distinct for different i, the highlighted rows in (8) imply that $x_2 = x_8 = x_{11} = x_{17} = x_{20} = x_{26} = 0$. Eliminating these variables from (8) we obtain a system of equations given by (9), as shown at the top of this page.

Looking at the first three rows in (9), and treating $x_1 + x_9$ as a new variable, we conclude that $x_0 = x_1 + x_9 = x_{18} = 0$. Similarly, the second group of three rows implies that $\gamma x_1 + x_9 = x_{10} = x_{19} = 0$. Taking these results together and noting that $\gamma \neq 1$, we see that $x_0 = x_1 = x_{18} = x_9 = x_{10} = x_{19} = 0$.

A similar argument used for the last 9 rows in (9) shows that $x_5 = x_{14} = x_{23} = x_6 = x_7 + x_{15} = x_{24} = \gamma x_7 + x_{15} = x_{16} = x_{25} = 0$, and so $x_5 = x_{14} = x_{23} = x_6 = x_7 = x_{24} = x_{15} = x_{16} = x_{25} = 0$. Writing out the remaining equations,

we obtain the following set of equations:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ \lambda_1 & \lambda_2 & \lambda_2 & 0 & \lambda_5 & 0 \\ \lambda_1^2 & \lambda_2^2 & \lambda_2^2 & 0 & \lambda_5^2 & 0 \\ 0 & \gamma & 1 & 1 & 0 & 1 \\ 0 & \gamma\lambda_1 & \lambda_1 & \lambda_2 & 0 & \lambda_5 \\ 0 & \gamma\lambda_1^2 & \lambda_1^2 & \lambda_2^2 & 0 & \lambda_5^2 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \end{bmatrix} = 0.$$

From this set of equations we obtain that $x_4 + x_{12} = x_3 = x_{21} = \gamma x_4 + x_{12} = x_{13} = x_{22} = 0$. Thus, $x_3 = x_4 = x_{21} = x_{12} = x_{13} = x_{22} = 0$. Overall these arguments prove that $X = 0$, and so B is invertible. □

B. A Proof of the MDS Property

Let us fix s, r, and m, so $n = sm$ and $l = s^m$. The code $\mathcal{C}$ given by Construction 1 is an MDS array code if for any $1 \le i_1 < i_2 < \ldots < i_r \le n$, the matrix

$$B_{s,r,m}[i_1, i_2, \ldots, i_r] := \begin{bmatrix} A_{0,i_1} & A_{0,i_2} & \ldots & A_{0,i_r} \\ A_{1,i_1} & A_{1,i_2} & \ldots & A_{1,i_r} \\ \vdots & \vdots & \vdots & \vdots \\ A_{r-1,i_1} & A_{r-1,i_2} & \ldots & A_{r-1,i_r} \end{bmatrix}$$

is invertible. Below we suppress the parameters s, r, m, and $i_1, i_2, \ldots, i_r$ from the notation and write B to refer to this matrix. In other words, given a vector $X \in F^{rl}$ we need to prove that

$$BX = 0 \tag{10}$$

implies that $X = 0$, where X is a vector in F^{rl} with coordinates $X = (x_0, x_1, \ldots, x_{rl-1})^T$. The proof essentially follows the example in Sect. III-A. We begin with a preview which also serves to introduce some notation.

In order to transform B into a matrix of the form (8), let us define a permutation matrix $(P_{ij})_{0 \le i,j < rl}$ by setting

$$P_{ij} = 1 \text{ iff } i = (j - j \bmod l)/l + r(j \bmod l); \tag{11}$$

compare with our example in (7).

Define a matrix $D = PB$. We shall prove that

$$DX = 0 \tag{12}$$

implies that $X = 0$. (The full notation for D should be $D_{s,r,m}[i_1, i_2, \ldots, i_r]$, but we again suppress the parameters.) For $a = 0, 1, \ldots, l-1$, define $D^{(a)}$ to be the $r \times rl$ submatrix of D consisting of rows $ar, ar+1, \ldots, ar+r-1$. Define a column vector

$$L_i = (1, \lambda_i, \ldots, \lambda_i^{r-1})^T, \quad i \in [n]. \tag{13}$$

On account of (4), for every $a \in \{0, 1, \ldots, l-1\}$, all the nonzero columns of $D^{(a)}$ belong to the set

$$\{L_1, L_2, \ldots, L_n, \gamma L_1, \gamma L_2, \ldots, \gamma L_n\}.$$

We proceed by defining several subsets of the set of column indices of $D^{(a)}$ for every $a \in \{0, 1, \ldots, l-1\}$:

- Let $\mathcal{U}^{(a)} \subset [n]$ be a subset such that $i \in \mathcal{U}^{(a)}$ if and only if there is a nonzero column in $D^{(a)}$ equal to either L_i or γL_i;
- Let $\mathcal{J}^{(a)} \subset \{0, 1, \ldots, rl\}$ be the set of indices of the nonzero columns in $D^{(a)}$;
- For $i \in [n]$, define the set $\mathcal{J}^{(a)}(i) \subset \mathcal{J}^{(a)}$ as $\mathcal{J}^{(a)}(i) = \{j \in \mathcal{J}^{(a)} : \text{ the } j\text{th column of } D^{(a)} \text{ is either } L_i \text{ or } \gamma L_i\}$.

In our example above, let $D^{(0)}$ be the first $r = 3$ rows of the matrix $D = D_{3,3,2}[1, 2, 5]$ in (8). Then the only nonzero columns in $D^{(0)}$ are of type L_1, L_2, L_3, or L_5, and so $\mathcal{U}^{(0)} = \{1, 2, 3, 5\}$. The indices of the nonzero columns are given by $\mathcal{J}^{(0)} = \{0, 1, 2, 9, 18\}$, and $\mathcal{J}^{(0)}(1) = \{0\}, \mathcal{J}^{(0)}(2) = \{1, 9\}, \mathcal{J}^{(0)}(3) = \{2\}, \mathcal{J}^{(0)}(5) = \{18\}$.

Clearly the sets $\mathcal{J}^{(a)}(i)$ form a partition of the set $\mathcal{J}^{(a)}$, so

$$\mathcal{J}^{(a)} = \bigcup_{i \in \mathcal{U}^{(a)}} \mathcal{J}^{(a)}(i).$$

According to (4), for every $i \in [n]$, every diagonal entry of $A_{t,i}$ is either λ_i^t or $\gamma\lambda_i^t$ (see also the example in Sect. III-A where we explicitly write out the matrices $A_{t,1}, \ldots, A_{t,6}$). Therefore, for every $i \in [n]$, every row of $A_{t,i}$ contains at least one of the elements λ_i^t and $\gamma\lambda_i^t$. As an immediate consequence, for every $i \in \{i_1, \ldots, i_r\}$, the set of nonzero columns in the strip of r rows $D^{(a)}$, $a = 0, 1, \ldots, l-1$ contains at least one column out of the pair $(L_i, \gamma L_i)$. This implies that $\{i_1, i_2, \ldots, i_r\} \subseteq \mathcal{U}^{(a)}$ for all $a \in \{0, 1, \ldots, l-1\}$. Our strategy of proving that (12) is satisfied only for $X = 0$ will be to find a set of indices

$$\mathcal{S} = \{a : a \in \{0, 1, \ldots, l-1\}, |\mathcal{U}^{(a)}| = r\}$$

(as before $\mathcal{S} = \mathcal{S}_{s,r,m}[i_1, i_2, \ldots, i_r]$). For every $a \in \mathcal{S}$, all the nonzero columns of $D^{(a)}$ belong to the set

$$\{L_{i_1}, L_{i_2}, \ldots, L_{i_r}, \gamma L_{i_1}, \gamma L_{i_2}, \ldots, \gamma L_{i_r}\}.$$

Since the columns $L_{i_1}, L_{i_2}, \ldots, L_{i_r}$ form a Vandermonde matrix, we conclude that the corresponding variables or their linear combinations are 0, and therefore we can eliminate some of the variables in (12). Referring to our example, this set is exactly the set of highlighted rows in the matrix in (8), and thus in this case the set of strip labels equals $\mathcal{S} = \{2, 8\}$.

Suppose that such a set $\mathcal{S}$ is found. Then the equations

$$D^{(a)}X = 0, \quad a \in \mathcal{S},$$

will imply that

$$x_j = 0 \text{ for all } j \in \bigcup_{a \in \mathcal{S}} \mathcal{J}^{(a)}. \tag{14}$$

Using (14), we can eliminate some of the variables in (12) and obtain the system

$$\widetilde{D}\widetilde{X} = 0$$

(in the example this corresponds to obtaining (9) from (8)).

Let $\tilde{l} = l - |\mathcal{S}|$ be the remaining count of variables x_i, so that $\widetilde{D}$ is an $r\tilde{l} \times r\tilde{l}$ matrix and $\widetilde{X} \in F^{r\tilde{l}}$. We iterate the above steps and define subsets $\widetilde{\mathcal{J}}^{(a)}, \widetilde{\mathcal{U}}^{(a)}$ for all $a \in \{0, 1, \ldots, \tilde{l}-1\}$. In the example the matrix $\widetilde{D}$ is given in (9), and the new set $\widetilde{\mathcal{S}}$ of the groups of r equations is given by $\widetilde{\mathcal{S}} = \{0, 1, 4, 5, 6\}$. Restricting our attention to the equations in these groups, we eliminate another subset of variables by proving that they are necessarily equal to zero, and continue this procedure until finally all of the variables have been shown to be zero.

A rigorous proof uses induction and is given below.

Theorem 6: The code $\mathcal{C}$ given by Construction 1 is an $(n = sm, k = n - r, l = s^m)$ MDS array code.

To prove this theorem we need to show that for every choice of the indices $1 \le i_1 < i_2 < \ldots < i_r \le n$, the only X that satisfies (12) is the all-zero vector. This will follow from the next two lemmas.

Lemma 7: For any $1 \le i_1 < i_2 < \ldots < i_r \le n$,

$$\min_{a \in \{0,1,\ldots,l-1\}} |\mathcal{U}^{(a)}[i_1, i_2, \ldots, i_r]| = r.$$

Proof: As mentioned above, $\{i_1, i_2, \ldots, i_r\} \subseteq \mathcal{U}^{(a)}$ for any $a \in \{0, 1, \ldots, l-1\}$, so $\min_a |\mathcal{U}^{(a)}| \ge r$ (we again simplify the notation by dropping the indices $i_1, \ldots, i_r$). We claim that there always exists an index $a \in \{0, 1, \ldots, l-1\}$ such that $\mathcal{U}^{(a)} = \{i_1, i_2, \ldots, i_r\}$. To see this, define the (possibly empty) set

$$\mathcal{G} = \{v : v \in [m], \{(v-1)s+1, (v-1)s+2, \ldots, vs\} \subseteq \{i_1, i_2, \ldots, i_r\}\}. \quad (15)$$

Now choose $a \in \{0, 1, \ldots, l-1\}$ such that

$$\text{if } v \in [m] \backslash \mathcal{G} \text{ then } (v-1)s + a_v + 1 \notin \{i_1, i_2, \ldots, i_r\}. \quad (16)$$

To see that we can always find such an a, notice that by (15), for every $v \in [m]\backslash\mathcal{G}$, there exists a number $y_v \in \{1, 2, \ldots, s\}$ such that $(v-1)s + y_v \notin \{i_1, i_2, \ldots, i_r\}$. In order to satisfy (16), it suffices to set the v-th digit of a to be $y_v - 1$, i.e., to set $a_v = y_v - 1$.

Next we prove that $\mathcal{U}^{(a)} = \{i_1, i_2, \ldots, i_r\}$, which is equivalent to the following statement:

Claim: For any $i \in \{i_1, i_2, \ldots, i_r\}$, all the nonzero entries of the a-th row of $A_{t,i}$ belong to the set

$$\{\lambda^t_{i_1}, \lambda^t_{i_2}, \ldots, \lambda^t_{i_r}, \gamma\lambda^t_{i_1}, \gamma\lambda^t_{i_2}, \ldots, \gamma\lambda^t_{i_r}\}.$$

Let us write $i = (v-1)s + u + 1$, where $v \in [m]$ and $u \in \{0, 1, \ldots, s-1\}$. We consider the following two cases:

Case 1 ($v \in [m]\backslash\mathcal{G}$): In this case (16) states that $(v-1)s + a_v + 1 \notin \{i_1, i_2, \ldots, i_r\}$, and therefore $(v-1)s + a_v + 1 \ne i$ (keep in mind that $i \in \{i_1, i_2, \ldots, i_r\}$). As a result, $a_v \ne u$. According to (4), if $a_v < u$, then the a-th row of $A_{t,i}$ contains a single nonzero entry λ^t_i; if $a_v > u$, then the a-th row of $A_{t,i}$ contains a single nonzero entry $\gamma\lambda^t_i$. Both λ^t_i and $\gamma\lambda^t_i$ belong to the set $\{\lambda^t_{i_1}, \lambda^t_{i_2}, \ldots, \lambda^t_{i_r}, \gamma\lambda^t_{i_1}, \gamma\lambda^t_{i_2}, \ldots, \gamma\lambda^t_{i_r}\}$. This establishes the claim for this case.

Case 2 ($v \in \mathcal{G}$): If $a_v \ne u$, then the claim holds by the argument in Case 1, so let $a_v = u$. The a-th row of $A_{t,i}$ contains s nonzero entries $\lambda^t_{(v-1)s+1}, \lambda^t_{(v-1)s+2}, \ldots, \lambda^t_{vs}$. By (15), they all belong to $\{\lambda^t_{i_1}, \lambda^t_{i_2}, \ldots, \lambda^t_{i_r}\}$. This establishes the claim for this case and completes the proof of the lemma. ∎

Lemma 8: For every $a \in \{0, 1, \ldots, l-1\}$, $x_j = 0$ for all $j \in \mathcal{J}^{(a)}$.

Proof: We will argue by induction on the cardinality of the set $\mathcal{U}^{(a)}$. By Lemma 7, to establish the induction basis we need to prove that the lemma holds for all a such that $|\mathcal{U}^{(a)}| = r$. Let a be one of the values that have this property. We will prove that for every $t \in [r]$, $x_j = 0$ for all $j \in \mathcal{J}^{(a)}(i_t)$.

Let us write the index $i_t, t \in [r]$ in the form

$$i_t = (v_t - 1)s + u_t + 1,$$

where $v_t \in [m]$ and $0 \le u_t \le s-1$. Let us further partition $[r]$ into three disjoint subsets $\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3$ as follows:

$$\mathcal{K}_1 = \{t : t \in [r], a_{v_t} = u_t\} \cup \{t : t \in [r], \nexists\, p \in [r] \text{ s.t. } v_p = v_t \text{ and } u_p = a_{v_t}\},$$
$$\mathcal{K}_2 = \{t : t \in [r], a_{v_t} > u_t\} \cap \{t : t \in [r], \exists\, p \in [r] \text{ s.t. } v_p = v_t \text{ and } u_p = a_{v_t}\},$$
$$\mathcal{K}_3 = \{t : t \in [r], a_{v_t} < u_t\} \cap \{t : t \in [r], \exists\, p \in [r] \text{ s.t. } v_p = v_t \text{ and } u_p = a_{v_t}\}.$$

We will prove our claim separately for each of these subsets, starting with $\mathcal{K}_1$. By definition (4), all the nonzero entries in the matrix $A_{h,(v-1)s+u+1}$ belong to the set $\{\lambda^h_{(v-1)s+1}, \lambda^h_{(v-1)s+2}, \ldots, \lambda^h_{vs}, \gamma\lambda^h_{(v-1)s+1}, \gamma\lambda^h_{(v-1)s+2}, \ldots, \gamma\lambda^h_{vs}\}$, where $h = 0, 1, \ldots, r-1$. Moreover, if $a_v \ne u$, then the a-th row of the matrix $A_{h,(v-1)s+u+1}$ contains only a single nonzero entry, either $\lambda^h_{(v-1)s+u+1}$ or $\gamma\lambda^h_{(v-1)s+u+1}$. On the other hand, if $a_v = u$, then the a-th row of the matrix $A_{h,(v-1)s+u+1}$ contains s nonzero entries. In particular, if $a_v = u$, then the only appearance of the element $\lambda^h_{(v-1)s+u+1}$ in the a-th row of all the matrices $A_{h,i}, i \in [n]$ is in position (a, a) of $A_{h,(v-1)s+u+1}$, i.e., on the diagonal of $A_{h,(v-1)s+u+1}$ (and $\gamma\lambda^h_{(v-1)s+u+1}$ does not appear in the a-th row of any one of the matrices $A_{h,i}, i \in [n]$). As an immediate consequence, if $a_v = u$, then the column $L_{(v-1)s+u+1}$ appears at most once in the matrix $D^{(a)}$ (it appears when $(v-1)s + u + 1 \in \{i_1, \ldots, i_r\}$), and $\gamma L_{(v-1)s+u+1}$ does not appear in $D^{(a)}$. Therefore, for each $t \in \mathcal{K}_1$, we have $\mathcal{J}^{(a)}(i_t) = \{(t-1)l + a\}$. The vectors $L_{i_j}, j = 1, \ldots, r$ are linearly independent, which implies that $x_{(t-1)l+a} = 0, t \in \mathcal{K}_1$.

Moving to the case $t \in \mathcal{K}_2$, observe that $\mathcal{J}^{(a)}(i_t) = \{(t-1)l + a, (p-1)l + a(v_t, u_t)\}$ which implies that

$$\gamma x_{(t-1)l+a} + x_{(p-1)l+a(v_t,u_t)} = 0. \quad (17)$$

Let us consider the submatrix $D^{(a(v_t,u_t))}$. Its nonzero columns are described as follows:

$$\mathcal{U}^{(a(v_t,u_t))} = \{i_1, i_2, \ldots, i_r\},$$
$$\mathcal{J}^{(a(v_t,u_t))}(i_p) = \{(t-1)l + a, (p-1)l + a(v_t, u_t)\}.$$

From this we see that

$$x_{(t-1)l+a} + x_{(p-1)l+a(v_t,u_t)} = 0. \quad (18)$$

Since $\gamma \ne 1$, conditions (17) and (18) imply that $x_{(t-1)l+a} = x_{(p-1)l+a(v_t,u_t)} = 0$, exhausting the case of $t \in \mathcal{K}_2$.

The last remaining case $t \in \mathcal{K}_3$ is very similar to $\mathcal{K}_2$, and we again obtain that $x_j = 0$ for all $j \in \mathcal{J}^{(a)}(i_t)$. This concludes the proof of the induction basis.

Now suppose that the statement of the lemma holds true for all a such that $|\mathcal{U}^{(a)}| \le w - 1$ for some $w \ge r$ and let us prove it for all a such that $|\mathcal{U}^{(a)}| = w$. We again aim to show that for every $i \in \mathcal{U}^{(a)}$, $x_j = 0$ for all $j \in \mathcal{J}^{(a)}(i)$.

For $i \in \mathcal{U}^{(a)}\backslash\{i_1, i_2, \ldots, i_r\}$, there exists a unique $t \in [r]$ such that $a_{v_t} = u_t$ and $1 \le i - (v_t - 1)s \le s$, and

$\mathcal{J}^{(a)}(i) = \{(t-1)l + a(v_t, \alpha)\}$, where $\alpha = i - (v_t - 1)s - 1$. Consider the matrix $D^{(a(v_t,\alpha))}$. By our choice of i there is no $p \in [r]$ such that $v_p = v_t$ and $u_p = \alpha$. Therefore, $\mathcal{U}^{(a(v_t,\alpha))} \subset \mathcal{U}^{(a)}$ and $i \notin \mathcal{U}^{(a(v_t,\alpha))}$. This implies that $|\mathcal{U}^{(a(v_t,\alpha))}| \leq w - 1$, so the induction hypothesis applies, and $x_j = 0$ for all $j \in \mathcal{J}^{(a(v_t,\alpha))}$. Furthermore, $(t-1)l + a(v_t, \alpha) \in \mathcal{J}^{(a(v_t,\alpha))}$, and thus $x_{(t-1)l+a(v_t,\alpha)} = 0$. Rephrasing this, we have shown that for every $i \in \mathcal{U}^{(a)} \backslash \{i_1, i_2, \ldots, i_r\}$, $x_j = 0$ for all $j \in \mathcal{J}^{(a)}(i)$.

We are left to consider the variables $\cup_{t\in[r]}\{x_j : j \in \mathcal{J}^{(a)}(i_t)\}$. To show that they must be 0 to satisfy $DX = 0$, we note that the left-hand side of $D^{(a)}X = 0$ reduces to a linear combination of the linearly independent columns L_{i_j}, $j = 1, \ldots, r$. Therefore, the coefficients of this linear combination are all 0. This implies that for every $t \in [r]$, $x_j = 0$ for all $j \in \mathcal{J}^{(a)}(i_t)$. This claim is proved in exactly the same way as the induction basis above, so we omit the proof. This completes the induction step. ∎

Corollary 9: The code $\mathcal{C}'$ given by Construction 2 is an $(n = rm + r', k = n - r, l = r^{m+1})$ MDS array code.

Proof: Consider the MDS code $\mathcal{C}$ of length $n = r(m+1)$ given by Construction 1. The code $\mathcal{C}'$ is obtained from $\mathcal{C}$ by discarding $r - r'$ coordinates, and so is also MDS. ∎

We finish this section by a brief remark on the complexity of node repair (decoding) and encoding of the constructed codes. From the coding-theoretic perspective, the repair problem is erasure correction, and is very similar to the encoding problem (the encoding is correction of $r = n - k$ erasures from k known nodes). From the example in Section III-A and the proof of Theorem 6, we immediately see that the encoding and decoding procedures of codes given by either of Constructions 1 or 2 rely on inversion of $r \times r$ matrices over F, and thus have low complexity.

IV. The Optimal Access Property

Consider the code $\mathcal{C}$ given by Construction 1. Recall that we write the i-th node as $C_i = (c_{i,0}, c_{i,1}, \ldots, c_{i,l-1})^T$. For every $i \in [n]$, define the following set of coordinates of the i-th node:

$$\mathcal{C}_i^{(v,u)} = \{c_{i,a} : a \in \{0, 1, \ldots, l-1\}, a_v = u\}. \tag{19}$$

where a_v is the v-th digit of a.

Theorem 10: Consider the code $\mathcal{C}$ given by Construction 1. For any $v \in [m]$ and $u \in \{0, 1, \ldots, s-1\}$, the node $C_{(v-1)s+u+1}$ can be recovered from the elements in the set

$$\mathcal{C}^{(v,u)} = \bigcup_{\substack{i\in[n] \\ i\neq(v-1)s+u+1}} \mathcal{C}_i^{(v,u)}.$$

Proof: Fix a value of t. Let us write out the a-th row of the equation $\sum_{i=1}^n A_{t,i}C_i = 0$ (cf. (3)). We first notice that since $n = sm$, the equation $\sum_{i=1}^n A_{t,i}C_i = 0$ is equivalent to $\sum_{q=1}^m \sum_{w=0}^{s-1} A_{t,(q-1)s+w+1}C_{(q-1)s+w+1} = 0$. By definition (4), if $a_q > w$, then the a-th row of $A_{t,(q-1)s+w+1}$ contains a single nonzero entry $\gamma\,\lambda^t_{(q-1)s+w+1}$ located in the a-th column; if $a_q < w$, then the a-th row of $A_{t,(q-1)s+w+1}$ contains a single nonzero entry $\lambda^t_{(q-1)s+w+1}$ located in the a-th column; if $a_q = w$, then the a-th row of $A_{t,(q-1)s+w+1}$ contains s nonzero entries located in columns $a(q,0)$ to $a(q,s-1)$. Thus, the a-th row of the equation $\sum_{i=1}^n A_{t,i}C_i = 0$ can be written as follows:

$$\sum_{q\in[m]} \Big(\sum_{w=0}^{a_q-1} \gamma\,\lambda^t_{(q-1)s+w+1} c_{(q-1)s+w+1,a} + \sum_{w=0}^{s-1} \lambda^t_{(q-1)s+w+1} c_{(q-1)s+a_q+1,a(q,w)} + \sum_{w=a_q+1}^{s-1} \lambda^t_{(q-1)s+w+1} c_{(q-1)s+w+1,a} \Big) = 0, \tag{20}$$

where the first sum in the parentheses corresponds to the case $a_q > w$; the second sum corresponds to the case $a_q = w$; and the third sum corresponds to the case $a_q < w$. Since our aim is to repair the node $C_{(v-1)s+u+1}$, let us break the sum on $q \in [m]$ in (20) into two parts: $q \neq v$ and $q = v$. We obtain the equation given in (21), as shown at the bottom of this page. For all $t = 0, 1, \ldots, r-1$ and all a satisfying $a_v = u$, all the terms in (21) apart from the underlined term can be found from the elements in the set $\mathcal{C}^{(v,u)}$. Indeed, on account of (22), as shown at the bottom of this page, and the fact that $r \geq s$, the coordinates in the set $\{c_{(v-1)s+u+1,a(v,w)} : w = 0, 1, \ldots, s-1\}$ can be found from the set $\{\sum_{w=0}^{s-1} \lambda^t_{(v-1)s+w+1} c_{(v-1)s+u+1,a(v,w)} :$

$$\sum_{q\neq v, q\in[m]} \Big(\sum_{w=0}^{a_q-1} \gamma\,\lambda^t_{(q-1)s+w+1} c_{(q-1)s+w+1,a} + \sum_{w=0}^{s-1} \lambda^t_{(q-1)s+w+1} c_{(q-1)s+a_q+1,a(q,w)} + \sum_{w=a_q+1}^{s-1} \lambda^t_{(q-1)s+w+1} c_{(q-1)s+w+1,a} \Big)$$
$$+ \Big(\sum_{w=0}^{a_v-1} \gamma\,\lambda^t_{(v-1)s+w+1} c_{(v-1)s+w+1,a} + \underline{\sum_{w=0}^{s-1} \lambda^t_{(v-1)s+w+1} c_{(v-1)s+a_v+1,a(v,w)}} + \sum_{w=a_v+1}^{s-1} \lambda^t_{(v-1)s+w+1} c_{(v-1)s+w+1,a} \Big) = 0 \tag{21}$$

$$\begin{bmatrix} \sum_{w=0}^{s-1} c_{(v-1)s+u+1,a(v,w)} \\ \sum_{w=0}^{s-1} \lambda_{(v-1)s+w+1} c_{(v-1)s+u+1,a(v,w)} \\ \vdots \\ \sum_{w=0}^{s-1} \lambda^{r-1}_{(v-1)s+w+1} c_{(v-1)s+u+1,a(v,w)} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_{(v-1)s+1} & \lambda_{(v-1)s+2} & \ldots & \lambda_{(v-1)s+s} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda^{r-1}_{(v-1)s+1} & \lambda^{r-1}_{(v-1)s+2} & \ldots & \lambda^{r-1}_{(v-1)s+s} \end{bmatrix} \begin{bmatrix} c_{(v-1)s+u+1,a(v,0)} \\ c_{(v-1)s+u+1,a(v,1)} \\ \vdots \\ c_{(v-1)s+u+1,a(v,s-1)} \end{bmatrix}. \tag{22}$$

$t = 0, 1, \ldots, r-1\}$ for all $a = 0, 1, \ldots, l-1$. In particular, the values in the set

$$\{c_{(v-1)s+u+1,a} : a = 0, 1, \ldots, l-1\} = \{c_{(v-1)s+u+1,a(v,w)} : a_v = u, w = 0, 1, \ldots, s-1\}$$

can be found from the values $\left\{\sum_{w=0}^{s-1} \lambda_{(v-1)s+w+1}^{t} c_{(v-1)s+u+1,a(v,w)} : a_v = u, t = 0, 1, \ldots, r-1\right\}$. As mentioned above, the values $\{\sum_{w=0}^{s-1} \lambda_{(v-1)s+w+1}^{t} c_{(v-1)s+u+1,a(v,w)} : a_v = u, t = 0, 1, \ldots, r-1\}$ are uniquely determined by the elements in the set $\mathcal{C}^{(v,u)}$. We conclude that the entire node $C_{(v-1)s+u+1}$ can be determined by the elements in $\mathcal{C}^{(v,u)}$. ∎

Theorem 11: Let $s = r$, then the code $\mathcal{C}$ given by Construction 1 has the optimal access property.

Proof: Since $\mathcal{C}_i^{(v,u)}$ contains exactly a $(1/r)$th fraction of the coordinates of C_i, by Theorem 10 we only need to access a $1/r$ proportion of the data stored in each of the surviving nodes in order to repair a single node failure. ∎

Theorem 12: The code $\mathcal{C}'$ given by Construction 2 has the optimal access property.

Proof: Using the same approach as in Theorem 10, we can show that for all $v \in [m]$ and $u \in \{0, 1, \ldots, r-1\}$, the node $C_{(v-1)r+u+1}$ can be determined by the elements in the set $\{c_{i,a} : i \neq (v-1)r + u + 1, a_v = u\}$, and that for all $u \in \{0, 1, \ldots, r'-1\}$, the node C_{mr+u+1} can be determined by the elements in the set $\{c_{i,a} : i \neq mr + u + 1, a_{m+1} = u\}$. ∎

Note that upon setting $s = r$ in Construction 1, we obtain optimal-access MDS array codes with code length divisible by r, and Construction 2 gives optimal-access MDS array codes with code length not divisible by r. Thus we can construct optimal-access $(n, n-r, r^{\lceil n/r \rceil})$ MDS array codes for any n and r.

A. Group Optimal Access

In the second part of this section we examine the group optimal access property of the codes considered above and prove the following result.

Theorem 13: The code $\mathcal{C}$ given by Construction 1 has the $(s, s+k-1)$-group optimal access property.

This theorem will follow from Theorem 10 and a lemma proved below in this section.

Let us define the following subset of indices

$$\mathcal{N}^{(v)} = [n] \backslash \{(v-1)s+1, (v-1)s+2, \ldots, vs\}, \quad v \in [m].$$

Lemma 14: Consider the code $\mathcal{C}$ given by Construction 1. For every $v \in [m]$ and $u \in \{0, 1, \ldots, s-1\}$, and every $\mathcal{M} \subseteq \mathcal{N}^{(v)}$ with cardinality $|\mathcal{M}| = k$, the values of elements in the set $\cup_{i \in \mathcal{N}^{(v)}} \mathcal{C}_i^{(v,u)}$ (cf. (19)) are determined by the elements in the set $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$.

Before proving Lemma 14, let us explain how this lemma together with Theorem 10 implies Theorem 13. The group optimal access property simply means that for every $v \in [m]$ and every $u \in \{0, 1, \ldots, s-1\}$, the node $C_{(v-1)s+u+1}$ can be repaired by connecting to $\{C_{(v-1)s+u'+1} : u' \in \{0, 1, \ldots, s-1\} \setminus \{u\}\}$ together with any other k helper nodes in the set $\{C_i : i \in \mathcal{N}^{(v)}\}$, and accessing exactly $(1/s)$th fraction of coordinates of each helper node. In Theorem 10, we have shown that the node $C_{(v-1)s+u+1}$ can be repaired if we know the values of all the elements in the set $\bigcup_{i \neq (v-1)s+u+1} \mathcal{C}_i^{(v,u)}$ from all the surviving nodes. By definition each set $\mathcal{C}_i^{(v,u)}$ contains exactly $(1/s)$th fraction of the coordinates of C_i. This is where we need Lemma 14 which states that the values of the elements in the set $\cup_{i \in \mathcal{N}^{(v)}} \mathcal{C}_i^{(v,u)}$ (cf. (19)) can be calculated from the elements in the set $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$ for any $\mathcal{M} \subseteq \mathcal{N}^{(v)}$ such that $|\mathcal{M}| = k$. This establishes that the code $\mathcal{C}$ given by Construction 1 has the $(s, s+k-1)$-group optimal access property.

Proof of Lemma 14: The case $s = r$ is trivially true, so we assume that $s < r$. Let us write (3) in matrix form:

$$\begin{bmatrix} A_{0,1} & A_{0,2} & \ldots & A_{0,n} \\ A_{1,1} & A_{1,2} & \ldots & A_{1,n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{r-1,1} & A_{r-1,2} & \ldots & A_{r-1,n} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} = 0. \quad (23)$$

Let us permute the equations in this system using the permutation matrix defined in (11) and denote the resulting matrix of coefficients by H. As before, let $H^{(a)}, a = 0, 1, \ldots, l-1$ be a submatrix of H formed of rows $ar, ar+1, \ldots, (a+1)r-1$. Writing out the equation $H^{(a)}[C_1, C_2, \ldots, C_n]^T = 0$ in full form, we obtain (24), as shown at the bottom of this page, where the vectors $L_i, i \in [n]$ are defined in (13) (this equation amounts to taking columns $(v-1)s+1, (v-1)s+2, \ldots, vs$ to the right-hand side of (23)).

Define polynomials $g_0^{(v)}(x) = \prod_{w=1}^{s} (x - \lambda_{(v-1)s+w})$, and $g_j^{(v)}(x) = x^j g_0^{(v)}(x)$ for $j = 0, 1, \ldots, r-s-1$. Since the degree of $g_j^{(v)}(x)$ is less than r for all $j = 0, 1, \ldots, r-s-1$, we can write

$$g_j^{(v)}(x) = \sum_{t=0}^{r-1} g_{j,t}^{(v)} x^t.$$

Define the $(r-s) \times r$ matrix

$$G^{(v)} = \begin{bmatrix} g_{0,0}^{(v)} & g_{0,1}^{(v)} & \cdots & g_{0,r-1}^{(v)} \\ g_{1,0}^{(v)} & g_{1,1}^{(v)} & \cdots & g_{1,r-1}^{(v)} \\ \vdots & \vdots & \vdots & \vdots \\ g_{r-s-1,0}^{(v)} & g_{r-s-1,1}^{(v)} & \cdots & g_{r-s-1,r-1}^{(v)} \end{bmatrix}.$$

$$\sum_{q \in [m] \backslash v} \Big(\sum_{w=0}^{a_q - 1} \gamma\, c_{(q-1)s+w+1,a} L_{(q-1)s+w+1} + \sum_{w=0}^{s-1} c_{(q-1)s+a_q+1,a(q,w)} L_{(q-1)s+w+1} + \sum_{w=a_q+1}^{s-1} c_{(q-1)s+w+1,a} L_{(q-1)s+w+1} \Big)$$
$$= -\Big(\sum_{w=0}^{a_v - 1} \gamma\, c_{(v-1)s+w+1,a} L_{(v-1)s+w+1} + \sum_{w=0}^{s-1} c_{(v-1)s+a_v+1,a(v,w)} L_{(v-1)s+w+1} + \sum_{w=a_v+1}^{s-1} c_{(v-1)s+w+1,a} L_{(v-1)s+w+1} \Big) \quad (24)$$

We have

$$G^{(v)}L_i = \begin{bmatrix} g_{0,0}^{(v)} & g_{0,1}^{(v)} & \cdots & g_{0,r-1}^{(v)} \\ g_{1,0}^{(v)} & g_{1,1}^{(v)} & \cdots & g_{1,r-1}^{(v)} \\ \vdots & \vdots & \vdots & \vdots \\ g_{r-s-1,0}^{(v)} & g_{r-s-1,1}^{(v)} & \cdots & g_{r-s-1,r-1}^{(v)} \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_i \\ \vdots \\ \lambda_i^{r-1} \end{bmatrix} = \hat{L}_i^{(v)}, \quad (25)$$

where $\hat{L}_i^{(v)}, i \in [n]$ is given by

$$\hat{L}_i^{(v)} = \begin{bmatrix} g_0^{(v)}(\lambda_i) \\ g_1^{(v)}(\lambda_i) \\ \vdots \\ g_{r-s-1}^{(v)}(\lambda_i) \end{bmatrix} = g_0^{(v)}(\lambda_i) \begin{bmatrix} 1 \\ \lambda_i \\ \vdots \\ \lambda_i^{r-s-1} \end{bmatrix}. \quad (26)$$

By definition, $g_0^{(v)}(\lambda_i) = 0$ for all $(v-1)s+1 \le i \le vs$, and so $G^{(v)}L_i = 0$ for all $(v-1)s+1 \le i \le vs$. Observe that every term on the right-hand-side of (24) contains one column vector from the set $\{L_i : (v-1)s+1 \le i \le vs\}$. As a result, multiplying equation (24) by $G^{(v)}$ on the left, we obtain

$$\sum_{q \ne v, q \in [m]} \Big(\sum_{w=0}^{a_q-1} \gamma\, c_{(q-1)s+w+1,a} G^{(v)} L_{(q-1)s+w+1} + \sum_{w=0}^{s-1} c_{(q-1)s+a_q+1,a(q,w)} G^{(v)} L_{(q-1)s+w+1} + \sum_{w=a_q+1}^{s-1} c_{(q-1)s+w+1,a} G^{(v)} L_{(q-1)s+w+1} \Big) = 0.$$

Using (25), this equation can be written as

$$\sum_{q \ne v, q \in [m]} \Big(\sum_{w=0}^{a_q-1} \gamma\, c_{(q-1)s+w+1,a} \hat{L}^{(v)}_{(q-1)s+w+1} + \sum_{w=0}^{s-1} c_{(q-1)s+a_q+1,a(q,w)} \hat{L}^{(v)}_{(q-1)s+w+1} + \sum_{w=a_q+1}^{s-1} c_{(q-1)s+w+1,a} \hat{L}^{(v)}_{(q-1)s+w+1} \Big) = 0. \quad (27)$$

In order to prove the theorem, we only need to prove that given any $v \in [m]$ and $u \in \{0, 1, \ldots, s-1\}$, and any $i_1 < i_2 < \ldots < i_{r-s}$ such that $\{i_1, i_2, \ldots, i_{r-s}\} \subseteq \mathcal{N}^{(v)}$, the values of elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$ can be determined by the elements in the set $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$, where $\mathcal{M} = \mathcal{N}^{(v)} \backslash \{i_1, i_2, \ldots, i_{r-s}\}$. We will prove that we can find the elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$ from $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$ using equation (27).

For a given $a = 0, 1, \ldots, l-1$, denote by $E^{(a)}$ the set of equations in (27). Each set $E^{(a)}$ contains $r-s$ scalar equations. Observe that if $a_v = u$, then $E^{(a)}$ contains only elements in $\cup_{i \in \mathcal{N}^{(v)}} \mathcal{C}_i^{(v,u)}$. Moreover, every element in the set $\cup_{i \in \mathcal{N}^{(v)}} \mathcal{C}_i^{(v,u)}$ appears at least once in the equations $\{E^{(a)} : a_v = u\}$. Therefore, equations (27) contain $(r-s)l/s$ scalar equations and $(r-s)l/s$ unknown elements, namely, the elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$.

Let us set all the elements in the set $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$ to 0 in $E^{(a)}$ and denote by $E_{\mathcal{M}}^{(a)}$ the obtained set of equations. (In other words, $E_{\mathcal{M}}^{(a)}$ are the equations obtained by eliminating all the terms which contain elements in the set $\cup_{i \in \mathcal{M}} \mathcal{C}_i^{(v,u)}$ in (27).) In order to prove the lemma, it suffices to show that the equations $\{E_{\mathcal{M}}^{(a)} : a_v = u\}$ imply that all the elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$ are 0.

Recall that equation (12) can be viewed as the set of equations $\{D^{(a)}[i_1, i_2, \ldots, i_r]X = 0 : a \in \{0, 1, \ldots, l-1\}\}$. Note that, once we form a vector X of the elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$, equations $\{E_{\mathcal{M}}^{(a)} : a_v = u\}$ have almost the same form as $\{D^{(a)}[i_1, i_2, \ldots, i_r]X = 0 : a \in \{0, 1, \ldots, l-1\}\}$. The only difference is that in the equations $\{E_{\mathcal{M}}^{(a)} : a_v = u\}$, the columns $\{\hat{L}_i^{(v)} : i \in \mathcal{N}^{(v)}\}$ take place of the vectors $\{L_i : i \in [n]\}$, and $r-s$ takes place of r.

Examining closely the proof of Theorem 6, we note that the property that any r columns in the set $\{L_i, i \in [n]\}$ are linearly independent, suffices to show that $X = 0$. Since $p_0^{(v)}(\lambda_i) \ne 0$ for any $i \in \mathcal{N}^{(v)}$, by Definition (26) any $(r-s)$ vectors in the set of vectors $\{\hat{L}_i^{(v)} : i \in \mathcal{N}^{(v)}\}$ are also linearly independent. Thus, using exactly the same arguments as in the proof of Theorem 6, we can show that the equations $\{E_{\mathcal{M}}^{(a)} : a_v = u\}$ imply that all the elements in the set $\cup_{t=1}^{r-s} \mathcal{C}_{i_t}^{(v,u)}$ are 0. This completes the proof of the theorem. ■

V. Conclusion

In this paper we presented an explicit construction of optimal-access MDS codes with nearly optimal sub-packetization $l = r^{\lceil n/r \rceil}$ and field size $|F| \ge r\lceil n/r \rceil$, which is just slightly greater than n. It is shown in [11] that if we only require systematic optimal repair property instead of optimal access property, there exist codes with even smaller sub-packetization $l = r^{\lceil k/(r+1) \rceil}$ if the base field F is sufficiently large. For $r = 2$ and 3, explicit constructions achieving this sub-packetization over fields of size linear in k are given in [11] and [17], respectively. It is an interesting open problem to give explicit constructions achieving this sub-packetization over small fields for general values of r.

References

[1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[2] M. Blaum, P. G. Farell, and H. van Tilborg, "Array codes," in *Handbook of Coding Theory*, vol. 2, V. Pless and W. C. Huffman, Eds. Amsterdam, The Netherlands: Elsevier, 1998, ch. 22, pp. 1855–1909.

[3] V. Cadambe and A. Mazumdar, "Alphabet-size dependent bounds for exact repair in distributed storage," in *Proc. IEEE Inf. Theory Workshop-Fall (ITW)*, Oct. 2015, pp. 1–3.

[4] 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. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.

[5] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2134–2158, Apr. 2012.

[6] Y. Wu and A. G. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun./Jul. 2009, pp. 2276–2280.

[7] V. R. Cadambe, C. Huang, and J. Li, "Permutation code: Optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 1225–1229.

[8] V. R. Cadambe, C. Huang, J. Li, and S. Mehrotra, "Polynomial length MDS codes with optimal repair in distributed storage," in *Proc. 45th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2011, pp. 1850–1854.
[9] D. Papailiopoulos, A. G. Dimakis, and V. Cadambe, "Repair optimal erasure codes through Hadamard designs," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3021–3037, May 2013.
[10] I. Tamo, Z. Wang, and J. Bruck, "Zigzag codes: MDS array codes with optimal rebuilding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1597–1616, Mar. 2013.
[11] Z. Wang, I. Tamo, and J. Bruck, "Explicit minimum storage regenerating codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4466–4480, Aug. 2016.
[12] Z. Wang, I. Tamo, and J. Bruck, "On codes for optimal rebuilding access," in *Proc. 49th Annu. Allerton Conf. Commun., Control Comput.*, Sep. 2011, pp. 1374–1381.
[13] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2001–2014, Apr. 2017.
[14] I. Tamo, Z. Wang, and J. Bruck, "Access versus bandwidth in codes for storage," *IEEE Trans. Inf. Theory*, vol. 60, no. 4, pp. 2028–2037, Apr. 2014.
[15] B. Sasidharan, G. K. Agarwal, and P. V. Kumar, "A high-rate MSR code with polynomial sub-packetization level," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 2051–2055.
[16] A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath. (2016). "Progress on high-rate MSR codes: Enabling arbitrary number of helper nodes." [Online]. Available: https://arxiv.org/abs/1601.06362
[17] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of high-rate minimum storage regenerating codes over small fields," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2015–2038, Apr. 2017.
[18] V. Guruswami and M. Wootters, "Repairing Reed–Solomon codes," *IEEE Trans. Inf. Theory*, to be published, doi: 10.1109/TIT.2017.2702660.
[19] M. Ye and A. Barg, "Explicit constructions of MDS array codes and RS codes with optimal repair bandwidth," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1202–1206.
[20] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: Achieving the cut-set bound," in *Proc. 58th Ann. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2017, to be published. [Online], Available: https://arxiv.org/abs/1706.00112
[21] D. S. Papailiopoulos and A. G. Dimakis, "Locally repairable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5843–5855, Oct. 2014.
[22] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, Aug. 2014.
[23] N. Silberstein, A. S. Rawat, O. O. Koyluoglu, and S. Vishwanath, "Optimal locally repairable codes via rank-metric codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 1819–1823.
[24] I. Ahmad and C.-C. Wang. (2016) "When can helper node selection improve regenerating codes? Part I: Graph-based analysis." [Online]. Available: https://arxiv.org/abs/1604.08231
[25] I. Ahmad and C.-C. Wang. (2016). "When can helper node selection improve regenerating codes? Part II: An explicit exact-repair code construction." [Online]. Available: https://arxiv.org/abs/1604.08230
[26] G. M. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4637–4660, Aug. 2014.
[27] H. D. L. Hollmann, "On the minimum storage overhead of distributed storage codes with a given repair locality," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 1041–1045.
[28] B. Sasidharan, M. Vajha, and P. V. Kumar. (2016). "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair." [Online]. Available: https://arxiv.org/abs/1607.07335

Min Ye received the B.S. degree in Electrical Engineering from Peking University, Beijing, China in 2012. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Maryland, College Park. His research interests include coding theory and information theory.

Alexander Barg (M'00–SM'01–F'08) received the M.Sc. degree in applied mathematics and the Ph.D. degree in electrical engineering, the latter from the Institute for Information Transmission Problems (IPPI) Moscow, Russia, in 1987. He has been a Senior Researcher at the IPPI since 1988. Since 2003 he has been a Professor in the Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park.

Alexander Barg was a co-recipient of the IEEE Information Theory Society Paper Award in 2015. He was the Technical Program Co-Chair of the 2006 IEEE International Symposium on Information Theory and of 2010 and 2015 IEEE ITWs. He is currently Executive Editor of IEEE TRANSACTIONS ON INFORMATION THEORY and serves on the Editorial Board of *Problems of Information Transmission* and *Advances in Mathematics of Communications*.

Alexander Barg's research interests are in coding and information theory, signal processing, and algebraic combinatorics.