Achieving Maximum Distance Separable Private Information Retrieval Capacity With Linear Codes

Citation for the original published paper (version of record):
Kumar, S., Lin, H., Rosnes, E. et al (2019). Achieving Maximum Distance Separable Private Information Retrieval Capacity With Linear Codes. IEEE Transactions on Information Theory, 65(7): 4243-4273. http://dx.doi.org/10.1109/TIT.2019.2900313

Achieving Maximum Distance Separable Private Information Retrieval Capacity With Linear Codes

Siddhartha Kumar, *Student Member, IEEE*, Hsuan-Yin Lin, *Senior Member, IEEE*,
Eirik Rosnes, *Senior Member, IEEE*, and Alexandre Graell i Amat, *Senior Member, IEEE*

***Abstract*—We propose three private information retrieval (PIR) protocols for distributed storage systems (DSSs) where data is stored using an arbitrary linear code. The first two protocols, named Protocol 1 and Protocol 2, achieve privacy for the scenario with noncolluding nodes. Protocol 1 requires a file size that is exponential in the number of files in the system, while Protocol 2 requires a file size that is independent of the number of files and is hence simpler. We prove that, for certain linear codes, Protocol 1 achieves the maximum distance separable (MDS) PIR capacity, i.e., the maximum PIR rate (the ratio of the amount of retrieved stored data per unit of downloaded data) for a DSS that uses an MDS code to store any given (finite and infinite) number of files, and Protocol 2 achieves the *asymptotic* MDS-PIR capacity (with infinitely large number of files in the DSS). In particular, we provide a necessary and a sufficient condition for a code to achieve the MDS-PIR capacity with Protocols 1 and 2 and prove that cyclic codes, Reed-Muller (RM) codes, and a class of distance-optimal local reconstruction codes achieve both the *finite* MDS-PIR capacity (i.e., with any given number of files) and the asymptotic MDS-PIR capacity with Protocols 1 and 2, respectively. Furthermore, we present a third protocol, Protocol 3, for the scenario with multiple colluding nodes, which can be seen as an improvement of a protocol recently introduced by Freij-Hollanti *et al.*. Similar to the noncolluding case, we provide a necessary and a sufficient condition to achieve the maximum possible PIR rate of Protocol 3. Moreover, we provide a particular class of codes that is suitable for this protocol and show that RM codes achieve the maximum possible PIR rate for the protocol. For all three protocols, we present an algorithm to optimize their PIR rates.**

***Index Terms*—Code automorphisms, colluding servers, generalized Hamming weight, distributed storage, linear codes, local reconstruction codes, Reed-Muller codes, private information retrieval.**

I. Introduction

In data storage applications, besides resilience against disk failures and data protection against illegitimate users, the privacy may also be of concern. For instance, one may be interested in designing a storage system in which a file can be downloaded without revealing any information of which file is actually downloaded to the servers storing it. This form of privacy is usually referred to as *private information retrieval* (PIR). PIR is important to, e.g., protect users from surveillance and monitoring.

PIR protocols were first studied in the computer science literature by Chor *et al.* in [1], [2], which introduced the concept of an n-server PIR protocol, where a binary storage node is replicated among n servers (referred to as nodes) and the aim is to privately retrieve a single bit from the storage nodes while minimizing the total upload and download communication cost. Additionally, an n-server PIR protocol assumes that the n nodes do not collude in order to reveal the identity of the requested bit. The communication cost in [1] was further reduced in [3]–[5]. Since then, coded PIR schemes have been introduced, where data is encoded (as opposed to simply being replicated) across several nodes [6]. With the advent of distributed storage systems (DSSs), where the user data is encoded and then stored on n nodes, there has been an increasing interest in implementing coded PIR protocols for these systems.

In recent years PIR has become an active research area in the information theory community with a fundamental difference in the measurement of efficiency. In the information-theoretic sense, the message sizes are much larger than the size of all queries sent to the storage nodes. Thus, rather than accounting for both the upload and the download cost, efficiency is measured in terms of download cost only as the upload cost can be neglected. The ratio of the requested file size to the amount of downloaded data is referred to as the PIR rate, where a higher PIR rate means a higher efficiency. The highest achievable PIR rate for any n-server PIR protocol is referred to as the PIR capacity.

In the information theory literature, the authors in [7] were the first to present PIR protocols for DSSs where data is stored using codes from two explicit linear code constructions. In [8], the authors presented upper bounds on the tradeoff between the storage and the PIR rates for a certain class of linear PIR protocols. In [9], Fazeli *et al.* introduced PIR codes which, when used in conjunction with traditional n-server PIR protocols, allow to achieve PIR on DSSs. These codes achieve high code rates without sacrificing on the communication cost of an n-server PIR protocol. In [10], given an arbitrary number of files, the authors derived the PIR capacity for noncolluding and replicated databases, where the data can be seen as being encoded by a trivial class of maximum distance separable (MDS) codes, i.e., repetition codes. For the case of noncolluding nodes, Banawan and Ulukus [11] derived the

This work was partially funded by the Research Council of Norway (grant 240985/F20) and the Swedish Research Council (grant #2016-04253). This paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, June 2017, at the IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, June 2018, and at the IEEE Information Theory Workshop (ITW), Guangzhou, China, November 2018.

S. Kumar, H.-Y. Lin, and E. Rosnes are with Simula UiB, N-5008 Bergen, Norway (e-mail: kumarsi@simula.no; lin@simula.no; eirikrosnes@simula.no).

A. Graell i Amat is with the Department of Electrical Engineering, Chalmers University of Technology, SE-41296 Gothenburg, Sweden (e-mail: alexandre.graell@chalmers.se).

PIR capacity for DSSs using an $[n, k]$ MDS code to store a given number of files, referred to as the MDS-PIR capacity. In this paper, we will refer to the MDS-PIR capacity for a given finite number of files as the *finite MDS-PIR capacity*, and to the MDS-PIR capacity for an infinite number of files as the *asymptotic MDS-PIR capacity*. In [12], a PIR protocol for MDS-coded DSSs and noncolluding nodes was proposed and shown to achieve the asymptotic MDS-PIR capacity. PIR protocols for the case of colluding nodes were proposed in [12]–[15]. The MDS-PIR capacity for the colluding case is still unknown in general, except for some special cases [16] and for repetition codes [17]. The problem of *symmetric* PIR for DSSs was recently considered in [18], where an expression for the symmetric PIR capacity for linear schemes in the general case of colluding nodes and an MDS linear storage code was derived. In the symmetric case, the user should not only be able to privately retrieve the requested file from the system, but also learn nothing about the other files stored from the retrieved data. See also the related work [19], [20], which deals with replicated databases. The PIR capacity for the case where a given number of storage nodes fail to respond (so-called robust PIR) was given in [17] for the scenario of colluding servers with replication coding.

In the storage community, it is well known that MDS codes are inefficient in the repair of failed nodes. In particular, they have large repair locality, i.e., the repair of a failed node requires contacting a large number of nodes.[1] Repair is essential to maintain the initial state of reliability of the DSS. To address low repair locality, Pyramid codes [23], locally repairable codes [24], local reconstruction codes (LRCs) [25], [26], and locally recoverable codes [27] are some non-MDS codes that have been proposed. These four classes of codes follow the same design philosophy and for simplicity, we will refer to them generically as LRCs. Following the motivation of using non-MDS codes in DSSs, the authors of [28] presented a PIR protocol for DSSs that store data using arbitrary linear codes for the scenario of noncolluding nodes. Independently, Freij-Hollanti *et al.* in [29] presented a PIR protocol that ensures privacy even when a subset of at most $n - k$ nodes collude. The protocol is based on two codes, the storage code and the *query code*, which defines the queries. The retrieval process is then characterized by the *retrieval code*, which is the Hadamard product of these two codes. The PIR rate of the protocol is upperbounded by $(n - \tilde{k})/n$, where $\tilde{k}$ is the dimension of the retrieval code. The authors showed that with generalized Reed-Solomon (GRS) codes for the storage and query codes, the upper bound on the PIR rate is achieved. To the best of our knowledge, in the asymptotic regime when the number of files tends to infinity, the PIR rate $(n - \tilde{k})/n$ is the highest achievable PIR rate known so far. Moreover, they showed that their protocol could work with certain non-MDS codes. However, for non-MDS codes (e.g., Reed-Muller (RM) codes where considered in [30]) the PIR rates that can be achieved by the protocol in [29] are lower than the upper bound $(n - \tilde{k})/n$.

[1]In a parallel line of work, schemes for efficient repair (in terms of repair bandwidth) of Reed-Solomon codes have been proposed [21], [22].

In this paper, as an extension of [28], we present three PIR protocols for DSSs using arbitrary linear codes. These protocols share the fact that all of them are constructed by making use of correctable erasure patterns and information sets of the underlying storage code. We first focus on the noncolluding scenario and propose two PIR protocols, referred to as Protocol 1 and Protocol 2. Protocol 1 requires a file size that is exponential in the number of files in the system, while Protocol 2 requires a file size that is independent of the number of files and is therefore simpler. Furthermore, Protocol 1 is designed such that its PIR rate depends on the number of files in the system, while Protocol 2 is such that its PIR rate is independent of the number of files. We prove that, interestingly, for certain non-MDS code families, Protocol 1 achieves the finite MDS-PIR capacity (and hence the asymptotic MDS-PIR capacity as well) and Protocol 2 achieves the asymptotic MDS-PIR capacity. Thus, we show that the MDS property required to achieve the MDS-PIR capacity in [10]–[12] is not necessary and is overly restrictive. In particular, we give a sufficient condition based on code automorphisms and a necessary condition connected to the generalized Hamming weights of the underlying storage code to achieve the MDS-PIR capacity for Protocols 1 and 2. We prove that cyclic codes, RM codes, and distance-optimal information locality codes achieve the finite MDS-PIR capacity (and thus the asymptotic MDS-PIR capacity, too) with Protocol 1 and the asymptotic MDS-PIR capacity with Protocol 2. For other codes, we present an optimization algorithm for Protocols 1 and 2 to optimize their PIR rates.

We furthermore present a third protocol, Protocol 3, for the scenario of multiple colluding nodes and non-MDS storage codes. This protocol is based on and improves the protocol in [29], [30], in the sense that it achieves higher PIR rates. We extend the necessary and the sufficient condition from the noncolluding case to provide joint conditions on the storage and query codes to achieve the upper bound $(n-\tilde{k})/n$ on the PIR rate of Protocol 3. Moreover, we show that Protocol 3 achieves the upper bound $(n - \tilde{k})/n$ on the PIR rate for RM codes and some non-MDS codes. We also provide an optimization algorithm for the protocol to optimize the PIR rate. Such an optimization is in itself an extension of the optimization algorithm for Protocols 1 and 2 for the case of noncolluding nodes. Besides GRS and RM codes as in [29], [30], we also prove that $(\mathcal{U}|\mathcal{U}+\mathcal{V})$ codes [31] with $\mathcal{U}$ being an arbitrary binary linear code and $\mathcal{V}$ a binary repetition code can be used in conjunction with Protocol 3. We finally give examples of all-symbol locality LRCs with good PIR rates.

The main contributions of the paper are summarized in the following:

- For the noncolluding case, we propose two PIR protocols, Protocol 1 and Protocol 2 (Sections IV and V), and provide a necessary and a sufficient condition for a code to achieve the MDS-PIR capacity with these protocols (Theorems 3 and 4, respectively, in Section VI).
- For the noncolluding case, we show that important classes of non-MDS codes, namely cyclic codes, RM codes, and distance-optimal information locality codes achieve the finite MDS-PIR capacity and the asymptotic MDS-PIR

capacity with Protocols 1 and 2, respectively (Corollaries 7, 8, and Theorem 5, respectively, in Section VI).
- For the colluding case, we propose Protocol 3 that achieves higher asymptotic PIR rates for non-MDS codes (equal to its upper bound) than the best known protocol [29], [30]. Similar to the noncolluding case, a necessary and a sufficient condition for the protocol to achieve PIR rates equal to its upper bound is provided (Corollary 10 and Theorem 8, respectively, in Section VIII). Independently of this work, by using an approach similar to ours, in [32] the authors modified the protocol in [30] and showed that the PIR rate $(n-\tilde{k})/n$ is achievable for *transitive codes*.[2] However, the protocol in [32] requires a much larger number of stripes and query sizes than our proposed Protocol 3, since it is based on transitive subgroups of the automorphishm groups of the storage and query codes, and thus is less practical.
- For both the noncolluding and colluding cases, we provide an algorithm that optimizes the PIR rate of the underlying code (Sections VII and VIII-E).

The remainder of this paper is organized as follows. We provide some definitions and preliminaries in Section II. In Section III, we provide a general system model for the three PIR protocols proposed in the paper. In Sections IV and V, we present Protocols 1 and 2 for the scenario with noncolluding nodes. In Section VI, we give a necessary and a sufficient condition for codes to achieve the MDS-PIR capacity with Protocols 1 and 2 and prove that several families of codes achieve it. In Section VII, we give an optimization algorithm to optimize the PIR rate. In Section VIII, we consider the scenario with colluding nodes and propose Protocol 3. In the same section, we also present a family of storage codes that can be used with this protocol. Lastly, we provide a necessary and a sufficient condition to achieve an upper bound on the PIR rate for this protocol, and we show that RM codes satisfy the sufficient condition and thus achieve the upper bound on the PIR rate of Protocol 3. In Section IX, we optimize Protocols 1, 2, and 3 to maximize their PIR rates for different storage codes under the scenarios of noncolluding and colluding nodes. Finally, some conclusions are drawn in Section X.

A. Notation and Conventions

In this paper, we use the following notation. We use lowercase bold letters to denote vectors, uppercase bold letters to denote matrices, and calligraphic uppercase letters to denote sets. For example: $\boldsymbol{x}$, $\boldsymbol{X}$, and $\mathcal{X}$ denote a vector, a matrix, and a set, respectively. An identity matrix of dimensions $m \times m$ is denoted as $\boldsymbol{I}_m$. The all-zero matrix of dimensions $a \times b$ is denoted as $\boldsymbol{0}_{a\times b}$, while the all-one matrix of dimensions $a \times b$ is referred to as $\boldsymbol{1}_{a\times b}$. $(\cdot)^{\intercal}$ represents the transpose of its argument and $\langle \cdot, \cdot \rangle$ denotes the scalar dot product between two vectors. The operator $\circ$ represents the Hadamard product. As such, $\boldsymbol{x} \circ \boldsymbol{y}$ represents the Hadamard product of two length-n vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Consider the column vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_a$, then $(\boldsymbol{x}_1 | \ldots | \boldsymbol{x}_a)$ represents the horizontal concatenation of the column vectors. Similarly, the horizontal concatenation of the matrices $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_a$, all with the same number of rows, will be denoted by $(\boldsymbol{X}_1 | \ldots | \boldsymbol{X}_a)$. We represent a submatrix of $\boldsymbol{X}$ that is restricted in columns by the set $\mathcal{I}$ and in rows by the set $\mathcal{J}$ by $\boldsymbol{X}|_{\mathcal{I}}^{\mathcal{J}}$, and the matrix rank of $\boldsymbol{X}$ by $\operatorname{rank}(\boldsymbol{X})$. The function $\mathsf{LCM}(a, b)$ computes the lowest common multiple of two positive integers a and b, and $a \mid b$ denotes that a is a divisor of b, while the function $\mathsf{H}(\cdot)$ represents the entropy of its argument.

In the rest of the paper, $\mathcal{C}$ will denote a linear code over a finite field $\mathrm{GF}(q)$. The operations over $\mathrm{GF}(q)$, such as addition, multiplication, etc., will be clearly understood from the context. We use the customary code parameters $[n, k]$ to refer to a code of block length n and dimension k, having code rate $R^{\mathcal{C}} = k/n$. The dimension of a code $\mathcal{C}$ will sometimes be denoted by $\dim(\mathcal{C})$. Furthermore, $[n, k, d_{\mathsf{min}}^{\mathcal{C}}]$ represents an $[n, k]$ code of minimum Hamming distance $d_{\mathsf{min}}^{\mathcal{C}}$. Since a code $\mathcal{C}$ can be seen as a codebook matrix, the shortened and punctured codes are denoted by $\mathcal{C}|_{\mathcal{I}}^{\mathcal{J}}$, with column indices $\mathcal{I}$ and row coordinates $\mathcal{J}$. In addition, $\boldsymbol{H}^{\mathcal{C}}$, $\boldsymbol{G}^{\mathcal{C}}$, and $\mathcal{C}^{\perp}$ represent a parity-check matrix, a generator matrix, and the dual code, respectively, of $\mathcal{C}$. We denote by $\mathbb{N}$ the set of all positive integers, $\mathbb{N}_a \triangleq \{1, 2, \ldots, a\}$, and $\mathbb{N}_{n_1:n_2} \triangleq \{n_1, n_1+1, \ldots, n_2\}$ for two positive integers $n_1 \leq n_2$, $n_1, n_2 \in \mathbb{N}$. The Hamming weight of a binary vector $\boldsymbol{x}$ is denoted by $w_{\mathsf{H}}(\boldsymbol{x})$, while the support of a vector $\boldsymbol{x}$, i.e., the set of nonzero entries of $\boldsymbol{x}$, will be denoted by $\chi(\boldsymbol{x})$. Note that sometimes, for the sake of convenience, we will omit the superscripts and/or the subscripts if the arguments we refer to are contextually unambiguous. Also, with some abuse of language, the index of a coordinate of a vector is sometimes referred to simply as the coordinate. An erasure pattern is a binary vector where the ones represent erased positions, while the zeros represent nonerased positions. The weight of an erasure pattern is the number of erased positions, and an erasure pattern $\boldsymbol{x}$ is said to be correctable by a code $\mathcal{C}$ if $\boldsymbol{H}^{\mathcal{C}}|_{\chi(\boldsymbol{x})}$ has rank $|\chi(\boldsymbol{x})|$. Finally, for ease of notation, we will refer to a matrix with constant row weight, constant column weight, and constant row and column weight equal to a as an a-row regular, a-column regular, and a-regular matrix, respectively.

II. Definitions and Preliminaries

In this section, we review some basic notions in coding theory and some classes of codes that will be used throughout the paper.

Definition 1. *Let $\mathcal{C}$ be an $[n, k]$ code defined over $\mathrm{GF}(q)$. A set of coordinates of $\mathcal{C}$, $\mathcal{I} \subseteq \mathbb{N}_n$, of size k is said to be an* information set *if and only if $\boldsymbol{G}^{\mathcal{C}}|_{\mathcal{I}}$ is invertible.*

Definition 2. *Let $\mathcal{D}$ be a subcode of an $[n, k]$ code $\mathcal{C}$. The* support *of $\mathcal{D}$ is defined as*

$$\chi(\mathcal{D}) \triangleq \{j \in \mathbb{N}_n \colon \exists\, \boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{D}, x_j \neq 0\}.$$

It is noted that

$$\chi(\mathcal{D}) = \bigcup_{\boldsymbol{x} \in \mathcal{D}} \chi(\boldsymbol{x}).$$

[2] Note that the proposed sufficient condition (Theorem 8) is not equivalent to the concept of transitive codes in [32] when there are at least 2 colluding nodes (in the noncolluding case the concept of transitive codes in [32] reduces to our sufficient condition (Theorem 4)).

Next, we introduce the concept of generalized Hamming weights [33].

Definition 3. *The s-th generalized Hamming weight of an $[n,k]$ code $\mathcal{C}$, denoted by $d_s^{\mathcal{C}}$, $s \in \mathbb{N}_k$, is defined as the cardinality of the smallest support of an s-dimensional subcode of $\mathcal{C}$, i.e.,*

$$d_s^{\mathcal{C}} \triangleq \min\big\{|\chi(\mathcal{D})| : \mathcal{D} \text{ is an } [n,s] \text{ subcode of } \mathcal{C}\big\}.$$

For the sequel, we introduce the notion of Hadamard product [34] of vector spaces.

Definition 4. *Let $\mathcal{X}$ and $\mathcal{Y}$ be two vector spaces in $\mathrm{GF}(q)^n$. The Hadamard product of $\mathcal{X}$ and $\mathcal{Y}$, denoted by $\mathcal{X} \circ \mathcal{Y}$, is defined as the space in $\mathrm{GF}(q)^n$ generated by the Hadamard products $\boldsymbol{x} \circ \boldsymbol{y}$ for all $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$.*

A. Reed-Muller Codes

We review the family of binary linear RM codes [35] and then quickly summarize a result related to information sets of an RM code. We adapt the concept and definition from [31, Ch. 13], and the details can be found therein.

Definition 5. *For a given $m \in \mathbb{N}$, the v-th order binary RM code $\mathcal{R}(v,m)$ is an $[n,k]$ code with length $n = 2^m$ and code dimension $k = \sum_{i=0}^{v} \binom{m}{i}$ for $v \in \{0\} \cup \mathbb{N}_m$, constructed as the linear space spanned by the set of all m-variable Boolean monomials of degree at most v.*

For example, $\mathcal{R}(2,3)$ can be viewed as the linear space spanned by the set of Boolean monomials $\{1, z_1, z_2, z_3, z_1z_2, z_1z_3, z_2z_3\}$.

Next, we introduce a way to number the coordinate index of an RM codeword. Without loss of generality, since there are in total $n = 2^m$ codeword coordinates, each coordinate index $i \in \mathbb{N}_{2^m}$ can be described by a binary column m-tuple $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^{\mathsf{T}}$, $\mu_j \in \mathrm{GF}(2)$, such that

$$i \triangleq 1 + \sum_{j=1}^{m} \mu_j 2^{j-1}. \tag{1}$$

For instance, for $m = 4$, the 7-th coordinate of an RM code corresponds to $(0\ 1\ 1\ 0)^{\mathsf{T}}$. Hence, a set of coordinates $\mathcal{I} \subseteq \mathbb{N}_n$ can alternatively be written as a set of corresponding m-tuples for RM codes.

Let $\boldsymbol{V}$ be an $m \times m$ invertible matrix over $\mathrm{GF}(2)$ and $\boldsymbol{\sigma} \in \mathrm{GF}(2)^{m\times 1}$ be a length-m binary column vector. It is well known that the coordinate transformation mapping $\boldsymbol{\mu}$ onto $g(\boldsymbol{\mu}) \triangleq \boldsymbol{V}\boldsymbol{\mu} + \boldsymbol{\sigma}$ is an *automorphism* for the RM code [31, Ch. 13].

For the sake of simplicity, throughout the paper we assume $\boldsymbol{V} = \boldsymbol{I}_m$.

Example 1. *Consider the RM code $\mathcal{R}(1,3)$ with generator matrix*

$$\boldsymbol{G}^{\mathcal{R}(1,3)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

The i-th row of $\boldsymbol{G}^{\mathcal{R}(1,3)}$ corresponds to the i-th monomial of the set of Boolean monomials $\{1, z_1, z_2, z_3\}$, $i \in \mathbb{N}_4$. It can be seen that any codeword of $\mathcal{R}(1,3)$ corresponds to a linear combination of the Boolean monomials as $w_0 1 + w_1 z_1 + w_2 z_3 + w_3 z_3$, where $w_i \in \mathrm{GF}(2)$, $i \in \mathbb{N}_4$. Clearly, $\mathcal{I} = \{(0\ 0\ 0)^{\mathsf{T}}, (1\ 0\ 0)^{\mathsf{T}}, (0\ 1\ 0)^{\mathsf{T}}, (0\ 0\ 1)^{\mathsf{T}}\}$ forms an information set for $\mathcal{R}(1,3)$. Pick an automorphism g with $\boldsymbol{V} = \boldsymbol{I}_3$ and $\boldsymbol{\sigma} = (0\ 0\ 1)^{\mathsf{T}}$. Then,

$$\begin{aligned}\mathcal{I}' &= \{g(\boldsymbol{\mu}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \colon \boldsymbol{\mu} \in \mathcal{I}\} \\ &= \{(0\ 0\ 1)^{\mathsf{T}}, (1\ 0\ 1)^{\mathsf{T}}, (0\ 1\ 1)^{\mathsf{T}}, (0\ 0\ 0)^{\mathsf{T}}\}\end{aligned}$$

is also an information set of $\mathcal{R}(1,3)$.

The following lemma shows how to determine an information set for an RM code.

Lemma 1. *Consider the RM code $\mathcal{R}(v,m)$ with $v \in \{0\} \cup \mathbb{N}_m$, $m \in \mathbb{N}$. Then, the set of m-tuples given by*

$$\mathcal{I} \triangleq \{\boldsymbol{\mu} \in \mathrm{GF}(2)^{m\times 1} \colon w_{\mathsf{H}}(\boldsymbol{\mu}) \le v\}$$

is an information set for $\mathcal{R}(v,m)$.

Proof: The proof is based on the definition of RM codes. The details are given in Appendix A. ∎

Lemma 1 can be extended to nonbinary generalized RM codes (see the comprehensive work in [36] that determines the information sets for generalized RM codes).

B. Local Reconstruction Codes

LRCs are a family of codes that are used in DSSs because of their low repair locality, i.e., they need to contact a relatively low number of nodes in order to repair a failed node. Systematic codes that focus on lowering the locality for the systematic nodes (i.e., the nodes that store the systematic code symbols; see the system model in Section III) are referred to as *information locality* codes. Examples of such codes are presented in [23]–[26]. On the contrary, LRCs that achieve low locality for all nodes are referred to as *all-symbol locality* codes. The codes presented in [27] are examples of all-symbol locality codes. Formally, information locality codes are defined as follows.

Definition 6 ((r,δ) information locality code [26, Def. 2])**.** *An $[n,k]$ code is said to be an (r,δ) information locality code if there exist L_{c} punctured codes $\mathcal{C}_j \triangleq \mathcal{C}|_{\mathcal{S}_j}$ of $\mathcal{C}$ with column coordinate set $\mathcal{S}_j \subset \mathbb{N}_n$ for $j \in \mathbb{N}_{L_{\mathsf{c}}}$. Furthermore, $\{\mathcal{C}|_{\mathcal{S}_j}\}_{j\in\mathbb{N}_{L_{\mathsf{c}}}}$ must satisfy the following conditions:*

1) $|\mathcal{S}_j| \le r + \delta - 1$, $\forall\, j \in \mathbb{N}_{L_{\mathsf{c}}}$,
2) $d_{\mathsf{min}}^{\mathcal{C}_j} \ge \delta$, $\forall\, j \in \mathbb{N}_{L_{\mathsf{c}}}$, *and*
3) $\mathrm{rank}\big(\boldsymbol{G}|_{\bigcup_j \mathcal{S}_j}\big) = k$.

In other words, Definition 6 says that there are L_{c} local codes in $\mathcal{C}$ each having a block length of at most $r + \delta - 1$, a minimum Hamming distance at least δ, and the union of all coordinate sets of the local codes contains an information set. The overall code $\mathcal{C}$ has minimum Hamming distance $d_{\mathsf{min}}^{\mathcal{C}} \le n - k + 1 - (\lceil k/r \rceil - 1)(\delta - 1)$ and can repair up to $\delta - 1$ systematic nodes by contacting r storage nodes. Codes that achieve the upper bound on the minimum Hamming distance

are known as distance-optimal (r, δ) information locality codes and have the following structure.

Definition 7 (Distance-optimal (r, δ) information locality code [26, Th. 2.2])**.** *Let $r \mid k$ such that $L_{\mathsf{c}} = k/r$. An (r, δ) information locality code $\mathcal{C}$ as defined in Definition 6 is distance-optimal if:*

1) *Each local code $\mathcal{C}|_{\mathcal{S}_j}$, $j \in \mathbb{N}_{L_{\mathsf{c}}}$, is an $[r + \delta - 1, r]$ MDS code defined by a parity-check matrix $\boldsymbol{H}^{\mathcal{C}|_{\mathcal{S}_j}} = (\boldsymbol{P}_j | \boldsymbol{I}_{\delta-1})$ of dimensions $(\delta-1)\times(r+\delta-1)$ and minimum Hamming distance $d_{\min}^{\mathcal{C}|_{\mathcal{S}_j}} = \delta$.*
2) *The sets $\{\mathcal{S}_j\}_{j\in\mathbb{N}_{L_{\mathsf{c}}}}$ are disjoint, i.e., $\mathcal{S}_j \cap \mathcal{S}_{j'} = \emptyset$ for all $j, j' \in \mathbb{N}_{L_{\mathsf{c}}}$, $j \neq j'$.*
3) *The code $\mathcal{C}$ has a parity-check matrix of the form*

$$\boldsymbol{H} = \left(\begin{array}{cccccccc|c} \boldsymbol{P}_1 & \boldsymbol{I}_{\delta-1} & & & & & & & \\ & & \boldsymbol{P}_2 & \boldsymbol{I}_{\delta-1} & & & & & \\ & & & & \ddots & & & & \\ & & & & & & \boldsymbol{P}_{L_{\mathsf{c}}} & \boldsymbol{I}_{\delta-1} & \\ \hline \boldsymbol{M}_1 & \boldsymbol{0} & \boldsymbol{M}_2 & \boldsymbol{0} & \cdots & & \boldsymbol{M}_{L_{\mathsf{c}}} & \boldsymbol{0} & \boldsymbol{I}_a \end{array}\right) \tag{2}$$

where the matrices $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{L_{\mathsf{c}}}$ are arbitrary matrices in $\mathrm{GF}(q)$ of dimensions $(n - L_{\mathsf{c}}(r + \delta - 1)) \times r$, and $a \triangleq n - L_{\mathsf{c}}(r + \delta - 1)$.

For ease of exposition, we refer to the local parities as the parity symbols that take part in the local codes, while the parity symbols that are not part of the L_{c} local codes are referred to as global parity symbols. According to Definition 7, there exist $n - L_{\mathsf{c}}(r + \delta - 1)$ global parities and $L_{\mathsf{c}}(\delta - 1)$ local parities. We partition the coordinates of these parities into $L + 1$ sets, where $L \triangleq \left\lfloor \frac{n}{r+\delta-1} \right\rfloor$. For $i \in \mathbb{N}_{L+1}$, we have

$$\mathcal{P}_j = \begin{cases} \{(j-1)n_{\mathsf{c}} + r + 1, \ldots, jn_{\mathsf{c}}\} & \text{if } j \in \mathbb{N}_{L_{\mathsf{c}}}, \\ \{(j-1)n_{\mathsf{c}} + 1, \ldots, jn_{\mathsf{c}}\} & \text{if } j \in \mathbb{N}_{L_{\mathsf{c}}+1:L}, \\ \{Ln_{\mathsf{c}} + 1, \ldots, n\} & \text{if } j = L + 1, \end{cases} \tag{3}$$

where $n_{\mathsf{c}} \triangleq r + \delta - 1$ is the block length of each local code. The set $\mathcal{P}_j$, $j \in \mathbb{N}_{L_{\mathsf{c}}}$, represents the coordinates of the local parities of the j-th local code $\mathcal{C}_j$. The remaining sets $\mathcal{P}_j$, $j \in \mathbb{N}_{L_{\mathsf{c}}+1:L+1}$, represent the coordinates of the global parities of $\mathcal{C}$. As such, the set $\mathcal{P} = \bigcup_{j=1}^{L+1} \mathcal{P}_j$ represents the parity coordinates of $\mathcal{C}$.

C. UUV Codes

Consider an $[n_1, k_1]$ code $\mathcal{U}$ and an $[n_1, k_2]$ code $\mathcal{V}$ both over $\mathrm{GF}(q)$. An $[n = 2n_1, k = k_1 + k_2]$ $(\mathcal{U} \mid \mathcal{U} + \mathcal{V})$ code [31] (herein referred to as a UUV code) has codewords of the form $(\boldsymbol{u} \mid \boldsymbol{u} + \boldsymbol{v})$, where $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{v} \in \mathcal{V}$. A UUV code has generator matrix

$$\boldsymbol{G}^{\mathsf{UUV}} = \begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{k_2 \times n_1} & \boldsymbol{G}^{\mathcal{V}} \end{pmatrix},$$

where $\boldsymbol{G}^{\mathcal{U}}$ and $\boldsymbol{G}^{\mathcal{V}}$ are the generator matrices of $\mathcal{U}$ and $\mathcal{V}$, respectively. One can construct RM codes using UUV codes in an iterative manner [31, p. 374].

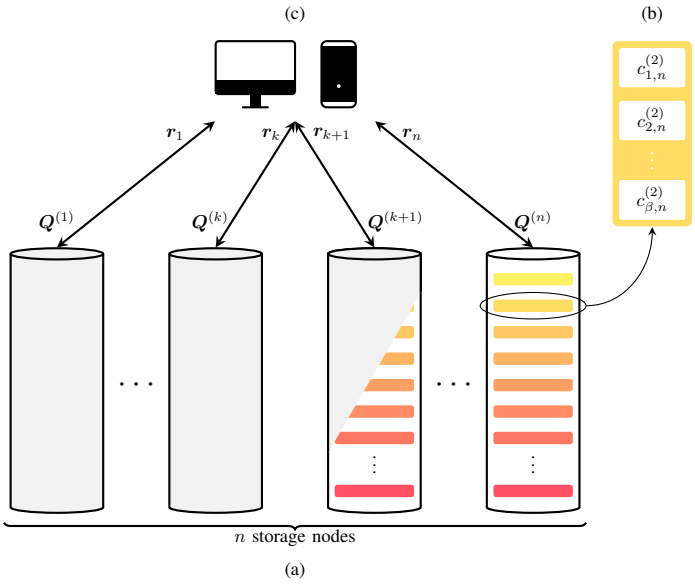

Fig. 1. System Model. (a) The colored boxes in each storage node represent the f coded chunks pertaining to the f files. (b) Coded chunk corresponding to the 2nd file in the n-th node. It consists of β code symbols, $c_{i,n}^{(2)}$, $i \in \mathbb{N}_\beta$. (c) The user sends the queries $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_n$, to the storage nodes and receives responses $\boldsymbol{r}_l$.

III. System Model

We consider a DSS that stores f files $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(f)}$, where each file $\boldsymbol{X}^{(m)} = (x_{i,j}^{(m)})$, $m \in \mathbb{N}_f$, can be seen as a $\beta \times k$ matrix over $\mathrm{GF}(p^{\alpha\ell})$, with β, k, α, $\ell \in \mathbb{N}$, and p being some prime number. Each file is encoded using a linear code as follows. Let $\boldsymbol{x}_i^{(m)} = (x_{i,1}^{(m)}, \ldots, x_{i,k}^{(m)})$, $i \in \mathbb{N}_\beta$, be a message vector corresponding to the i-th row of $\boldsymbol{X}^{(m)}$. Each $\boldsymbol{x}_i^{(m)}$ is encoded by an $[n, k]$ code $\mathcal{C}$ over $\mathrm{GF}(q)$ with $q \triangleq p^\alpha$, having *subpacketization* α, into a length-n codeword $\boldsymbol{c}_i^{(m)} = \left(c_{i,1}^{(m)}, \ldots, c_{i,n}^{(m)}\right)$, where $c_{i,j}^{(m)} \in \mathrm{GF}(q^\ell)$, $j \in \mathbb{N}_n$. For $\alpha = 1$, the code $\mathcal{C}$ is referred to as a *scalar* code. Otherwise, the code is called a *vector* code [37]. The βf generated codewords $\boldsymbol{c}_i^{(m)}$ are then arranged in the array $\boldsymbol{C} = \left((\boldsymbol{C}^{(1)})^\intercal | \ldots | (\boldsymbol{C}^{(f)})^\intercal\right)^\intercal$ of dimensions $\beta f \times n$, where $\boldsymbol{C}^{(m)} = \left((\boldsymbol{c}_1^{(m)})^\intercal | \ldots | (\boldsymbol{c}_\beta^{(m)})^\intercal\right)^\intercal$ for $m \in \mathbb{N}_f$. For a given column j of $\boldsymbol{C}$, we denote the column vector $\left(c_{1,j}^{(m)}, \ldots, c_{\beta,j}^{(m)}\right)^\intercal$ as a coded chunk pertaining to file $\boldsymbol{X}^{(m)}$. The f coded chunks in column j are stored in the j-th storage node, $j \in \mathbb{N}_n$, as shown in Fig. 1(a). In case the $[n, k]$ code $\mathcal{C}$ is systematic, the nodes that store the systematic code symbols are referred to as systematic nodes.

A. Privacy Model

We consider a DSS where a set of T nodes may act as spies. Further, they may collude and hence they are referred to as colluding nodes. In addition, it is assumed that the remaining nonspy nodes do not collaborate with the spy nodes. The scenario of a single spy node ($T = 1$) in the DSS is analogous to having a system with no colluding nodes. Let $\mathcal{T} \subset \mathbb{N}_n$, $|\mathcal{T}| = T$, denote the set of spy nodes in the DSS. The role of the spy nodes is to determine which file $\boldsymbol{X}^{(m)}$ is accessed by the user. We assume that the user does not know $\mathcal{T}$, since

otherwise it can trivially achieve PIR by not contacting the spy nodes. To retrieve file $\boldsymbol{X}^{(m)}$ from the DSS, the user sends a $d \times \beta f$ matrix query $\boldsymbol{Q}^{(l)}$ over $\mathrm{GF}(q) \subseteq \mathrm{GF}(q^\ell)$ to the l-th node for all $l \in \mathbb{N}_n$. The query matrices are represented in the form of d subquery vectors $\boldsymbol{q}_i^{(l)}$ of length βf as

$$\boldsymbol{Q}^{(l)} = \begin{pmatrix} \boldsymbol{q}_1^{(l)} \\ \vdots \\ \boldsymbol{q}_d^{(l)} \end{pmatrix} = \begin{pmatrix} q_{1,1}^{(l)} & \cdots & q_{1,\beta f}^{(l)} \\ \vdots & \cdots & \vdots \\ q_{d,1}^{(l)} & \cdots & q_{d,\beta f}^{(l)} \end{pmatrix}.$$

The i-th subqueries $\boldsymbol{q}_i^{(l)}$, $l \in \mathbb{N}_n$, of the n queries aim at recovering Γ unique code symbols[3] of the file $\boldsymbol{X}^{(m)}$. In response to the received query matrix, node l sends the column vector

$$\boldsymbol{r}_l = (r_{l,1}, \ldots, r_{l,d})^\intercal = \boldsymbol{Q}^{(l)} \big(c_{1,l}^{(1)}, \ldots, c_{\beta,l}^{(1)}, \ldots, c_{\beta,l}^{(f)}\big)^\intercal, \quad (4)$$

referred to as the response vector, back to the user as illustrated in Fig. 1(c). We refer to $r_{l,i}$ as the i-th subresponse of the l-th node. Perfect information-theoretic PIR for such a scheme is defined in the following.

Definition 8. *Consider a DSS with n nodes storing f files in which a set of T nodes $\mathcal{T} = \{t_1, \ldots, t_T\} \subset \mathbb{N}_n$, $1 \leq |\mathcal{T}| = T \leq n-k$, act as colluding spies. A user who wishes to retrieve the m-th file sends the queries $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_n$, to the storage nodes, which return the responses $\boldsymbol{r}_l$. This scheme achieves perfect information-theoretic PIR if and only if*

$$\textit{Privacy:} \qquad \mathsf{H}\big(m|\boldsymbol{Q}^{(t_1)}, \ldots, \boldsymbol{Q}^{(t_T)}\big) = \mathsf{H}(m); \quad (5a)$$

$$\textit{Recovery:} \qquad \mathsf{H}\big(\boldsymbol{X}^{(m)}|\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\big) = 0. \quad (5b)$$

Queries satisfying (5a) ensure that the file requested by the user is independent of the queries. Thus, the colluding nodes in $\mathcal{T}$ do not gain any additional information regarding which file is requested by the user by observing the queries. The recovery constraint in (5b) ensures that the user is able to recover the requested file from the responses sent by the DSS.

The efficiency of a PIR protocol is defined as the amount of retrieved data per unit of total amount of downloaded data, since it is assumed that the content of the retrieved file dominates the total communication cost [8], [12].

Definition 9. *The PIR rate of a PIR protocol, denoted by R, is the amount of information retrieved per downloaded symbol, i.e.,*

$$\mathsf{R} \triangleq \frac{\beta k}{nd}.$$

Since the size of each file is βk, the parameters d and Γ should be chosen such that $\beta k = \Gamma d$.[4] For Protocols 2 and 3 in Sections V and VIII to be practical, we may select

$$\beta = \frac{\mathsf{LCM}(k,\Gamma)}{k} \quad \text{and} \quad d = \frac{\mathsf{LCM}(k,\Gamma)}{\Gamma}, \quad (6)$$

as it ensures the smallest values of β and d. This is not the case for Protocol 1 in Section IV, where β is exponential in the number of files in order achieve optimal PIR rates. By choosing the values above for β and d, the PIR rate for Protocols 2 and 3 becomes

$$\mathsf{R} = \frac{\Gamma}{n}.$$

We will write $\mathsf{R}(\mathcal{C})$ to highlight that the PIR rate depends on the underlying storage code $\mathcal{C}$. The maximum achievable PIR rate is the PIR capacity. It was shown in [11] that for the noncolluding case and for a given number of files f stored using an $[n,k]$ MDS code, the MDS-PIR capacity, denoted by C_f, is

$$\mathsf{C}_f \triangleq \frac{n-k}{n}\left[1 - \left(\frac{k}{n}\right)^f\right]^{-1}. \quad (7)$$

Throughout the paper we refer to the capacity in (7) as the *finite MDS-PIR capacity* as it depends on the number of files. On the contrary, when the number of files $f \to \infty$, the *asymptotic MDS-PIR capacity* is

$$\mathsf{C}_\infty \triangleq \frac{n-k}{n}. \quad (8)$$

It was shown in [8, Th. 3] that the PIR rate for a DSS with noncolluding nodes is upperbounded by C_∞ for a special class of linear information retrieval schemes. In the case of colluding nodes, an explicit upper bound is currently unknown, as well as an expression for the MDS-PIR capacity. Some initial work for the case of two colluding nodes has recently been presented in [16].

IV. Finite MDS-PIR Capacity-Achieving Protocol for the Noncolluding Case

In this section, we propose a capacity-achieving protocol, named Protocol 1, that achieves the finite MDS-PIR capacity in (7) for the scenario of noncolluding nodes. The protocol is inspired by the protocol introduced in [11].

A. PIR Achievable Rate Matrix

In [10], the concept of exploiting *side information* for PIR problems was introduced. By side information we mean additional redundant symbols not related to the requested file but downloaded by the user in order to maintain privacy. These symbols can be exploited by the user to retrieve the requested file from the responses of the storage nodes. In [11, Sec. V.A], it was shown that a $[5,3,3]$ MDS storage code can be used to achieve the finite MDS-PIR capacity, where the side information is decoded by utilizing other code coordinates forming an information set in the code array. For instance, the authors chose the $\nu = 5$ information sets $\mathcal{I}_1 = \{1,2,3\}$, $\mathcal{I}_2 = \{1,4,5\}$, $\mathcal{I}_3 = \{2,3,4\}$, $\mathcal{I}_4 = \{1,2,5\}$, and $\mathcal{I}_5 = \{3,4,5\}$ of the $[5,3,3]$ MDS code in their PIR achievable scheme. Observe that in $\{\mathcal{I}_i\}_{i\in\mathbb{N}_5}$ each coordinate of the $[5,3,3]$ code appears exactly $\kappa = 3$ times. This motivates the following definition.

[3] In general, the i-th subqueries recover Γ_i unique code symbols such that among the $\sum_i \Gamma_i$ recovered code symbols there are βk distinct information symbols. However, for the sake of simplicity, we assume $\Gamma_i = \Gamma$ for all i for Protocols 2 and 3.

[4] For Protocol 1, d and Γ should be chosen such that $\beta k = \sum_{i=1}^{d} \Gamma_i$.

Definition 10. *Let $\mathcal{C}$ be an arbitrary $[n,k]$ code. A $\nu\times n$ binary matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})$ is said to be a* PIR achievable rate matrix *for $\mathcal{C}$ if the following conditions are satisfied.*

1) *The Hamming weight of each column of $\mathbf{\Lambda}_{\kappa,\nu}$ is κ, and*
2) *for each matrix row $\boldsymbol{\lambda}_i$, $i\in\mathbb{N}_\nu$, $\chi(\boldsymbol{\lambda}_i)$ always contains an information set.*

In other words, each coordinate j of $\mathcal{C}$, $j\in\mathbb{N}_n$, appears exactly κ times in $\{\chi(\boldsymbol{\lambda}_i)\}_{i\in\mathbb{N}_\nu}$, and every set $\chi(\boldsymbol{\lambda}_i)$ contains an information set.

Lemma 2. *If a matrix $\mathbf{\Lambda}_{\nu,\kappa}(\mathcal{C})$ exists for an $[n,k]$ code $\mathcal{C}$, then we have*

$$\frac{\kappa}{\nu}\geq\frac{k}{n},$$

where equality holds if $\chi(\boldsymbol{\lambda}_i)$, $i\in\mathbb{N}_\nu$, are all information sets.

Proof: Since by definition each row $\boldsymbol{\lambda}_i$ of $\mathbf{\Lambda}_{\nu,\kappa}$ always contains an information set, we have $w_{\mathsf{H}}(\boldsymbol{\lambda}_i)\geq k$, $i\in\mathbb{N}_\nu$. Let $\boldsymbol{v}_j$, $j\in\mathbb{N}_n$, be the j-th column of $\mathbf{\Lambda}_{\kappa,\nu}$. If we look at $\mathbf{\Lambda}_{\kappa,\nu}$ from both a row-wise and a column-wise point of view, we obtain

$$\nu k\leq\sum_{i=1}^{\nu}w_{\mathsf{H}}(\boldsymbol{\lambda}_i)=\sum_{j=1}^{n}w_{\mathsf{H}}(\boldsymbol{v}_j)=\kappa n,$$

from which the result follows. Clearly, equality holds if $\chi(\boldsymbol{\lambda}_i)$, $i\in\mathbb{N}_\nu$, are all information sets. ∎

Example 2. *Consider the $[5,3,2]$ systematic code with generator matrix*

$$\boldsymbol{G}=\begin{pmatrix}1&0&0&1&0\\0&1&0&1&0\\0&0&1&0&1\end{pmatrix}.$$

One can easily verify that

$$\mathbf{\Lambda}_{2,3}=\begin{pmatrix}0&1&1&1&1\\1&0&0&1&1\\1&1&1&0&0\end{pmatrix}$$

is a PIR achievable rate matrix for this code.

Before we state our main results, in order to clearly illustrate our example and the following achievability proof, we first introduce the following definition.

Definition 11. *For a given $\nu\times n$ PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})=(\lambda_{u,j})$, we define the PIR interference matrices $\boldsymbol{A}_{\kappa\times n}=(a_{i,j})$ and $\boldsymbol{B}_{(\nu-\kappa)\times n}=(b_{i,j})$ for the code $\mathcal{C}$ with*

$$a_{i,j}\triangleq u \text{ if } \lambda_{u,j}=1,\ \forall\, j\in\mathbb{N}_n,\ i\in\mathbb{N}_\kappa,u\in\mathbb{N}_\nu,$$
$$b_{i,j}\triangleq u \text{ if } \lambda_{u,j}=0,\ \forall\, j\in\mathbb{N}_n,\ i\in\mathbb{N}_{\nu-\kappa},u\in\mathbb{N}_\nu.$$

Note that in Definition 11, for each $j\in\mathbb{N}_n$, distinct values of $u\in\mathbb{N}_\nu$ should be assigned for all i. Thus, the assignment is not unique in the sense that the order of the entries of each column of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be permuted. For $j\in\mathbb{N}_n$, let $\mathcal{A}_j\triangleq\{a_{i,j}\colon i\in\mathbb{N}_\kappa\}$ and $\mathcal{B}_j\triangleq\{b_{i,j}\colon i\in\mathbb{N}_{\nu-\kappa}\}$. Note that the j-th column of $\boldsymbol{A}$ contains the row indices of $\mathbf{\Lambda}$ whose entries in the j-th column are equal to 1, while $\boldsymbol{B}$ contains the remaining row indices of $\mathbf{\Lambda}$. Hence, it can be observed that $\mathcal{B}_j=\mathbb{N}_\nu\setminus\mathcal{A}_j$, $\forall\, j\in\mathbb{N}_n$.

Definition 12. *By $\mathcal{S}(a|\boldsymbol{A}_{\kappa\times n})$ we denote the set of column coordinates of matrix $\boldsymbol{A}_{\kappa\times n}=(a_{i,j})$ in which at least one of its entries is equal to a, i.e.,*

$$\mathcal{S}(a|\boldsymbol{A}_{\kappa\times n})\triangleq\{j\in\mathbb{N}_n\colon\exists\, a_{i,j}=a,i\in\mathbb{N}_\kappa\}.$$

The following claim can be directly verified.

Claim 1. *$\mathcal{S}(a|\boldsymbol{A}_{\kappa\times n})$ contains an information set of code $\mathcal{C}$, $\forall\, a\in\mathbb{N}_\nu$. Moreover, for an arbitrary entry $b_{i,j}$ of $\boldsymbol{B}_{(\nu-\kappa)\times n}$, $\mathcal{S}(b_{i,j}|\boldsymbol{A}_{\kappa\times n})=\mathcal{S}(a|\boldsymbol{A}_{\kappa\times n})\subseteq\mathbb{N}_n\setminus\{j\}$ if $b_{i,j}=a$.*

We illustrate the previous points in the following example.

Example 3. *Continuing with Example 2 and following Definition 11, we obtain*

$$\boldsymbol{A}_{2\times5}=\begin{pmatrix}2&1&1&1&1\\3&3&3&2&2\end{pmatrix}\text{ and }\boldsymbol{B}_{1\times5}=\begin{pmatrix}1&2&2&3&3\end{pmatrix}$$

for $\mathbf{\Lambda}_{2,3}$. One can see that $\mathcal{A}_j\cup\mathcal{B}_j=\mathbb{N}_3$, $\forall\, j\in\mathbb{N}_5$. Moreover, for instance, take $a=1$, then $\mathcal{S}(1|\boldsymbol{A}_{2\times5})=\{2,3,4,5\}$ contains an information set of the $[5,3,2]$ systematic code of Example 2.

Now consider the two matrices

$$\begin{pmatrix}c^{(m)}_{\mu+a_{1,1},1}&c^{(m)}_{\mu+a_{1,2},2}&\cdots&c^{(m)}_{\mu+a_{1,n},n}\\\vdots&\cdots&&\vdots\\c^{(m)}_{\mu+a_{\kappa,1},1}&c^{(m)}_{\mu+a_{\kappa,2},2}&\cdots&c^{(m)}_{\mu+a_{\kappa,n},n}\end{pmatrix}\quad\text{and}$$

$$\begin{pmatrix}c^{(m)}_{\mu+b_{1,1},1}&c^{(m)}_{\mu+b_{1,2},2}&\cdots&c^{(m)}_{\mu+b_{1,n},n}\\\vdots&\cdots&&\vdots\\c^{(m)}_{\mu+b_{\nu-\kappa,1},1}&c^{(m)}_{\mu+b_{\nu-\kappa,2},2}&\cdots&c^{(m)}_{\mu+b_{\nu-\kappa,n},n}\end{pmatrix}$$

of code symbols of the m-th file, where $\mu\in\mathbb{N}_{\beta-\nu}\cup\{0\}$. Observe that if the user knows the first matrix of code symbols, from Claim 1, since the coordinate set $\mathcal{S}(b_{i,j}|\boldsymbol{A}_{\kappa\times n})\subseteq\mathbb{N}_n\setminus\{j\}$ contains an information set and the user knows the structure of the storage code $\mathcal{C}$, the code symbols $c^{(m)}_{\mu+b_{i,j},j}$ of the second matrix can be obtained. The intuition behind the definition of the interference matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is as follows. Assume that $\boldsymbol{X}^{(1)}$ is requested. Protocol 1 requires the user to download the side information $\sum_{m\neq1}c^{(m)}_{\mu+a_{i,j},j}$ based on $\boldsymbol{A}$ and also to download code symbols as sums of code symbols from the requested file and the side information $\sum_{m\neq1}c^{(m)}_{\mu+b_{i,j},j}$ based on $\boldsymbol{B}$. Claim 1 then indicates that the side information $\sum_{m\neq1}c^{(m)}_{\mu+b_{i,j},j}$ based on $\boldsymbol{B}$ can be reliably decoded and hence we can obtain the requested file by cancelling the side information. Here, the entries of $\boldsymbol{A}$ and $\boldsymbol{B}$ are respectively marked in red and blue. We are now ready to state Protocol 1.

B. Protocol 1

The proposed Protocol 1 generalizes the MDS-coded PIR protocol in [11] to DSSs where files are stored using an arbitrary linear code. Inspired by [10] and [11], a PIR capacity-achievable scheme should follow three important principles: 1) enforcing symmetry across storage nodes, 2) enforcing file symmetry within each storage node, and 3) exploiting

side information of undesired symbols to retrieve new desired symbols. Note that principle 1) is in general not a necessary requirement for a feasible PIR protocol. However, as pointed out in [11] and [16], any PIR scheme can be made symmetric, hence we keep this principle for the purpose of simplifying the implementation.

The PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}$ for the given storage code $\mathcal{C}$ plays a central role in the proposed PIR protocol. Moreover, the protocol requires $\beta = \nu^f$ stripes and exploits the corresponding PIR interference matrices $\boldsymbol{A}_{\kappa\times n}$ and $\boldsymbol{B}_{(\nu-\kappa)\times n}$. Note that the number of stripes depends on the number of files f, hence Protocol 1 depends on f as well. We first outline the steps of the protocol, and then we will prove that the proposed protocol satisfies the perfect privacy condition of (5a) and results in the PIR rate of Theorem 1 below. Without loss of generality, we assume that the user wants to download the first file, i.e., $m = 1$. The algorithm is composed of four steps as described below. In Appendix B, we show that the algorithm generates $d \times \beta f$ query matrices $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_n$, with

$$d = \frac{\kappa}{\nu-\kappa}\Big[\nu^f - \kappa^f\Big].$$

Step 1. *Index Preparation:* For all files, the user interleaves the query indices for requesting the rows of $\boldsymbol{C}^{(m)}$ randomly and independently of each other. This is equivalent to generating the interleaved code array $\boldsymbol{Y}^{(m)} = \big((\boldsymbol{y}_1^{(m)})^\mathsf{T}|\dots|(\boldsymbol{y}_\beta^{(m)})^\mathsf{T}\big)^\mathsf{T}$, $\forall\, m \in \mathbb{N}_f$, with rows

$$\boldsymbol{y}_i^{(m)} = \boldsymbol{c}_{\pi(i)}^{(m)}, \quad i \in \mathbb{N}_\beta,$$

where $\pi(\cdot) : \mathbb{N}_\beta \to \mathbb{N}_\beta$ is a random permutation, which is privately known to the user only. Therefore, when the user requests code symbols from each storage node, this procedure is designed to make the requested row indices to be random and independent of the requested file index.

Step 2. *Download Symbols in the i-th Repetition:* The user downloads the needed symbols in κ repetitions. In the i-th repetition, $i \in \mathbb{N}_\kappa$, the user downloads the required symbols in a total of f rounds. Each repetition comprises f rounds. In the m-th round, the user downloads symbols that are linear sums of code symbols from any m files, $m \in \mathbb{N}_f$. Using the terminology in [11], the user downloads two types of symbols in each round, *desired symbols*, which are directly related to the requested file index $m = 1$, and *undesired symbols*, which are not related to the requested file index $m = 1$, but are exploited to decode the requested file from the desired symbols. For the desired symbols, we will distinguish between round $\ell = 1$ and round $\ell \in \mathbb{N}_{2:f}$.

Undesired symbols. The undesired symbols refer to sums of code symbols which do not contain symbols from the requested file. For every round ℓ, $\ell \in \mathbb{N}_{f-1}$, the user downloads the code symbols

$$\Bigg\{ \sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+a_{1,j},j},\ \dots,\ \sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+a_{\kappa,j},j},\ \dots,\ \sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+a_{1,j},j},\ \dots,\ \sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+a_{\kappa,j},j} \Bigg\} \tag{9}$$

for all $j \in \mathbb{N}_n$ and for all possible subsets $\mathcal{M} \subseteq \mathbb{N}_{2:f}$, where $|\mathcal{M}| = \ell$ and

$$\mathsf{U}(\ell) \triangleq \sum_{h=1}^{\ell} \kappa^{f-(h+1)}(\nu-\kappa)^{h-1}.$$

In contrast to undesired symbols, desired symbols are sums of code symbols which contain symbols of the requested file. The main idea of the protocol is that the user downloads desired symbols that are linear sums of requested symbols and undesired symbols from the previous round.

Desired symbols in the first round. In the first round, the user downloads $\kappa \cdot \mathsf{U}(1) = \kappa\kappa^{f-(1+1)}(\nu-\kappa)^{1-1} = \kappa^{f-1}$ undesired symbols from each storage node. However, these symbols cannot be exploited directly. Hence, due to symmetry, in round $\ell = 1$, the user downloads the κ^{f-1} desired symbols

$$\Big\{ y^{(1)}_{\kappa^{f-1}(a_{i,j}-1)+1,j}, \dots, y^{(1)}_{\kappa^{f-1}(a_{i,j}-1)+\kappa^{f-1},j} \Big\} \tag{10}$$

from the j-th storage node, $j \in \mathbb{N}_n$, i.e., the user also downloads κ^{f-1} symbols for $m = 1$ from each storage node.

Desired symbols in higher rounds. In the $(\ell+1)$-th round, $\ell \in \mathbb{N}_{f-1}$, in order to exploit the side information, i.e., the undesired symbols from the previous round, the user downloads the symbols

$$\Bigg\{ y^{(1)}_{\mathsf{D}(\ell-1)\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_1} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+b_{1,j},j},\ y^{(1)}_{(\mathsf{D}(\ell-1)+1)\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_1} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+b_{2,j},j},\ \dots, y^{(1)}_{(\mathsf{D}(\ell-1)+(\nu-\kappa)-1)\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_1} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+b_{\nu-\kappa,j},j},$$
$$y^{(1)}_{(\mathsf{D}(\ell-1)+(\nu-\kappa))\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_1} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1)+1)\cdot\nu+b_{1,j},j},\ \dots, y^{(1)}_{\left[\mathsf{D}(\ell-1)+(\mathsf{U}(\ell)-\mathsf{U}(\ell-1))(\nu-\kappa)-1\right]\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_1} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+b_{\nu-\kappa,j},j},$$
$$\dots, y^{(1)}_{(\mathsf{D}(\ell)-(\nu-\kappa))\cdot\nu+a_{i,j},j} + \sum_{m'\in\mathcal{M}_{\mathsf{N}(\ell)}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+b_{1,j},j},\ \dots, y^{(1)}_{(\mathsf{D}(\ell)-1)\cdot\nu+a_{i,j},j}$$

$$+\sum_{m'\in\mathcal{M}_{\mathsf{N}(\ell)}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+b_{\nu-\kappa,j},j}\Bigg\} \quad (11)$$

for all distinct ℓ-sized subsets $\mathcal{M}_1,\dots,\mathcal{M}_{\mathsf{N}(\ell)}\subseteq\mathbb{N}_{2:f}$, where $j\in\mathbb{N}_n$, $\mathsf{N}(\ell)\triangleq\binom{f-1}{\ell}$, and

$$\mathsf{D}(\ell)\triangleq\kappa^{f-1}+\sum_{h=1}^{\ell}\binom{f-1}{h}\kappa^{f-(h+1)}(\nu-\kappa)^h.$$

This indicates that for each combination of files indexed by $\mathcal{M}_l$, $l\in\mathbb{N}_{\mathsf{N}(\ell)}$, the user downloads $\big[\mathsf{U}(\ell)-1-\mathsf{U}(\ell-1)+1\big](\nu-\kappa)$ new desired symbols from each storage node, and since there are in total $\mathsf{N}(\ell)$ combinations of files, in each round $\mathsf{D}(\ell)-1-\mathsf{D}(\ell-1)+1$ extra desired symbols are downloaded from each storage node.

Exploiting the side information. Using the fact that for a linear code $\mathcal{C}$ any linear combination of codewords is also a codeword, and together with Claim 1, it is not too hard to see that by fixing an arbitrary coordinate $j\in\mathbb{N}_n$, there always exist some coordinates $\mathcal{S}\subset\mathbb{N}_n\setminus\{j\}$ (see Claim 1) such that for a subset $\mathcal{M}\subseteq\mathbb{N}_{2:f}$ with $|\mathcal{M}|=\ell$, the so-called *aligned sum*

$$\Bigg\{\sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell-1))\cdot\nu+b_{1,j},j},\\ \dots,\sum_{m'\in\mathcal{M}} y^{(m')}_{((i-1)\mathsf{U}(f-1)+\mathsf{U}(\ell)-1)\cdot\nu+b_{\nu-\kappa,j},j}\Bigg\}$$

for $\ell\in\mathbb{N}_{f-1}$ and $i\in\mathbb{N}_\kappa$, can be decoded. Consequently, in the $(\ell+1)$-th round, from each storage node j we can collect code symbols related to $m=1$ from the desired symbols, i.e.,

$$\Big\{y^{(1)}_{\mathsf{D}(\ell-1)\cdot\nu+a_{i,j},j},\dots,y^{(1)}_{(\mathsf{D}(\ell)-1)\cdot\nu+a_{i,j},j}\Big\} \quad (12)$$

is obtained.

Symmetry across storage nodes. In the previous steps, since the user downloads the same amount of required symbols for each $j\in\mathbb{N}_n$ and for every round, symmetry across storage nodes is ensured.

File symmetry within each storage node. To ensure that the privacy condition (5a) is fulfilled, we have to make sure that in each round $\ell\in\mathbb{N}_f$ of each repetition, for each storage node and for every combination of files indexed by $\mathcal{M}\subseteq\mathbb{N}_f$ with $|\mathcal{M}|=\ell$, the user requests the same number of linear sums $\eta(\mathcal{M})\triangleq\sum_{m\in\mathcal{M}}y^{(m)}_{\eta_m,j}$, where η_m depends on m. This will be shown to be inherent from the protocol (see proof of Theorem 1 in Appendix B). In addition, since the user always requests the same number of linear sums for each combination of files, the scheme also implies that the frequencies of requested code symbols pertaining to each individual file index $m\in\mathbb{N}_f$ among all the linear sums are the same for each storage node.

Step 3. *Complete κ Repetitions:* The user repeats Step 2 until $i=\kappa$. We will show that by our designed parameters $\mathsf{U}(\ell)$ and $\mathsf{D}(\ell)$, the user indeed downloads in total $\beta=\nu^f$ stripes for the requested file (see again Appendix B).

Step 4. *Shuffling the Order of Queries to Each Node:* The order of the queries to each storage node is uniformly shuffled to prevent the storage node to be able to identify which file is requested from the index of the first downloaded symbol.

C. Achievable PIR Rate

The PIR rate, $\mathsf{R}(\mathcal{C})$, of Protocol 1 in Section IV-B for a DSS where f files are stored using an arbitrary $[n,k]$ code $\mathcal{C}$ is given in the following theorem.

Theorem 1. *Consider a DSS that uses an $[n,k]$ code $\mathcal{C}$ to store f files. If a PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})$ exists, then the PIR rate*

$$\mathsf{R}(\mathcal{C})=\frac{(\nu-\kappa)k}{\kappa n}\left[1-\left(\frac{\kappa}{\nu}\right)^f\right]^{-1} \quad (13)$$

is achievable.

Proof: See Appendix B. ∎

We remark that from Lemma 2, (13) is smaller than or equal to the finite MDS-PIR capacity in (7) since

$$\mathsf{R}(\mathcal{C})=\frac{\frac{\nu k}{\kappa n}\left[1-\frac{\kappa}{\nu}\right]}{\left[1-\left(\frac{\kappa}{\nu}\right)^f\right]}=\frac{\nu k}{\kappa n}\left[1+\frac{\kappa}{\nu}+\dots+\left(\frac{\kappa}{\nu}\right)^{f-1}\right]^{-1}\\ \le\left[1+\frac{k}{n}+\dots+\left(\frac{k}{n}\right)^{f-1}\right]^{-1}, \quad (14)$$

and it becomes the finite MDS-PIR capacity in (7) if there exists a matrix $\mathbf{\Lambda}_{\kappa,\nu}$ for $\mathcal{C}$ with $\frac{\kappa}{\nu}=\frac{k}{n}$. The inequality in (14) follows from Lemma 2.

Corollary 1. *If a PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})$ with $\frac{\kappa}{\nu}=\frac{k}{n}$ exists for an $[n,k]$ code $\mathcal{C}$, then the finite MDS-PIR capacity in (7) is achievable.*

This gives rise to the following definition.

Definition 13. *A PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})$ with $\frac{\kappa}{\nu}=\frac{k}{n}$ for an $[n,k]$ code $\mathcal{C}$ is called an* MDS-PIR capacity-achieving *matrix, and $\mathcal{C}$ is referred to as an* MDS-PIR capacity-achieving *code.*

We remark that there might exist codes that are MDS-PIR capacity-achieving for which an MDS-PIR capacity-achieving matrix does not exist.

Note that the largest achievable PIR rate in the noncolluding case where data is stored using an arbitrary linear code is still unknown. Interestingly, it is observed from Lemma 2 and (14) that the largest possible achievable PIR rate for an arbitrary linear code with Protocol 1 strongly depends on the smallest possible value of $\frac{\kappa}{\nu}$ for which a PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}$ exists. We stress that the existence of an MDS-PIR capacity-achieving matrix $\mathbf{\Lambda}_{\kappa,\nu}$ does not necessarily require $(\nu,\kappa)=(n,k)$, but $\frac{\kappa}{\nu}=\frac{k}{n}$.

Since the existence of a PIR achievable rate matrix is connected to the information sets of a code, we review a widely known result in coding theory.

Proposition 1 ([38, Th. 1.4.15])**.** *Let $\mathcal{C}$ be an $[n,k,d^{\mathcal{C}}_{\mathsf{min}}]$ code. Then, every set of $n-d^{\mathcal{C}}_{\mathsf{min}}+1$ coordinates of $\mathcal{C}$ contains an*

information set. Furthermore, $n - d^{\mathcal{C}}_{\min} + 1$ is the smallest number of coordinates with this property.

Lemma 3. *For a given $[n, k, d^{\mathcal{C}}_{\min}]$ code $\mathcal{C}$, there always exists a PIR achievable rate matrix $\mathbf{\Lambda}_{k,\nu}$ with*

$$\nu = k + \min\left(k, d^{\mathcal{C}}_{\min} - 1\right).$$

Proof: See Appendix C. ∎

A lower bound on the largest possible achievable PIR rate obtained from Theorem 1 and Lemma 3 is given as follows.

Corollary 2. *Consider a DSS that uses an $[n, k, d^{\mathcal{C}}_{\min}]$ code $\mathcal{C}$ to store f files. Then, the PIR rate*

$$\mathsf{R}(\mathcal{C}) = \frac{\min\left(k, d^{\mathcal{C}}_{\min} - 1\right)}{n}\left[1 - \left(\frac{k}{k + \min\left(k, d^{\mathcal{C}}_{\min} - 1\right)}\right)^{f}\right]^{-1}$$

is achievable.

We remark that because every set of k coordinates of an $[n, k]$ MDS code is an information set, we can construct n information sets by cyclically shifting an arbitrary information set n times, hence an MDS-PIR capacity-achieving matrix $\mathbf{\Lambda}_{k,n}$ of an MDS code can be easily constructed. In other words, Protocol 1 with MDS codes is MDS-PIR capacity-achieving (see Corollary 1) and MDS codes are a class of MDS-PIR capacity-achieving codes.

Remark 1. *Since minimum storage regenerating (MSR) codes are MDS codes [39] and can be viewed as scalar linear codes over a larger extension field, it follows that MSR codes are also MDS-PIR capacity-achieving codes.*

In Section VI, we provide a necessary and a sufficient condition for an arbitrary linear code to achieve the MDS-PIR capacity with Protocol 1 and give certain families of MDS-PIR capacity-achieving codes. For illustration purposes, in the next subsection, we give an example of an MDS-PIR capacity-achieving code.

D. A $[5, 3, 2]$ MDS-PIR Capacity-Achieving Code for $f = 2$

In this subsection, we compute the PIR achievable rate of a $[5, 3, 2]$ non-MDS code for a DSS that stores two files, $f = 2$, and show that it is MDS-PIR capacity-achieving.

Let $\mathcal{C}$ be a non-MDS $[5, 3, 2]$ binary code with generator matrix

$$\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{15}$$

One can see that the $\nu \times n = 5 \times 5$ matrix

$$\mathbf{\Lambda}_{3,5} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is a PIR achievable rate matrix. From $\mathbf{\Lambda}_{3,5}$, we obtain the following sets:

$$\chi(\boldsymbol{\lambda}_1) = \{1, 2, 3\}, \chi(\boldsymbol{\lambda}_2) = \{1, 4, 5\}, \chi(\boldsymbol{\lambda}_3) = \{2, 4, 5\},$$
$$\chi(\boldsymbol{\lambda}_4) = \{2, 3, 4\}, \chi(\boldsymbol{\lambda}_5) = \{1, 3, 5\}.$$

All of these sets contain an information set of $\mathcal{C}$ (see Definition 10). Furthermore, we get the following PIR interference matrices

$$\boldsymbol{A}_{3\times 5} = \begin{pmatrix} 1 & 1 & 1 & 2 & 2 \\ 2 & 3 & 4 & 3 & 3 \\ 5 & 4 & 5 & 4 & 5 \end{pmatrix},$$

$$\boldsymbol{B}_{2\times 5} = \begin{pmatrix} 3 & 2 & 2 & 1 & 1 \\ 4 & 5 & 3 & 5 & 4 \end{pmatrix}.$$

One can see that Claim 1 holds. For example, $\mathcal{S}(3|\boldsymbol{A}_{3\times 5}) = \{2, 4, 5\}$ contains an information set for $\mathcal{C}$.

In the next step, for each $m \in \mathbb{N}_2$ and for $\beta = \nu^f = 5^2$, we first generate the interleaved code array $\boldsymbol{Y}^{(m)}$ with row vectors $\boldsymbol{y}^{(m)}_i = \boldsymbol{c}^{(m)}_{\pi(i)}$, $i \in \mathbb{N}_{5^2}$, by a randomly selected permutation function $\pi(\cdot)$. Suppose that the user wishes to obtain $\boldsymbol{X}^{(1)}$. We list all downloaded sums of code symbols in Table I, which is similar to [11, Table II]. Similar to the PIR protocol in [11], Protocol 1 requires $f = 2$ rounds in each repetition, and the scheme needs to be repeated $\kappa = 3$ times. Note that since the protocol requests an equal amount of code symbols associated with $\boldsymbol{X}^{(1)}$ and $\boldsymbol{X}^{(2)}$, it is straightforward to see that the privacy constraint is satisfied.

It should be mentioned that here we strongly make use of the PIR interference matrices. For example, in round 2 of repetition 1 (see Table I), since the user knows $\mathcal{C}$, the code symbols $y^{(2)}_{5\cdot 0+3,1}$, $y^{(2)}_{5\cdot 0+2,2}$, and $y^{(2)}_{5\cdot 0+2,3}$ can be obtained by knowing $\{y^{(2)}_{5\cdot 0+3,2}, y^{(2)}_{5\cdot 0+3,4}, y^{(2)}_{5\cdot 0+3,5}\}$ and $\{y^{(2)}_{5\cdot 0+2,1}, y^{(2)}_{5\cdot 0+2,4}, y^{(2)}_{5\cdot 0+2,5}\}$, from which the corresponding coded symbols $\{y^{(1)}_{3\cdot 5+1,1}, y^{(1)}_{3\cdot 5+1,2}, y^{(1)}_{3\cdot 5+1,3}\}$ can be obtained by cancelling the side information. Since $\{1, 2, 3\}$ is an information set, the corresponding requested file vector of length $k = 3$ can also be decoded. Hence, in summary, it is sufficient to reliably decode $5^2 = 25$ different length-k requested file vectors for $m = 1$. In summary, for $f = 2$, the user downloads 3×5 undesired symbols based on (9) and $(3 + 2) \times 5 = 25$ desired symbols according to (10) and (11) in each repetition. Hence, the PIR achievable rate is equal to

$$\mathsf{R}(\mathcal{C}) = \frac{3 \cdot 25}{3 \cdot (25 + 15)} = \frac{5}{8} = \frac{1 - \frac{3}{5}}{1 - \left(\frac{3}{5}\right)^2},$$

which corresponds to the finite MDS-PIR capacity in (7) with $f = 2$, i.e., the $[5, 3, 2]$ non-MDS code given by (15) is MDS-PIR capacity-achieving.

V. Asymptotic MDS-PIR Capacity-Achieving Protocol for the Noncolluding Case

In this section, we present Protocol 2, a PIR protocol with PIR rate independent of the number of files that achieves the asymptotic MDS-PIR capacity in (8) for the case of noncolluding nodes. We assume that the DSS uses an $[n, k]$ code $\mathcal{C}$ over $\mathrm{GF}(q)$ of rate $R^{\mathcal{C}}$ and subpacketization α. For such a code $\mathcal{C}$, the user designs the l-th, $l \in \mathbb{N}_n$, query as

$$\boldsymbol{Q}^{(l)} = \boldsymbol{U} + \boldsymbol{V}^{(l)}, \tag{16}$$

TABLE I
PROTOCOL 1 WITH A $[5, 3, 2]$ NON-MDS CODE FOR $f = 2$.

		Server 1	Server 2	Server 3	Server 4	Server 5
repetition 1	round 1	$y^{(1)}_{3(1-1)+1,1}$ $y^{(1)}_{3(1-1)+2,1}$ $y^{(1)}_{3(1-1)+3,1}$	$y^{(1)}_{3(1-1)+1,2}$ $y^{(1)}_{3(1-1)+2,2}$ $y^{(1)}_{3(1-1)+3,2}$	$y^{(1)}_{3(1-1)+1,3}$ $y^{(1)}_{3(1-1)+2,3}$ $y^{(1)}_{3(1-1)+3,3}$	$y^{(1)}_{3(2-1)+1,4}$ $y^{(1)}_{3(2-1)+2,4}$ $y^{(1)}_{3(2-1)+3,4}$	$y^{(1)}_{3(2-1)+1,5}$ $y^{(1)}_{3(2-1)+2,5}$ $y^{(1)}_{3(2-1)+3,5}$
		$y^{(2)}_{5\cdot0+1,1}$ $y^{(2)}_{5\cdot0+2,1}$ $y^{(2)}_{5\cdot0+5,1}$	$y^{(2)}_{5\cdot0+1,2}$ $y^{(2)}_{5\cdot0+3,2}$ $y^{(2)}_{5\cdot0+4,2}$	$y^{(2)}_{5\cdot0+1,3}$ $y^{(2)}_{5\cdot0+4,3}$ $y^{(2)}_{5\cdot0+5,3}$	$y^{(2)}_{5\cdot0+2,4}$ $y^{(2)}_{5\cdot0+3,4}$ $y^{(2)}_{5\cdot0+4,4}$	$y^{(2)}_{5\cdot0+2,5}$ $y^{(2)}_{5\cdot0+3,5}$ $y^{(2)}_{5\cdot0+5,5}$
	rnd. 2	$y^{(1)}_{3\cdot5+1,1}+y^{(2)}_{5\cdot0+3,1}$ $y^{(1)}_{4\cdot5+1,1}+y^{(2)}_{5\cdot0+4,1}$	$y^{(1)}_{3\cdot5+1,2}+y^{(2)}_{5\cdot0+2,2}$ $y^{(1)}_{4\cdot5+1,2}+y^{(2)}_{5\cdot0+5,2}$	$y^{(1)}_{3\cdot5+1,3}+y^{(2)}_{5\cdot0+2,3}$ $y^{(1)}_{4\cdot5+1,3}+y^{(2)}_{5\cdot0+3,3}$	$y^{(1)}_{3\cdot5+2,4}+y^{(2)}_{5\cdot0+1,4}$ $y^{(1)}_{4\cdot5+2,4}+y^{(2)}_{5\cdot0+5,4}$	$y^{(1)}_{3\cdot5+2,5}+y^{(2)}_{5\cdot0+1,5}$ $y^{(1)}_{4\cdot5+2,5}+y^{(2)}_{5\cdot0+4,5}$
repetition 2	round 1	$y^{(1)}_{3(2-1)+1,1}$ $y^{(1)}_{3(2-1)+2,1}$ $y^{(1)}_{3(2-1)+3,1}$	$y^{(1)}_{3(3-1)+1,2}$ $y^{(1)}_{3(3-1)+2,2}$ $y^{(1)}_{3(3-1)+3,2}$	$y^{(1)}_{3(4-1)+1,3}$ $y^{(1)}_{3(4-1)+2,3}$ $y^{(1)}_{3(4-1)+3,3}$	$y^{(1)}_{3(3-1)+1,4}$ $y^{(1)}_{3(3-1)+2,4}$ $y^{(1)}_{3(3-1)+3,4}$	$y^{(1)}_{3(3-1)+1,5}$ $y^{(1)}_{3(3-1)+2,5}$ $y^{(1)}_{3(3-1)+3,5}$
		$y^{(2)}_{5\cdot1+1,1}$ $y^{(2)}_{5\cdot1+2,1}$ $y^{(2)}_{5\cdot1+5,1}$	$y^{(2)}_{5\cdot1+1,2}$ $y^{(2)}_{5\cdot1+3,2}$ $y^{(2)}_{5\cdot1+4,2}$	$y^{(2)}_{5\cdot1+1,3}$ $y^{(2)}_{5\cdot1+4,3}$ $y^{(2)}_{5\cdot1+5,3}$	$y^{(2)}_{5\cdot1+2,4}$ $y^{(2)}_{5\cdot1+3,4}$ $y^{(2)}_{5\cdot1+4,4}$	$y^{(2)}_{5\cdot1+2,5}$ $y^{(2)}_{5\cdot1+3,5}$ $y^{(2)}_{5\cdot1+5,5}$
	rnd. 2	$y^{(1)}_{3\cdot5+2,1}+y^{(2)}_{5+3,1}$ $y^{(1)}_{4\cdot5+2,1}+y^{(2)}_{5+4,1}$	$y^{(1)}_{3\cdot5+3,2}+y^{(2)}_{5+2,2}$ $y^{(1)}_{4\cdot5+3,2}+y^{(2)}_{5+5,2}$	$y^{(1)}_{3\cdot5+4,3}+y^{(2)}_{5+2,3}$ $y^{(1)}_{4\cdot5+4,3}+y^{(2)}_{5+3,3}$	$y^{(1)}_{3\cdot5+3,4}+y^{(2)}_{5+1,4}$ $y^{(1)}_{4\cdot5+3,4}+y^{(2)}_{5+5,4}$	$y^{(1)}_{3\cdot5+3,5}+y^{(2)}_{5+1,5}$ $y^{(1)}_{4\cdot5+3,5}+y^{(2)}_{5+4,5}$
repetition 3	round 1	$y^{(1)}_{3(5-1)+1,1}$ $y^{(1)}_{3(5-1)+2,1}$ $y^{(1)}_{3(5-1)+3,1}$	$y^{(1)}_{3(4-1)+1,2}$ $y^{(1)}_{3(4-1)+2,2}$ $y^{(1)}_{3(4-1)+3,2}$	$y^{(1)}_{3(5-1)+1,3}$ $y^{(1)}_{3(5-1)+2,3}$ $y^{(1)}_{3(5-1)+3,3}$	$y^{(1)}_{3(4-1)+1,4}$ $y^{(1)}_{3(4-1)+2,4}$ $y^{(1)}_{3(4-1)+3,4}$	$y^{(1)}_{3(5-1)+1,5}$ $y^{(1)}_{3(5-1)+2,5}$ $y^{(1)}_{3(5-1)+3,5}$
		$y^{(2)}_{5\cdot2+1,1}$ $y^{(2)}_{5\cdot2+2,1}$ $y^{(2)}_{5\cdot2+5,1}$	$y^{(2)}_{5\cdot2+1,2}$ $y^{(2)}_{5\cdot2+3,2}$ $y^{(2)}_{5\cdot2+4,2}$	$y^{(2)}_{5\cdot2+1,3}$ $y^{(2)}_{5\cdot2+4,3}$ $y^{(2)}_{5\cdot2+5,3}$	$y^{(2)}_{5\cdot2+2,4}$ $y^{(2)}_{5\cdot2+3,4}$ $y^{(2)}_{5\cdot2+4,4}$	$y^{(2)}_{5\cdot2+2,5}$ $y^{(2)}_{5\cdot2+3,5}$ $y^{(2)}_{5\cdot2+5,5}$
	rnd. 2	$y^{(1)}_{3\cdot5+5,1}+y^{(2)}_{5\cdot2+3,1}$ $y^{(1)}_{4\cdot5+5,1}+y^{(2)}_{5\cdot2+4,1}$	$y^{(1)}_{3\cdot5+4,2}+y^{(2)}_{5\cdot2+2,2}$ $y^{(1)}_{4\cdot5+4,2}+y^{(2)}_{5\cdot2+5,2}$	$y^{(1)}_{3\cdot5+5,3}+y^{(2)}_{5\cdot2+2,3}$ $y^{(1)}_{4\cdot5+5,3}+y^{(2)}_{5\cdot2+3,3}$	$y^{(1)}_{3\cdot5+4,4}+y^{(2)}_{5\cdot2+1,4}$ $y^{(1)}_{4\cdot5+4,4}+y^{(2)}_{5\cdot2+5,4}$	$y^{(1)}_{3\cdot5+5,5}+y^{(2)}_{5\cdot2+1,5}$ $y^{(1)}_{4\cdot5+5,5}+y^{(2)}_{5\cdot2+4,5}$

where $\boldsymbol{U} = (u_{i,j})$ is a $d \times \beta f$ matrix whose elements $u_{i,j}$ are chosen independently and uniformly at random from $\mathrm{GF}(q)$ and whose purpose is to make $\boldsymbol{Q}^{(l)}$ appear random and thus ensure privacy. $\boldsymbol{V}^{(l)} = \left(v^{(l)}_{i,j}\right)$ is a $d \times \beta f$ deterministic binary matrix over $\mathrm{GF}(q)$, where $v^{(l)}_{i,j} = 1$ means that the j-th symbol in node l is accessed by the i-th subquery of $\boldsymbol{Q}^{(l)}$, that allows recovery of the requested data by the user. Matrix $\boldsymbol{V}^{(l)}$ is constructed from a $d \times n$ matrix $\hat{\boldsymbol{E}}$, as explained below.

Let $\mathcal{I}_1, \ldots, \mathcal{I}_\beta$ be β information sets for $\mathcal{C}$ (which are implicitly linked to the β stripes of each file) and define $\mathcal{F}_l \triangleq \{i \in \mathbb{N}_\beta \colon l \in \mathcal{I}_i\}$ to be the set of indices of the information sets $\mathcal{I}_1, \ldots, \mathcal{I}_\beta$ containing the l-th coordinate of $\mathcal{C}$. Then, $\hat{\boldsymbol{E}} = (\hat{e}_{i,l})$ is a binary matrix of size $d \times n$ that has the following structure.

- C1. Each row, denoted by $\hat{\boldsymbol{e}}_i$, $i \in \mathbb{N}_d$, has Hamming weight $w_{\mathsf{H}}(\hat{\boldsymbol{e}}_i) = \Gamma$.
- C2. Each row $\hat{\boldsymbol{e}}_i$ is an erasure pattern that is correctable by $\mathcal{C}$.
- C3. Each column, denoted by $\boldsymbol{t}_l$, $l \in \mathbb{N}_n$, has weight $w_{\mathsf{H}}(\boldsymbol{t}_l) = |\mathcal{F}_l|$, i.e., the weight of the l-th column of $\hat{\boldsymbol{E}}$ is the number of times the l-th coordinate of the storage code $\mathcal{C}$ appears in the β information sets $\mathcal{I}_1, \ldots, \mathcal{I}_\beta$.

For later use, we call the vector $(w_{\mathsf{H}}(\boldsymbol{t}_1), \ldots, w_{\mathsf{H}}(\boldsymbol{t}_n))$ the *column weight profile of* $\hat{\boldsymbol{E}}$.

Matrix $\boldsymbol{V}^{(l)}$ is constructed from $\hat{\boldsymbol{E}}$ such that if $\hat{e}_{i,l} = 1$, then the i-th subquery of the l-th query, $\boldsymbol{q}^{(l)}_i$, accesses a code symbol stored in the l-th node. Additionally, $\hat{\boldsymbol{E}}$ is a matrix having strictly Γd nonzero entries, ensuring that Γd code symbols are downloaded by the protocol. We defer the intuition behind the three conditions above until later in this section. More precisely, matrix $\boldsymbol{V}^{(l)}$ is constructed from $\hat{\boldsymbol{E}}$ as follows. For $l \in \mathbb{N}_n$, $\boldsymbol{V}^{(l)}$ has the form

$$\boldsymbol{V}^{(l)} = \left(\boldsymbol{0}_{d\times(m-1)\beta} \mid \boldsymbol{\Delta}_l \mid \boldsymbol{0}_{d\times(f-m)\beta}\right),$$

where $\boldsymbol{\Delta}_l$ is the $d \times \beta$ binary matrix

$$\boldsymbol{\Delta}_l = \left(\boldsymbol{\omega}^{\mathsf{T}}_{j^{(l)}_1} \mid \boldsymbol{\omega}^{\mathsf{T}}_{j^{(l)}_2} \mid \cdots \mid \boldsymbol{\omega}^{\mathsf{T}}_{j^{(l)}_d}\right)^{\mathsf{T}}, \tag{17}$$

with $\boldsymbol{\omega}_j$, $j \in \mathbb{N}_\beta$, being the j-th β-dimensional unit vector, i.e., a length-β weight-1 binary vector with a single 1 at the j-th position and $\boldsymbol{\omega}_0 = \boldsymbol{0}_{1\times\beta}$. Also, given a chosen $d \times n$ matrix $\hat{\boldsymbol{E}}$,

$$j^{(l)}_i = \begin{cases} s^{(l)}_i & \text{if } \hat{e}_{i,l} = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where $s^{(l)}_i \in \mathcal{F}_l$ and $s^{(l)}_i \neq s^{(l)}_{i'}$ for $i \neq i'$, $i, i' \in \mathbb{N}_d$. This completes the construction of the protocol.

Now, we provide the intuition behind conditions C1, C2, and C3 above.

- Condition C1 stems from the fact that the user should be able to recover Γ unique code symbols of the requested file $\boldsymbol{X}^{(m)}$ from the i-th subqueries $\boldsymbol{q}_i^{(l)}$ that are sent to the n nodes. Thus, each row of $\hat{\boldsymbol{E}}$ should have exactly Γ ones.
- For C2, consider an arbitrary row $\hat{\boldsymbol{e}}_i$ of $\hat{\boldsymbol{E}}$. The corresponding set of n subqueries $\{\boldsymbol{q}_i^{(1)}, \ldots, \boldsymbol{q}_i^{(n)}\}$ trigger a response from the n nodes of the form

$$r_{l,i} = \begin{cases} Y_l + \phi_l & \text{if } \hat{e}_{i,l} = 1, \\ Y_l & \text{otherwise,} \end{cases}$$

  where ϕ_l represents a code symbol present in the l-th node, and Y_l is some interference symbol generated due to the product between $\boldsymbol{q}_i^{(l)}$ and the content of the l-th node. The vector $(Y_1, \ldots, Y_n)$ represents a codeword of $\mathcal{C}$ (see also Theorem 2 below and its proof in Appendix D for further details). In order to recover ϕ_l, $l \in \chi(\hat{\boldsymbol{e}}_i)$, we need to know Y_l. This can be seen as a decoding problem over the binary erasure channel. In other words, the i-th row of $\hat{\boldsymbol{E}}$ should be an erasure pattern that is correctable by $\mathcal{C}$.
- Condition C3 comes from the fact that the protocol should be able to recover $w_{\mathsf{H}}(\boldsymbol{t}_l)$ unique code symbols from the l-th node.

The idea behind the construction of $\boldsymbol{V}^{(l)}$ from $\hat{\boldsymbol{E}}$ is that the retrieval process can be cast as the correction of an erasure pattern. Thus, we design $\boldsymbol{V}^{(l)}$ (and subsequently the responses) so that erasure correction is possible.

We remark that for a code $\mathcal{C}$, $\hat{\boldsymbol{E}}$ and $\{\mathcal{I}_i\}_{i \in \mathbb{N}_\beta}$ need not be unique. Furthermore, each set $\mathcal{I}_i$, $i \in \mathbb{N}_\beta$, can alternatively be represented as a correctable erasure pattern $\bar{\boldsymbol{e}}_i = (\bar{e}_{i,1}, \ldots, \bar{e}_{i,n})$, where $\bar{e}_{i,l} = 0$, $\forall\, l \in \mathcal{I}_i$. Also, the information sets $\{\mathcal{I}_i\}_{i \in \mathbb{N}_\beta}$ can alternatively be defined by a matrix $\bar{\boldsymbol{E}}$ of size $\beta \times n$ as

$$\bar{\boldsymbol{E}} = \begin{pmatrix} \bar{\boldsymbol{e}}_1 \\ \vdots \\ \bar{\boldsymbol{e}}_\beta \end{pmatrix}.$$

The two matrices $\hat{\boldsymbol{E}}$ and $\bar{\boldsymbol{E}}$ can be stacked into the matrix $\boldsymbol{E} = (e_{i,l})$ as

$$\boldsymbol{E} = \begin{pmatrix} \hat{\boldsymbol{E}} \\ \bar{\boldsymbol{E}} \end{pmatrix}. \tag{19}$$

To meet condition C3, for each $l \in \mathbb{N}_n$, $w_{\mathsf{H}}(\boldsymbol{t}_l) = \beta - w_{\mathsf{H}}(\boldsymbol{w}_l)$, where $\boldsymbol{t}_l$ and $\boldsymbol{w}_l$ are columns of $\hat{\boldsymbol{E}}$ and $\bar{\boldsymbol{E}}$, respectively. It follows that meeting all three conditions C1, C2, and C3 is equivalent to finding a $(\beta + d) \times n$ β-column regular matrix $\boldsymbol{E}$ in which each row is a correctable erasure pattern. Hence, we conclude that the requirements for $\boldsymbol{E}$ are equivalent to finding a PIR achievable rate matrix

$$\boldsymbol{\Lambda}_{d,\beta+d}(\mathcal{C}) = \mathbf{1}_{(\beta+d)\times n} - \boldsymbol{E}_{(\beta+d)\times n}, \tag{20}$$

where β and d are chosen according to $\beta k = \Gamma d$.

In the following lemma, we prove that our construction of the queries ensures that the privacy condition (5a) is satisfied.

Lemma 4. *Consider a DSS that uses an $[n, k]$ code with subpacketization α to store f files, each divided into β stripes. Then, the queries $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_n$, designed as in* (16) *satisfy $\mathsf{H}\big(m|\boldsymbol{Q}^{(l)}\big) = \mathsf{H}(m)$, where $l \in \mathbb{N}_n$ represents the spy node.*

Proof: The queries $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_n$, are a sum of a random matrix $\boldsymbol{U}$ and a deterministic matrix $\boldsymbol{V}^{(l)}$. The resulting queries have elements that are independently and uniformly distributed at random from $\mathrm{GF}(q)$. Hence, any $\boldsymbol{Q}^{(l)}$ obtained by the spy node is statistically independent of m. This ensures that $\mathsf{H}\big(m|\boldsymbol{Q}^{(l)}\big) = \mathsf{H}(m)$. ∎

The following theorem shows that Protocol 2 achieves perfect information-theoretic PIR, and it gives its achievable PIR rate, $\mathsf{R}(\mathcal{C})$. Note that to prove perfect information-theoretic PIR it remains to be shown that from the responses $\boldsymbol{r}_l$ in (4) sent by the nodes back to the user, one can recover the requested file, i.e., that the constructed PIR protocol satisfies the recovery condition in (5b).

Theorem 2. *Consider a DSS that uses an $[n, k]$ code with subpacketization α to store f files, each divided into β stripes. If there exists a Γ-row regular matrix $\hat{\boldsymbol{E}}$ satisfying conditions* C1*,* C2*, and* C3*, then $\mathsf{H}\big(\boldsymbol{X}^{(m)}|\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\big) = 0$ and the PIR rate*

$$\mathsf{R}(\mathcal{C}) = \frac{\Gamma}{n} \leq \frac{n-k}{n}$$

is achievable.

Proof: See Appendix D. ∎

Theorem 2 generalizes [12, Th. 1] to any linear code.

Corollary 3. *If for an $[n, k]$ code $\mathcal{C}$ there exists an $(n-k)$-regular matrix $\boldsymbol{E}$ satisfying conditions* C1*,* C2*, and* C3*, then Protocol 2 achieves the asymptotic MDS-PIR capacity C_∞ in* (8)*.*

Remark 2. *From* (20)*, if there exists an $(n-k)$-regular matrix $\boldsymbol{E}$ satisfying conditions* C1*,* C2*, and* C3*, a $\boldsymbol{\Lambda}_{\kappa,\nu}$ MDS-PIR capacity-achieving matrix with $\frac{\kappa}{\nu} = \frac{k}{n}$ exists. Thus, if a code achieves the asymptotic MDS-PIR capacity C_∞ with Protocol 2, it also achieves the finite MDS-PIR capacity C_f with Protocol 1.*

Note that the parameters Γ, β mentioned in Theorem 2, and d (which is not explicitly mentioned) have to be carefully selected such that $\beta k = \Gamma d$ and such that a Γ-row regular matrix $\hat{\boldsymbol{E}}$ (satisfying condition C3) actually exists with a valid collection of information sets $\{\mathcal{I}_i\}_{i \in \mathbb{N}_\beta}$. In the following corollary, we provide a valid set of values.

Corollary 4. *Let $\mathcal{C}$ be an $[n, k, d_{\mathsf{min}}^{\mathcal{C}}]$ code. For $\Gamma = \min(k, d_{\mathsf{min}}^{\mathcal{C}} - 1)$, it holds that*

$$\mathsf{H}\big(\boldsymbol{X}^{(m)}|\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\big) = 0, \tag{21}$$

and the PIR rate $\mathsf{R}(\mathcal{C}) = \frac{\min\left(k, d_{\mathsf{min}}^{\mathcal{C}}-1\right)}{n}$ is achievable.

Proof: Let $d = k$ and $\beta = \Gamma$. Then, (21) follows directly from Theorem 2, since we have shown in Lemma 3 that the required matrix $\boldsymbol{\Lambda}_{k,\Gamma+k}(\mathcal{C})$ exists for $\mathcal{C}$, and the existence of $\boldsymbol{E}_{(\Gamma+k)\times n}$ follows from (20). ∎

The above corollary provides a lower bound on the value of Γ for any code. In other words, it allows us to design a PIR protocol with PIR rate greater than or equal to $\min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1)/n$. We remark that with a better designed $\hat{\boldsymbol{E}}$, it may be possible to achieve a higher PIR rate. For systematic codes with rate $R^{\mathcal{C}} > 1/2$, a better lower bound on the maximum achievable PIR rate compared to that of Corollary 4 is given below.

Corollary 5. *Let $\mathcal{C}$ be an $[n, k]$ systematic code with $R^{\mathcal{C}} > 1/2$ and $\boldsymbol{H}^{\mathcal{C}} = (\boldsymbol{P} \mid \boldsymbol{I}_{n-k})$. Consider the $[n = k, k']$ code $\mathcal{C}'$ with parity-check matrix $\boldsymbol{H}^{\mathcal{C}'} = \boldsymbol{P}$. For $\Gamma = d^{\mathcal{C}'}_{\mathsf{min}} - 1$, it holds that*

$$\mathsf{H}\big(\boldsymbol{X}^{(m)} | \boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\big) = 0, \tag{22}$$

and the PIR rate $\mathsf{R}(\mathcal{C}) = \frac{d^{\mathcal{C}'}_{\mathsf{min}} - 1}{n}$ is achievable.

Proof: As for the proof of Corollary 4, let $d = k$ and $\beta = \Gamma$. Then, (22) follows directly from Theorem 2. Select k erasure patterns $\hat{\boldsymbol{e}}'_i$, $i \in \mathbb{N}_k$, of length k, such that $w_{\mathsf{H}}(\hat{\boldsymbol{e}}'_i) = d^{\mathcal{C}'}_{\mathsf{min}} - 1$ and $\hat{\boldsymbol{e}}'_{i+1}$ is a right cyclic shift of $\hat{\boldsymbol{e}}'_i$, $i \in \mathbb{N}_{k-1}$. The patterns are all correctable by the code $\mathcal{C}'$. Thus, the erasure patterns

$$\hat{\boldsymbol{e}}_i = (\hat{\boldsymbol{e}}'_i, \underbrace{0, \cdots, 0}_{n-k})$$

are also correctable by $\mathcal{C}$. Choosing the information sets $\mathcal{I}_i = \mathbb{N}_k$, $i \in \mathbb{N}_\Gamma$, the required Γ-row regular matrix $\hat{\boldsymbol{E}}$ satisfying conditions C1, C2, and C3 can then be constructed from $\{\hat{\boldsymbol{e}}_i\}_{i \in \mathbb{N}_k}$ and $\{\mathcal{I}_i\}_{i \in \mathbb{N}_\Gamma}$. ∎

Observe that $R^{\mathcal{C}} > \frac{1}{2}$ implies $k > d^{\mathcal{C}}_{\mathsf{min}} - 1$. In [29], under the assumption that $k > d^{\mathcal{C}}_{\mathsf{min}} - 1$, a PIR protocol achieving a PIR rate of $(d^{\mathcal{C}}_{\mathsf{min}} - 1)/n$ was given. Note that $d^{\mathcal{C}}_{\mathsf{min}} \leq d^{\mathcal{C}'}_{\mathsf{min}}$, and thus $\mathsf{R}(\mathcal{C}) \geq (d^{\mathcal{C}}_{\mathsf{min}} - 1)/n$ for our construction.

Below we give two examples to elucidate Protocol 2. Example 4 illustrates the PIR protocol when the underlying code has rate $R^{\mathcal{C}} > 1/2$, with parameters $d = k$ and $\beta = \Gamma$. On the other hand, Example 5 uses an underlying code that has rate $R^{\mathcal{C}} < 1/2$, again with parameters $d = k$ and $\beta = \Gamma$.

Example 4. *Consider a DSS that uses the $[5, 3, 2]$ scalar ($\alpha = 1$) binary code $\mathcal{C}$ in Section IV-D (with generator matrix given in (15)) to store a single file by dividing it into β stripes. Its parity-check matrix is given by*

$$\boldsymbol{H}^{\mathcal{C}} = (\boldsymbol{P} | \boldsymbol{I}_{n-k}) = \left(\begin{array}{ccc|cc} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right).$$

To determine the value of the parameter β, we compute the minimum Hamming distance $d^{\mathcal{C}'}_{\mathsf{min}}$ of the $[n' = 3, k' = 1]$ code $\mathcal{C}'$ with parity-check matrix $\boldsymbol{H}^{\mathcal{C}'} = \boldsymbol{P}$. From $\boldsymbol{H}^{\mathcal{C}'}$ it follows that $d^{\mathcal{C}'}_{\mathsf{min}} = 3$. Hence, from Corollary 5, $\beta = 2$. Let the file to be stored be denoted by the 2×3 matrix $\boldsymbol{X} = (x_{i,j})$, where the message symbols $x_{i,j} \in \mathrm{GF}(2^\ell)$ for $\ell \in \mathbb{N}$. Then,

$$\boldsymbol{C} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,1} + x_{1,2} & x_{1,2} + x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,1} + x_{2,2} & x_{2,2} + x_{2,3} \end{pmatrix}.$$

The user wants to download the file $\boldsymbol{X}$ from the DSS and sends a query $\boldsymbol{Q}^{(l)}$, $l \in \mathbb{N}_5$, to the l-th storage node. The queries take the form shown in (16). For $l \in \mathbb{N}_5$, we construct the matrix $\boldsymbol{V}^{(l)} = \boldsymbol{\Delta}_l$ by choosing an appropriate matrix $\hat{\boldsymbol{E}}$. To do this, we carefully choose the information sets $\mathcal{I}_1 = \{1, 2, 3\}$ and $\mathcal{I}_2 = \{1, 2, 3\}$ (and hence $\boldsymbol{V}^{(4)} = \boldsymbol{V}^{(5)} = \boldsymbol{0}_{d \times \beta}$). This allows us to generate a column weight profile in $\hat{\boldsymbol{E}}$. More specifically, let $\boldsymbol{t}_l$ be the l-th column of $\hat{\boldsymbol{E}}$, $l \in \mathbb{N}_5$. We have $w_{\mathsf{H}}(\boldsymbol{t}_1) = w_{\mathsf{H}}(\boldsymbol{t}_2) = w_{\mathsf{H}}(\boldsymbol{t}_3) = 2$ and $w_{\mathsf{H}}(\boldsymbol{t}_4) = w_{\mathsf{H}}(\boldsymbol{t}_5) = 0$. A valid matrix $\hat{\boldsymbol{E}}$ is

$$\hat{\boldsymbol{E}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

and we construct $\boldsymbol{\Delta}_1$ according to (17). Focusing on the first column of $\hat{\boldsymbol{E}}$, we can see that the first two rows have a one in the first position. Thus, we choose $j^{(1)}_1 = s^{(1)}_1$, $j^{(1)}_2 = s^{(1)}_2$, and $j^{(1)}_3 = 0$, since $\hat{e}_{1,1} = 1$, $\hat{e}_{2,1} = 1$, and $\hat{e}_{3,1} = 0$. We take $s^{(1)}_1, s^{(1)}_2 \in \mathbb{N}_2$. We arbitrarily choose $s^{(1)}_1 = 1$ and $s^{(1)}_2 = 2$ to get

$$\boldsymbol{\Delta}_1 = \begin{pmatrix} \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Similarly, we construct

$$\boldsymbol{\Delta}_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \boldsymbol{\Delta}_3 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}.$$

The queries $\boldsymbol{Q}^{(l)}$ are sent to the respective nodes and the responses

$$\boldsymbol{r}_1 = \begin{pmatrix} u_{1,1}x_{1,1} + u_{1,2}x_{2,1} + x_{1,1} \\ u_{2,1}x_{1,1} + u_{2,2}x_{2,1} + x_{2,1} \\ u_{3,1}x_{1,1} + u_{3,2}x_{2,1} \end{pmatrix} = \begin{pmatrix} I_1 + x_{1,1} \\ I_4 + x_{2,1} \\ I_7 \end{pmatrix},$$

$$\boldsymbol{r}_2 = \begin{pmatrix} u_{1,1}x_{1,2} + u_{1,2}x_{2,2} \\ u_{2,1}x_{1,2} + u_{2,2}x_{2,2} + x_{1,2} \\ u_{3,1}x_{1,2} + u_{3,2}x_{2,2} + x_{2,2} \end{pmatrix} = \begin{pmatrix} I_2 \\ I_5 + x_{1,2} \\ I_8 + x_{2,2} \end{pmatrix},$$

$$\boldsymbol{r}_3 = \begin{pmatrix} u_{1,1}x_{1,3} + u_{1,2}x_{2,3} + x_{2,3} \\ u_{2,1}x_{1,3} + u_{2,2}x_{2,3} \\ u_{3,1}x_{1,3} + u_{3,2}x_{2,3} + x_{1,3} \end{pmatrix} = \begin{pmatrix} I_3 + x_{2,3} \\ I_6 \\ I_9 + x_{1,3} \end{pmatrix},$$

$$\boldsymbol{r}_4 = \begin{pmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \\ u_{3,1} & u_{3,2} \end{pmatrix} \begin{pmatrix} x_{1,1} + x_{1,2} \\ x_{2,1} + x_{2,2} \end{pmatrix} = \begin{pmatrix} I_1 + I_2 \\ I_4 + I_5 \\ I_7 + I_8 \end{pmatrix},$$

$$\boldsymbol{r}_5 = \begin{pmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \\ u_{3,1} & u_{3,2} \end{pmatrix} \begin{pmatrix} x_{1,2} + x_{1,3} \\ x_{2,2} + x_{2,3} \end{pmatrix} = \begin{pmatrix} I_2 + I_3 \\ I_5 + I_6 \\ I_8 + I_9 \end{pmatrix},$$

where $I_i = \sum_{j=1}^{2} u_{h,j} x_{j,h'}$ and $i = 3(h-1) + h'$, with $h, h' \in \mathbb{N}_3$, are collected by the user. Notice that each storage node sends back $d = k = 3$ symbols. The user obtains the requested file as follows. Knowing I_2, the user obtains I_1 and I_3 from the first components of $\boldsymbol{r}_4$ and $\boldsymbol{r}_5$. This allows the user to obtain $x_{1,1}$ and $x_{2,3}$. In a similar fashion, knowing I_6 the user gets I_5 from the second component of $\boldsymbol{r}_5$, then uses this to obtain I_4 from the second component of $\boldsymbol{r}_4$. This allows the user to obtain $x_{2,1}$ and $x_{1,2}$. Similarly, knowing I_7 allows the user to get I_8 from the third component of $\boldsymbol{r}_4$. Knowing I_8 allows the user to obtain I_9 from the third component of $\boldsymbol{r}_5$, which then allows to recover the symbols $x_{2,2}$ and $x_{1,3}$. In this way, the user recovers all symbols of the file and hence recovers $\boldsymbol{X}$. Note that $\mathsf{R}(\mathcal{C}) = \frac{2 \cdot 3}{5 \cdot 3} = \frac{2}{5}$, which is equal to the asymptotic MDS-PIR capacity C_∞ in (8).

Example 5. *Consider a DSS consisting of $n = 7$ storage nodes that store a single file $\boldsymbol{X}$. The DSS uses a $[7, 3, 4]$ scalar binary code $\mathcal{C}$. The parity-check matrix of the code is*

$$\boldsymbol{H}^{\mathcal{C}} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

We take $\beta = \Gamma = n-k = 4$. File $\boldsymbol{X}$ is of size $\beta \times k$ and hence consists of βk symbols in $\mathrm{GF}(2^\ell)$. Accordingly, the code array is

$$\boldsymbol{C} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,2}+x_{1,3} & x_{1,1}+x_{1,3} & x_{1,1}+x_{1,2} & x_{1,1}+x_{1,2}+x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,2}+x_{2,3} & x_{2,1}+x_{2,3} & x_{2,1}+x_{2,2} & x_{2,1}+x_{2,2}+x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,2}+x_{3,3} & x_{3,1}+x_{3,3} & x_{3,1}+x_{3,2} & x_{3,1}+x_{3,2}+x_{3,3} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,2}+x_{4,3} & x_{4,1}+x_{4,3} & x_{4,1}+x_{4,2} & x_{4,1}+x_{4,2}+x_{4,3} \end{pmatrix}.$$

The queries sent to each node, each consisting of $d = k = 3$ subqueries, take the form in (16)*. The aim of each subquery is to recover Γ code symbols using the PIR protocol. In order to do so, we construct the information sets $\{\mathcal{I}_i\}_{i\in\mathbb{N}_4}$. With careful consideration, we choose $\mathcal{I}_1 = \{3, 4, 6\}$, $\mathcal{I}_2 = \{2, 6, 7\}$, $\mathcal{I}_3 = \{1, 3, 4\}$, and $\mathcal{I}_4 = \{1, 5, 6\}$. The column weight profile of $\hat{\boldsymbol{E}}$ is $(2, 1, 2, 2, 1, 3, 1)$. A valid matrix $\hat{\boldsymbol{E}}$ is*

$$\hat{\boldsymbol{E}} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Note that each erasure pattern in $\hat{\boldsymbol{E}}$ (each row) is correctable by the code $\mathcal{C}$. As in Example 4, we map the columns of $\hat{\boldsymbol{E}}$ and $\{\mathcal{I}_i\}_{i\in\mathbb{N}_4}$ to the matrix $\boldsymbol{V}^{(l)}$ and obtain

$$\boldsymbol{\Delta}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \boldsymbol{\Delta}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$
$$\boldsymbol{\Delta}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \boldsymbol{\Delta}_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$
$$\boldsymbol{\Delta}_5 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \boldsymbol{\Delta}_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$
$$\boldsymbol{\Delta}_7 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

As an example, we next show the reconstruction of symbols from the first pair of subqueries and subresponses. We have

$$r_{1,1} = I_1,\ r_{2,1} = I_2,\ r_{3,1} = I_3 + x_{1,3},$$
$$r_{4,1} = I_2 + I_3 + x_{1,2} + x_{1,3},\ r_{5,1} = I_1 + I_3 + x_{4,1} + x_{4,3},$$
$$r_{6,1} = I_1 + I_2 + x_{1,1} + x_{1,2},\ r_{7,1} = I_1 + I_2 + I_3,$$

where $r_{l,1}$ denotes the first subresponse from the l-th node, $I_i = \sum_{j=1}^{4} u_{h,j} x_{j,h'}$ and $i = 3(h-1) + h'$, $h, h' \in \mathbb{N}_3$. Clearly, the subresponses $r_{1,1}$, $r_{2,1}$, and $r_{7,1}$ allow the user to obtain the three interference symbols I_1, I_2, and I_3. This is solely because the first row of $\hat{\boldsymbol{E}}$ (pertaining to the first subqueries) is an erasure pattern correctable by $\mathcal{C}$. Having this knowledge, the user obtains the symbols $x_{1,3}$, $x_{1,2}+x_{1,3}$, $x_{4,1}+x_{4,3}$, and $x_{1,1}+x_{1,2}$ from the remaining subresponses. From the obtained code symbols the user can decode $x_{1,3}$, $x_{1,2}$, and $x_{1,1}$, hence obtaining the message symbols in the first row of $\boldsymbol{C}$. The code symbol $x_{4,1}+x_{4,3}$ is used to decode $x_{4,1}$, $x_{4,2}$, and $x_{4,3}$ from the code symbols that are further obtained from the third subresponse. In the same way, the remaining two subresponses allow the recovery of $\beta k = 12$ message symbols.

The PIR rate is $\mathsf{R}(\mathcal{C}) = \frac{4\cdot 3}{7\cdot 3} = \frac{4}{7}$, which is equal to the asymptotic MDS-PIR capacity C_∞ in (8)*.*

VI. MDS-PIR Capacity-Achieving Codes

For given values of n and k, whether an $[n, k]$ code is MDS-PIR capacity-achieving or not is of great interest. In this section, we provide a necessary condition for an arbitrary linear code to achieve the MDS-PIR capacities C_f and C_∞ with Protocols 1 and 2, respectively. Furthermore, we prove that certain important families of codes, namely cyclic codes, RM codes, and a class of distance-optimal LRCs are MDS-PIR capacity-achieving. For Protocol 2, the MDS-PIR capacity-achieving proofs for these classes of codes assume $\beta = n - k$ and $d = k$, which are not necessary the minimum values given in (6). However, in the numerical results section (see Tables II and III) we show examples for which Protocol 2 also achieves the MDS-PIR capacity for β and d in (6).

As shown in the previous sections, the only requirement for a code $\mathcal{C}$ to achieve the MDS-PIR capacity with Protocol 1 is that there exists an MDS-PIR capacity-achieving matrix $\boldsymbol{\Lambda}_{\kappa,\nu}(\mathcal{C})$ (or a $(\Gamma = n-k)$-regular matrix $\boldsymbol{E}$ of size $(\beta+d)\times n$). In other words, the code $\mathcal{C}$ should be able to correct $\beta + d$ erasure patterns of $n - k$ erasures that satisfy the regularity condition of $\boldsymbol{E}$.

Let us first consider a fact for any information set of an $[n, k]$ code.

Proposition 2 ([38, Th. 1.6.2])**.** *If $\mathcal{I}$ is an information set of an $[n, k]$ code $\mathcal{C}$, then $\mathbb{N}_n \setminus \mathcal{I}$ is an information set of its $[n, n-k]$ dual code $\mathcal{C}^\perp$.*

Based on Proposition 2, the subsequent result follows.

Corollary 6. *The dual of an $[n, k]$ MDS-PIR capacity-achieving code is an $[n, n-k]$ MDS-PIR capacity-achieving code.*

To check if a linear code achieves the MDS-PIR capacity with Protocol 1, sometimes it might be easier to verify the MDS-PIR capacity-achieving condition for its dual code.

Next, we derive a useful result that gives the relation between an information set and a subcode of dimension s.

Lemma 5. *Given an $[n, k]$ code $\mathcal{C}$, for any information set $\mathcal{I}$ and an s-dimensional subcode $\mathcal{D} \subseteq \mathcal{C}$, we have*

$$\left|\mathcal{I} \cap \chi(\mathcal{D})\right| \geq s.$$

Proof: See Appendix E. ∎

Now, we are able to provide a necessary condition for a code to achieve the MDS-PIR capacity with Protocol 1.

Theorem 3. *If an MDS-PIR capacity-achieving matrix exists for an $[n, k]$ code $\mathcal{C}$, then*

$$d_s^{\mathcal{C}} \geq \frac{n}{k} s, \quad \forall\, s \in \mathbb{N}_k. \tag{23}$$

Proof: By definition there exists a PIR achievable rate matrix $\mathbf{\Lambda}_{\kappa,\nu}(\mathcal{C})$ with $\frac{\kappa}{\nu} = \frac{k}{n}$. This means that there exist information sets $\mathcal{I}_i$, $i \in \mathbb{N}_\nu$, such that in $\{\mathcal{I}_i\}_{i\in\mathbb{N}_\nu}$ each coordinate j of $\mathcal{C}$, $j \in \mathbb{N}_n$, appears exactly κ times. Let $\mathcal{D}$ be any subcode of dimension s of the $[n,k]$ code $\mathcal{C}$. This implies that

$$\kappa\,|\chi(\mathcal{D})| = \sum_{i=1}^{\nu} \big|\mathcal{I}_i \cap \chi(\mathcal{D})\big| \overset{(a)}{\geq} \nu s,$$

where (a) follows from Lemma 5. Based on the definition of $d_s^{\mathcal{C}}$, $s \in \mathbb{N}_k$, there exists a rank-s subcode $\mathcal{D}^*$ that achieves $d_s^{\mathcal{C}}$. We then have

$$d_s^{\mathcal{C}} \geq \frac{\nu}{\kappa} s = \frac{n}{k} s, \quad \forall\, s \in \mathbb{N}_k.$$

■

Based on the necessary condition, it can be shown that the code $\mathcal{C}$ in Example 2 is not MDS-PIR capacity-achieving with Protocol 1, since $d_2^{\mathcal{C}} = 3 < \frac{5}{3} \cdot 2$, i.e., it is impossible to find an MDS-PIR capacity-achieving matrix $\mathbf{\Lambda}_{\kappa,\nu}$ for this code.

We would like to emphasize that it seems that the necessary condition for MDS-PIR capacity-achieving matrices in Theorem 3 is also a sufficient condition. We have performed an exhaustive search for codes with parameters $k \in \mathbb{N}_n$ and $n \in \mathbb{N}_{11}$ (except for $[n,k] = [10,5]$ and $[n,k] = [11, 4 \leq k \leq 7]$) and seen that for codes that satisfy the necessary condition, there always exists an MDS-PIR capacity-achieving matrix. Therefore, we conjecture that (23) in Theorem 3 is an if and only if condition for the existence of an MDS-PIR capacity-achieving matrix.

Conjecture 1. *An MDS-PIR capacity-achieving matrix exists for an $[n,k]$ code $\mathcal{C}$ if and only if*

$$d_s^{\mathcal{C}} \geq \frac{n}{k} s, \quad \forall\, s \in \mathbb{N}_k.$$

In the following, we provide a sufficient condition for an $[n,k]$ code $\mathcal{C}$ to achieve the MDS-PIR capacity with Protocol 1 by using code automorphisms [31, Ch. 8].

Theorem 4. *Given an $[n,k]$ code $\mathcal{C}$, if there exist n distinct automorphisms $\pi_1, \ldots, \pi_n$ of $\mathcal{C}$ such that for every code coordinate $j \in \mathbb{N}_n$, $\{\pi_1(j), \ldots, \pi_n(j)\} = \mathbb{N}_n$, then the code $\mathcal{C}$ is an MDS-PIR capacity-achieving code.*

Proof: Since any $[n,k]$ code $\mathcal{C}$ contains at least one information set $\mathcal{I}$, the automorphisms $\{\pi_i\}_{i\in\mathbb{N}_n}$ guarantee that

$$\mathcal{I}_i \triangleq \{\pi_i(j) \colon j \in \mathcal{I}\}, \quad i \in \mathbb{N}_n,$$

are all information sets of $\mathcal{C}$. By assumption, for a given $j \in \mathcal{I}$, we have $\{\pi_1(j), \ldots, \pi_n(j)\} = \mathbb{N}_n$. Since there are in total k coordinates in $\mathcal{I}$, every coordinate appears exactly k times in $\{\mathcal{I}_i\}_{i\in\mathbb{N}_n}$, and hence an MDS-PIR capacity-achieving matrix $\mathbf{\Lambda}_{k,n}(\mathcal{C})$ satisfying Definition 13 exists. ■

Using their known code automorphisms and Theorem 4, it is easy to prove that the families of cyclic codes and RM codes achieve the MDS-PIR capacity.

A. Cyclic Codes

Corollary 7. *Cyclic codes are MDS-PIR capacity-achieving codes.*

Proof: Let π_i denote the automorphism of an $[n,k]$ cyclic code $\mathcal{C}$ that cyclically shifts each coordinate to the right by i positions. Clearly, for every code coordinate $j \in \mathbb{N}_n$, $\{\pi_1(j), \ldots, \pi_n(j)\} = \mathbb{N}_n$, and the result follows from Theorem 4. ■

B. Reed-Muller Codes

Corollary 8. *RM codes are MDS-PIR capacity-achieving codes.*

Proof: Consider an arbitrary RM code $\mathcal{R}(v,m)$ with $v \in \{0\} \cup \mathbb{N}_m$ for some $m \in \mathbb{N}$. Consider the n distinct automorphisms $g_i(\boldsymbol{\mu}) \triangleq \boldsymbol{\mu} + \boldsymbol{\sigma}_i$, where $\boldsymbol{\sigma}_i$ is the i-th m-tuple in $\mathrm{GF}(2)^{m\times 1}$, $i \in \mathbb{N}_n$, $n = 2^m$ (see Section II-A). For any $\boldsymbol{\mu} \in \mathrm{GF}(2)^{m\times 1}$,

$$\{g_1(\boldsymbol{\mu}), \ldots, g_n(\boldsymbol{\mu})\} = \{\boldsymbol{\mu} + \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\mu} + \boldsymbol{\sigma}_n\}$$

forms the vector space $\mathrm{GF}(2)^{m\times 1}$, and the result follows from Theorem 4. ■

We remark here that because of the property of invertible and affine automorphisms for binary RM codes, it is not too hard to see that Corollary 8 can be extended to nonbinary generalized RM codes [40]. The detailed discussion is omitted. Furthermore, note that in the independent work [32] it was also shown that RM codes can achieve the asymptotic MDS-PIR capacity, albeit with a protocol that requires a much larger β and d.

Besides cyclic codes and RM codes, there exist other families of codes satisfying Theorem 4, for instance, the class of low-density parity-check (LDPC) codes constructed from array codes [41], [42]. We further emphasize that the proof of Theorem 4 indicates that the automorphisms of an $[n,k]$ code are very important to design an MDS-PIR capacity-achieving matrix.

C. Local Reconstruction Codes

In this subsection, we prove that a certain family of LRCs achieves the MDS-PIR capacity by directly showing the existence of an $(n-k)$-regular $n \times n$ matrix $\boldsymbol{E}$.

Consider an $[n,k]$ distance-optimal (r,δ) information locality code (see Definition 7) for which the $(n'-k) \times n'$ matrix

$$\left(\begin{array}{cccc|c} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \cdots & \boldsymbol{P}_{L_{\mathsf{c}}} & \\ \boldsymbol{M}_1 & \boldsymbol{M}_2 & \cdots & \boldsymbol{M}_{L_{\mathsf{c}}} & \boldsymbol{I}_{n'-k} \end{array}\right) \triangleq \boldsymbol{H}^{\mathsf{MDS}} \tag{24}$$

is the parity-check matrix of an $[n',k]$ MDS code over $\mathrm{GF}(q)$, where $n' = n - (L_{\mathsf{c}} - 1)(\delta - 1)$.[5] For such a class of codes, we give an explicit construction of the matrix $\boldsymbol{E}$ in order to design the PIR protocol.

[5]Examples of codes that satisfy (24) are Pyramid codes, the LRCs in [25], and codes from the parity-splitting construction of [26].

Recall that $L = \left\lfloor \frac{n}{n_{\mathsf{c}}} \right\rfloor$, $n_{\mathsf{c}} = r+\delta-1$, and let $\bar{r} \triangleq n \bmod n_{\mathsf{c}}$. We consider

$$\boldsymbol{E} = \begin{pmatrix} \boldsymbol{E}_{1,1} & \boldsymbol{E}_{1,2} & \dots & \boldsymbol{E}_{1,L+1} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{E}_{L+1,1} & \boldsymbol{E}_{L+1,2} & \dots & \boldsymbol{E}_{L+1,L+1} \end{pmatrix}$$

having $(L+1)^2$ submatrices $\boldsymbol{E}_{l,h}$, $l,h \in \mathbb{N}_{L+1}$. For any $l,h \in \mathbb{N}_L$, the submatrices $\boldsymbol{E}_{l,h}$ have dimensions $n_{\mathsf{c}} \times n_{\mathsf{c}}$, $\boldsymbol{E}_{l,L+1}$ has dimensions $n_{\mathsf{c}} \times \bar{r}$, $\boldsymbol{E}_{L+1,h}$ has dimensions $\bar{r} \times n_{\mathsf{c}}$, and $\boldsymbol{E}_{L+1,L+1}$ has dimensions $\bar{r} \times \bar{r}$. We denote by $\boldsymbol{e}_i^{(l)}$, $l \in \mathbb{N}_{L+1}$, the i-th row of $\big(\boldsymbol{E}_{l,1}|\dots|\boldsymbol{E}_{l,L+1}\big)$. The coordinates of $\boldsymbol{e}_i^{(l)}$ represent the coordinates of the code $\mathcal{C}$ defined by its parity-check matrix in (2). Furthermore, each row vector is subdivided into $L+1$ subvectors $\boldsymbol{e}_{i,j}^{(l)}$, $j \in \mathbb{N}_{L+1}$, as

$$\boldsymbol{e}_i^{(l)} = (e_{i,1}^{(l)}, \dots, e_{i,n}^{(l)}) = (\boldsymbol{e}_{i,1}^{(l)}, \dots, \boldsymbol{e}_{i,L}^{(l)}, \boldsymbol{e}_{i,L+1}^{(l)}).$$

The subvectors $\boldsymbol{e}_{i,1}^{(l)}, \dots, \boldsymbol{e}_{i,L}^{(l)}$ are of length n_{c}, while $\boldsymbol{e}_{i,L+1}^{(l)}$ is of length $\bar{r}$. Correspondingly, we can think about $\boldsymbol{E}$ as partitioned into $L+1$ column partitions, where the first L_{c} partitions correspond to the L_{c} local codes and the remaining $L+1-L_{\mathsf{c}}$ partitions correspond to global parities (see also (3)). We can write $\boldsymbol{E}$ as

$$\boldsymbol{E} \triangleq \begin{pmatrix} \boldsymbol{e}_1^{(1)} \\ \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}}}^{(1)} \\ \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}}}^{(L)} \\ \boldsymbol{e}_1^{(L+1)} \\ \vdots \\ \boldsymbol{e}_{\bar{r}}^{(L+1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{e}_{1,1}^{(1)} & \boldsymbol{e}_{1,2}^{(1)} & \cdots & \boldsymbol{e}_{1,L}^{(1)} & \boldsymbol{e}_{1,L+1}^{(1)} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}},1}^{(1)} & \boldsymbol{e}_{n_{\mathsf{c}},2}^{(1)} & \cdots & \boldsymbol{e}_{n_{\mathsf{c}},L}^{(1)} & \boldsymbol{e}_{n_{\mathsf{c}},L+1}^{(1)} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}},1}^{(L)} & \boldsymbol{e}_{n_{\mathsf{c}},2}^{(L)} & \cdots & \boldsymbol{e}_{n_{\mathsf{c}},L}^{(L)} & \boldsymbol{e}_{n_{\mathsf{c}},L+1}^{(L)} \\ \boldsymbol{e}_{1,1}^{(L+1)} & \boldsymbol{e}_{1,2}^{(L+1)} & \cdots & \boldsymbol{e}_{1,L}^{(L+1)} & \boldsymbol{e}_{1,L+1}^{(L+1)} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \boldsymbol{e}_{\bar{r},1}^{(L+1)} & \boldsymbol{e}_{\bar{r},2}^{(L+1)} & \cdots & \boldsymbol{e}_{\bar{r},L}^{(L+1)} & \boldsymbol{e}_{\bar{r},L+1}^{(L+1)} \end{pmatrix}.$$

We refer to the set of rows $\boldsymbol{e}_1^{(l)}, \dots, \boldsymbol{e}_{n_{\mathsf{c}}}^{(l)}$ as the l-th row partition of $\boldsymbol{E}$.

For convenience, we divide $\boldsymbol{E}$ into four submatrices $\tilde{\boldsymbol{E}}$, $\boldsymbol{W}$, $\boldsymbol{Z}$, and $\boldsymbol{O}$ defined as

$$\tilde{\boldsymbol{E}} \triangleq \begin{pmatrix} \boldsymbol{e}_{1,1}^{(1)} & \boldsymbol{e}_{1,2}^{(1)} & \cdots & \boldsymbol{e}_{1,L}^{(1)} \\ \boldsymbol{e}_{2,1}^{(1)} & \boldsymbol{e}_{2,2}^{(1)} & \cdots & \boldsymbol{e}_{2,L}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}},1}^{(L)} & \boldsymbol{e}_{n_{\mathsf{c}},2}^{(L)} & \cdots & \boldsymbol{e}_{n_{\mathsf{c}},L}^{(L)} \end{pmatrix}, \boldsymbol{Z} \triangleq \begin{pmatrix} \boldsymbol{e}_{1,L+1}^{(1)} \\ \boldsymbol{e}_{2,L+1}^{(1)} \\ \vdots \\ \boldsymbol{e}_{n_{\mathsf{c}},L+1}^{(L)} \end{pmatrix},$$

$$\boldsymbol{W} \triangleq \begin{pmatrix} \boldsymbol{e}_{1,1}^{(L+1)} & \boldsymbol{e}_{1,2}^{(L+1)} & \cdots & \boldsymbol{e}_{1,L}^{(L+1)} \\ \vdots & \vdots & \dots & \vdots \\ \boldsymbol{e}_{\bar{r},1}^{(L+1)} & \boldsymbol{e}_{\bar{r},2}^{(L+1)} & \cdots & \boldsymbol{e}_{\bar{r},L}^{(L+1)} \end{pmatrix}, \boldsymbol{O} \triangleq \begin{pmatrix} \boldsymbol{e}_{1,L+1}^{(L+1)} \\ \vdots \\ \boldsymbol{e}_{\bar{r},L+1}^{(L+1)} \end{pmatrix}, \quad (25)$$

where $\tilde{\boldsymbol{E}}$ is an $n_{\mathsf{c}}L \times n_{\mathsf{c}}L$ matrix, having L^2 submatrices $\boldsymbol{E}_{l,h}$, $l,h \in \mathbb{N}_L$.

In the following, we give a systematic construction of $\boldsymbol{E}$ such that it is $(n-k)$-regular. The construction involves two steps.

a) **Initialize matrices $\tilde{\boldsymbol{E}}$, $\boldsymbol{W}$, $\boldsymbol{Z}$, and $\boldsymbol{O}$.** Matrix $\boldsymbol{Z}$ is initialized to the all-zero matrix of dimensions $n_{\mathsf{c}}L \times \bar{r}$. Matrices $\boldsymbol{W}$ and $\boldsymbol{O}$ are initialized by setting $e_{i,j}^{(L+1)} = 1$, $i \in \mathbb{N}_{\bar{r}}$, $j \in \mathcal{P} = \bigcup_{j'=1}^{L+1} \mathcal{P}_{j'}$, where $\mathcal{P}$ corresponds to the parity coordinates of $\mathcal{C}$ and the sets $\mathcal{P}_{j'}$ are defined in Section II-B (see (3)). Let $m = \left\lfloor \frac{n-k}{L} \right\rfloor$, $m_1 = m+1$, $\rho_1 = \dots = \rho_t = m_1$, and $\rho_{t+1} = \dots = \rho_L = m$, where $t = (n-k) \bmod L$. Matrix $\tilde{\boldsymbol{E}}$ is initialized with the structure

$$\tilde{\boldsymbol{E}} = \begin{pmatrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \cdots & \boldsymbol{\pi}_L \\ \boldsymbol{\pi}_L & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{L-1} \\ \vdots & \vdots & \dots & \vdots \\ \boldsymbol{\pi}_2 & \boldsymbol{\pi}_3 & \cdots & \boldsymbol{\pi}_1 \end{pmatrix}, \quad (26)$$

where each matrix entry $\boldsymbol{\pi}_l$, $l \in \mathbb{N}_L$, is a ρ_l-regular square matrix of dimensions $n_{\mathsf{c}} \times n_{\mathsf{c}}$. Notice that due to the structure in (26), $\tilde{\boldsymbol{E}}$ has row and column weight equal to $n-k$, and subsequently each row of $\boldsymbol{E}$ has weight $n-k$. Note also that the columns of $\boldsymbol{E}$ with coordinates in $\mathcal{P}_j$, $j \in \mathbb{N}_L$, have column weight $n-k+\bar{r}$, while the columns with coordinates in $\mathcal{P}_{L+1}$ have weight $\bar{r}$.

b) **Swapping elements between $\tilde{\boldsymbol{E}}$ and $\boldsymbol{Z}$.** The swapping of elements is performed iteratively with $\bar{r}$ iterations. For each iteration, in the i-th row partition and j-th column partition, we consider a set of row coordinates $\mathcal{R}_j^{(i)}$ of size $|\mathcal{P}_j|$ from which $s_j^{(i)} \in \{0,1\}$ ones from columns with coordinates in $\mathcal{P}_j$, $j \in \mathbb{N}_L$, are swapped with zeroes in the corresponding rows of $\boldsymbol{Z}$. For convenience, we define $\boldsymbol{s}^{(i)} = (s_1^{(i)}, \dots, s_L^{(i)})$ and require that $\sum_{j=1}^{L} s_j^{(i)} = 1$. Note that $\mathcal{R}_j^{(i)}$ and $\boldsymbol{s}^{(i)}$ depend on the iteration number. We describe the procedure for iteration $j' \in \mathbb{N}_{\bar{r}}$. For the first row partition, select $\boldsymbol{s}^{(1)}$ with $s_j^{(1)} = 1$ and $s_z^{(1)} = 0$, $\forall\, z \in \mathbb{N}_L \backslash \{j\}$, for some $j \in \mathbb{N}_L$, such that if $j \in \mathbb{N}_{L_{\mathsf{c}}}$ there exist $\delta-1$ rows in the first row partition and j-th column partition such that their individual weight is strictly larger than $\delta-1$, and otherwise if $j \in \mathbb{N}_{L_{\mathsf{c}}+1:L}$, all rows in the first row partition and j-th column partition must have weight larger than or equal to $\max(1, m-(\delta-1))$. This will ensure that the resulting erasure patterns after the swap (as described next) are correctable by $\mathcal{C}$ (see Appendix F-B). Such an $\boldsymbol{s}^{(1)}$ will also always exist for all $\bar{r}$ iterations as shown in Appendix F-B. Next, for all $i' \in \mathcal{R}_j^{(1)}$ and $p \in \mathcal{P}_j$ (where different p's are chosen for different i''s, and index j is such that $s_j^{(1)} = 1$) the one at coordinate (i',p) of $\tilde{\boldsymbol{E}}$ is swapped with a zero at coordinate (i',j') of $\boldsymbol{Z}$ (this corresponds to coordinate $(i', n_{\mathsf{c}}L+j')$ of $\boldsymbol{E}$). Then, for the remaining row partitions $i = 2, \dots, L$, consider $\boldsymbol{s}^{(i)}$ to be the $(i-1)$-th right cyclic shift of $\boldsymbol{s}^{(1)}$ and repeat the swapping procedure for the first row partition. Due to the specific selection of $\boldsymbol{s}^{(1)}$, the corresponding erasure patterns for all row partitions after the swaps are correctable by $\mathcal{C}$ (see Appendix F-B). Note that we have performed $\sum_{j=1}^{L} |\mathcal{P}_j| = n-k-\bar{r}$ swaps from the columns of $\tilde{\boldsymbol{E}}$ with coordinates in the set $\cup_{j=1}^{L} \mathcal{P}_j$ to the j'-th column of $\boldsymbol{Z}$. Thus, each column in $\cup_{j=1}^{L} \mathcal{P}_j$ has column weight $n-k+\bar{r}-1$ and the $(n_{\mathsf{c}}L+j')$-th column has column weight $n-k-\bar{r}+\bar{r} = n-k$. Letting $j' = j'+1$ and repeating the above procedure $\bar{r}$ times ensures $\boldsymbol{E}$ to be $(n-k)$-regular.

This completes the construction of $\boldsymbol{E}$, which has row and column weight $n-k$. In the following theorem, we show that each row of $\boldsymbol{E}$ (considered as an erasure pattern) can be corrected by any code from the class of distance-optimal (r,δ) information locality codes whose parity-check matrices are as in (2) and are compliant with (24). Thus, this class of codes is MDS-PIR capacity-achieving.

Theorem 5. *An $[n,k]$ distance-optimal (r,δ) information locality code $\mathcal{C}$ with parity-check matrix as in* (2) *and satisfying* (24) *is an MDS-PIR capacity-achieving code.*

Proof: See Appendix F. ∎

In the following, we present an example to illustrate the construction of the matrix $\boldsymbol{E}$. The existence of such a matrix ensures that the PIR protocols presented in Sections IV and V achieve the finite MDS-PIR capacity C_f in (7) and the asymptotic MDS-PIR capacity C_∞ in (8), respectively.

Example 6. *Consider an $[n=7,k=4]$ Pyramid code $\mathcal{C}$ that is constructed from an $[n'=6,4]$ Reed-Solomon (RS) code over $\mathrm{GF}(2^3)$ with parity-check matrices*

$$\boldsymbol{H}^{\mathcal{C}}=\begin{pmatrix} z^3 & 1 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & z^3 & z & 1 & 0\\ z^4 & 1 & 0 & z^5 & z^5 & 0 & 1\end{pmatrix}$$

and

$$\boldsymbol{H}^{\mathsf{MDS}}=\begin{pmatrix} z^3 & 1 & z^3 & z & 1 & 0\\ z^4 & 1 & z^5 & z^5 & 0 & 1\end{pmatrix},$$

respectively, where z denotes a primitive element of $\mathrm{GF}(2^3)$. It is easy to see that $\mathcal{C}$ is a distance-optimal $(r=2,\delta=2)$ information locality code. We have $n_{\mathsf{c}}=3$, $L=L_{\mathsf{c}}=2$, and $\bar{r}\triangleq n \bmod n_{\mathsf{c}}=1$. Since $\rho_1=2$ and $\rho_2=1$, we get

$$\tilde{\boldsymbol{E}}=\begin{pmatrix}\boldsymbol{\pi}_1 & \boldsymbol{\pi}_2\\ \boldsymbol{\pi}_2 & \boldsymbol{\pi}_1\end{pmatrix}=\left(\begin{array}{ccc|ccc} 1&1&0&1&0&0\\ 0&1&1&0&1&0\\ 1&0&1&0&0&1\\ \hline 1&0&0&1&1&0\\ 0&1&0&0&1&1\\ 0&0&1&1&0&1\end{array}\right),\ \boldsymbol{Z}=\begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix},$$

where $\boldsymbol{\pi}_1$ is a 2-regular 3×3 matrix and $\boldsymbol{\pi}_2$ is picked as the identity matrix. The set of parity coordinates is $\mathcal{P}=\{3,6,7\}$, and we set $e^{(3)}_{1,3}=e^{(3)}_{1,6}=e^{(3)}_{1,7}=1$. As such, we get

$$\boldsymbol{W}=\begin{pmatrix}0&0&1&0&0&1\end{pmatrix} \text{ and } \boldsymbol{O}=\begin{pmatrix}1\end{pmatrix}.$$

This completes Step a) of the construction above. Note that each row of $\boldsymbol{E}$ has now weight 3. The second step of the procedure (Step b)) is as follows. Consider the first iteration, $j'=1$. In the first row partition we choose $\boldsymbol{s}^{(1)}=(s^{(1)}_1=1, s^{(1)}_2=0)$. Taking $\mathcal{R}^{(1)}=\{2\}$, we do the swap between the coordinates $(i'=2, p=3\in\mathcal{P}_1)$ and $(i', 6+j')$. For the second row partition we have $\boldsymbol{s}^{(2)}=(0,1)$ which is a right cyclic shift of $\boldsymbol{s}^{(1)}$. Taking $\mathcal{R}^{(2)}=\{6\}$, we do the swap between the coordinates $(i'=6, p=6\in\mathcal{P}_2)$ and $(i', 6+j')$. Thus, we have

$$e^{(1)}_{2,3}=0,\ e^{(1)}_{2,7}=1,$$
$$e^{(2)}_{3,6}=0,\ e^{(2)}_{3,7}=1.$$

Algorithm 1: Optimizing the PIR rate

Input : Distributed storage code $\mathcal{C}$ of length n
Output: Optimized matrix $\boldsymbol{E}_{\mathsf{opt}}$ and largest possible Γ

$\Gamma \leftarrow \min\left(k, d^{\mathcal{C}}_{\mathsf{min}}-1\right)$
$\boldsymbol{E}_{\mathsf{opt}}\leftarrow\emptyset$, $\Gamma_{\mathsf{opt}}\leftarrow\Gamma$
$\mathcal{L}_{n-k}\leftarrow$ `ComputeErasurePatternList`$(\mathcal{C}, n-k)$
while $\Gamma\le n-k$ **do**
  $\mathcal{L}_\Gamma\leftarrow$ `ComputeErasurePatternList`$(\mathcal{C},\Gamma)$
  if $\mathcal{L}_\Gamma\neq\emptyset$ **then**
    $\boldsymbol{E}\leftarrow$ `ComputeMatrix`$(\mathcal{L}_\Gamma,\mathcal{L}_{n-k})$
    if $\boldsymbol{E}\neq\emptyset$ **then**
      $\boldsymbol{E}_{\mathsf{opt}}\leftarrow\boldsymbol{E}$, $\Gamma_{\mathsf{opt}}\leftarrow\Gamma$
    else
      return $(\boldsymbol{E}_{\mathsf{opt}},\Gamma_{\mathsf{opt}})$
    end
  end
  $\Gamma\leftarrow\Gamma+1$
end
return $(\boldsymbol{E}_{\mathsf{opt}},\Gamma_{\mathsf{opt}})$

Since $\bar{r}=1$, this completes Step b), which results in

$$\boldsymbol{E}=\left(\begin{array}{ccc|ccc|c} 1&1&0&1&0&0&0\\ 0&1&0&0&1&0&1\\ 1&0&1&0&0&1&0\\ \hline 1&0&0&1&1&0&0\\ 0&1&0&0&1&1&0\\ 0&0&1&1&0&0&1\\ \hline 0&0&1&0&0&1&1\end{array}\right).$$

The entries in red indicate the swapped values within each row. It can easily be verified that each row of $\boldsymbol{E}$ is an erasure pattern that is correctable by code $\mathcal{C}$.

VII. Optimizing the PIR Rate for the Noncolluding Case

For codes for which we are not able to prove that they achieve the MDS-PIR capacity, in this section we provide an algorithm to optimize Protocols 1 and 2 in order to achieve the highest possible PIR rate $\mathsf{R}(\mathcal{C})$ for a given code $\mathcal{C}$ by taking the structure of the underlying code into consideration. The algorithm is given in Algorithm 1 and is based on Theorem 2. In particular, we need to find a $(d+\beta)\times n$ matrix $\boldsymbol{E}$ in (19) for which $\hat{\boldsymbol{E}}$ consists of erasure patterns of weight Γ that are all correctable by $\mathcal{C}$, and for which $\bar{\boldsymbol{E}}$ corresponds to information sets of $\mathcal{C}$ (the support of each row is the complement of an information set). In addition, it is required that each column weight of $\hat{\boldsymbol{E}}$ is equal to the corresponding column weight of $\mathbf{1}-\bar{\boldsymbol{E}}$ (see Section V). Note that from the resulting matrix $\boldsymbol{E}$ we can find a PIR achievable rate matrix by taking its binary complement as in (20) (see Section V), thus optimizing Protocols 1 and 2 in Sections IV and V, respectively.

The main issues that need to be addressed are the efficient enumeration of the set of erasure patterns of a given weight Γ (corresponding to the rows of $\hat{\boldsymbol{E}}$) and also of weight $n-k$ ($\bar{\boldsymbol{E}}$, corresponding to information

sets) that can be corrected by $\mathcal{C}$, and the efficient computation of the matrix $\boldsymbol{E}$. These issues are addressed by the subprocedures `ComputeErasurePatternList`$(\mathcal{C},\cdot)$ and `ComputeMatrix`$(\mathcal{L}_\Gamma,\mathcal{L}_{n-k})$, in Lines 3, 5, and 7 of Algorithm 1, and discussed below in Sections VII-A and VII-B, respectively. Here, $\mathcal{L}_\Gamma$ and $\mathcal{L}_{n-k}$ correspond to erasure patterns for $\hat{\boldsymbol{E}}$ and $\bar{\boldsymbol{E}}$, respectively. We remark that the algorithm will always return a valid $\boldsymbol{E} \neq \emptyset$, since initially $\Gamma = \min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1)$. This follows directly from the fact that we can construct an arbitrary $\hat{\boldsymbol{E}}$ with row weights Γ such that its column weights match the corresponding weights in $\mathbf{1} - \bar{\boldsymbol{E}}$. Each row in $\hat{\boldsymbol{E}}$ is an erasure pattern that is correctable by $\mathcal{C}$.

Let $d = k$ and $\beta = \Gamma$. In the particular case of $\mathcal{C}$ being a rate $R^{\mathcal{C}} > 1/2$ systematic MDS code, $d^{\mathcal{C}}_{\mathsf{min}} = n - k + 1$, and the algorithm will do exactly one iteration of the main loop. This follows directly from the construction of $\boldsymbol{E}$: the matrix $\boldsymbol{E}$ can be constructed by taking the support of an arbitrary information set of $\mathcal{C}$ and cyclically shifting it n times to construct an $n \times n$ PIR achievable rate matrix, after which the resulting matrix is complemented as (20) to get $\boldsymbol{E}$. In this case, the overall PIR scheme reduces to the scheme described in [12, Sec. IV] for systematic MDS codes of rate $R^{\mathcal{C}} > 1/2$. Clearly, for general MDS codes (including nonsystematic codes) of rate $R^{\mathcal{C}} > 1/2$, the same construction of $\boldsymbol{E}$ works, and the algorithm will perform exactly one iteration of the main loop also for nonsystematic MDS codes. In the case of $\mathcal{C}$ being a rate $R^{\mathcal{C}} \leq 1/2$ general MDS code, the initial value of Γ becomes k (since $\Gamma = \min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1) = \min(k, n-k) = k$), but the algorithm will also find a valid matrix $\boldsymbol{E}$ for $\Gamma = n - k \geq k$. Again, the existence of $\boldsymbol{E}$ follows from the same argument of cyclically shifting an existing information set n times. In the general case of $d \neq k$ and $\beta \neq \Gamma$, a similar argument to the one above can be made.

A. `ComputeErasurePatternList`$(\mathcal{C},\cdot)$

Computing a list of erasure patterns that are correctable for a given short code can be done using any maximum likelihood (ML) decoding algorithm. For small codes, all length-n binary vectors (or erasure patterns) of weight Γ (or $n-k$) that are correctable can be found by exhaustive search, while for longer codes a random search can be performed, in the sense of picking length-n binary vectors (or erasure patterns) of weight Γ (or $n-k$) at random, and then verifying whether they are correctable or not. Alternatively, one can apply a random permutation π to the columns of $\boldsymbol{H}^{\mathcal{C}}$, apply the Gauss-Jordan algorithm to the resulting matrix to transform it into *row echelon form*, collect a subset of size Γ of the column indices of *leading-one-columns*,[6] and finally apply the inverse permutation π^{-1} to this subset of column indices. The resulting set corresponds to erased coordinates in $\mathcal{C}$ that can be recovered by the code. Finally, one can check whether all cyclic shifts of the added erasure pattern are correctable or not and add the correctable cyclic shifts to $\mathcal{L}_\Gamma$ (or $\mathcal{L}_{n-k}$).

[6]The leading-one-columns are the columns containing a *leading one*, where the first nonzero entry in each matrix row of a matrix in row echelon form is called a leading one.

B. `ComputeMatrix`$(\mathcal{L}_\Gamma,\mathcal{L}_{n-k})$

Given the lists $\mathcal{L}_\Gamma$ and $\mathcal{L}_{n-k}$ of erasure patterns of weight Γ and $n-k$, respectively, that are correctable for $\mathcal{C}$, we construct a $\big(|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|\big)\times n$ matrix, denoted by $\boldsymbol{\Psi} = (\psi_{i,j})$, in which each row $i \in \mathbb{N}_{|\mathcal{L}_\Gamma|}$ is one of the erasure patterns from $\mathcal{L}_\Gamma$ and each row $i \in \mathbb{N}_{|\mathcal{L}_\Gamma|+1:|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|}$ is one of the erasure patterns from $\mathcal{L}_{n-k}$. The problem is now to find a $d \times n$ submatrix $\hat{\boldsymbol{\Psi}}$ of the upper part of $\boldsymbol{\Psi}$ (rows 1 to $|\mathcal{L}_\Gamma|$) and a $\beta \times n$ submatrix $\bar{\boldsymbol{\Psi}}$ of the lower part of $\boldsymbol{\Psi}$ (rows $|\mathcal{L}_\Gamma|+1$ to $|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|$) such that the column weight of each of the n columns is the same for $\hat{\boldsymbol{\Psi}}$ and the binary complement of $\bar{\boldsymbol{\Psi}}$, where β and d are chosen such that $\beta k = \Gamma d$.

This can be formulated as an integer program (in the integer variables $\eta_1, \ldots, \eta_{|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|}$) in the following way,

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|} \eta_i \\
\text{s.t.} \quad & \sum_{i=1}^{|\mathcal{L}_\Gamma|} \eta_i \psi_{i,j} = \sum_{i=|\mathcal{L}_\Gamma|+1}^{|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|} \eta_i (1-\psi_{i,j}),\ \forall\, j \in \mathbb{N}_n, \\
& \eta_i \in \{0,1\},\ \forall\, i \in \mathbb{N}_{|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|}, \\
& \sum_{i=1}^{|\mathcal{L}_\Gamma|} \eta_i = d, \text{ and } \sum_{i=|\mathcal{L}_\Gamma|+1}^{|\mathcal{L}_\Gamma|+|\mathcal{L}_{n-k}|} \eta_i = \beta.
\end{aligned} \tag{27}$$

A valid $(d+\beta)\times n$ matrix $\boldsymbol{E}$ is constructed from the rows of $\boldsymbol{\Psi}$ with η_i-values equal to one in any feasible solution of (27). When $|\mathcal{L}_\Gamma| + |\mathcal{L}_{n-k}|$ is large, solving (27) may become impractical (solving a general integer program is known to be NP-hard), in which case one can take several random subsets (of some size) of the lists $\mathcal{L}_\Gamma$ and $\mathcal{L}_{n-k}$, construct the corresponding matrices $\boldsymbol{\Psi}$, and try to solve the program in (27).

VIII. Multiple Colluding Nodes

In this section, we consider the scenario where $T > 1$ nodes act as spies and have the ability to collude. In particular, we propose a protocol for this scenario that improves upon the PIR protocol in [29]. We refer to the protocol in [29] as the $(\mathcal{C},\bar{\mathcal{C}})$-*retrieval protocol (or scheme)*, since it is based on two linear codes: an $[n,k]$ code $\mathcal{C}$ and an $[n,\bar{k}]$ code $\bar{\mathcal{C}}$, where $\mathcal{C}$ is the underlying storage code of the DSS and $\bar{\mathcal{C}}$ defines the queries. Furthermore, the retrieval process is defined by an $[n,\tilde{k}]$ code $\tilde{\mathcal{C}}$ that is the Hadamard product of $\mathcal{C}$ and $\bar{\mathcal{C}}$, $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$. The protocol yields privacy against at most $T = d^{\bar{\mathcal{C}}^\perp}_{\mathsf{min}} - 1$ colluding nodes under the assumption that the code $\bar{\mathcal{C}}$ with $d^{\bar{\mathcal{C}}^\perp}_{\mathsf{min}} = T+1$ exists for the given T (existing in the sense that the Hadamard product of $\mathcal{C}$ and $\bar{\mathcal{C}}$ has rate strictly smaller than 1).

Originally, the protocol was designed to work with GRS codes, a class of MDS codes, i.e., both codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ are GRS codes. In this case $\bar{\mathcal{C}}$ has parameters $[n, \bar{k} = T]$, the retrieval code $\tilde{\mathcal{C}}$ has parameters $[n, \tilde{k} = k + T - 1]$, and the PIR rate is

$$\mathsf{R}_{\mathsf{GRS}} = \frac{n-(k+T-1)}{n}.$$

For non-MDS codes, the protocol achieves a PIR rate

$$\mathsf{R}(\mathcal{C},\bar{\mathcal{C}}) = \frac{d_{\mathsf{min}}^{\bar{\mathcal{C}}} - 1}{n},$$

which is lower than $\mathsf{R}_{\mathsf{GRS}}$. In general, when the underlying codes are arbitrary codes, it can be shown that the PIR rate of the $(\mathcal{C},\bar{\mathcal{C}})$-retrieval protocol is upperbounded by

$$\mathsf{R}_{\mathsf{UB}} \triangleq \frac{n-\tilde{k}}{n}. \tag{28}$$

In particular, the $(\mathcal{C},\bar{\mathcal{C}})$-retrieval protocol in [29] achieves a PIR rate $\mathsf{R}(\mathcal{C},\bar{\mathcal{C}}) < \mathsf{R}_{\mathsf{UB}}$ for non-MDS codes. Furthermore, it was shown in [30] that if $\mathcal{C}$ is either a GRS code or an RM code, then $\bar{\mathcal{C}}$ always exists for any $T \leq n-k$. In this section, we look at this protocol from the perspective of arbitrary linear codes $\mathcal{C}$ and propose an improved protocol, referred to as Protocol 3, that achieves a higher PIR rate $\mathsf{R}_{\mathsf{P3}}(\mathcal{C},\bar{\mathcal{C}})$, where $\mathsf{R}(\mathcal{C},\bar{\mathcal{C}}) \leq \mathsf{R}_{\mathsf{P3}}(\mathcal{C},\bar{\mathcal{C}}) \leq \mathsf{R}_{\mathsf{UB}} \leq \mathsf{R}_{\mathsf{GRS}}$. In particular, we show that the upper bound R_{UB} can be achieved for some non-MDS codes. Also, for a given T we present a code family for $\mathcal{C}$ for which $\bar{\mathcal{C}}$ exists.

A. Protocol 3: The Multiple Colluding Nodes Case

The protocol presented here, referred to as Protocol 3, can be seen as an extension of Protocol 2 in Section V. We assume that each file $\boldsymbol{X}^{(m)} = \big(x_{i,j}^{(m)}\big)$, $m \in \mathbb{N}_f$, of size $\beta \times k$, is stored using an $[n,k]$ code $\mathcal{C}$ over $\mathrm{GF}(q)$, where $x_{i,j}^{(m)} \in \mathrm{GF}(q^\ell)$ for some $\ell \in \mathbb{N}$. Let $\bar{\mathcal{C}}$ be an $[n,\bar{k}]$ code over $\mathrm{GF}(q)$. The code $\bar{\mathcal{C}}$ is used to design the query matrix $\boldsymbol{Q}^{(l)}$, of dimensions $d \times \beta f$, where $\boldsymbol{q}_i^{(l)}$ is the i-th subquery of $\boldsymbol{Q}^{(l)}$ (see Section III-A). Furthermore, $\bar{\mathcal{C}}$ characterizes T, i.e., the maximum number of colluding nodes the PIR protocol can handle whilst maintaining information-theoretic privacy. As for Protocol 2, β and d are taken as small as possible according to (6). The response vector corresponding to the i-th subquery $\boldsymbol{q}_i^{(l)}$, denoted by $\boldsymbol{\rho}^{(i)} = (r_{1,i},\ldots,r_{n,i})^{\mathsf{T}}$, is a collection of the n response symbols $r_{l,i}$ from the n storage nodes and is related to the codewords of an $[n,\tilde{k}]$ code $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$. Furthermore, $\tilde{\mathcal{C}}$ characterizes the PIR rate of the protocol.

1) Query Construction: The protocol requires that the user constructs queries by choosing βf codewords $\bar{\boldsymbol{c}}_i^{(m)} = (\bar{c}_{i,1}^{(m)},\ldots,\bar{c}_{i,n}^{(m)})$, $i \in \mathbb{N}_\beta$ and $m \in \mathbb{N}_f$, drawn independently and uniformly at random from the code $\bar{\mathcal{C}}$. It then constructs the vector

$$\mathring{\boldsymbol{c}}_l = (\mathring{\boldsymbol{c}}_l^{(1)},\ldots,\mathring{\boldsymbol{c}}_l^{(f)}), \quad l \in \mathbb{N}_n,$$

where $\mathring{\boldsymbol{c}}_l^{(m)} = (\bar{c}_{1,l}^{(m)},\ldots,\bar{c}_{\beta,l}^{(m)})$. Thus, the vector $\mathring{\boldsymbol{c}}_l$ is of length βf. The vector $\mathring{\boldsymbol{c}}_l^{(m)}$ is a collection of the entries of the l-th coordinates of the codewords $\bar{\boldsymbol{c}}_1^{(m)},\ldots,\bar{\boldsymbol{c}}_\beta^{(m)}$ that pertain to the m-th file. We denote by $\mathcal{J}_i \subseteq \mathbb{N}_n$, $i \in \mathbb{N}_d$, $|\mathcal{J}_i| = \Gamma$, the set of nodes from which the protocol obtains code symbols pertaining to the m-th file from the i-th subresponses.

Similar to Protocol 2 presented in Section V for the case of noncolluding nodes, we need to construct a matrix $\hat{\boldsymbol{E}}$ and β information sets $\{\mathcal{I}_i\}_{i\in\mathbb{N}_\beta}$. The matrix $\hat{\boldsymbol{E}}$ is a $d \times n$ binary matrix where each row represents an erasure pattern of weight Γ correctable by $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$. The column weight profile of $\hat{\boldsymbol{E}}$ is determined from $\{\mathcal{I}_i\}_{i\in\mathbb{N}_\beta}$ as in Section V. Note that $\mathcal{J}_i$ is the support of the i-th row vector of $\hat{\boldsymbol{E}}$. Let m denote the index of the requested file. Then, the i-th subquery to node l is constructed as

$$\boldsymbol{q}_i^{(l)} = \mathring{\boldsymbol{c}}_l + \boldsymbol{\delta}_i^{(l)}, \tag{29}$$

where

$$\boldsymbol{\delta}_i^{(l)} = \begin{cases} \boldsymbol{\omega}_{\beta(m-1)+s_i^{(l)}} & \text{if } l \in \mathcal{J}_i, \\ \boldsymbol{\omega}_0 & \text{otherwise,} \end{cases} \tag{30}$$

for $l \in \mathbb{N}_n$, where $\boldsymbol{\omega}_t$, $t \in \mathbb{N}_{\beta f}$, is the t-th (βf)-dimensional unit vector and $\boldsymbol{\omega}_0 = \mathbf{0}_{1\times\beta f}$. The index $s_i^{(l)}$ is defined as

$$s_i^{(l)} \in \mathcal{F}_l = \{t \in \mathbb{N}_\beta \colon l \in \mathcal{I}_t\} \tag{31}$$

and $s_i^{(l)} \neq s_{i'}^{(l)}$ for $i \neq i'$, $i,i' \in \mathbb{N}_d$. The index $s_i^{(l)}$ denotes the symbol downloaded from the $s_i^{(l)}$-th row of the chunk pertaining to $\boldsymbol{X}^{(m)}$ of the l-th node in response to the i-th subquery. Clearly, we see that the symbols downloaded from all nodes form β information sets as $\sum_{i=1}^{d}|\mathcal{J}_i| = \sum_{i=1}^{\beta}|\mathcal{I}_i| = \beta k$.

Note that in (29), the vector $\mathring{\boldsymbol{c}}_l$ introduces randomness such that privacy is ensured, while the vector $\boldsymbol{\omega}$ is deterministic and is properly designed such that the requested file can be recovered by the user.

2) Response Construction: For the i-th subquery, the response symbol from the l-th node is constructed as

$$r_{l,i} = \langle \boldsymbol{q}_i^{(l)}, (c_{1,l}^{(1)},\ldots,c_{\beta,l}^{(f)})\rangle. \tag{32}$$

The response symbol in (32) is the dot product between the subquery vector to the l-th node and its content. The user obtains a response vector $\boldsymbol{\rho}^{(i)}$, consisting of response symbols from n nodes as

$$\boldsymbol{\rho}^{(i)} = \begin{pmatrix} r_{1,i} \\ r_{2,i} \\ \vdots \\ r_{n,i} \end{pmatrix} = \underbrace{\sum_{i'=1}^{\beta}\sum_{m'=1}^{f} \begin{pmatrix} \bar{c}_{i',1}^{(m')} c_{i',1}^{(m')} \\ \bar{c}_{i',2}^{(m')} c_{i',2}^{(m')} \\ \vdots \\ \bar{c}_{i',n}^{(m')} c_{i',n}^{(m')} \end{pmatrix}}_{\in\left\{\boldsymbol{x}\in(\mathrm{GF}(q^\ell))^n \colon \boldsymbol{H}^{\bar{\mathcal{C}}}\boldsymbol{x}=\mathbf{0}\right\}} + \begin{pmatrix} o_1^{(i)} \\ o_2^{(i)} \\ \vdots \\ o_n^{(i)} \end{pmatrix}, \tag{33}$$

where $\boldsymbol{H}^{\tilde{\mathcal{C}}}$ is a parity-check matrix of the code $\tilde{\mathcal{C}}$,[7] the symbol $o_l^{(i)}$ denotes the symbol obtained from the l-th node corresponding to the i-th subquery, and

$$o_l^{(i)} = \begin{cases} c_{i',l}^{(m)} & \text{if } l \in \mathcal{J}_i, \\ 0 & \text{otherwise,} \end{cases}$$

where $i' = s_i^{(l)}$. These symbols are obtained by post-processing (33) as follows,

$$\boldsymbol{H}^{\bar{\mathcal{C}}}\boldsymbol{\rho}^{(i)} = \boldsymbol{H}^{\bar{\mathcal{C}}} \begin{pmatrix} o_1^{(i)} \\ o_2^{(i)} \\ \vdots \\ o_n^{(i)} \end{pmatrix}. \tag{34}$$

[7]Note that the upload cost of the PIR scheme in [29], [30] grows linearly with f. However, when the file size is large (i.e., when ℓ is large; q^ℓ is the field size of the message symbols) the upload cost can be ignored.

This completes the construction of the PIR protocol. In the following, we prove that this protocol satisfies the PIR conditions (5a) and (5b) in Definition 8.

Lemma 6. *Consider a DSS that uses an $[n,k]$ code with subpacketization α to store f files, each divided into β stripes, and assume the privacy model of Section III-A with a set $\mathcal{T}=\{t_1,\dots,t_{|\mathcal{T}|}\}\subset\mathbb{N}_n$ of $|\mathcal{T}|\leq T\leq d_{\mathsf{min}}^{\bar{\mathcal{C}}^\perp}-1$ colluding nodes. Then, the subqueries $\boldsymbol{q}_i^{(l)}$, $l\in\mathbb{N}_n$, $i\in\mathbb{N}_d$, designed as in* (29) *and* (30) *satisfy* $\mathsf{H}\big(m|\boldsymbol{Q}^{(t_1)},\dots,\boldsymbol{Q}^{(t_{|\mathcal{T}|})}\big)=\mathsf{H}(m)$.

Proof: The addition of a deterministic vector in (29) does not change the probability distribution of the vectors $\boldsymbol{q}_i^{(t_1)},\dots,\boldsymbol{q}_i^{(t_{|\mathcal{T}|})}$. The same can be said about their joint distribution. Furthermore, in each query matrix $\boldsymbol{Q}^{(l)}$, $l\in\{t_1,\dots,t_{|\mathcal{T}|}\}$, the subqueries $\boldsymbol{q}_i^{(l)}$ are independent of each other. Thus, the proof follows the same lines as the proof of [29, Th. 8]. ∎

Theorem 6. *Consider a DSS that uses an $[n,k]$ code $\mathcal{C}$ with subpacketization α to store f files, each divided into β stripes. Let $\bar{\mathcal{C}}$ be an $[n,\bar{k}]$ code such that there exists an $[n,\tilde{k}]$ code $\tilde{\mathcal{C}}=\mathcal{C}\circ\bar{\mathcal{C}}$ of rate $R^{\tilde{\mathcal{C}}}<1$. If there exists a Γ-row regular $d\times n$ binary matrix $\hat{\boldsymbol{E}}$ in which each row is a correctable erasure pattern for $\tilde{\mathcal{C}}$ and satisfying condition* C3*, then* $\mathsf{H}(\boldsymbol{X}^{(m)}|\boldsymbol{\rho}^{(1)},\dots,\boldsymbol{\rho}^{(d)})=0$ *and the PIR rate*

$$\mathsf{R}_{\mathsf{P3}}(\mathcal{C},\bar{\mathcal{C}})=\frac{\Gamma}{n}\leq\mathsf{R}_{\mathsf{UB}} \tag{35}$$

is achievable.

Proof: By assumption there exists a matrix $\hat{\boldsymbol{E}}$ of size $d\times n$ having row weight Γ. Furthermore, again by assumption, each row of $\hat{\boldsymbol{E}}$ is an erasure pattern that is correctable by $\tilde{\mathcal{C}}$. From (30), (34) results in

$$\boldsymbol{H}^{\tilde{\mathcal{C}}}\boldsymbol{\rho}^{(i)}=\boldsymbol{H}^{\tilde{\mathcal{C}}}|_{\chi(\hat{\boldsymbol{e}}_i)}\begin{pmatrix}o_{l_1}^{(i)}\\ o_{l_2}^{(i)}\\ \vdots\\ o_{l_{|\mathcal{J}_i|}}^{(i)}\end{pmatrix},$$

where $\hat{\boldsymbol{e}}_i$ is the i-th row of $\hat{\boldsymbol{E}}$ and l_j, $j\in\mathbb{N}_{|\mathcal{J}_i|}$, denotes the elements of $\mathcal{J}_i$. The above linear system of equations is full rank as $\boldsymbol{H}^{\tilde{\mathcal{C}}}|_{\chi(\hat{\boldsymbol{e}}_i)}$ is full rank. This is because $\hat{\boldsymbol{e}}_i$ is a correctable erasure pattern for $\tilde{\mathcal{C}}$. As such, the Γ symbols $\{o_l^{(i)}\}_{l\in\mathcal{J}_i}$ are obtained. From all responses, the user obtains $\Gamma d=\beta k$ code symbols of the code $\mathcal{C}$. Furthermore, from (31), these Γd symbols are part of the β information sets $\{\mathcal{I}_i\}_{i\in\mathbb{N}_\beta}$ of $\mathcal{C}$. Thus, $\mathsf{H}\big(\boldsymbol{X}^{(m)}|\boldsymbol{\rho}^{(1)},\dots,\boldsymbol{\rho}^{(d)}\big)=0$. ∎

Unlike [29], where the authors consider sets $\mathcal{J}_i$ with a fixed structure, we generalize the sets to match arbitrary codes $\mathcal{C}$, $\bar{\mathcal{C}}$, and $\tilde{\mathcal{C}}$. In particular, the sets in [29] were constructed targeting MDS codes, in which case the PIR rate of the $(\mathcal{C},\bar{\mathcal{C}})$-retrieval protocol is upperbounded by R_{UB} in (28), as mentioned earlier. However, the use of these sets for arbitrary codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ does not allow to obtain the requested file $\boldsymbol{X}^{(m)}$. Thus, Theorem 6 can be seen as a generalization of [29, Th. 7], where the PIR rate for non-MDS codes was shown to be $\mathsf{R}(\mathcal{C},\bar{\mathcal{C}})=(d_{\mathsf{min}}^{\bar{\mathcal{C}}}-1)/n<\mathsf{R}_{\mathsf{UB}}$. Our proposed protocol can achieve higher rates as illustrated in the following corollary. In particular, we will show that the upper bound R_{UB} is achievable for some classes of non-MDS codes.

Corollary 9. *If for an $[n,k]$ code $\mathcal{C}$ and an $[n,\bar{k}]$ code $\bar{\mathcal{C}}$ there exists an $[n,\tilde{k}]$ code $\tilde{\mathcal{C}}=\mathcal{C}\circ\bar{\mathcal{C}}$ of rate $R^{\tilde{\mathcal{C}}}<1$ and an $(n-\tilde{k})$-row regular $d\times n$ binary matrix $\hat{\boldsymbol{E}}$ in which each row is a correctable erasure pattern by $\tilde{\mathcal{C}}$ and satisfying condition* C3*, then Protocol 3 achieves the upper bound* R_{UB}.

As for Protocol 2, the parameters Γ, β, and d mentioned in Theorem 6 have to be carefully selected such $\beta k=\Gamma d$ and such that a Γ-row regular matrix $\hat{\boldsymbol{E}}$ (satisfying condition C3) actually exists with a valid collection of information sets $\{\mathcal{I}_i\}_{i\in\mathbb{N}_\beta}$ for $\mathcal{C}$.

B. Example

Lemma 6 proves that the proposed protocol provides privacy up to $T=d_{\mathsf{min}}^{\bar{\mathcal{C}}^\perp}-1$ colluding nodes. This is illustrated in the example below.

Consider a DSS with $n=12$ nodes that stores a single file $\boldsymbol{X}^{(1)}$ of size 1×4. $\boldsymbol{X}^{(1)}$ is encoded using the $[12,4,6]$ binary code $\mathcal{C}$ with parity-check matrix

$$\boldsymbol{H}^{\mathcal{C}}=\begin{pmatrix}0&1&1&0&1&0&0&0&0&0&0&0\\ 1&0&1&0&0&1&0&0&0&0&0&0\\ 1&1&1&0&0&0&1&0&0&0&0&0\\ 1&1&0&1&0&0&0&1&0&0&0&0\\ 1&0&1&1&0&0&0&0&1&0&0&0\\ 0&1&1&1&0&0&0&0&0&1&0&0\\ 1&1&1&1&0&0&0&0&0&0&1&0\\ 0&0&1&1&0&0&0&0&0&0&0&1\end{pmatrix}.$$

Let $\bar{\mathcal{C}}=\mathcal{C}$, and the code $\tilde{\mathcal{C}}=\mathcal{C}\circ\bar{\mathcal{C}}$ has parity-check matrix

$$\boldsymbol{H}^{\tilde{\mathcal{C}}}=\begin{pmatrix}1&1&1&1&0&0&1&1&1&1&0&0\\ 1&1&0&1&1&1&0&1&0&0&1&1\end{pmatrix}.$$

Note that the dual code $\bar{\mathcal{C}}^\perp$ has minimum Hamming distance $d_{\mathsf{min}}^{\bar{\mathcal{C}}^\perp}=3$, thus Protocol 3 protects against $T=d_{\mathsf{min}}^{\bar{\mathcal{C}}^\perp}-1=2$ colluding nodes. Choosing $\Gamma=d_{\mathsf{min}}^{\tilde{\mathcal{C}}}-1=1$, one can use the PIR protocol as presented in [29] to get a PIR rate of $\mathsf{R}(\mathcal{C},\bar{\mathcal{C}})=\frac{1}{12}$. However, we can set $\Gamma=2$ and use Protocol 3 to achieve a higher PIR rate. Note that the value of Γ cannot be greater than 2 as the number of redundant symbols in $\tilde{\mathcal{C}}$ is 2. We choose

$$\hat{\boldsymbol{E}}=\begin{pmatrix}0&0&0&0&0&0&0&0&1&0&0&1\\ 0&1&1&0&0&0&0&0&0&0&0&0\end{pmatrix}$$

and $\mathcal{I}_1=\{2,3,9,12\}$. Thus, $\beta=1$ and $d=2$. Note that each row of $\hat{\boldsymbol{E}}$ is an erasure pattern that is correctable by the $[12,10,2]$ code $\tilde{\mathcal{C}}$, and that $\mathcal{I}_1$ is an information set of $\mathcal{C}$. In order to form all the queries (each query consists of $d=2$ subqueries), we need to choose $s_1^{(9)}$, $s_1^{(12)}$, $s_2^{(2)}$, and $s_2^{(3)}$. From (31), we have

$$s_1^{(9)}=1, s_1^{(12)}=1, s_2^{(2)}=1, \text{ and } s_2^{(3)}=1.$$

Now, consider the first subqueries. The query vectors $\boldsymbol{q}_1^{(9)}$ and $\boldsymbol{q}_1^{(12)}$ are

$$\boldsymbol{q}_1^{(9)}=\mathring{\boldsymbol{c}}_9+\boldsymbol{\omega}_1=\mathring{\boldsymbol{c}}_9+1,$$
$$\boldsymbol{q}_1^{(12)}=\mathring{\boldsymbol{c}}_{12}+\boldsymbol{\omega}_1=\mathring{\boldsymbol{c}}_{12}+1,$$

and $\boldsymbol{q}_1^{(l)} = \mathring{\boldsymbol{c}}_l$, $\forall\, l \in \mathbb{N}_{12}\backslash\{9,12\}$. The corresponding response vector is

$$\boldsymbol{\rho}^{(1)} = \underbrace{\sum_{i'=1}^{1}\sum_{m'=1}^{1} \begin{pmatrix} \bar{c}_{i',1}^{(m')} c_{i',1}^{(m')} \\ \bar{c}_{i',2}^{(m')} c_{i',2}^{(m')} \\ \vdots \\ \bar{c}_{i',12}^{(m')} c_{i',12}^{(m')} \end{pmatrix}}_{\in\left\{\boldsymbol{x}\in(\mathrm{GF}(q^\ell))^{12}\colon \boldsymbol{H}^{\bar{\mathcal{C}}}\boldsymbol{x}=\boldsymbol{0}\right\}} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c_{1,9}^{(1)} \\ 0 \\ 0 \\ c_{1,12}^{(1)} \end{pmatrix}.$$

Finally, the user computes

$$\boldsymbol{H}^{\tilde{\mathcal{C}}}\boldsymbol{\rho}^{(1)} = \boldsymbol{H}^{\tilde{\mathcal{C}}} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c_{1,9}^{(1)} \\ 0 \\ 0 \\ c_{1,12}^{(1)} \end{pmatrix} = \begin{pmatrix} c_{1,9}^{(1)} \\ c_{1,12}^{(1)} \end{pmatrix}.$$

In a similar manner, from $\boldsymbol{\rho}^{(2)}$ the user obtains $c_{1,2}^{(1)}$ and $c_{1,3}^{(1)}$. Clearly, the indices of the symbols downloaded by the user form the information set $\mathcal{I}_1$, from which we can obtain the requested file $\boldsymbol{X}^{(1)}$. The PIR rate of the scheme is $\mathsf{R}_{\mathsf{P3}}(\mathcal{C},\bar{\mathcal{C}}) = \frac{4}{24} = \frac{1}{6}$, i.e., double of the PIR rate of the protocol in [29]. Furthermore, it achieves the upper bound in (35).

A limiting factor for Protocol 3 is that the upper bound on the PIR rate R_{UB} in (35) depends on the dimension of $\tilde{\mathcal{C}}$. Furthermore, in order to achieve the PIR property with large T, one requires $\bar{\mathcal{C}}$ to be of large dimension such that $\bar{\mathcal{C}}^\perp$ has large minimum Hamming distance. Therefore, for arbitrary $\mathcal{C}$ and $\bar{\mathcal{C}}$ the chances that $\tilde{\mathcal{C}} = \mathcal{C}\circ\bar{\mathcal{C}}$ has rate $R^{\tilde{\mathcal{C}}} = 1$ (the code does not even correct a single erasure) are quite high for large values of T, since the Hadamard product is highly nonlinear. In other words, the probability that $\mathsf{R}_{\mathsf{UB}} = 0$ is high. Below, we present code constructions for which $\mathsf{R}_{\mathsf{UB}} > 0$.

C. Codes for Protocol 3

As seen in the preceding subsections, the codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ must be chosen such that the code $\tilde{\mathcal{C}} = \mathcal{C}\circ\bar{\mathcal{C}}$ has rate $R^{\tilde{\mathcal{C}}} < 1$ for the PIR protocol to work. In this subsection, we provide a family of codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ that satisfy $R^{\tilde{\mathcal{C}}} < 1$ for a given T.

In particular, we show that a special class of UUV codes can be used for the codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ to obtain a valid $\tilde{\mathcal{C}}$. Let $\mathcal{U}$ be an $[n_1,k_1]$ code and $\mathcal{V}$ an $[n_1,1]$ repetition code, both over $\mathrm{GF}(2)$. We construct the $[n=2n_1, k=k_1+1]$ code $\mathcal{C} = (\mathcal{U} \mid \mathcal{U}+\mathcal{V})$ with generator matrix

$$\boldsymbol{G}^{\mathcal{C}} = \begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{1\times n_1} & \boldsymbol{1}_{1\times n_1} \end{pmatrix}. \tag{36}$$

Theorem 7. *Let $\mathcal{U}$ be an $[n_1,k_1]$ binary code where $n_1 \geq k_1+2$ and $\mathcal{V}$ an $[n_1,1]$ binary repetition code. Then, the $[n=2n_1, k=k_1+1]$ codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ constructed using (36) ensure that the vector space $\tilde{\mathcal{C}} = \mathcal{C}\circ\bar{\mathcal{C}}$ of length n is a linear code of dimension strictly less than n, i.e., $R^{\tilde{\mathcal{C}}} < 1$.*

Proof: See Appendix G. ∎

Theorem 7 proves that for an arbitrary linear code $\mathcal{U}$, the UUV code ensures that $\tilde{k} < n$ and thus $\tilde{\mathcal{C}}$ is a valid code. The fact that any code $\mathcal{U}$ can be used in the protocol makes the UUV construction attractive. Also, the UUV construction may produce d_{min}-optimal binary linear codes. For instance, the codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ in Section VIII-B are d_{min}-optimal binary linear codes that can be constructed through the UUV construction. One drawback of the UUV construction, however, is that the constructed codes are in general low rate codes.

In [30], the authors showed that choosing $\mathcal{C}$ and $\bar{\mathcal{C}}$ to be RM codes with carefully selected parameters ensures that $\tilde{\mathcal{C}}$ is also an RM code of dimension $\tilde{k} < n$. However, the PIR rate is very low [30, Th. 15]. In the following subsection, we show that RM codes can indeed achieve a higher PIR rate of $\mathsf{R}_{\mathsf{P3}}(\mathcal{C},\bar{\mathcal{C}}) = (n-\tilde{k})/n = \mathsf{R}_{\mathsf{UB}}$.

D. Codes Achieving the Maximum PIR Rate of Protocol 3

In order to consider the codes achieving the maximum possible PIR rate for Protocol 3, we give a definition similar to Definition 10 in Section IV.

Definition 14. *Let $\mathcal{C}$ be an $[n,k]$ code and $\bar{\mathcal{C}}$ an $[n,\bar{k}]$ code. Denote by $\tilde{\mathcal{C}} = \mathcal{C}\circ\bar{\mathcal{C}}$ the $\tilde{k}$-dimensional code generated by the Hadamard product of $\mathcal{C}$ and $\bar{\mathcal{C}}$. A $(k+n-\tilde{k})\times n$ binary matrix $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ is called a PIR maximum rate matrix for Protocol 3 if the following conditions are satisfied.*

1) *$\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ is a k-column regular matrix, and*
2) *there are exactly k rows $\{\boldsymbol{\lambda}_i\}_{i\in\mathbb{N}_k}$ and $n-\tilde{k}$ rows $\{\boldsymbol{\lambda}_{i+k}\}_{i\in\mathbb{N}_{n-\tilde{k}}}$ of $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ such that $\forall\, i\in\mathbb{N}_k$, $\chi(\boldsymbol{\lambda}_i)$ is an information set for $\tilde{\mathcal{C}}$ and $\forall\, i\in\mathbb{N}_{n-\tilde{k}}$, $\chi(\boldsymbol{\lambda}_{i+k})$ is an information set for $\mathcal{C}$.*

Similar to the case of noncolluding nodes in Section V, it is not difficult to show that the existence of a $k\times n$ matrix $\hat{\boldsymbol{E}}$ for the code $\tilde{\mathcal{C}} = \mathcal{C}\circ\bar{\mathcal{C}}$ and an $(n-\tilde{k})\times n$ matrix $\bar{\boldsymbol{E}}$ for the code $\mathcal{C}$ is equivalent to the existence of $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$.

The following corollary follows immediately from a similar reasoning as for Theorem 3.

Corollary 10. *If a PIR maximum rate matrix $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ exists for Protocol 3, then*

$$d_s^{\mathcal{C}} \geq \frac{n-\tilde{k}}{k}s, \quad \forall\, s\in\mathbb{N}_k,$$
$$d_s^{\tilde{\mathcal{C}}} \geq s, \qquad \forall\, s\in\mathbb{N}_{\tilde{k}}.$$

Proof: Using an argumentation similar to the proof of Theorem 3, the existence of a PIR maximum rate matrix for Protocol 3 implies that there exist k information sets $\{\tilde{\mathcal{I}}_i\}_{i\in\mathbb{N}_k}$ of $\tilde{\mathcal{C}}$ and $n-\tilde{k}$ information sets $\{\mathcal{I}_{i'}\}_{i'\in\mathbb{N}_{n-\tilde{k}}}$ of $\mathcal{C}$ such that each coordinate j of $\mathcal{C}$ appears exactly k times in $\{\tilde{\mathcal{I}}_i\}_{i\in\mathbb{N}_k}\cup\{\mathcal{I}_{i'}\}_{i'\in\mathbb{N}_{n-\tilde{k}}}$, $j\in\mathbb{N}_n$. Hence, we obtain

$$k\,|\chi(\mathcal{D})| = \underbrace{\sum_{i=1}^{k}\big|\tilde{\mathcal{I}}_i\cap\chi(\mathcal{D})\big|}_{\geq 0} + \sum_{i'=1}^{n-\tilde{k}}\big|\mathcal{I}_{i'}\cap\chi(\mathcal{D})\big|$$
$$\geq \sum_{i'=1}^{n-\tilde{k}}\big|\mathcal{I}_{i'}\cap\chi(\mathcal{D})\big| \geq (n-\tilde{k})s;$$

$$k\left|\chi(\tilde{\mathcal{D}})\right| = \sum_{i=1}^{k}\left|\tilde{\mathcal{I}}_i \cap \chi(\tilde{\mathcal{D}})\right| + \underbrace{\sum_{i'=1}^{n-\tilde{k}}\left|\mathcal{I}_{i'} \cap \chi(\tilde{\mathcal{D}})\right|}_{\geq 0}$$

$$\geq \sum_{i=1}^{k}\left|\tilde{\mathcal{I}}_i \cap \chi(\tilde{\mathcal{D}})\right| \geq ks,$$

where $\mathcal{D}$ is an $[n, s]$ subcode of $\mathcal{C}$, $s \in \mathbb{N}_k$, and $\tilde{\mathcal{D}}$ is an $[n, s]$ subcode of $\tilde{\mathcal{C}}$, $s \in \mathbb{N}_{\tilde{k}}$. ∎

It can be seen from the proof above that we can only have $|\tilde{\mathcal{I}} \cap \chi(\mathcal{D})| \geq 0$ for an information set $\tilde{\mathcal{I}}$ of $\tilde{\mathcal{C}}$ and a subcode $\mathcal{D} \subseteq \mathcal{C}$ (or $|\mathcal{I} \cap \chi(\tilde{\mathcal{D}})| \geq 0$ for an information set $\mathcal{I}$ of $\mathcal{C}$ and a subcode $\tilde{\mathcal{D}} \subseteq \tilde{\mathcal{C}}$). Hence, unlike in Conjecture 1, we do not conjecture this necessary condition to be sufficient.

Similar to Theorem 4 for the noncolluding case, we provide a sufficient condition for codes to achieve the maximum possible PIR rate of Protocol 3 by using code automorphisms of $\mathcal{C}$ and $\tilde{\mathcal{C}}$.

Theorem 8. *Let $\mathcal{C}$ be an $[n, k]$ code, $\bar{\mathcal{C}}$ an $[n, \bar{k}]$ code, and $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$. If there exist k information sets $\tilde{\mathcal{I}}_1, \ldots, \tilde{\mathcal{I}}_k$ of $\tilde{\mathcal{C}}$, an information set $\mathcal{I}$ of $\mathcal{C}$, and $n - \tilde{k}$ distinct automorphisms of $\mathcal{C}$ such that for every code coordinate $j_i \in \mathcal{I}$, $i \in \mathbb{N}_k$,*

$$\tilde{\mathcal{I}}_i \cup \{\pi_1(j_i), \ldots, \pi_{n-\tilde{k}}(j_i)\} = \{1, 2, \ldots, n\},$$

then the codes $\mathcal{C}$ and $\bar{\mathcal{C}}$ achieve the maximum possible PIR rate of Protocol 3, i.e., R_{UB}.

Proof: Since there exist $n - \tilde{k}$ distinct automorphisms of $\mathcal{C}$ such that $\mathcal{I}_j \triangleq \{\pi_j(j_i) \colon j_i \in \mathcal{I}\}$, $j \in \mathbb{N}_{n-\tilde{k}}$, are information sets of $\mathcal{C}$, and for every code coordinate $j_i \in \mathcal{I}$, $i \in \mathbb{N}_k$,

$$\tilde{\mathcal{I}}_i \cup \{\pi_1(j_i), \ldots, \pi_{n-\tilde{k}}(j_i)\} = \{1, 2, \ldots, n\},$$

each code coordinate $h \in \mathbb{N}_n$ appears exactly k times in $\{\tilde{\mathcal{I}}_i\}_{i\in\mathbb{N}_k} \cup \{\mathcal{I}_j\}_{j\in\mathbb{N}_{n-\tilde{k}}}$, which shows the existence of a PIR maximum rate matrix $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ for Protocol 3. ∎

We now show that RM codes achieve the maximum PIR rate of Protocol 3.

Corollary 11. *Let $\mathcal{C}$ be an $[n, k]$ RM code $\mathcal{R}(v, m)$, $\bar{\mathcal{C}}$ an $[n, \bar{k}]$ RM code $\mathcal{R}(\bar{v}, m)$, and $\tilde{k} = k + \bar{k}$, where $n = 2^m$, $k = \sum_{i=0}^{v}\binom{m}{i}$, and $\bar{k} = \sum_{i=0}^{\bar{v}}\binom{m}{i}$. Then, a PIR maximum rate matrix $\tilde{\boldsymbol{\Lambda}}_{k,k+n-\tilde{k}}$ exists for Protocol 3, and its PIR rate is*

$$\mathsf{R}_{\mathsf{P3}}(\mathcal{C}, \bar{\mathcal{C}}) = \frac{n - \tilde{k}}{n} = \mathsf{R}_{\mathsf{UB}}.$$

Proof: It can be easily shown that $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$ is an RM code $\mathcal{R}(\tilde{v}, m)$ with $\tilde{v} = v + \bar{v}$. Consider two information sets $\mathcal{I}$ and $\tilde{\mathcal{I}}$ of $\mathcal{C}$ and $\tilde{\mathcal{C}}$, respectively. (Lemma 1 gives one way to construct these two information sets.) We construct the $k + n - \tilde{k}$ information sets

$$\tilde{\mathcal{I}}_i \triangleq \{\boldsymbol{\sigma} + \boldsymbol{\mu}_i \colon \boldsymbol{\sigma} \in \tilde{\mathcal{I}}\}, \quad i \in \mathbb{N}_k,$$
$$\mathcal{I}_j \triangleq \{\boldsymbol{\mu} + \bar{\boldsymbol{\sigma}}_j \colon \boldsymbol{\mu} \in \mathcal{I}\}, \quad j \in \mathbb{N}_{n-\tilde{k}},$$

for $\tilde{\mathcal{C}}$ and $\mathcal{C}$, respectively, where $\{\boldsymbol{\mu}_i\}_{i\in\mathbb{N}_k}$ and $\{\bar{\boldsymbol{\sigma}}_j\}_{j\in\mathbb{N}_{n-\tilde{k}}}$ are the numbered binary m-tuples in $\mathcal{I}$ and $\mathrm{GF}(2)^{m\times 1} \setminus \tilde{\mathcal{I}}$, respectively.

Without loss of generality, the i-th information set $\tilde{\mathcal{I}}_i$, $i \in \mathbb{N}_k$, can be written as

$$\tilde{\mathcal{I}}_i = \{\boldsymbol{\mu}_i + \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\mu}_i + \boldsymbol{\sigma}_{\tilde{k}}\},$$

where $\boldsymbol{\sigma}_{j'} \in \tilde{\mathcal{I}}$, $j' \in \mathbb{N}_{\tilde{k}}$. Furthermore, consider the i-th elements across all sets $\mathcal{I}_j$, $j \in \mathbb{N}_{n-\tilde{k}}$. They have the form $\boldsymbol{\mu}_i + \bar{\boldsymbol{\sigma}}_j$, where $\boldsymbol{\mu}_i \in \mathcal{I}$. Since $\bar{\boldsymbol{\sigma}}_j \in \mathrm{GF}(2)^{m\times 1} \setminus \tilde{\mathcal{I}}$ and $\boldsymbol{\sigma}_i \in \tilde{\mathcal{I}}$, the set

$$\{\boldsymbol{\mu}_i + \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\mu}_i + \boldsymbol{\sigma}_{\tilde{k}}\} \cup \{\boldsymbol{\mu}_i + \bar{\boldsymbol{\sigma}}_1, \ldots, \boldsymbol{\mu}_i + \bar{\boldsymbol{\sigma}}_{n-\tilde{k}}\}$$

with cardinality $n = 2^m$ is equal to $\mathrm{GF}(2)^{m\times 1}$, i.e., the set containing the elements of the i-th information set $\tilde{\mathcal{I}}_i$ and the i-th elements $\boldsymbol{\mu}_i + \bar{\boldsymbol{\sigma}}_j$ in all sets $\mathcal{I}_j$ is equal to the set of all binary $n = 2^m$ tuples. Therefore, we are able to find k information sets $\{\tilde{\mathcal{I}}_i\}_{i\in\mathbb{N}_k}$ of $\tilde{\mathcal{C}}$, an information set $\mathcal{I}$ of $\mathcal{C}$, and $n - \tilde{k}$ distinct automorphisms $\pi_j(\boldsymbol{\mu}) = \boldsymbol{\mu} + \bar{\boldsymbol{\sigma}}_j$ of $\mathcal{C}$, $j \in \mathbb{N}_{n-\tilde{k}}$, satisfying Theorem 8. This completes the proof. ∎

We remark again that Corollary 11 can be extended to nonbinary generalized RM codes. Finally, note that in the independent work [32] it was also shown that RM codes can achieve the maximum possible PIR rate of the $(\mathcal{C}, \bar{\mathcal{C}})$-retrieval protocol, i.e., R_{UB}, for transitive codes. However, it is important to highlight that our Protocol 3 requires a much smaller β (number of stripes) and a significant smaller d (number of subqueries). Indeed, the protocol in [32] requires very large β and d (in the order of 10000 for the example provided), and thus our protocol is more practical.

E. Optimizing the PIR rate

For those codes for which we do not have a proof that R_{UB} is achieved, we now provide an algorithm to optimize the PIR rate $\mathsf{R}_{\mathsf{P3}}(\mathcal{C}, \bar{\mathcal{C}})$ for a given storage code $\mathcal{C}$ and query code $\bar{\mathcal{C}}$ such that it comes closer to the upper bound R_{UB}. The algorithm is identical to Algorithm 1 for the case of noncolluding nodes with some key differences which we highlight below.

- In Line 1, Γ is initialized to 1.
- The while loop in Line 4 runs up to $n - \tilde{k}$.
- The first argument to the subprocedure `ComputeErasurePatternList`$(\cdot, \Gamma)$ is changed from $\mathcal{C}$ to $\mathcal{C} \circ \bar{\mathcal{C}}$ in Line 5.

With these minor modifications, Algorithm 1 can be used to optimize the PIR rate in the case of T colluding nodes. Numerical results are presented below in Section IX. Note that Γ is initialized to 1 as opposed to $\min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1)$ in the case of noncolluding nodes. This is because $\bar{\boldsymbol{E}}$ and $\bar{\boldsymbol{E}}$ of $\boldsymbol{E}$ are based on different codes. This also guarantees that the algorithm always returns $\boldsymbol{E}_{\mathsf{opt}} \neq \emptyset$ (assuming $d^{\bar{\mathcal{C}}}_{\mathsf{min}} \geq 2$), since in this case all weight-1 erasure patterns are correctable by $\tilde{\mathcal{C}}$, and a valid matrix $\boldsymbol{E}$ can be trivially constructed.

IX. Numerical Results

In this section, we present maximized PIR rates for the PIR protocols described in Sections IV, V, and VIII. Unless specified otherwise, these protocols are optimized using Algorithm 1 with minimum possible values for the parameters

TABLE II
OPTIMIZED VALUES FOR THE PIR RATE FOR DIFFERENT CODES HAVING CODE RATES STRICTLY LARGER THAN $1/2$ FOR THE CASE OF NONCOLLUDING NODES.

Code	$d_{\mathsf{min}}^{\mathcal{C}}$	$d_{\mathsf{min}}^{\mathcal{C}'}$	$\mathsf{R}_{\mathsf{non-opt}}$	$\mathsf{R}_{\mathsf{opt}}$	C_{∞}
$\mathcal{C}_1 : [5, 3]$ (Example 4)	2	3	0.4	0.4	0.4
$\mathcal{C}_2 : [11, 6]$	4	4	0.2727	0.4545	0.4545
$\mathcal{C}_3 : [12, 8]$ Pyramid	4	4	0.25	0.3333	0.3333
$\mathcal{C}_4 : [18, 12]$ Pyramid	5	5	0.2222	0.3333	0.3333
$\mathcal{C}_5 : [16, 10]$ LRC	5	5	0.25	0.3750	0.3750
$\mathcal{C}_6 : [154, 121]$ LRC	4	6	0.0325	0.2013	0.2143
$\mathcal{C}_7 : [187, 121]$ LRC	7	16	0.0802	0.3262	0.3529

TABLE III
OPTIMIZED VALUES FOR THE PIR RATE FOR DIFFERENT CODES HAVING CODE RATES AT MOST $1/2$ FOR THE CASE OF NONCOLLUDING NODES.

Code	$d_{\mathsf{min}}^{\mathcal{C}}$	$\mathsf{R}_{\mathsf{non-opt}}$	$\mathsf{R}_{\mathsf{opt}}$	C_{∞}
$\mathcal{C}_8 : [7, 3]$ (Example 5)	4	0.4286	0.5714	0.5714
$\mathcal{C}_9 : [9, 4]$ LRC ([27, Ex. 1])	5	0.4444	0.5555	0.5555
$\mathcal{C}_{10} : [12, 6]$ LRC ([27, Ex. 2])	6	0.4167	0.5	0.5

β and d as given in (6). In contrast to Sections VI to VI-C, where different classes of codes were proved to be MDS-PIR capacity-achieving, we consider here other codes (with two exceptions as detailed below) and their highest possible PIR rates. The results are tabulated in Tables II and III for the case of noncolluding nodes, and in Table IV for the colluding case. Results in Table II are for code rates strictly larger than $1/2$, while codes of rate at most $1/2$ are tabulated in Table III.

In Tables II and III, C_{∞} (see (8)) is the asymptotic MDS-PIR capacity and $\mathsf{R}_{\mathsf{opt}}$ is the optimized PIR rate computed from Algorithm 1. In Table II, $\mathsf{R}_{\mathsf{non-opt}} = (d_{\mathsf{min}}^{\mathcal{C}'} - 1)/n$, while in Table III, $\mathsf{R}_{\mathsf{non-opt}} = (d_{\mathsf{min}}^{\mathcal{C}} - 1)/n$. In Table IV, $\mathsf{C}_{\mathsf{LB},\infty} \triangleq (n - (k + T - 1))/n$ is a lower bound (taken from [29]) on the asymptotic MDS-PIR capacity in the case of at most T colluding nodes, while $\mathsf{R}_{\mathsf{opt}}$ is the optimized PIR rate computed from Algorithm 1 and $\mathsf{R}_{\mathsf{non-opt}} = (d_{\mathsf{min}}^{\bar{\mathcal{C}}} - 1)/n$.

The code $\mathcal{C}_1$ in Table II is from Example 4, $\mathcal{C}_2$ is an $[11, 6]$ binary linear code with optimum minimum Hamming distance, while codes $\mathcal{C}_3$ and $\mathcal{C}_4$ are Pyramid codes, taken from [23], of locality 4 and 6, respectively, $\mathcal{C}_5$ is an LRC of locality 5 borrowed from [24]. In [43], a construction of optimal (in terms of minimum Hamming distance) binary LRCs with multiple repair groups was given. In particular, in [43, Constr. 3], a construction based on array LDPC codes was provided. The minimum Hamming distance of array LDPC codes is known for certain sets of parameters (see, e.g., [44] and references therein). Codes $\mathcal{C}_6$ and $\mathcal{C}_7$ in Table II are *optimal* LRCs based on array LDPC codes constructed using [43, Constr. 3] and having information locality 11. The protocols for these two underlying codes have $\beta = \Gamma$ and $d = k$.

Code $\mathcal{C}_8$ in Table III is the dual code of the $[7, 4, 3]$ Hamming code and is taken from Example 5, while the codes $\mathcal{C}_9$ and $\mathcal{C}_{10}$ are d_{min}-optimal LRCs over $\mathrm{GF}(13)$ of all-symbol locality 2 and 3, respectively, taken from [27] (see Examples 1 and 2, respectively, in [27]). These two codes are also tabulated in Table IV. The corresponding $\bar{\mathcal{C}}$ codes are RS codes and their parameters are given in Table IV (an RS code of length n and dimension k is denoted by $\mathrm{RS}[n, k]$). Code $\mathcal{C}_{11}$ (from Table IV) is taken from Section VIII-B, while code $\mathcal{C}_{12}$ (also from Table IV) is taken from [27] (see Example 5 in [27]). Note that $\mathcal{C}_{12}$ is an LRC of length 12 over $\mathrm{GF}(13)$ with two disjoint recovering sets of sizes 2 and 3, respectively, for every symbol of the code (all-symbol locality). Code $\mathcal{C}_{13}$ (from Table IV) is a $[26, 9, 8]$ binary UUV code that is close to an optimal binary linear code (the best known code for these parameters has a minimum Hamming distance of 9), while the code $\mathcal{C}_{14}$ is a UUV code where $\mathcal{U}$ is a $[16, 5, 8]$ RM code (the code $\mathcal{R}(1, 4)$). Note that $\mathcal{C}_{14}$ becomes the RM code $\mathcal{R}(1, 5)$. Due to the high computational complexity of Algorithm 1 for $\mathcal{C}_{14}$, we are unable to compute the maximum rate of Protocol 3 for the minimum values of β and d. Instead, we take $\beta = \Gamma$ and $d = k$ and use Corollary 11 to obtain the maximum rate of the protocol.

It is observed in Tables II and III that in the case of noncolluding nodes, the optimized PIR rate $\mathsf{R}_{\mathsf{opt}}$ is equal to the asymptotic capacity C_{∞} for all tabulated codes except $\mathcal{C}_6$ and $\mathcal{C}_7$. Note that the codes $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_5$–$\mathcal{C}_{10}$ do not fall within the code families that we proved are MDS-PIR capacity-achieving (see Section VI). Thus, the results in Tables II and III show that, interestingly, other codes can achieve the asymptotic MDS-PIR capacity as well. On the other hand, $\mathcal{C}_3$ and $\mathcal{C}_4$ satisfy the conditions of Theorem 5. Thus, they are MDS-PIR capacity-achieving with $\beta = \Gamma$ and $d = k$. The results in the table show that they also achieve C_{∞} for β and d as in (6). Also, note that by the nature of the optimization procedure (see Remark 2), MDS-PIR capacity-achieving matrices $\mathbf{\Lambda}_{\kappa,\nu}$ with $\frac{\kappa}{\nu} = \frac{k}{n}$ of all tabulated codes except $\mathcal{C}_6$ and $\mathcal{C}_7$ are found. This implies that they are also MDS-PIR capacity-achieving codes for any finite number of files and must satisfy the necessary condition based on generalized Hamming weights in Theorem 3. Due to the high computational complexity of Algorithm 1, it is difficult to maximize the PIR rates of $\mathcal{C}_6$ and $\mathcal{C}_7$. Therefore, it is an open problem whether or not they are MDS-PIR capacity-achieving codes. For the colluding case (see Table IV) the lower bound $\mathsf{C}_{\mathsf{LB},\infty}$ on the asymptotic MDS-PIR capacity is not achieved, even after optimization. To the best of our knowledge, GRS codes are the only known class of codes where this bound is actually achieved [29]. On the other hand, the upper bound R_{UB} (from (35)) is attained in all cases.

X. CONCLUSION

We presented three different PIR protocols, namely Protocol 1, Protocol 2, and Protocol 3, for DSSs where data is stored using an arbitrary linear code. We first considered the case where no nodes in the DSS collude. Under this scenario, Protocols 1 and 2 achieve the PIR property. We proved that, for certain non-MDS codes, Protocol 1 achieves the finite MDS-PIR capacity (and also the asymptotic MDS-PIR capacity) and Protocol 2, which is a much simpler protocol compared to Protocol 1, achieves the asymptotic MDS-PIR capacity. Thus, the MDS property is not necessary in order to achieve the MDS-PIR capacity (both finite and asymptotic). We also provided a necessary and a sufficient condition for codes

TABLE IV
OPTIMIZED VALUES FOR THE PIR RATE FOR DIFFERENT CODES FOR THE COLLUDING CASE WITH $T = 2$ AND $T = 3$.

Code $\mathcal{C}$	$\bar{\mathcal{C}}$	$\tilde{\mathcal{C}}$	$d^{\mathcal{C}}_{\mathsf{min}}$	T	$\mathsf{R}_{\mathsf{non-opt}}$	$\mathsf{R}_{\mathsf{opt}}$	R_{UB}	$\mathsf{C}_{\mathsf{LB},\infty}$
$\mathcal{C}_9 : [9, 4]$ LRC ([27, Ex. 1])	RS$[9, 2]$	RS$[9, 6]$	5	2	0.3333	0.3333	0.3333	0.4444
$\mathcal{C}_{10} : [12, 6]$ LRC ([27, Ex. 2])	RS$[12, 2]$	RS$[12, 8]$	6	2	0.3333	0.3333	0.3333	0.4167
$\mathcal{C}_{11} : [12, 4]$ (Section VIII-B)	$\mathcal{C}$	$[12, 10, 2]$	6	2	0.0833	0.1667	0.1667	0.5833
$\mathcal{C}_{12} : [12, 4]$ LRC ([27, Ex. 5])	RS$[12, 2]$	$[12, 7, 5]$	6	2	0.3333	0.4167	0.4167	0.5833
$\mathcal{C}_{13} : [26, 9]$ $(\mathcal{U} \mid \mathcal{U} + \mathcal{V})$	$\mathcal{C}$	$[26, 22, 1]$	8	3	0	0.1538	0.1538	0.5769
$\mathcal{C}_{14} : [32, 6]$ $(\mathcal{U} \mid \mathcal{U} + \mathcal{V})$	$\mathcal{C}$	$[32, 16, 8]$	16	3	0.2188	0.5	0.5	0.75

to be MDS-PIR capacity-achieving with Protocols 1 and 2. The necessary condition is based on generalized Hamming weights while the sufficient condition is obtained from code automorphisms of the linear storage code. We proved that cyclic codes, RM codes, and a class of distance-optimal information locality codes are MDS-PIR capacity-achieving codes. For other codes, we provided an optimization algorithm that optimizes Protocols 1 and 2 in order to maximize their PIR rates. We also considered the scenario where a subset of nodes in the DSS collude. For such a scenario, we proposed Protocol 3, which is an improvement of the PIR protocol by Freij-Hollanti *et al.*. The improvement allows the protocol to achieve higher PIR rates, and the PIR rates for non-MDS codes are no longer limited by the minimum Hamming distance of the retrieval code. Subsequently, we presented an optimization algorithm to optimize the PIR rate of the protocol, and a family of codes based on the classical $(\mathcal{U} \mid \mathcal{U} + \mathcal{V})$ construction that can be used with this protocol. Furthermore, as for the noncolluding case, we provided a necessary and a sufficient condition to achieve the maximum possible PIR rate of Protocol 3. Moreover, we proved that RM codes satisfy the sufficient condition and can achieve much higher PIR rates than previously reported by Freij-Hollanti *et al.*. Finally, we presented some numerical results on the PIR rates for several linear codes, including distance-optimal all-symbol locality LRCs constructed by Tamo and Barg.

APPENDIX A
PROOF OF LEMMA 1

We need to ensure that given a $k \times n$ generator matrix $\boldsymbol{G}$ of $\mathcal{R}(v, m)$ with $k = \sum_{i=0}^{v} \binom{m}{i}$ and $n = 2^m$, the $k \times k$ matrix $\boldsymbol{G}|_{\mathcal{I}}$ that comprises the columns of the generator matrix indexed by the coordinates of $\mathcal{I}$ is invertible. We are going to elaborate on this by considering all the monomials $z_1^{\mu_1} \cdots z_m^{\mu_m}$, $\mu_i \in \mathrm{GF}(2)$, in a so-called *graded lexicographic order*, where each vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)^{\intercal} \in \mathrm{GF}(2)^{m \times 1}$ defines a column of the generator matrix $\boldsymbol{G}$ according to (1). Formally speaking, denote $z_1^{\mu_1} \cdots z_m^{\mu_m}$ by $\boldsymbol{z}^{\boldsymbol{\mu}}$. We say $\boldsymbol{z}^{\boldsymbol{\mu}} \prec \boldsymbol{z}^{\boldsymbol{\mu}'}$ either if $w_{\mathsf{H}}(\boldsymbol{\mu}) < w_{\mathsf{H}}(\boldsymbol{\mu}')$ or if $w_{\mathsf{H}}(\boldsymbol{\mu}) = w_{\mathsf{H}}(\boldsymbol{\mu}')$ and the topmost nonzero entry of $\boldsymbol{\mu} - \boldsymbol{\mu}'$ (subtraction is over the reals) is positive. For instance, in graded lexicographical ordering we have $z_1 \prec z_2 \prec z_3 \prec z_1 z_2 \prec z_1 z_3 \prec z_2 z_3 \prec z_1 z_2 z_3$ for $m = 3$.

Now we are ready for the proof. It is noted that a basis of $\mathcal{R}(v, m)$ can be viewed as $\mathcal{B} \triangleq \{1, z_1, z_2, z_3, \ldots\} = \{\boldsymbol{z}^{\boldsymbol{\mu}'} : w_{\mathsf{H}}(\boldsymbol{\mu}') \leq v\}$. Let us list the monomials in $\mathcal{B}$ in graded lexicographical order, and let the ℓ-th monomial $f_\ell(\boldsymbol{z})$ of the ordered list represent the ℓ-th row of $\boldsymbol{G}$, $\ell \in \mathbb{N}_k$. According to the generator matrix construction of $\mathcal{R}(v, m)$, it is known that the $(\ell, \boldsymbol{\mu})$ entry of $\boldsymbol{G}$ is equal to the value of the ℓ-th monomial $f_\ell(\boldsymbol{z})$ at $\boldsymbol{z} = \boldsymbol{\mu}$ [31, Ch. 13]. Furthermore, given a column coordinate $\boldsymbol{\mu} \in \mathcal{I}$, for $\ell \in \mathbb{N}_k$, we have

$$f_\ell(\boldsymbol{\mu}) = \begin{cases} 1 & \text{if } \boldsymbol{z}^{\boldsymbol{\mu}} = f_\ell(\boldsymbol{z}), \\ 0 & \text{if } \boldsymbol{z}^{\boldsymbol{\mu}} \prec f_\ell(\boldsymbol{z}). \end{cases}$$

Thus, the $(\boldsymbol{z}^{\boldsymbol{\mu}}, \boldsymbol{\mu})$ entry can be seen as a pivot of $\boldsymbol{G}|_{\mathcal{I}}$ and $\boldsymbol{G}|_{\mathcal{I}}$ is obviously invertible.

APPENDIX B
PROOF OF THEOREM 1

The proof is completed by showing that the following statements are true.

File symmetry within each storage node. For all repetitions, we investigate file symmetry for every possible combination of files in each round within each storage node. In the first round ($\ell = 1$) of all κ repetitions, it follows from (9) that, for each $m' \in \mathbb{N}_{2:f}$, the downloaded number of undesired symbols $y_{s,j}^{(m')}$ is equal to $\kappa \mathsf{U}(1) = \kappa^{f-1}$, while for the desired symbols, from (10), it follows that the user requests κ^{f-1} code symbols $y_{s,j}^{(1)}$. In the $(\ell+1)$-th round of all κ repetitions, $\ell \in \mathbb{N}_{f-1}$, arbitrarily choose a combination of files indexed by $\mathcal{M} \subseteq \mathbb{N}_{2:f}$, where $|\mathcal{M}| = \ell$. It follows from (11) that the total number of requested desired symbols for files pertaining to $\{1\} \cup \mathcal{M}$ is equal to

$$\begin{aligned}(\nu - \kappa)\big[\big(\mathsf{U}(\ell) - 1\big) - \mathsf{U}(\ell - 1) + 1\big] \\ = (\nu - \kappa)\kappa^{f-(\ell+1)}(\nu - \kappa)^{\ell-1} = \kappa^{f-(\ell+1)}(\nu - \kappa)^{\ell}.\end{aligned}$$

On the other hand, for the undesired symbols, it follows from (9) that in the $(\ell + 1)$-th round the user requests

$$\begin{aligned}\kappa\big[\big(\mathsf{U}(\ell + 1) - 1\big) - \mathsf{U}(\ell) + 1\big] \\ = \kappa\kappa^{f-(\ell+2)}(\nu - \kappa)^{\ell} = \kappa^{f-(\ell+1)}(\nu - \kappa)^{\ell}\end{aligned}$$

linear sums for a combination of files indexed by $\mathcal{M} \subseteq \mathbb{N}_{2:f}$, $|\mathcal{M}| = \ell + 1$. Thus, in rounds $\mathbb{N}_{f-1}$, an equal number of linear sums for all combinations of files indexed by $\mathcal{M} \subseteq \mathbb{N}_f$ are downloaded. By construction, these are linear sums of unique code symbols pertaining to f files. Thus, symmetry in all $f - 1$ rounds is ensured. In the f-th round, only desired symbols are downloaded. Since each desired symbol is a linear combination of code symbols from all f files, an equal number of linear sums is again downloaded for the combination of files indexed by $\mathbb{N}_f$. Therefore, symmetry within each node and in each round is ensured.

The $\beta \times k$ file $\boldsymbol{X}^{(1)}$ can be reliably decoded. In the first round ($\ell = 1$) of all κ repetitions, $\forall\, s \in \mathbb{N}_{\kappa^{f-1}}$, the user has downloaded the matrix

$$\begin{pmatrix} y^{(1)}_{\kappa^{f-1}(a_{1,1}-1)+s,1} & \cdots & y^{(1)}_{\kappa^{f-1}(a_{1,n}-1)+s,n} \\ \vdots & \cdots & \vdots \\ y^{(1)}_{\kappa^{f-1}(a_{\kappa,1}-1)+s,1} & \cdots & y^{(1)}_{\kappa^{f-1}(a_{\kappa,n}-1)+s,n} \end{pmatrix}$$

of code symbols. Given an $a \in \mathbb{N}_\nu$, recalling Definitions 11 and 12, it follows that for each $s \in \mathbb{N}_{\kappa^{f-1}}$, the coordinate set $\mathcal{S}\big(\kappa^{f-1}(a-1)+s \big| \kappa^{f-1}(\boldsymbol{A}_{\kappa\times n} - \boldsymbol{1}_{\kappa\times n}) + s\boldsymbol{1}_{\kappa\times n}\big)$ contains an information set. Hence, the $(\kappa^{f-1}(a-1)+1)$-th, $\ldots$, $(\kappa^{f-1}(a-1)+\kappa^{f-1})$-th stripes are recovered. Since $a_{i,j} \in \mathbb{N}_\nu$, we know until now that the user has obtained the 1-st, 2-nd, $\ldots$, $(\kappa^{f-1}(\nu-1)+\kappa^{f-1})$-th stripes. Note that $\kappa^{f-1}(\nu-1)+\kappa^{f-1} = \mathsf{D}(0)\nu$. Moreover, owing to (12), in the $(\ell' = \ell+1)$-th round of all κ repetitions with $\ell \in \mathbb{N}_{f-1}$, $\forall\, s \in \mathbb{N}_{\mathsf{D}(\ell-1):(\mathsf{D}(\ell)-1)}$ the matrices

$$\begin{pmatrix} y^{(1)}_{s\cdot\nu+a_{1,1},1} & \cdots & y^{(1)}_{s\cdot\nu+a_{1,n},n} \\ \vdots & \cdots & \vdots \\ y^{(1)}_{s\cdot\nu+a_{\kappa,1},1} & \cdots & y^{(1)}_{s\cdot\nu+a_{\kappa,n},n} \end{pmatrix}$$

of code symbols are downloaded. Similarly, fix an $s \in \mathbb{N}_{\mathsf{D}(\ell-1):(\mathsf{D}(\ell)-1)}$. Then, $\forall\, a \in \mathbb{N}_\nu$, the coordinate set $\mathcal{S}\big(s\nu + a \big| s\nu\boldsymbol{1}_{\kappa\times n} + \boldsymbol{A}_{\kappa\times n}\big)$ must contain an information set, and the user can recover the $(s\nu+1)$-th, $\ldots$, $(s\nu+\nu)$-th stripes. Observe that in the last $(\ell' = (f-1)+1)$-th round, the row index of the last recovered stripe is equal to $(\mathsf{D}(f-1)-1)\nu+\nu$. Hence, the total number of stripes the user has recovered is

$$\begin{aligned} &\big(\mathsf{D}(f-1)-1\big)\nu + \nu \\ &= \left[\sum_{\ell=0}^{f-1} \binom{f-1}{\ell} \kappa^{f-(\ell+1)}(\nu-\kappa)^\ell - 1\right]\nu + \nu \\ &= (\nu^{f-1}-1)\nu + \nu = \nu^f. \end{aligned}$$

This indicates that the user has recovered all ν^f stripes for $\boldsymbol{X}^{(1)}$, and $\boldsymbol{X}^{(1)}$ is in fact reliably reconstructed.

The PIR achievable rate is expressed as (13)**.** According to (9), since there are $\binom{f-1}{\ell}$ combinations of files other than the first file with index $m = 1$, the user has downloaded

$$\begin{aligned} &\kappa\binom{f-1}{\ell}\big[\mathsf{U}(\ell) - 1 - \mathsf{U}(\ell-1) + 1\big] \\ &= \kappa\binom{f-1}{\ell}\kappa^{f-(\ell+1)}(\nu-\kappa)^{\ell-1} \\ &= \binom{f-1}{\ell}\kappa^{f-\ell}(\nu-\kappa)^{\ell-1} \end{aligned}$$

undesired symbols from each storage node in the ℓ-th round, $\ell \in \mathbb{N}_{f-1}$, of each repetition. Moreover, from (10) and (11), the user has downloaded κ^{f-1} desired symbols from each storage node in round $\ell = 1$ of each repetition, and

$$\mathsf{D}(\ell) - 1 - \mathsf{D}(\ell-1) + 1 = \binom{f-1}{\ell}\kappa^{f-(\ell+1)}(\nu-\kappa)^\ell$$

extra desired symbols from each storage node in the $(\ell+1)$-th round, $\ell \in \mathbb{N}_{f-1}$, of each repetition. In summary, the total download cost for Protocol 1 using $\boldsymbol{\Lambda}_{\kappa,\nu}(\mathcal{C})$ is equal to

$$\begin{aligned} nd &= \text{total number of undesired symbols} \\ &\quad + \text{total number of desired symbols} \\ &= \kappa n \sum_{\ell=1}^{f-1} \binom{f-1}{\ell} \kappa^{f-\ell}(\nu-\kappa)^{\ell-1} \\ &\quad + \kappa n \sum_{\ell=0}^{f-1} \binom{f-1}{\ell} \kappa^{f-(\ell+1)}(\nu-\kappa)^{\ell} \\ &= \kappa n \left[\frac{\kappa}{\nu-\kappa} \sum_{\ell=1}^{f-1} \binom{f-1}{\ell} \kappa^{f-(\ell+1)}(\nu-\kappa)^{\ell} \right. \\ &\qquad \left. + \sum_{\ell=0}^{f-1} \binom{f-1}{\ell} \kappa^{f-(\ell+1)}(\nu-\kappa)^{\ell}\right] \\ &= \kappa n \left[\frac{\kappa}{\nu-\kappa}(\nu^{f-1} - \kappa^{f-1}) + \nu^{f-1}\right] \\ &= \frac{\kappa n}{\nu-\kappa}\left[\kappa\nu^{f-1} - \kappa^f + \nu^f - \kappa\nu^{f-1}\right] \\ &= \frac{\kappa n}{\nu-\kappa}\left[\nu^f - \kappa^f\right]. \end{aligned}$$

Therefore, the PIR achievable rate $\mathsf{R}(\mathcal{C})$ is given by

$$\mathsf{R}(\mathcal{C}) = \frac{\beta k}{nd} = \frac{\nu^f k}{\frac{\kappa n}{\nu-\kappa}\left[\nu^f - \kappa^f\right]} = \frac{\frac{(\nu-\kappa)k}{\kappa n}}{\left[1 - \left(\frac{\kappa}{\nu}\right)^f\right]}.$$

Appendix C
Proof of Lemma 3

By setting $\kappa = k$ and using Definition 11, we will prove the existence of $\boldsymbol{\Lambda}_{k,\nu}$ with $\nu = k + \min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1)$. In fact, given an $[n, k, d^{\mathcal{C}}_{\mathsf{min}}]$ code $\mathcal{C}$, observe that for an interference matrix $\boldsymbol{A}_{k\times n}$ derived from a valid $\boldsymbol{\Lambda}_{k,\nu}$, $\mathcal{S}(a|\boldsymbol{A}_{k\times n})$ must contain an information set $\forall\, a \in \mathbb{N}_\nu$. We first choose $\Gamma = \min(k, d^{\mathcal{C}}_{\mathsf{min}} - 1)$ information sets of $\mathcal{C}$. Note that since every code contains at least one information set, one can always arbitrarily choose Γ information sets even if some of them are repeatedly chosen. Let us denote the selected information sets by $\mathcal{I}_i$, $i \in \mathbb{N}_\Gamma$, and start to construct the corresponding matrix $\boldsymbol{A}_{k\times n}$ with

$$a_{i,j} = k + i, \text{ if } j \in \mathcal{I}_i,\ i \in \mathbb{N}_\Gamma. \tag{37}$$

In this way, $k\Gamma$ entries of $\boldsymbol{A}_{k\times n}$ are constructed. Next, denote the remaining nonconstructed entries in each column of $\boldsymbol{A}_{k\times n}$ by

$$\mathcal{A}_j \triangleq \left\{a_{i_1^{(j)},j}, \ldots, a_{i_{s(j)}^{(j)},j}\right\}, \quad j \in \mathbb{N}_n,$$

where $s(j) \le k$ is the total number of nonconstructed entries in each column. Hence, there are in total $kn - k\Gamma = k(n-\Gamma)$ nonconstructed entries as follows,

$$\left\{a_{i_1^{(1)},1}, \ldots, a_{i_{s(1)}^{(1)},1}, \ldots, a_{i_1^{(n)},n}, \ldots, a_{i_{s(n)}^{(n)},n}\right\}. \tag{38}$$

If we consecutively assign $1, \ldots, k$ to the entries of $\boldsymbol{A}_{k\times n}$ in (38) and repeat this process $n - \Gamma$ times, the remaining $k(n-\Gamma)$ entries of $\boldsymbol{A}_{k\times n}$ will certainly be constructed. Note

that since we consecutively assign values of $\mathbb{N}_k$ and the largest number of empty entries of each column of $\boldsymbol{A}_{k\times n}$ is k, it is impossible to have repeated values of $\mathbb{N}_k$ in each column of the constructed $\boldsymbol{A}_{k\times n}$. From (37) and (38), it can be seen that each $a \in \mathbb{N}_k$ occurs in $n-\Gamma$ columns of $\boldsymbol{A}_{k\times n}$. From Proposition 1, we can then say that the set $\mathcal{S}(a|\boldsymbol{A}_{k\times n})$ of cardinality $n-\Gamma \geq n-(d^{\mathcal{C}}_{\mathsf{min}}-1)$ contains an information set. For the remaining $a \in \mathbb{N}_{k+1:k+\Gamma}$, (37) ensures that $\mathcal{S}(a|\boldsymbol{A}_{k\times n})$ contains an information set. Thus, this procedure will result in a valid PIR interference matrix $\boldsymbol{A}_{k\times n}$. The proof is then completed, since we can construct a PIR achievable rate matrix $\boldsymbol{\Lambda}_{k,k+\Gamma}$ from $\boldsymbol{A}_{k\times n}$.

Appendix D
Proof of Theorem 2

Consider the i-th subresponse of each response $\boldsymbol{r}_l$. Out of the n subresponses generated from the n storage nodes, there are Γ subresponses originating from a subset of nodes $\mathcal{J} \subset \mathbb{N}_n$, $|\mathcal{J}|=\Gamma$, of the form

$$r_{l,i} = Y_l + c^{(m)}_{s,l}, \quad \forall\, l \in \mathcal{J},\, s \in \mathbb{N}_\beta. \tag{39}$$

Y_l is referred to as *code interference symbol*. Considering $\boldsymbol{G}^{\mathcal{C}} = (g_{i',l})$, for $l \in \mathbb{N}_n$, each code symbol and code interference symbol have the form

$$c^{(m)}_{s,l} = \sum_{i'=1}^{k} g_{i',l} x^{(m)}_{s,i'}, \tag{40}$$

$$Y_l = \sum_{i'=1}^{k} g_{i',l} I_{(i-1)k+i'}, \tag{41}$$

where $x^{(m)}_{s,i'}$ is an information symbol of $\mathcal{C}$, and $I_{(i-1)k+i'} = \sum_{m=1}^{f}\sum_{i''=(m-1)\beta+1}^{m\beta} u_{i,i''} x^{(m)}_{i''-(m-1)\beta,i'}$ is an interference symbol. To obtain Γ code symbols from (39), the user requires the knowledge of the code interference symbols Y_l. This is obtained from the remaining $n-\Gamma$ subresponses of the nodes in $\overline{\mathcal{J}} \triangleq \mathbb{N}_n \setminus \mathcal{J}$, which are

$$r_{l,i} = Y_l, \quad \forall\, l \in \overline{\mathcal{J}}. \tag{42}$$

From (40) and (41) we can observe that the interference symbols Y_l have the same form as the code symbols of $\mathcal{C}$. Since there are Γ unknowns, solving (42) resembles ML decoding of the code $\mathcal{C}$. (42) is a full rank system in the unknowns $I_{(i-1)k+1},\ldots,I_{ik}$ (from the third requirement $\mathsf{C3}$ of $\hat{\boldsymbol{E}}$ in Section V) in $\mathrm{GF}(q^\ell)$. Hence, knowing the interference symbols allows the recovery of Γ *unique* (from the first requirement $\mathsf{C1}$ for $\hat{\boldsymbol{E}}$ in Section V) code symbols from the i-th subquery as the user has the knowledge of Y_l, $l \in \mathcal{J}$. In a similar way, from all subqueries, the user obtains $d\Gamma = \beta k$ *unique* code symbols pertaining to file $\boldsymbol{X}^{(m)}$. These βk code symbols are part of β information sets (from the second requirement $\mathsf{C2}$ of $\hat{\boldsymbol{E}}$ in Section V and (18)). Furthermore, since each information set is implicitly linked to a unique stripe of the requested file and $s \in \mathbb{N}_\beta$ (see (39)) is selected (without repetition) from $\mathcal{F}_l$ (see (18)), k code symbols from each stripe are obtained, and the user can recover the whole file $\boldsymbol{X}^{(m)}$, from which it follows that $\mathsf{H}\big(\boldsymbol{X}^{(m)}|\boldsymbol{r}_1,\ldots,\boldsymbol{r}_n\big)=0$.

Appendix E
Proof of Lemma 5

We prove the inequality by using the well known Sylvester's rank inequality:[8] If $\boldsymbol{U}$ is an $s\times k$ matrix and $\boldsymbol{G}$ is a matrix of size $k\times n$, then

$$\mathrm{rank}\,(\boldsymbol{U}\boldsymbol{G}) \geq \mathrm{rank}\,(\boldsymbol{U}) + \mathrm{rank}\,(\boldsymbol{G}) - k.$$

Let $\mathcal{C}$ be an $[n,k]$ code with generator matrix $\boldsymbol{G}$. Given an arbitrary information set $\mathcal{I}$, $\boldsymbol{G}|_{\mathcal{I}}$ is by definition invertible (see Definition 1). We next choose an arbitrary subcode $\mathcal{D}\subseteq\mathcal{C}$ of dimension s that can be generated by $\boldsymbol{U}\boldsymbol{G}$ for some $s\times k$ matrix $\boldsymbol{U}$ of rank s.

Applying Sylvester's rank inequality, we have

$$\mathrm{rank}\,(\boldsymbol{U}(\boldsymbol{G}|_{\mathcal{I}})) \geq s+k-k = s.$$

Because each basis vector of the space $\boldsymbol{U}(\boldsymbol{G}|_{\mathcal{I}})$ must at least contain one nonzero component, this leads to

$$|\mathcal{I}\cap\chi(\mathcal{D})| = |\chi(\mathcal{D}|_{\mathcal{I}})| = |\chi(\boldsymbol{U}(\boldsymbol{G}|_{\mathcal{I}}))| \geq s,$$

where $\chi(\mathcal{D})$ is the support of $\mathcal{D}$ (see Definition 2).

Appendix F
Proof of Theorem 5

The proof is a two-step procedure. First, we prove that all rows in $\boldsymbol{E}$ after Step a) are correctable by $\mathcal{C}$. Secondly, we prove that the swaps in certain rows in Step b) ensure that the resulting rows are correctable erasure patterns. We start by proving two key lemmas (Lemmas 7 and 8 below), which will form the basis of the overall proof of the theorem.

Lemma 7. *Let $\mathcal{C}$ be an $[n,k]$ distance-optimal (r,δ) information locality code consisting of L_{c} local codes and with parity-check matrix as in* (2)*. Additionally, it adheres to the condition in* (24)*. Then, $\mathcal{C}$ can simultaneously correct $\delta-1+\nu_j$ erasures, $\nu_j\geq 0$, in each local code $\mathcal{C}|_{\mathcal{S}_j}$ provided that the number of global parities available is at least $\nu_1+\cdots+\nu_{L_{\mathsf{c}}}$.*

Proof: We begin by defining $\boldsymbol{H}^{\mathcal{C}}|^{\mathcal{I}}_{\mathcal{J}}$ as the submatrix of $\boldsymbol{H}^{\mathcal{C}}$ restricted in columns by the set $\mathcal{J}$ and in rows by the set $\mathcal{I}$. For $j\in\mathbb{N}_{L_{\mathsf{c}}}$, consider the j-th local code. Let $\mathcal{E}_j$ denote the set of coordinates that are erased in the j-th local code, where $|\mathcal{E}_j| = \delta-1+\nu_j$. Let

$$\mathcal{R}_j = \{(\delta-1)(j-1)+1,\ldots,(\delta-1)j\}\cup\mathcal{A}_j$$

be a set of rows of $\boldsymbol{H}^{\mathcal{C}}$ of cardinality $|\mathcal{R}_j| = |\mathcal{E}_j|$, where $\mathcal{A}_j \subset \mathbb{N}_{L_{\mathsf{c}}(\delta-1)+1:(n-k)}$, $|\mathcal{A}_j|=\nu_j$, is a set of rows of $\boldsymbol{H}^{\mathcal{C}}$ (which correspond to parity-check equations of the available global parities). In order to prove the lemma one needs to prove that

$$\mathrm{rank}\Big(\boldsymbol{H}^{\mathcal{C}}\big|^{\cup_j\mathcal{R}_j}_{\cup_j\mathcal{E}_j}\Big) = \sum_{j=1}^{L_{\mathsf{c}}}(\delta-1+\nu_j). \tag{43}$$

[8]The proof of this inequality is available in the literature on linear algebra, so here we omit the proof.

For each $j \in \mathbb{N}_{L_\mathsf{c}}$ and $j' \neq j$, assume that there exists a set $\mathcal{A}_{j'} \subset \mathbb{N}_{L_\mathsf{c}(\delta-1)+1:(n-k)}$ such that $\mathcal{A}_j \cap \mathcal{A}_{j'} = \emptyset$. Then, it follows that $\mathcal{R}_j \cap \mathcal{R}_{j'} = \emptyset$, and since $\mathcal{E}_j \cap \mathcal{E}_{j'} = \emptyset$,

$$\operatorname{rank}\Big(\boldsymbol{H}^{\mathcal{C}}\big|_{\cup_j \mathcal{E}_j}^{\cup_i \mathcal{R}_j}\Big) = \sum_{j=1}^{L_\mathsf{c}} \operatorname{rank}\big(\boldsymbol{H}^{\mathcal{C}}\big|_{\mathcal{E}_j}^{\mathcal{R}_j}\big).$$

Thus, to show (43) it is sufficient to show that

$$\operatorname{rank}\big(\boldsymbol{H}^{\mathcal{C}}\big|_{\mathcal{E}_j}^{\mathcal{R}_j}\big) = \delta - 1 + \nu_j \tag{44}$$

for all $j \in \mathbb{N}_{L_\mathsf{c}}$.

To show this, consider now the $[n', k]$ MDS code $\mathcal{C}'$ whose parity-check matrix is given by $\boldsymbol{H}^{\mathsf{MDS}}$ in (24). Let $\mathcal{S}'_j \subset \mathbb{N}_{n'}$ denote a set of coordinates of $\mathcal{C}'$ of cardinality $\big|\mathcal{S}'_j\big| = k+\delta-1+\nu_j$. More specifically,

$$\mathcal{S}'_j = \{1, \ldots, k\} \cup \{k+1, \ldots, k+\delta-1\} \cup \mathcal{B}_j,$$

where $\mathcal{B}_j = \{a - L_\mathsf{c}(\delta-1) + (\delta-1) + k \colon a \in \mathcal{A}_j\} \subset \mathbb{N}_{\delta+k:n'}$. In other words, the set $\mathcal{S}'_j$ consists of k systematic coordinates and $\delta - 1 + \nu_j$ parity coordinates of $\mathcal{C}'$. The punctured code $\mathcal{C}'_j = \mathcal{C}'|_{\mathcal{S}'_j}$ is defined by a parity-check matrix $\boldsymbol{H}^{\mathcal{C}'_j}$ of dimensions $(\delta-1+\nu_j) \times (k+\delta-1+\nu_j)$ that is a submatrix of $\boldsymbol{H}^{\mathsf{MDS}}$. Since the punctured code of an MDS code is also an MDS code [45], $\mathcal{C}'_j$ has minimum Hamming distance $d_{\mathsf{min}}^{\mathcal{C}'_j} = \delta + \nu_j = \delta + |\mathcal{A}_j|$. Note that for some column index set $\mathcal{J} \subset \mathbb{N}_{k+\delta-1+\nu_j}$, $|\mathcal{J}| = |\mathcal{E}_j|$, one can build $\boldsymbol{H}^{\mathcal{C}}\big|_{\mathcal{E}_j}^{\mathcal{R}_j} = \boldsymbol{H}^{\mathcal{C}'_j}|_{\mathcal{J}}$. From the MDS property, it follows that

$$\operatorname{rank}\big(\boldsymbol{H}^{\mathcal{C}}\big|_{\mathcal{E}_j}^{\mathcal{R}_j}\big) = \operatorname{rank}\big(\boldsymbol{H}^{\mathcal{C}'_j}|_{\mathcal{J}}\big) = \delta - 1 + \nu_j.$$

Finally, if the total number of global parities is at least $\sum_{j=1}^{L_\mathsf{c}} \nu_j$, we can assign to the set $\mathcal{A}_j$, $j \in \mathbb{N}_{L_\mathsf{c}}$, a set of ν_j rows of $\boldsymbol{H}^{\mathcal{C}}$ corresponding to global parity-checks such that the sets $\mathcal{A}_j$ are all disjoint, hence (44) holds for all $j \in \mathbb{N}_{L_\mathsf{c}}$, and (43) follows, which completes the proof. ∎

Lemma 8. *Consider an erasure pattern $\boldsymbol{e}$ of length n of the form*

$$\boldsymbol{e} = (e_1, \ldots, e_n) = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_L, \boldsymbol{e}_{L+1}),$$

where the subvectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_L$ are all of length $n_\mathsf{c} = r+\delta-1$ and $\boldsymbol{e}_{L+1}$ is of length $\bar{r} = n \bmod n_\mathsf{c}$. Let $\chi(\boldsymbol{e}_j)$, $j \in \mathbb{N}_{L+1}$, be the support of $\boldsymbol{e}_j$ and $t = (n-k) \bmod L$. If $|\chi(\boldsymbol{e}_1)| = \cdots = |\chi(\boldsymbol{e}_t)| = m_1$, $|\chi(\boldsymbol{e}_{t+1})| = \cdots = |\chi(\boldsymbol{e}_L)| = m$, and $|\chi(\boldsymbol{e}_{L+1})| = 0$, where $m = \left\lfloor \frac{n-k}{L} \right\rfloor$ and $m_1 = m+1$, then $\boldsymbol{e}$ is correctable by $\mathcal{C}$.

Proof: The erasure pattern $\boldsymbol{e}$ is divided into $L+1$ partitions represented by $\boldsymbol{e}_j = (e_{n_\mathsf{c}(j-1)+1}, \ldots, e_{n_\mathsf{c}j})$, $j \in \mathbb{N}_L$, and $\boldsymbol{e}_{L+1} = (e_{n_\mathsf{c}L+1}, \ldots, e_n)$, where $\boldsymbol{e}_j$, $j \in \mathbb{N}_{L_\mathsf{c}}$, corresponds to the coordinates of the j-th local code, and $\boldsymbol{e}_{L_\mathsf{c}+1}, \ldots, \boldsymbol{e}_{L+1}$ correspond to the coordinates of the global parities of $\mathcal{C}$.

The set $\chi(\boldsymbol{e}_j)$, $j \in \mathbb{N}_{L+1}$, is the set of coordinates erased from the j-th partition, and we construct the erasure patterns $\boldsymbol{e}_j$, $j \in \mathbb{N}_L$, such that $|\chi(\boldsymbol{e}_j)| = \delta - 1 + \nu_j$ with

$$\nu_j = \begin{cases} m_1 - (\delta-1) & \text{if } j \in \mathbb{N}_t, \\ m - (\delta-1) & \text{if } j \in \mathbb{N}_{t+1:L}, \end{cases}$$

where $t = (n-k) \bmod L$, and let $\chi(\boldsymbol{e}_{L+1}) = \emptyset$. In other words, we construct the erasure patterns such that the erasures are distributed as equally as possible across the first L partitions.

From Definition 7, it follows that $n - k \geq (\delta-1)L_\mathsf{c} + (L - L_\mathsf{c})(r + \delta - 1) \geq L(\delta-1)$ (where the last inequality follows from $L \geq L_\mathsf{c}$), hence $\delta - 1$ is an integer satisfying the inequality $L(\delta-1) \leq n-k$, and subsequently $\delta - 1 \leq \frac{n-k}{L}$. The integer m is the largest integer such that $m \leq \frac{n-k}{L}$. Therefore, $\delta - 1 \leq m$. To show that $\boldsymbol{e}$ is correctable it is enough to show that the erasures in the L_c local codes can be corrected, since in this case we have a nonerased information set for $\mathcal{C}$, which allows to correct the remaining erasures in $\boldsymbol{e}$.

From Lemma 7, to correct $\delta - 1 + \nu_j$ erasures in the j-th local code for all $j \in \mathbb{N}_{L_\mathsf{c}}$, the number of global parities available, $\gamma_{\mathsf{tot}} + \bar{r}$, must be

$$\gamma_{\mathsf{tot}} + \bar{r} \geq \sum_{j=1}^{L_\mathsf{c}} \nu_j = \begin{cases} m_1 t + m(L_\mathsf{c} - t) - L_\mathsf{c}(\delta-1) & \text{if } t \leq L_\mathsf{c}, \\ m_1 L_\mathsf{c} - L_\mathsf{c}(\delta-1) & \text{if } t > L_\mathsf{c}, \end{cases} \tag{45}$$

where γ_{tot} is the number of global parities available in the $(L_\mathsf{c}+1)$-th, $\ldots$, L-th partitions and $\bar{r} = n - n_\mathsf{c}L$ is the number of global parities in the $(L+1)$-th partition. By counting the number of global parities not erased in $L - L_\mathsf{c}$ partitions, we get

$$\gamma_{\mathsf{tot}} = \begin{cases} (n_\mathsf{c} - m)(L - L_\mathsf{c}) & \text{if } t \leq L_\mathsf{c}, \\ (n_\mathsf{c} - m_1)(t - L_\mathsf{c}) + (n_\mathsf{c} - m)(L - t) & \text{if } t > L_\mathsf{c}. \end{cases} \tag{46}$$

By substituting (46) into (45), we get (after performing some simple arithmetic) the condition

$$n - k - mL \geq t,$$

which is valid for both cases of t ($t \leq L_\mathsf{c}$ and $t > L_\mathsf{c}$). By definition of t and m, the above inequality is met with equality, and it follows that $\boldsymbol{e}$ is a correctable erasure pattern. ∎

A. Proof of Step a)

Let $\tilde{\boldsymbol{E}}$, $\boldsymbol{W}$, $\boldsymbol{Z}$, and $\boldsymbol{O}$ be submatrices of $\boldsymbol{E}$ as shown in (25). We begin the proof by proving that each of the $n_\mathsf{c}L$ rows of the matrix $\big(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z}\big)$ is a correctable erasure pattern, where $\tilde{\boldsymbol{E}}$ is defined in (26). This is proved by induction on the row partitions of $\big(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z}\big)$.

Base Case. Consider the first row partition of $\big(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z}\big)$, given by

$$\begin{pmatrix} \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \cdots & \boldsymbol{\pi}_L & \boldsymbol{0}_{n_\mathsf{c} \times \bar{r}} \end{pmatrix}.$$

For each row vector $\boldsymbol{e}_i^{(1)}$, $i \in \mathbb{N}_{n_\mathsf{c}}$, in this row partition, where the subscript i indicates the row index and the superscript the row partition, consider the subvectors $\boldsymbol{e}_{i,1}^{(1)}, \ldots, \boldsymbol{e}_{i,L}^{(1)}$. From Step a) in Section VI-C, for all $i \in \mathbb{N}_{n_\mathsf{c}}$, the j-th subvectors $\boldsymbol{e}_{i,j}^{(1)}$ have support of cardinality $|\chi(\boldsymbol{e}_{i,j}^{(1)})| = m_1$ for all $j \in \mathbb{N}_t$, where $t = (n-k) \bmod L$, $|\chi(\boldsymbol{e}_{i,j}^{(1)})| = m$ for $j \in \mathbb{N}_{t+1:L}$, and $|\chi(\boldsymbol{e}_{i,L+1}^{(1)})| = 0$. Thus, the vectors $\boldsymbol{e}_i^{(1)}$ in the first row

partition of $\left(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z}\right)$ have the same structure as the erasure pattern $\boldsymbol{e}$ from Lemma 8 and are therefore erasure patterns that are correctable by $\mathcal{C}$. Note that the number of global parities available in the $(L_{\mathsf{c}}+1)$-th, $\ldots$, L-th subvectors of vector $\boldsymbol{e}_i^{(1)}$, $\gamma_{\mathsf{tot}}^{(1)}$, is $\gamma_{\mathsf{tot}}^{(1)} = \gamma_{\mathsf{tot}}$, hence $\gamma_{\mathsf{tot}}^{(1)} + \bar{r} = \gamma_{\mathsf{tot}} + \bar{r} \geq \sum_{j=1}^{L_{\mathsf{c}}} \nu_j$ and from the proof of Lemma 8 the error pattern $\boldsymbol{e}_i^{(1)}$ is correctable.

Inductive Step. Assume that the vectors $\boldsymbol{e}_i^{(l)}$, $i \in \mathbb{N}_{n_{\mathsf{c}}}$, in the l-th row partition of $\left(\tilde{\boldsymbol{E}}|\boldsymbol{Z}\right)$ are correctable by $\mathcal{C}$ and that each local code $\mathcal{C}|_{\mathcal{S}_j}$ can correct $\delta - 1 + \nu_j^{(l)}$ erasures, $j \in \mathbb{N}_{L_{\mathsf{c}}}$. The row vectors are taken from the matrix

$$\left(\boldsymbol{\pi}_{\sigma^{l-1}(1)} \quad \boldsymbol{\pi}_{\sigma^{l-1}(2)} \quad \cdots \quad \boldsymbol{\pi}_{\sigma^{l-1}(L)} \quad \boldsymbol{0}_{n_{\mathsf{c}}\times\bar{r}}\right),$$

where $\sigma \triangleq (L\,(L-1)\cdots 1)$ denotes a *cycle* whose mapping is $L \mapsto (L-1) \mapsto \cdots \mapsto 1 \mapsto L$. The $(L+1)$-th subvectors satisfy $|\chi(\boldsymbol{e}_{i,L+1}^{(l)})| = 0$. From Lemma 7, the underlying characteristic of the vectors $\boldsymbol{e}_i^{(l)}$ is that they are correctable erasure patterns if the number of global parities not erased in $\boldsymbol{e}_i^{(l)}$, $\gamma_{\mathsf{tot}}^{(l)} + \bar{r}$, is larger than or equal to $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l)}$.
In the $(l+1)$-th row partition of $\left(\tilde{\boldsymbol{E}}|\boldsymbol{Z}\right)$, the n_{c} rows have the form

$$\left(\boldsymbol{\pi}_{\sigma^{l}(1)} \quad \boldsymbol{\pi}_{\sigma^{l}(2)} \quad \cdots \quad \boldsymbol{\pi}_{\sigma^{l}(L)} \quad \boldsymbol{0}_{n_{\mathsf{c}}\times\bar{r}}\right).$$

Due to the cyclic shifts, for $j \in \mathbb{N}_L$, all the j-th subvectors of the vectors $\boldsymbol{e}_i^{(l+1)}$ in row partition $l+1$, $l \in \mathbb{N}_{L-1}$, have support size $|\chi(\boldsymbol{e}_{i,j}^{(l+1)})| = |\chi(\boldsymbol{e}_{i,\sigma^{-1}(j)}^{(l)})|$. Thus, there exist two indices $j', j'' \in \mathbb{N}_L$, $j' \neq j''$, such that

$$\begin{aligned} |\chi(\boldsymbol{e}_{i,j'}^{(l+1)})| - |\chi(\boldsymbol{e}_{i,j'}^{(l)})| &= |\chi(\boldsymbol{e}_{i,j''}^{(l)})| - |\chi(\boldsymbol{e}_{i,j''}^{(l+1)})|, \\ |\chi(\boldsymbol{e}_{i,j}^{(l+1)})| &= |\chi(\boldsymbol{e}_{i,j}^{(l)})|, \quad \forall\, j \in \mathbb{N}_L \setminus \{j', j''\}. \end{aligned} \tag{47}$$

One can see that there are at most 4 (depending on t and L_{c}) choices for the pair (j', j'') as follows.

Case 1. $j', j'' \in \mathbb{N}_{L_{\mathsf{c}}}$: From (47), it follows that $\nu_{j'}^{(l+1)} - \nu_{j'}^{(l)} = \nu_{j''}^{(l)} - \nu_{j''}^{(l+1)}$, $\nu_j^{(l+1)} = \nu_j^{(l)}$, and $\gamma_{\mathsf{tot}}^{(l)} = \gamma_{\mathsf{tot}}^{(l+1)}$. Thus, we have $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)} = \sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l)} = \gamma_{\mathsf{tot}}^{(l)} + \bar{r} = \gamma_{\mathsf{tot}}^{(l+1)} + \bar{r}$.

Case 2. $j', j'' \in \mathbb{N}_{L_{\mathsf{c}}+1:L}$: From (47), it follows that $\gamma_{\mathsf{tot}}^{(l+1)} = \gamma_{\mathsf{tot}}^{(l)}$ and $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)} = \sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l)}$. Therefore, $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)} = \gamma_{\mathsf{tot}}^{(l+1)} + \bar{r}$.

Case 3. $j' \in \mathbb{N}_{L_{\mathsf{c}}}$, $j'' \in \mathbb{N}_{L_{\mathsf{c}}+1:L}$: From (47), it follows that $\nu_{j'}^{(l+1)} - \nu_{j'}^{(l)} = \gamma_{\mathsf{tot}}^{(l+1)} - \gamma_{\mathsf{tot}}^{(l)}$. Moreover, it can be seen that $\sum_{j\neq j', j\in\mathbb{N}_{L_{\mathsf{c}}}} \nu_j^{(l+1)} = \sum_{j\neq j', j\in\mathbb{N}_{L_{\mathsf{c}}}} \nu_j^{(l)}$. Hence, we have

$$\begin{aligned} \sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)} &= \sum_{j\neq j', j\in\mathbb{N}_{L_{\mathsf{c}}}} \nu_j^{(l+1)} + \nu_{j'}^{(l+1)} \\ &= \sum_{j\neq j', j\in\mathbb{N}_{L_{\mathsf{c}}}} \nu_j^{(l+1)} + (\nu_{j'}^{(l+1)} - \nu_{j'}^{(l)}) + \nu_{j'}^{(l)} \\ &= \sum_{j\neq j', j\in\mathbb{N}_{L_{\mathsf{c}}}} \nu_j^{(l)} + (\gamma_{\mathsf{tot}}^{(l+1)} - \gamma_{\mathsf{tot}}^{(l)}) + \nu_{j'}^{(l)} \\ &= \sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l)} + (\gamma_{\mathsf{tot}}^{(l+1)} - \gamma_{\mathsf{tot}}^{(l)}) \\ &\overset{(b)}{=} \gamma_{\mathsf{tot}}^{(l)} + \bar{r} + (\gamma_{\mathsf{tot}}^{(l+1)} - \gamma_{\mathsf{tot}}^{(l)}) \\ &= \gamma_{\mathsf{tot}}^{(l+1)} + \bar{r}, \end{aligned}$$

where (b) holds since $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l)} = \gamma_{\mathsf{tot}}^{(l)} + \bar{r}$.

Case 4. $j' \in \mathbb{N}_{L_{\mathsf{c}}+1:L}$, $j'' \in \mathbb{N}_{L_{\mathsf{c}}}$: Following an argumentation similar to Case 3, we have $\sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)} = \gamma_{\mathsf{tot}}^{(l+1)} + \bar{r}$.
In each of the above cases we see that the condition $\gamma_{\mathsf{tot}} + \bar{r} \geq \sum_{j=1}^{L_{\mathsf{c}}} \nu_j^{(l+1)}$ is satisfied (with equality). From the proof of Lemma 8, the n_{c} rows in the $(l+1)$-th row partition of $\left(\tilde{\boldsymbol{E}}|\boldsymbol{Z}\right)$ are correctable by $\mathcal{C}$, which completes the inductive step.

The rows of $(\boldsymbol{W} \mid \boldsymbol{O})$ as shown in Step a) in Section VI-C have support corresponding to only the parity symbols of $\mathcal{C}$. Thus, these rows are all correctable by $\mathcal{C}$, and it follows from the above arguments that each row of $\boldsymbol{E}$ is an erasure pattern that is correctable by $\mathcal{C}$.

B. Proof of Step b)

We now address the second part of the proof. Note that the columns with coordinates in $\mathcal{P}_j$, $j \in \mathbb{N}_L$, have column weight $n - k + \bar{r}$ after Step a). Step b) involves the swapping of one entries from these coordinates with zero entries in the column coordinates of $\boldsymbol{Z}$. The swapping is done to ensure that the column weight of the columns indexed by $\mathcal{P}_j$, $j \in \mathbb{N}_L$, is reduced to $n-k$, while those of the columns of $\boldsymbol{Z}$ are increased to $n - k - \bar{r}$. Since $\boldsymbol{O}$ is an all-one matrix, the columns of $\boldsymbol{E}$ with indices in $\mathcal{P}_{L+1}$ have also weight $n-k$. It is possible to show that such a swapping always exists. Overall, the resulting matrix $\boldsymbol{E}$ is $(n-k)$-column regular. To ensure that the erasure patterns are correctable, we use Lemma 7. For each row,

$$\sum_{j=1}^{L_{\mathsf{c}}} \nu_j \leq \gamma_{\mathsf{tot}} + \gamma_{L+1}, \tag{48}$$

where γ_{L+1} is number of nonerased parity coordinates in column partition $L+1$, must hold. Clearly, if for a certain row of $(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z})$ a one from a column from a column partition in $\mathbb{N}_{L_{\mathsf{c}}+1:L}$ (corresponding to $\tilde{\boldsymbol{E}}$) is swapped with a zero in a column from partition $L+1$ (corresponding to $\boldsymbol{Z}$), then the resulting erasure pattern is still correctable by $\mathcal{C}$ as (48) is still valid. On the other hand, for $j \in \mathbb{N}_{L_{\mathsf{c}}}$, if for a certain row of $(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z})$ a one from the j-th column partition is swapped with a zero in the $(L+1)$-th column partition, then such a row is still a correctable erasure pattern provided that $\nu_j > 0$ before the swap. This is easy to see as the swapping procedure reduces ν_j and γ_{L+1} by one. Thus, (48) is still satisfied. From the aforementioned arguments and the fact that each row of any row partition of $(\tilde{\boldsymbol{E}} \mid \boldsymbol{Z})$ has at most $\bar{r}$ swaps of ones occurring from the set of $\mathbb{N}_L$ column partitions and zeroes from the $(L+1)$-th partition, it follows that the swaps according to Step b) are valid over all $\bar{r}$ iterations (valid in the sense that the resulting erasure patterns are correctable by $\mathcal{C}$) if

$$\sum_{j=1}^{L_{\mathsf{c}}} \nu_j + \sum_{j=L_{\mathsf{c}}+1}^{L} (m - (\delta - 1)) \geq \bar{r}. \tag{49}$$

This is a counting argument, where according to Step b) for each row we restrict swapping ν_j coordinates in the j-th

column partition, $j \in \mathbb{N}_{L_c}$, and $m-(\delta-1)$ coordinates in the column partitions $\mathbb{N}_{L_c+1:L}$ to make sure (following the arguments above) that the resulting erasure pattern after the swap is correctable by $\mathcal{C}$. Using that $\nu_j = \rho_j - (\delta-1)$ and $t = n-k-mL$, it can be shown that the left hand side of (49) can be lowerbounded by $n-k-L(\delta-1)$ when $t \leq L_c$. Setting $n = \bar{r}+L(r+\delta-1)$ and $k = L_c r$, it follows that (49) reduces to $L \geq L_c$. By definition, this is always true. When $t > L_c$, the left hand side of (49) is equal to $n-k-L(\delta-1)+L_c-t$, and it can be shown that this is always larger than or equal to $\bar{r}$, since $t \leq L$ (details omitted for brevity). It follows that for all $\bar{r}$ iterations and for all row partitions in the systematic procedure in Step b) there exists a valid swap such that the resulting erasure patterns are still correctable by $\mathcal{C}$.

Appendix G
Proof of Theorem 7

To prove the theorem we need the following lemma.

Lemma 9. *Let $\mathcal{C}$ be an $[n = 2n_1, k = k_1+1]$ binary code constructed from an $[n_1, k_1]$ code $\mathcal{U}$ through the $(\mathcal{U} \mid \mathcal{U}+\mathcal{V})$ construction, where $\mathcal{V}$ is an $[n, 1]$ binary repetition code. The generator matrix $\boldsymbol{G}^{\mathcal{C}}$ of $\mathcal{C}$ is given in* (36)*. Let $\bar{\mathcal{C}} = \mathcal{C}$ and $\boldsymbol{G}^{\bar{\mathcal{C}}} = \boldsymbol{G}^{\mathcal{C}}$. Then, the code $\tilde{\mathcal{C}} = \mathcal{C} \circ \bar{\mathcal{C}}$ is a vector space of dimension*

$$\dim(\tilde{\mathcal{C}}) \leq \begin{cases} k_1+n_1+1 & \text{if } n_1-k_1 \leq \binom{k_1}{2}, \\ 2k_1+\binom{k_1}{2}+1 & \text{otherwise}. \end{cases} \tag{50}$$

Proof: From Definition 4, we know that $\tilde{\boldsymbol{c}} \in \tilde{\mathcal{C}}$ has the form $\tilde{\boldsymbol{c}} = (c_1\bar{c}_1, \ldots, c_n\bar{c}_n)$, where $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathcal{C}$ and $\bar{\boldsymbol{c}} = (\bar{c}_1, \ldots, \bar{c}_n) \in \bar{\mathcal{C}}$. Considering $\boldsymbol{G}^{\mathcal{C}} = (g^c_{i,j})$ and $\boldsymbol{G}^{\bar{\mathcal{C}}} = (g^{\bar{c}}_{i,j})$, the vector space $\tilde{\mathcal{C}}$ is spanned by the row space of

$$\boldsymbol{G}^{\tilde{\mathcal{C}}} = \begin{pmatrix} g^c_{1,1}\boldsymbol{g}^{\bar{c}}_1 & g^c_{1,2}\boldsymbol{g}^{\bar{c}}_2 & \cdots & g^c_{1,n}\boldsymbol{g}^{\bar{c}}_n \\ g^c_{2,1}\boldsymbol{g}^{\bar{c}}_1 & g^c_{2,2}\boldsymbol{g}^{\bar{c}}_2 & \cdots & g^c_{2,n}\boldsymbol{g}^{\bar{c}}_n \\ \vdots & \vdots & \cdots & \vdots \\ g^c_{k,1}\boldsymbol{g}^{\bar{c}}_1 & g^c_{k,2}\boldsymbol{g}^{\bar{c}}_2 & \cdots & g^c_{k,n}\boldsymbol{g}^{\bar{c}}_n \end{pmatrix}, \tag{51}$$

where the vector $\boldsymbol{g}^{\bar{c}}_j$, $j \in \mathbb{N}_n$, denotes the j-th column vector of $\boldsymbol{G}^{\bar{\mathcal{C}}}$. The matrix $\boldsymbol{G}^{\tilde{\mathcal{C}}}$ is a matrix consisting of k^2 row vectors (corresponding to codewords of $\tilde{\mathcal{C}}$) of length n. We divide $\boldsymbol{G}^{\tilde{\mathcal{C}}}$ into k submatrices $\boldsymbol{G}^{\tilde{\mathcal{C}}}_i$, where $\boldsymbol{G}^{\tilde{\mathcal{C}}}_i = (g^c_{i,1}\boldsymbol{g}^{\bar{c}}_1 | g^c_{i,2}\boldsymbol{g}^{\bar{c}}_2 | \ldots | g^c_{i,n}\boldsymbol{g}^{\bar{c}}_n)$, $i \in \mathbb{N}_k$ (see (51)). From (36) and since $\boldsymbol{G}^{\mathcal{C}} = \boldsymbol{G}^{\bar{\mathcal{C}}}$, we have $g^c_{k,j} = g^{\bar{c}}_{k,j} = 0$, $j \in \mathbb{N}_{n_1}$, and $g^c_{k,j} = g^{\bar{c}}_{k,j} = 1$, $j \in \mathbb{N}_{n_1+1:n}$. Therefore, (51) can be expanded to

$$\boldsymbol{G}^{\tilde{\mathcal{C}}} = \begin{pmatrix} g^c_{1,1}\begin{pmatrix} g^{\bar{c}}_{1,1} \\ g^{\bar{c}}_{2,1} \\ \vdots \\ g^{\bar{c}}_{k_1,1} \\ 0 \end{pmatrix} & \cdots & g^c_{1,n_1}\begin{pmatrix} g^{\bar{c}}_{1,n_1} \\ g^{\bar{c}}_{2,n_1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1} \\ 0 \end{pmatrix} & g^c_{1,n_1+1}\begin{pmatrix} g^{\bar{c}}_{1,n_1+1} \\ g^{\bar{c}}_{2,n_1+1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1+1} \\ 1 \end{pmatrix} & \cdots & g^c_{1,n}\begin{pmatrix} g^{\bar{c}}_{1,n} \\ g^{\bar{c}}_{2,n} \\ \vdots \\ g^{\bar{c}}_{k_1,n} \\ 1 \end{pmatrix} \\ g^c_{2,1}\begin{pmatrix} g^{\bar{c}}_{1,1} \\ g^{\bar{c}}_{2,1} \\ \vdots \\ g^{\bar{c}}_{k_1,1} \\ 0 \end{pmatrix} & \cdots & g^c_{2,n_1}\begin{pmatrix} g^{\bar{c}}_{1,n_1} \\ g^{\bar{c}}_{2,n_1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1} \\ 0 \end{pmatrix} & g^c_{2,n_1+1}\begin{pmatrix} g^{\bar{c}}_{1,n_1+1} \\ g^{\bar{c}}_{2,n_1+1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1+1} \\ 1 \end{pmatrix} & \cdots & g^c_{2,n}\begin{pmatrix} g^{\bar{c}}_{1,n} \\ g^{\bar{c}}_{2,n} \\ \vdots \\ g^{\bar{c}}_{k_1,n} \\ 1 \end{pmatrix} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 0\begin{pmatrix} g^{\bar{c}}_{1,1} \\ g^{\bar{c}}_{2,1} \\ \vdots \\ g^{\bar{c}}_{k_1,1} \\ 0 \end{pmatrix} & \cdots & 0\begin{pmatrix} g^{\bar{c}}_{1,n_1} \\ g^{\bar{c}}_{2,n_1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1} \\ 0 \end{pmatrix} & 1\begin{pmatrix} g^{\bar{c}}_{1,n_1+1} \\ g^{\bar{c}}_{2,n_1+1} \\ \vdots \\ g^{\bar{c}}_{k_1,n_1+1} \\ 1 \end{pmatrix} & \cdots & 1\begin{pmatrix} g^{\bar{c}}_{1,n} \\ g^{\bar{c}}_{2,n} \\ \vdots \\ g^{\bar{c}}_{k_1,n} \\ 1 \end{pmatrix} \end{pmatrix}. \tag{52}$$

Furthermore, let $\boldsymbol{G}^{\mathcal{U}} = (g^u_{i,j})$ be the generator matrix of $\mathcal{U}$. From (36), we have $g^c_{i,j} = g^{\bar{c}}_{i,j} = g^u_{i,j}$ for $i \in \mathbb{N}_{k_1}$ and $j \in \mathbb{N}_{n_1}$. For $i, j \in \mathbb{N}_k$, we denote the i-th row of the j-th submatrix $\boldsymbol{G}^{\bar{\mathcal{C}}}_j$ as $\boldsymbol{w}^{(j)}_i$. For $i \in \mathbb{N}_{k-1}$, the i-th row of the i-th submatrix $\boldsymbol{G}^{\bar{\mathcal{C}}}_i$ is given as

$$\boldsymbol{w}^{(i)}_i = (g^c_{i,1}g^{\bar{c}}_{i,1}, g^c_{i,2}g^{\bar{c}}_{i,2}, \ldots, g^c_{i,n}g^{\bar{c}}_{i,n}). \tag{53}$$

Since $g^{\bar{c}}_{i,j} = g^c_{i,j} \in \mathrm{GF}(2)$, (53) reduces to $\boldsymbol{w}^{(i)}_i = (g^c_{i,1}, g^c_{i,2}, \ldots, g^c_{i,n})$. Furthermore, from (36) we see that $g^c_{i,j} = g^c_{i,n_1+j} = g^u_{i,j}$, $j \in \mathbb{N}_{n_1}$, $i \in \mathbb{N}_{k_1}$. Therefore, these $k_1 = k-1$ rows form the k_1 basis vectors of the code space $(\mathcal{U}, \mathcal{U})$ and can be arranged in a matrix as

$$\begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} & \boldsymbol{G}^{\mathcal{U}} \end{pmatrix}. \tag{54}$$

The k-th row of $\boldsymbol{G}^{\bar{\mathcal{C}}}_i$ can be written as

$$\begin{aligned} \boldsymbol{w}^{(i)}_k &= (\overbrace{0, 0, \ldots, 0}^{n_1}, g^c_{i,n_1+1}, g^c_{i,n_1+2}, \ldots, g^c_{i,n}) \\ &\overset{(c)}{=} (0, 0, \ldots, 0, g^u_{i,1}, g^u_{i,2}, \ldots, g^u_{i,n_1}), \end{aligned}$$

where (c) results from the structure of $\boldsymbol{G}^{\mathcal{C}}$ in (36). Stacking together the k-th row of all k_1 submatrices $\boldsymbol{G}^{\tilde{\mathcal{C}}}_i$, $i \in \mathbb{N}_{k_1}$, results in the k_1 row vectors

$$\begin{pmatrix} \boldsymbol{0}_{k_1 \times n_1} & \boldsymbol{G}^{\mathcal{U}} \end{pmatrix}. \tag{55}$$

In a similar way, the rows $\boldsymbol{w}^{(k)}_i$, $i \in \mathbb{N}_k$, of the k-th submatrix $\boldsymbol{G}^{\bar{\mathcal{C}}}_k$ result in the matrix

$$\begin{pmatrix} \boldsymbol{0}_{k_1 \times n_1} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{1 \times n_1} & \boldsymbol{1}_{1 \times n_1} \end{pmatrix}. \tag{56}$$

Of the remaining $(k-1)(k-2)$ rows in (52), since $\mathcal{C} = \bar{\mathcal{C}}$, there exist $\binom{k_1}{2}$ distinct rows as follows,

$$\boldsymbol{\Theta} = \begin{pmatrix} g^c_{1,1}g^{\bar{c}}_{2,1} & g^c_{1,2}g^{\bar{c}}_{2,2} & \cdots & g^c_{1,n}g^{\bar{c}}_{2,n} \\ g^c_{1,1}g^{\bar{c}}_{3,1} & g^c_{1,2}g^{\bar{c}}_{3,2} & \cdots & g^c_{1,n}g^{\bar{c}}_{3,n} \\ \vdots & \vdots & \cdots & \vdots \\ g^c_{1,1}g^{\bar{c}}_{k_1,1} & g^c_{1,2}g^{\bar{c}}_{k_1,2} & \cdots & g^c_{1,n}g^{\bar{c}}_{k_1,n} \\ g^c_{2,1}g^{\bar{c}}_{3,1} & g^c_{2,2}g^{\bar{c}}_{3,2} & \cdots & g^c_{2,n}g^{\bar{c}}_{3,n} \\ \vdots & \vdots & \cdots & \vdots \\ g^c_{2,1}g^{\bar{c}}_{k_1,1} & g^c_{2,2}g^{\bar{c}}_{k_1,2} & \cdots & g^c_{2,n}g^{\bar{c}}_{k_1,n} \\ \vdots & \vdots & \cdots & \vdots \\ g^c_{k_1-1,1}g^{\bar{c}}_{k_1,1} & g^c_{k_1-1,2}g^{\bar{c}}_{k_1,2} & \cdots & g^c_{k_1-1,n}g^{\bar{c}}_{k_1,n} \end{pmatrix}.$$

Furthermore, from the construction of $\boldsymbol{G}^{\mathcal{C}}$ in (36), we have $(g^c_{i,1}, \ldots, g^c_{i,n_1}) = (g^c_{i,n_1+1}, \ldots, g^c_{i,n}) = (g^u_{i,1}, \ldots, g^u_{i,n_1})$, $i \in \mathbb{N}_{k_1}$ and because $\bar{\mathcal{C}} = \mathcal{C}$, we have $(g^{\bar{c}}_{i,1}, \ldots, g^{\bar{c}}_{i,n_1}) = (g^{\bar{c}}_{i,n_1+1}, \ldots, g^{\bar{c}}_{i,n})$. Therefore,

$$\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\theta}_{\binom{k_1}{2} \times n_1} & \boldsymbol{\theta}_{\binom{k_1}{2} \times n_1} \end{pmatrix}, \tag{57}$$

where $\boldsymbol{\theta}_{\binom{k_1}{2}\times n_1}$ is a binary matrix of size $\binom{k_1}{2} \times n_1$. From (54)–(57), $\boldsymbol{G}^{\tilde{\mathcal{C}}}$ can be written as

$$\boldsymbol{G}^{\tilde{\mathcal{C}}} = \begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{k_1\times n_1} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{k_1\times n_1} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} & \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} \\ \boldsymbol{0}_{1\times n_1} & \boldsymbol{1}_{1\times n_1} \end{pmatrix}.$$

Using Gaussian elimination, $\boldsymbol{G}^{\tilde{\mathcal{C}}}$ can be reduced to

$$\boldsymbol{G}^{\tilde{\mathcal{C}}} = \begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} & \boldsymbol{0}_{k_1\times n_1} \\ \boldsymbol{0}_{k_1\times n_1} & \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{0}_{k_1\times n_1} & \boldsymbol{0}_{k_1\times n_1} \\ \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} & \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} \\ \boldsymbol{0}_{1\times n_1} & \boldsymbol{1}_{1\times n_1} \end{pmatrix}. \tag{58}$$

Let $\boldsymbol{G}^{\mathcal{U}} = \big(\boldsymbol{I}_{k_1} | \boldsymbol{P}_{k_1\times(n_1-k_1)}\big)$, where $\boldsymbol{P}_{k_1\times(n_1-k_1)}$ is the parity matrix of size $k_1 \times (n_1 - k_1)$. We now count the number of independent rows in the matrix

$$\begin{pmatrix} \boldsymbol{G}^{\mathcal{U}} \\ \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_{k_1} & \boldsymbol{P}_{k_1\times(n_1-k_1)} \\ & \boldsymbol{\theta}_{\binom{k_1}{2}\times n_1} \end{pmatrix}.$$

Upon performing Gaussian elimination, we get

$$\begin{pmatrix} \boldsymbol{I}_{k_1} & \boldsymbol{P}_{k_1\times(n_1-k_1)} \\ \boldsymbol{0}_{\binom{k_1}{2}\times k_1} & \boldsymbol{\Delta}_{\binom{k_1}{2}\times(n_1-k_1)} \end{pmatrix},$$

where $\boldsymbol{\Delta}_{\binom{k_1}{2}\times(n_1-k_1)}$ is a matrix of dimensions $\binom{k_1}{2} \times (n_1 - k_1)$ with elements in $\mathrm{GF}(2)$. Hence, we have $\mathrm{rank}\,(\boldsymbol{\Delta}) \leq \min\big(\binom{k_1}{2}, (n_1 - k_1)\big)$. From this and (58), we can easily see that

$$\begin{aligned} \tilde{k} &= \mathrm{rank}\big(\boldsymbol{G}^{\tilde{\mathcal{C}}}\big) \\ &= k_1 + k_1 + \mathrm{rank}\,(\boldsymbol{\Delta}) + 1 \\ &\leq \begin{cases} k_1 + n_1 + 1 & \text{if } n_1 - k_1 \leq \binom{k_1}{2}, \\ 2k_1 + \binom{k_1}{2} + 1 & \text{otherwise.} \end{cases} \end{aligned}$$

■

Lemma 9 gives an upper bound on the dimension of $\tilde{\mathcal{C}}$. In order to prove $\dim(\tilde{\mathcal{C}}) < n$, we check when the upper bound in (50) is at most $n - 1$. For the first case in (50), we need to show

$$\tilde{k} \leq k_1 + n_1 + 1 \leq 2n_1 - 1.$$

Clearly, this is true since $n_1 \geq k_1 + 2$ by assumption. For the second case in (50) we have to show

$$\tilde{k} \leq 2k_1 + \binom{k_1}{2} + 1 \leq 2n_1 - 1.$$

Since $n_1 > \binom{k_1}{2} + k_1$, the above inequality reduces to

$$\binom{k_1}{2} > 2.$$

Clearly, this is true for $k_1 \in \mathbb{N}_{3:\infty}$. In the following, we argue for $k_1 \in \mathbb{N}_2$. Since $n_1 \geq k_1 + 2$ by assumption, we have

$$2n_1 - 1 \geq 2(k_1 + 2) - 1 = 2k_1 + 3 > 2k_1 + \binom{k_1}{2} + 1,$$

for $k_1 \in \mathbb{N}_2$. Therefore, $\dim(\tilde{\mathcal{C}}) < n$ for $n_1 \geq k_1 + 2$.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annual IEEE Symp. Found. Comp. Sci. (FOCS)*, Milwaukee, WI, USA, Oct. 1995, pp. 41–50.

[2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM*, vol. 45, no. 6, pp. 965–981, Nov. 1998.

[3] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond, "Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval," in *Proc. 43rd Annual IEEE Symp. Found. Comp. Sci. (FOCS)*, Vancouver, BC, Canada, Nov. 2002, pp. 261–270.

[4] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *Journal of the ACM*, vol. 55, no. 1, pp. 1–16, Feb. 2008.

[5] K. Efremenko, "3-query locally decodable codes of subexponential length," in *Proc. 41st Annual ACM Symp. Theory Comput. (STOC)*, Bethesda, MD, USA, May/Jun. 2009, pp. 39–44.

[6] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Batch codes and their applications," in *Proc. 36th Annual ACM Symp. Theory Comput. (STOC)*, Chicago, IL, USA, Jun. 2004, pp. 262–271.

[7] N. B. Shah, K. V. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 856–860.

[8] T. H. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, China, Jun. 2015, pp. 2842–2846.

[9] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, China, Jun. 2015, pp. 2852–2856.

[10] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[11] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[12] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[13] Y. Zhang and G. Ge, "A general private information retrieval scheme for MDS coded databases with colluding servers," Apr. 2017, arXiv:1704.06785v1 [cs.IT]. [Online]. Available: https://arxiv.org/abs/1704.06785

[14] ——, "Private information retrieval from MDS coded databases with colluding servers under several variant models," May 2017, arXiv:1705.03186v2 [cs.IT]. [Online]. Available: https://arxiv.org/abs/1705.03186

[15] R. G. L. D'Oliveira and S. El Rouayheb, "Lifting private information retrieval from two to any number of messages," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 2018, pp. 1744–1748.

[16] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti *et al.*" *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1000–1022, Feb. 2018.

[17] ——, "The capacity of robust private information retrieval with colluding databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 2361–2370, Apr. 2018.

[18] Q. Wang and M. Skoglund, "Linear symmetric private information retrieval for MDS coded distributed storage with colluding servers," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Kaohsiung, Taiwan, Nov. 2017, pp. 71–75.

[19] ——, "On PIR and symmetric PIR from colluding databases with adversaries and eavesdroppers," 2019, to app. in *IEEE Trans. Inf. Theory*.

[20] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 322–329, Jan. 2019.

[21] V. Guruswami and M. Wootters, "Repairing Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5684–5698, Sep. 2017.

[22] I. Tamo, M. Ye, and A. Barg, "Optimal repair of Reed-Solomon codes: Achieving the cut-set bound," in *Proc. 58th Annual IEEE Symp. Found. Comp. Sci. (FOCS)*, Berkeley, CA, USA, Oct. 2017, pp. 216–227.

[23] C. Huang, M. Chen, and J. Li, "Pyramid codes: Flexible schemes to trade space for access efficiency in reliable data storage systems," in *Proc. IEEE Int. Symp. Net. Comp. Appl. (NCA)*, Cambridge, MA, USA, Jul. 2007, pp. 79–86.

[24] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "XORing elephants: Novel erasure codes for big data," in *Proc. 39th Very Large Data Bases Endowment (VLDB)*, Trento, Italy, Aug. 2013, pp. 325–336.

[25] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, and S. Yekhanin, "Erasure coding in Windows Azure storage," in *Proc. USENIX Annual Tech. Conf.*, Boston, MA, USA, Jun. 2012.

[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] 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.

[28] S. Kumar, E. Rosnes, and A. Graell i Amat, "Private information retrieval in distributed storage systems using an arbitrary linear code," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1421–1425.

[29] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geom.*, vol. 1, no. 1, pp. 647–664, Nov. 2017.

[30] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, A.-L. Horlemann-Trautmann, D. Karpuk, and I. Kubjas, "Reed-Muller codes for private information reterival," in *Proc. 10th Int. Workshop Coding Cryptography (WCC)*, Saint-Petersburg, Russia, Sep. 2017.

[31] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[32] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, A.-L. Horlemann-Trautmann, D. Karpuk, and I. Kubjas, "t-private information retrieval schemes using transitive codes," 2019, to app. in *IEEE Trans. Inf. Theory*.

[33] V. K. Wei, "Generalized Hamming weights for linear codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, pp. 1412–1418, Sep. 1991.

[34] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2013.

[35] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Trans. IRE Prof. Gro. Inf. Theory*, vol. 4, no. 4, pp. 38–49, Sep. 1954.

[36] J. D. Key, T. P. McDonough, and V. C. Mavron, "Information sets and partial permutation decoding for codes from finite geometries," *Finite Fields Th. App.*, vol. 12, no. 2, pp. 232–247, Apr. 2006.

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

[38] W. C. Huffman and V. Pless, Eds., *Fundamentals of Error-Correcting Codes*. Cambridge, UK: Cambridge University Press, 2010.

[39] 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.

[40] P. Delsarte, J. M. Goethals, and F. J. MacWilliams, "On generalized Reed-Muller codes and their relatives," *Inf. Contr.*, vol. 16, no. 5, pp. 403–442, Jul. 1970.

[41] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes & Rel. Topics (ISTC)*, Brest, France, Sep. 2000, pp. 543–546.

[42] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3268–3271, Dec. 2003.

[43] J. Hao and S.-T. Xia, "Constructions of optimal binary locally repairable codes with multiple repair groups," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1060–1063, Jun. 2016.

[44] E. Rosnes, M. A. Ambroze, and M. Tomlinson, "On the minimum/stopping distance of array low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5204–5214, Sep. 2014.

[45] C. Feyling, "Punctured maximum distance separable codes," *Electron. Lett.*, vol. 29, no. 5, pp. 470–471, Mar. 1993.