Private Information Retrieval Schemes with With Product-Matrix MBR Codes

Julien Lavauzelle, Razane Tajeddine, Ragnar Freij-Hollanti, Camilla Hollanti

Private Information Retrieval Schemes with Regenerating Codes

Julien Lavauzelle*†, Razane Tajeddine*§, Ragnar Freij-Hollanti§, Camilla Hollanti§

† Laboratoire LIX, École Polytechnique, Inria & CNRS UMR 7161, University Paris-Saclay, Palaiseau, France

Email: julien.lavauzelle@inria.fr

§ Department of Mathematics and Systems Analysis, Aalto University School of Science, Espoo, Finland

Emails: {razane.tajeddine, ragnar.freij, camilla.hollanti}@aalto.fi

Abstract

A private information retrieval (PIR) scheme allows a user to retrieve a file from a database without revealing any information on the file being requested. As of now, PIR schemes have been proposed for several kinds of storage systems, including replicated and MDS-coded data. In this paper, the problem of constructing a PIR scheme on regenerating codes is considered.

A regenerating code is a storage code whose codewords are distributed among n nodes, enabling efficient storage of files, as well as low-bandwidth retrieval of files and repair of nodes. In this work, a PIR scheme on regenerating codes is constructed, using the product-matrix (PM) framework of Rashmi, Shah and Kumar. Both the minimum-bandwidth (MBR) and minimum-storage (MSR) settings are considered, and the structure given by the PM framework is used in order to reduce the download communication complexity of our schemes.

I. Introduction

Private information retrieval (PIR) allows a user to retrieve a file from a storage system without revealing what file she is interested in. The problem of constructing PIR schemes was introduced by Chor, Goldreich, Kushilevitz and Sudan [1], [2], where data was considered to be replicated on multiple servers. In the first model, it was assumed that the data is a bitstring $\boldsymbol{x} \in \{0,1\}^m$, and the user would

*: Both authors contributed equally to this manuscript.

like to retrieve a bit x^f without revealing the index f to the servers. Since its introduction, much work has been done on the replicated data model [3]–[7]. The asymptotic capacity for a PIR scheme over a storage system where the files are replicated on n servers was found to be $1 - 1/n$ [3].

On the other hand, there is a lot of interest in using codes for storage in order to minimize storage overhead. As a consequence, many works also considered the PIR model where the data is not replicated but coded and distributed over multiple servers, see *e.g.* [8]–[16]. The asymptotic capacity for a PIR scheme for a storage system where the files are coded on multiple servers using an $[n, k]$ MDS code was found to be $1 - k/n$ [10]. The present work will focus on the case of *regenerating codes* as storage codes.

Regenerating codes are a class of codes dedicated to distributed storage, achieving the optimal tradeoff between the bandwidth needed for a node repair and the amount of data each node needs to store. These codes were pioneered by Dimakis *et al.* [17] who notably produced a cut-set bound on the parameters of the codes. This bound materializes two interesting optimal settings: one for which the repair communication cost in minimized, called the minimum-bandwidth regenerating (MBR) point, and one for which the nodes store the least data, called the minimum-storage regenerating (MSR) point. Rashmi *et al.* [18] then proposed optimal constructions for these two specific settings, based on the so-called *product-matrix* (PM) framework. Many other works followed, including [19]–[21] for the construction of MBR/MSR codes. Also notice that security against eavesdroppers in regenerating codes have been intensively studied, *e.g.* in [22], [23].

In this paper, we propose PIR schemes for the optimal PM constructions of Rashmi *et al.* [18] in both MBR and MSR settings. The protocols we give use the symmetry and the redundancy inherent to the PM constructions, in order to decrease the number of symbols downloaded from the servers. As a consequence, we outperform the very recent constructions of PIR schemes over PM codes given by Dorkson and Ng in [24], [25], which represent the only existing works on PIR schemes for MBR/MSR codes, to the best of our knowledge.

Concerning PM-MBR codes, we obtain a PIR rate strictly larger than $1 - \frac{k}{n}$, where n is the total number of servers and k is the smallest number of servers it is necessary to contact in order to retrieve a file in a regenerating code. This can be compared to the capacity of *scalar* $[n, k]$ MDS-coded PIR schemes for an unbounded number of messages, which is exactly $1 - \frac{k}{n}$ [9], [10]. Thus, this presents another incentive to use MBR codes for storage systems. It is important to note that, though our result might seem contradictory, PM codes are *vector* codes, hence the bound in [9], [10] does not apply. In this work, the PIR rate we obtain remains below $1 - \frac{k}{n} + \frac{k(k-1)}{2nd}$, which can be considered as an upper

bound on the capacity of PIR schemes based on (n, k, d) MBR codes[1].

In the PM-MSR setting, we construct a PIR scheme similar to the scheme in the PM-MBR setting, where we consider $d = 2k - 2$ for simplicity. The PIR scheme achieves a PIR rate which is between $1 - d/n$, the rate obtained by Dorkson and Ng [24] which is also the PIR capacity of an $[n, d]$ MDS code, and $1 - k/n$, the PIR capacity of an $[n, k]$ MDS code.

II. Preliminaries

A. Notation and definitions

For $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q^n$, we denote their inner product by $\langle \boldsymbol{a}, \boldsymbol{b} \rangle \coloneqq \sum_{i=1}^n a_i b_i \in \mathbb{F}_q$ and their component-wise (star) product by $\boldsymbol{a} \star \boldsymbol{b} \coloneqq (a_1 b_1, \ldots, a_n b_n) \in \mathbb{F}_q^n$. For $I \subset [1, n]$, we denote by $\boldsymbol{a}_{|I}$ the tuple obtained by restricting $\boldsymbol{a}$ to coordinates in I. The *Reed-Solomon code* of dimension k with distinct evaluation points $\boldsymbol{x} = (x_1, \ldots, x_n)$, where $x_i \in \mathbb{F}_q$, is defined by

$$\mathrm{RS}_k(\boldsymbol{x}) \coloneqq \{(f(x_1), \ldots, f(x_n)), f \in \mathbb{F}_q[X], \deg f \leq k - 1\} \subseteq \mathbb{F}_q^n \,.$$

It is well-known that for any $1 \leq k \leq n$, the code $\mathrm{RS}_k(\boldsymbol{x})$ is maximum-distance separable (MDS), and that $\mathrm{RS}_j(\boldsymbol{x}) \subseteq \mathrm{RS}_k(\boldsymbol{x})$ for every $j \leq k$. Therefore there exists a basis $\Gamma = \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k\}$ of $\mathrm{RS}_k(\boldsymbol{x})$, such that, for every $j \leq k$ and every subset $I \subset [1, n]$ for cardinality $|I| \geq j$, the family $\Gamma^{(I,j)} \coloneqq \{(\boldsymbol{\gamma}_1)_{|I}, \ldots, (\boldsymbol{\gamma}_j)_{|I}\}$ is a basis of $\mathrm{RS}_j(\boldsymbol{x}_{|I}) \subseteq \mathbb{F}_q^{|I|}$. For instance, one can take a degree-ordered monomial basis, explicitly given by $\boldsymbol{\gamma}_j \coloneqq (x_1^j, \ldots, x_n^j) \in \mathbb{F}_q^n$.

Throughout this paper, we will refer to the asymptotic PIR capacity simply as the PIR capacity, as this is the only definition of PIR capacity we consider.

The *Vandermonde matrix* with distinct basis elements $\boldsymbol{x} \in \mathbb{F}_q^n$ is the $n \times k$ matrix $\boldsymbol{\Psi} \in \mathbb{F}_q^{n \times k}$ such that $\Psi_{i,j} = x_i^j$ for $1 \leq i \leq n$ and $1 \leq j \leq k$. We know that $\boldsymbol{\Psi}$ generates the code $\mathrm{RS}_k(\boldsymbol{x})$ by columns. More precisely, these columns form the monomial basis we mentioned earlier.

The nomenclature used in this paper is summarized in the following table.

B. Private information retrieval

Consider a scheme between a user and n servers storing an encoded version of F files $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^F$. In the scheme, *queries* $\boldsymbol{Q}[1], \ldots, \boldsymbol{Q}[n]$ are sent to servers, which in return compute *responses* $\boldsymbol{R}[1], \ldots, \boldsymbol{R}[n]$ accordingly. Now, assume the user wants to retrieve a specific file $\boldsymbol{X}^{f_0}$, for $1 \leq f_0 \leq F$. We say the

[1]Indeed, under the constraint $\beta = 1$, a PM-MBR code is an $[nd, B]$ linear code over $\mathbb{F}_q$, where $B = kd - \frac{k(k-1)}{2}$. Moreover it is known that $1 - \frac{B}{nd}$ is an upper bound on the PIR capacity of an $[nd, B]$ linear code with such parameters, since it is the capacity of an $[nd, B]$ MDS code [10].

TABLE I
NOMENCLATURE

$\mathcal{C}$	Regenerating code
F	Number of files
n	Number of servers
k	Reconstruction parameter of the regenerating code
d	Repair parameter of the regenerating code
B	Number of symbols in a regenerating codeword
α	Storage capacity of a single server
β	Repair-bandwidth of a single server
$\boldsymbol{X} = (\boldsymbol{X}^1, \dots, \boldsymbol{X}^F)$	Set of files (database)
$\boldsymbol{X}^{f_0}$	Specific file requested by the user
$\boldsymbol{M}^f$	Redundant arrangement of file $\boldsymbol{X}^f$ in a matrix, as in the PM framework
$\boldsymbol{C}^f$	Regenerating codeword associated to $\boldsymbol{X}^f$, as stored on the DSS
$\boldsymbol{C}^f[\cdot, \cdot, s]$	s-th stripe of codeword $\boldsymbol{C}^f$
$\boldsymbol{C}^f[i, \cdot, s]$	Sub-array of $\boldsymbol{C}^f[\cdot, \cdot, s]$ stored by i-th server
$\boldsymbol{C}^f[i, j, s]$	j-th $\mathbb{F}_q$-symbol of sub-array $\boldsymbol{C}^f[i, \cdot, s]$
$\boldsymbol{Q}_\ell$	ℓ-th query sent to servers
R	Rate of a PIR scheme
$H(\cdot)$	Entropy function

scheme achieves information-theoretic PIR against non-colluding servers, if the following requirements hold:

$$\text{Privacy:} \quad H(f_0 \mid \boldsymbol{Q}[i]) = H(f_0), \quad i = 1, \dots, n.$$
$$\text{Recovery:} \quad H(\boldsymbol{X}^{f_0} \mid \boldsymbol{R}[1], \dots, \boldsymbol{R}[n]) = 0\,.$$

Here, $H(\cdot)$ denoted the entropy function. Concerning the recovery constraint, it is also desirable that the user is able to reconstruct $\boldsymbol{X}^{f_0}$ explicitly from $\boldsymbol{R}[1], \dots, \boldsymbol{R}[n]$. Finally, we define the (download) PIR rate of a scheme by $R \coloneqq \frac{|\boldsymbol{X}^{f_0}|}{\sum_i |\boldsymbol{R}[i]|}$ where $|\cdot|$ represents the bitsize of a vector. The PIR capacity is the maximum achievable PIR rate.

C. Regenerating codes

Regenerating codes were introduced by Dimakis *et al.* in the context of distributed storage [17]. In an $(n, k, d, B, \alpha, \beta)$ regenerating code, a coded version of a file of size B is stored on n servers (or nodes), each storing α symbols. Besides, two additional constraints are required. The first is to give any external user the ability to retrieve the file by contacting any subset of k servers. The second is to allow repair of any failed server by contacting any subset of $d \geq k$ servers and downloading β symbols from each,

i.e., $\gamma \coloneqq \beta d$ symbols in total. Parameters of regenerating codes are sometimes shortly denoted (n, k, d), but one should take care that d is *not* the minimum distance of the code, and k is *not* the dimension of the code.

Dimakis *et al.* [17] proved that any storage (erasure) code must satisfy the so-called *cut-set bound*

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}\,, \tag{1}$$

and codes achieving this bound are called *regenerating codes*. Dimakis *et al.* also showed that equality in (1) defines a tradeoff between parameters α and $\gamma = \beta d$, which cannot be minimized simultaneously. Optimal codes minimizing $\gamma = \beta d$ reach the minimum-bandwidth regeneration (MBR) point, while those minimizing α attain the minimum-storage regeneration (MSR) point.

D. Product-Matrix constructions

In this work, we focus on the regenerating codes built by Rashmi *et al.* in [18], through the *product-matrix* (PM) framework. In their constructions, the authors set $\beta = 1$ without loss of generality, since regenerating codes with $\beta \neq 1$ can be built by *striping* files in regenerating codes with $\beta = 1$. Therefore, for convenience we also consider the setting $\beta = 1$ in what follows.

1) PM codes in the MBR setting: At the MBR point with $\beta = 1$, we have the following constraints on the parameters:

$$\alpha = d \quad \text{and} \quad B = k(d-k) + \frac{k(k+1)}{2}\,.$$

The construction of Rashmi *et al.* [18] can be presented as follows. Firstly, file (message) symbols are arranged in a $d \times d$ matrix

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{S} & \boldsymbol{T} \\ \boldsymbol{T}^\top & \boldsymbol{0} \end{pmatrix} \tag{2}$$

where $\boldsymbol{S}$ is a $k \times k$ symmetric matrix containing $\frac{k(k+1)}{2}$ distinct file symbols, and $\boldsymbol{T}$ is a $k \times (d-k)$ matrix containing the remaining $k(d-k)$ file symbols. Let now $\boldsymbol{\Psi}$ be an $n \times d$ Vandermonde matrix over a large enough finite field $\mathbb{F}_q$. The code is defined as $\mathcal{C} \coloneqq \boldsymbol{\Psi}\boldsymbol{M} \in \mathbb{F}_q^{n \times d}$. The j-th row of a codeword in $\mathcal{C}$ is stored on server S_j, for $j = 1, \ldots, n$, and contains at most $\alpha = d$ information symbols. Notice that $\mathcal{C}$ is an $[nd, B]$ linear code over $\mathbb{F}_q$. For clarity, let us now rewrite the example given by the authors in [18, Sec. IV.A.].

Example 1 (Optimal PM-MBR code)**.** *Consider the setting* $(n,k,d)=(6,3,4)$ *over the field* $\mathbb{F}_7$*. The original file contains* $B=k(d-k)+\frac{k(k+1)}{2}=9$ *symbols. Let* $\boldsymbol{x}=(1,2,3,4,5,6)\in\mathbb{F}_7^6$*. The generator (Vandermonde) matrix and the message matrix are then given as:*

$$\boldsymbol{\Psi}=\begin{pmatrix}1&1&1&1\\1&2&4&1\\1&3&2&6\\1&4&2&1\\1&5&4&6\\1&6&1&6\end{pmatrix},\ \boldsymbol{M}=\begin{pmatrix}m_1&m_2&m_3&m_7\\m_2&m_4&m_5&m_8\\m_3&m_5&m_6&m_9\\m_7&m_8&m_9&0\end{pmatrix}.$$

2) PM codes in the MSR setting: In the MSR setting with $\beta=1$, parameters α and B are given by:

$$\alpha=d-k+1\quad\text{ and }\quad B=k(d-k+1)\,.$$

In [18], the authors construct PM codes at the MSR point, for $d\geq 2k-2$. In this setting, $d\leq 2\alpha$ and $B\leq\alpha(\alpha+1)$. In this work, for simplicity, we assume $d=2k-2$ as it is the case for the first construction given in [18]. Thus, d and B can be simplified as $d=2\alpha$ and $B=\alpha(\alpha+1)$. Note that the scheme we propose further in Section IV can be easily generalized to the case where $d\geq 2k-2$.

File symbols are arranged in a $2\alpha\times\alpha$ matrix

$$\boldsymbol{M}=\begin{pmatrix}\boldsymbol{S}_1\\\boldsymbol{S}_2\end{pmatrix}$$

where each $\boldsymbol{S}_i$ is an $\alpha\times\alpha$ symmetric matrix containing $\frac{\alpha(\alpha+1)}{2}$ file symbols. Let $\boldsymbol{\Psi}$ be an $n\times 2\alpha$ Vandermonde matrix over $\mathbb{F}_q$. As in the MBR setting, the j-th row of a codeword from the code $\mathcal{C}\coloneqq\boldsymbol{\Psi}\boldsymbol{M}$ is stored on server S_j, for $j=1,\ldots,n$.

This construction is referred to as PM-MSR codes. Let us also rewrite the example given in [18, Sec. V.A.].

Example 2 (Optimal PM-MSR code)**.** *Consider the setting* $(n,k,d)=(6,3,4)$ *over* $\mathbb{F}_{13}$*, which gives the file size* $B=6$*. Let* $\boldsymbol{x}=(1,2,3,4,5,6)\in\mathbb{F}_{13}^6$*. Matrices* $\boldsymbol{\Psi}$ *and* $\boldsymbol{M}$ *are then given by:*

$$\boldsymbol{\Psi}=\begin{pmatrix}1&1&1&1\\1&2&4&8\\1&3&9&1\\1&4&3&12\\1&5&12&8\\1&6&10&8\end{pmatrix},\ \boldsymbol{M}=\begin{pmatrix}m_1&m_2\\m_2&m_3\\m_4&m_5\\m_5&m_6\end{pmatrix}.$$

III. A PIR SCHEME IN THE MBR SETTING

In this section, we consider a PM-MBR code $\mathcal{C}$ over $\mathbb{F}_q$, with parameters (n, k, d). Recall that $\mathcal{C}$ is also a linear code over $\mathbb{F}_q$ of length nd and dimension $B = k(d-k) + \frac{k(k+1)}{2}$.

A. System setup

We consider a database $\boldsymbol{X}$ composed of F files $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^F$, such that each $\boldsymbol{X}^f$ consists of $B = k(d-k) + \frac{k(k+1)}{2}$ information symbols. For every $1 \leq f \leq F$, the symbols of file $\boldsymbol{X}^f$ are subdivided into $S \geq 1$ *stripes* (or subdivisions) and organized in a 3-dimensional array $\boldsymbol{M}^f$ (that we abusively name a *matrix*), such that

$$\boldsymbol{M}^f = \begin{pmatrix} & 1 \leq i \leq d \\ M^f[i,j,s], & 1 \leq j \leq d \\ & 1 \leq s \leq S \end{pmatrix} \in \mathbb{F}_q^{d \times d \times S},$$

where for every i, j, s, f, we have $M^f[i,j,s] \in \mathbb{F}_q$. Following the PM framework, every stripe $\boldsymbol{M}^f[\cdot, \cdot, s]$ must the form given in (2). Also notice that, by construction of the regenerating code $\mathcal{C}$, for all i, j, s, f, we have:

$$M^f[i,j,s] = M^f[j,i,s]\,,$$

and

$$M^f[i,j,s] = 0 \quad \text{if } i \geq k+1 \text{ and } j \geq k+1\,.$$

We also use the notation $\boldsymbol{M} \coloneqq (\boldsymbol{M}^1, \ldots, \boldsymbol{M}^F)$.

For every j, s, f, the column $\boldsymbol{M}^f[\cdot, j, s] \in \mathbb{F}_q^d$ is encoded using a Reed-Solomon code $\mathrm{RS}_d(\boldsymbol{x})$, resulting in a codeword

$$\boldsymbol{C}^f[\cdot, j, s] = \sum_{r=1}^{d} M^f[r,j,s]\boldsymbol{\gamma}_r,$$

where we recall that $\Gamma = \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d\}$ denotes a suitable basis for sequences of Reed-Solomon codes (see Section II-A). Due to the form of message matrices $\boldsymbol{M}^f$, one can also remark that $\boldsymbol{C}^f[\cdot, j, s] \in \mathrm{RS}_k(\boldsymbol{x})$ if $j \geq k+1$.

B. Intuition

The idea behind the constructed PIR scheme is to use the symmetric property of matrices $\boldsymbol{M}^f$ as a way to reuse information in order to decrease the download complexity of the scheme. We note that the servers are assumed not to collude. In this scheme, each file is divided into $S = n - k$ stripes. The user generates a set of k queries to the servers, similarly to the scheme in [13]. A query is defined as an $n \times S \times F$ vector that is sent by the user to retrieve information. Randomness is embedded in the

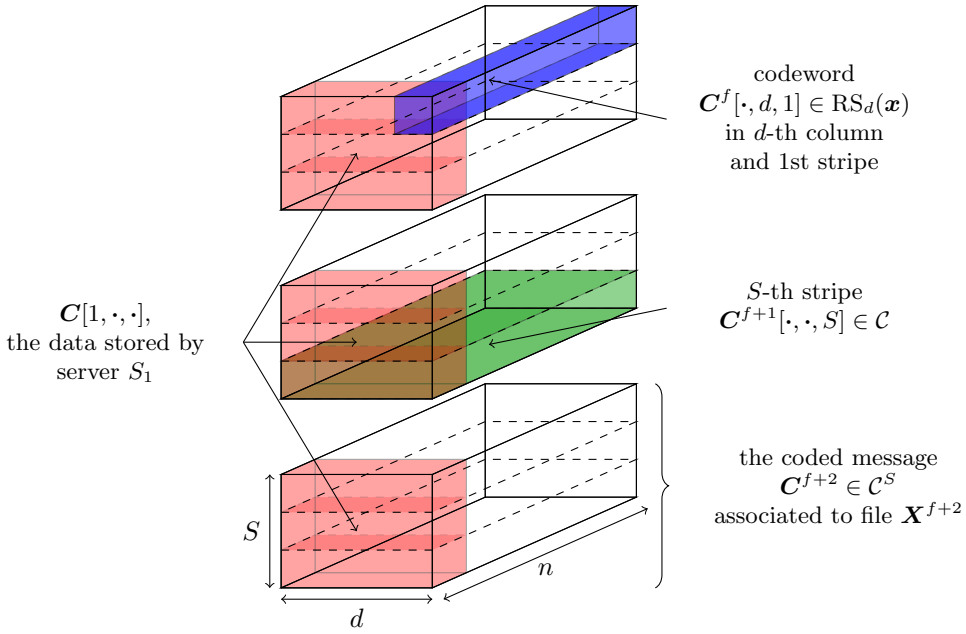


Fig. 1. An illustration of the arrangement of files, stripes and codewords in the storage system. A system of n servers stores encoded files represented by $S \times d \times n$ cuboids (in the figure, only three of them are represented). Foreground (red) blocks represent data stored by the first server. The horizontal block (in green) in the middle cuboid represents a stripe, which lies in the regenerating code $\mathcal{C}$. Top right block (in blue) is a column of a stripe, and typically lies in an MDS code.

queries as a way to hide the requested file's identity, in a similar manner to one-time padding. Naturally, if privacy were not a concern, a query to retrieve file $\boldsymbol{X}^{f_0}$ would be the vector of size $n \times S \times F$ with zeroes everywhere, except in positions $f = f_0$ corresponding to the requested file.

The queries are then sent to the servers which project queries on their stored data the following manner. For the last $d - k$ columns, since each of these columns stores file stripes encoded using an $[n, k]$ MDS code, servers are asked to project *all* the queries on the data they hold, similarly to [13]. For each of the other columns, stripes contain information already retrieved from the previously used columns, due to the nature of the product-matrix construction. Thus, from server S_d down to server S_1, servers are asked to project on their stored data a *decreasing* subset of the initial set of queries. This still enables the user to reconstruct the requested file, due to the fact that she had peeled off some randomness and information symbols from previous columns. Moreover, it allows her to run a more efficient PIR scheme on an $[n, k']$ MDS code with where $k' < k$. More details are given in the upcoming sections.

C. The PIR scheme

In this section, we describe the PIR scheme explicitly. Let us assume that the user wants to retrieve a file $\boldsymbol{X}^{f_0}$, for some $1 \leq f_0 \leq F$. We fix the number of stripes to $S = n - k$, and we consider a k-tuple of queries $\boldsymbol{Q} = (\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_k)$, such that for $1 \leq \ell \leq k$, query $\boldsymbol{Q}_\ell$ has the following form:

$$\boldsymbol{Q}_\ell = \begin{pmatrix} & 1 \leq i \leq n \\ Q_\ell^f[i,s], & 1 \leq s \leq S \\ & 1 \leq f \leq F \end{pmatrix}.$$

Notice that, since the same set of queries is meant to be used for every column, query $\boldsymbol{Q}_\ell$ does *not* depend on a column index $j \in [1, d]$. This property is fundamental for the privacy of the scheme.

The sub-query $\boldsymbol{Q}_\ell^f[i, \cdot]$ is then sent to server S_i, for each $1 \leq i \leq n$. The response $R_\ell[i,j] \in \mathbb{F}_q$ of server S_i with respect to the pair (ℓ, j), is then defined as:

$$R_\ell[i,j] \coloneqq \langle \boldsymbol{Q}_\ell[i,\cdot], \boldsymbol{C}[i,j,\cdot] \rangle = \sum_{s,f} Q_\ell^f[i,s] C^f[i,j,s]\,.$$

We also denote by $\boldsymbol{R}_\ell[\cdot, j] \coloneqq (R_\ell[1,j], \ldots, R_\ell[n,j]) \in \mathbb{F}_q^n$.

Generation of $\boldsymbol{Q}$. The random tuple of queries $\boldsymbol{Q}$ is defined as the sum of two components.

1) A random part $\boldsymbol{D}$, defined as follows. For every ℓ, s, f, a symbol $\lambda_{\ell,s,f} \in \mathbb{F}_q$ is picked uniformly at random and independently of others. Then, for every $1 \leq i \leq n$, we define:

$$D_\ell^f[i,s] = \lambda_{\ell,s,f}.$$

   In other words, $\boldsymbol{D}_\ell^f[\cdot, s] \in \mathbb{F}_q^n$ is a word picked uniformly at random in the repetition code of length n.

2) A deterministic part $\boldsymbol{E}^{(f_0)}$, also called the retrieval pattern. This pattern is defined by:

$$E_\ell^{(f_0),f}[i,s] = \begin{cases} 1 & \text{if } f = f_0 \text{ and } n - i = \ell + s - 2 \pmod S, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the tuple of queries $\boldsymbol{Q}$ is defined by $\boldsymbol{Q} \coloneqq \boldsymbol{D} + \boldsymbol{E}^{(f_0)}$. Notice here that each query is sent to the servers by hiding the deterministic part with a random vector. Therefore, the privacy of the scheme still holds.

Server responses to queries. We now assume that $\boldsymbol{Q}[i,\cdot]$ is sent to server S_i, for every $1 \leq i \leq n$. In the proposed protocol, the set of responses required by the user depend on the index $j \in [1, d]$ of the column, as described below:

- For columns $k+1 \leq j \leq d$, every server S_i, $1 \leq i \leq n$, must send back to the user the responses $R_\ell[i,j]$, where $1 \leq \ell \leq k$.

- For columns $1 \leq j \leq k$, only servers S_i such that $k-j+1 \leq i \leq n$ are required to respond to the user. Those servers S_i must compute and send the subset of responses $R_\ell[i,j]$, such that $1 \leq \ell \leq j$.

We here emphasize that, for these first columns $1 \leq j \leq k$, the *subset* of servers S_i, $i \in [k-j+1, n]$, send the *subset* of responses $R_\ell[i,j]$, $\ell \in [1,j]$ to the user. This is a key point in order to achieve a good PIR rate — see Example 3 for an illustration.

Reconstruction of $\boldsymbol{X}^{f_0}$. The recovery is run columnwise, from column d down to column 1. For each step j, $1 \leq j \leq d$, the goal is to retrieve $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ along with some random vectors.

- *For $k+1 \leq j \leq d$.* A precise description of the recovery algorithm is given in the proof of Lemma 1. In short, it consists of running, *independently on each column* $\boldsymbol{C}[\cdot, j, \cdot]$, the reconstruction of the PIR scheme over an MDS code described in [13]. Indeed, each $\boldsymbol{C}[\cdot, j, \cdot]$ can be viewed as a smaller database encoded and stored in an $[n, k]$ MDS storage system. This procedure allows the user to recover $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$, but one should notice that she can also collect random vectors $\sum_{s,f} M^f[r, j, s]\boldsymbol{D}^f_\ell[\cdot, s] \in \mathbb{F}_q^n$, for all $1 \leq r, \ell \leq k$.
- *For $1 \leq j \leq k$.* At step j, we can assume that for every $j' \geq j+1$, the user has already collected
  - $\boldsymbol{M}^{f_0}[\cdot, j', \cdot]$ and
  - the random vectors $\sum_{s,f} M^f[r, j', s]\boldsymbol{D}^f_\ell[\cdot, s] \in \mathbb{F}_q^{n-k+\min\{k,j'\}}$ for every $1 \leq r, \ell \leq \min\{k, j'\}$.

  Recall that $M^f[r, j', s] = M^f[j', r, s]$ and that every $\boldsymbol{D}^f_\ell[\cdot, s]$ lies in a repetition code. As a consequence, the user knows $\sum_{s,f} M^f[r, j, s]\boldsymbol{D}^f_\ell[\cdot, s] \in \mathbb{F}_q^{n-k+\min\{k,j'\}}$ for every $j+1 \leq r \leq d$ and every $1 \leq \ell \leq j$. The retrieval process described in the proof of Lemma 2 then ensures that the user can retrieve $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ and the random vectors $\sum_{s,f} M^f[r, j, s]\boldsymbol{D}^f_\ell[\cdot, s] \in \mathbb{F}_q^{n-k+j}$ for every $1 \leq r, \ell \leq j$.

We start by giving a simple example before diving into technical proofs.

Example 3. *We use the $(6, 3, 4)$ PM-MBR regenerating code described in Example 1. For this purpose, the files are divided into $S = n - k = 3$ stripes, and the user sends $k = 3$ query vectors:*

	Query 1	*Query* 2	*Query* 3
Server S_1	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}$
Server S_2	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}$
Server S_3	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}$
Server S_4	$\boldsymbol{u} + \boldsymbol{e}_{f_0,1}$	$\boldsymbol{v} + \boldsymbol{e}_{f_0,2}$	$\boldsymbol{w} + \boldsymbol{e}_{f_0,3}$
Server S_5	$\boldsymbol{u} + \boldsymbol{e}_{f_0,2}$	$\boldsymbol{v} + \boldsymbol{e}_{f_0,3}$	$\boldsymbol{w} + \boldsymbol{e}_{f_0,1}$
Server S_6	$\boldsymbol{u} + \boldsymbol{e}_{f_0,3}$	$\boldsymbol{v} + \boldsymbol{e}_{f_0,1}$	$\boldsymbol{w} + \boldsymbol{e}_{f_0,2}$

where $\boldsymbol{e}_{f_0,s_0} \in \mathbb{F}_q^{F\times S}$ *is the deterministic vector with all zeros, but one* 1 *in position* (f_0, s_0)*, which corresponds to stripe* s_0 *of what is stored from file* $\boldsymbol{X}^{f_0}$*. Vectors* $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{F}_q^{F\times S}$ *are uniformly random vectors.*

The servers project the data stored in columns 3 *and* 4 *on all the queries. Server* S_1 *does not respond to any other queries. Servers* $S_2, \ldots, S_6$ *project only the first* 2 *queries on the data stored in their second column. Server* S_2 *does not respond to any other queries. Servers* $S_3, \ldots, S_6$ *project only the first query on the data stored in column* 1*. Then the servers send this information back to the user.*

• *Decodability: In this example* $d - k = 1$*. For the last row, the user receives the responses from all three queries from all six servers. The storage code for the last row is a* $[6, 3]$ *MDS code. If we look at the responses to the first query from the last column, it will be:*

	Response 1
S_1	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + M^f[2,4,s] + M^f[3,4,s])$
S_2	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + 2M^f[2,4,s] + 4M^f[2,4,s])$
S_3	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + 3M^f[2,4,s] + 2M^f[3,4,s])$
S_4	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + 4M^f[2,4,s] + 2M^f[3,4,s]) + M^1[1,4,1] + 4M^1[2,4,1] + 2M^1[3,4,1]$
S_5	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + 5M^f[2,4,s] + 4M^f[3,4,s]) + M^1[1,4,2] + 5M^1[2,4,2] + 4M^1[3,4,2]$
S_6	$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}(M^f[1,4,s] + 6M^f[2,4,s] + M^f[3,4,s]) + M^1[1,4,3] + 6M^1[2,4,3] + 1M^1[3,4,3]$

From the above table, we can see that the user can recover the three random symbols

$$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[1,4,s],$$

$$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[2,4,s]$$

and

$$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[3,4,s],$$

along with the three required symbols

$$M^1[1,4,1], M^1[2,4,2], M^1[3,4,3].$$

Following the same reasoning, from the second and third queries the user can retrieve the random symbols

$$\sum_{f=1}^{F}\sum_{s=1}^{3} v_{f,s}M^f[1,4,s], \sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[1,4,s],$$

$$\sum_{f=1}^{F}\sum_{s=1}^{3} v_{f,s}M^f[2,4,s], \sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[2,4,s]$$

and

$$\sum_{f=1}^{F}\sum_{s=1}^{3} v_{f,s}M^f[3,4,s], \sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[3,4,s],$$

along with the required symbols,

$$M^1[1,4,2], M^1[2,4,3], M^1[3,4,1], M^1[1,4,3], M^1[2,4,1], M^1[3,4,2].$$

Notice that the PIR scheme run over the fourth column achieves a PIR rate of $3/6$.

For the third column, the storage code is a $[6,4]$ *MDS code. Recall that* $\boldsymbol{M}^f[3,4,\cdot] = \boldsymbol{M}^f[4,3,\cdot]$ *for every* f*, and the user has already collected information in the responses from column* 4*. As a consequence, the user knows the vector* $\boldsymbol{M}^1[4,3,\cdot]$ *as well as the random symbols*

$$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[4,3,s],$$

$$\sum_{f=1}^{F}\sum_{s=1}^{3} v_{f,s}M^f[4,3,s]$$

and

$$\sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[4,3,s].$$

Therefore, the responses from the third column allow the user to decode the symbols, just like the responses from the last column. The user, thus, recovers $\boldsymbol{M}^1[1,3,\cdot]$*,* $\boldsymbol{M}^1[2,3,\cdot]$*, and* $\boldsymbol{M}^1[3,3,\cdot]$ *with a rate* $3/6$.

For the second column, the storage code is also a $[6,4]$ *MDS code, but the user can use the information she collected from columns* 3 *and* 4*. More precisely, the user already knows vectors* $\boldsymbol{M}^1[2,3,\cdot]$*,* $M^1[2,4,\cdot]$*, and random symbols*

$$\sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[2,3,s], \sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[2,4,s],$$

$$\sum_{f=1}^{F}\sum_{s=1}^{3} v_{f,s}M^f[2,3,s], \sum_{f=1}^{F}\sum_{s=1}^{3} u_{f,s}M^f[2,4,s]$$

and

$$\sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[2,3,s], \sum_{f=1}^{F}\sum_{s=1}^{3} w_{f,s}M^f[2,4,s].$$

Thus, the user does not need the response from server S_1 *in order to decode the symbols. It means that the code can be assumed to be reduced to a* $[5,2]$ *MDS code. The user can then decode the parts* $\boldsymbol{M}^1[1,2,\cdot]$*, and* $\boldsymbol{M}^1[2,2,\cdot]$ *from servers* $S_2,\ldots,S_6$ *and from the first* 2 *queries, with rate* $6/10 = 3/5$.

Following the same reasoning for the first column, the user needs only the responses of servers $S_3,\ldots,S_6$ *to the first query only. The storage code can be seen as a* $[4,1]$ *MDS code on those servers,*

after introducing the already known information. This allows the user to decode the last part of the file, $\boldsymbol{M}^1[1,1,\cdot]$*, with rate* $3/4$*.*

Finally, the PIR rate of the scheme in this example is $R_{\mathrm{MBR}} = \frac{3+6+9+9}{4+10+18+18} = \frac{27}{50} = 0.54$*. We see this rate is larger than* $1-\frac{k}{n} = 1-\frac{3}{6} = \frac{1}{2} = 0.5$ *which is the capacity of scalar MDS-coded PIR schemes, but less than* $1-\frac{B}{nd} = 1-\frac{9}{6\times 4} = \frac{5}{8} = 0.625$*, which is an upper bound on the capacity of* $[nd, B]$*-coded PIR schemes.*

• *Privacy: Privacy follows from the fact that for any fixed desired file, every server gets a uniform random vector as a query.*

D. Analysis

We next prove the correctness of the PIR scheme proposed in previous section.

Lemma 1. *Let* $k+1 \le j \le d$*. Then, conditioned on* $(\boldsymbol{R}_1[\cdot,j],\dots,\boldsymbol{R}_k[\cdot,j])$*, the following is determined:*

- *the piece* $\boldsymbol{M}^{f_0}[\cdot,j,\cdot]$ *of the desired file;*
- *the random vectors* $\sum_{s,f} M^f[r,j,s]\boldsymbol{D}_\ell^f[\cdot,s] \in \mathbb{F}_q^n$ *for every* $1 \le r,\ell \le k$*.*

Proof. Let us fix $1 \le \ell \le k$. After receiving responses from servers, the user is able to build the response vector

$$\boldsymbol{R}_\ell[\cdot,j] \coloneqq (R_\ell[1,j],\dots,R_\ell[n,j]) \in \mathbb{F}_q^n\,.$$

Notice that we have

$$\boldsymbol{R}_\ell[\cdot,j] = \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot,s] \star \boldsymbol{C}^f[\cdot,j,s] + \sum_s \boldsymbol{E}_\ell^{(f_0),f_0}[\cdot,s] \star \boldsymbol{C}^{f_0}[\cdot,j,s]\,.$$

We can now define

$$\boldsymbol{B}_\ell[\cdot,j] \coloneqq \sum_s \boldsymbol{E}_\ell^{(f_0),f_0}[\cdot,s] \star \boldsymbol{C}^{f_0}[\cdot,j,s] \in \mathbb{F}_q^n\,,$$

and we see that

$$B_\ell[i,j] = \begin{cases} C^{f_0}[i,j,s'] & \text{if } i \ge k+1, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $s' \in [1,k]$ satisfies $n-i = (\ell+s'-2 \mod n-k)$. In particular $\boldsymbol{B}_\ell[\cdot,j]$ is supported on $[k+1,n]$, and therefore has weight at most $n-k$.

Now, denote by

$$\boldsymbol{A}_\ell[\cdot,j] \coloneqq \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot,s] \star \boldsymbol{C}^f[\cdot,j,s] \in \mathbb{F}_q^n\,.$$

Since every $\boldsymbol{D}_\ell^f[\cdot,s]$ belongs to the repetition code and $\boldsymbol{C}^f[\cdot,j,s] \in \mathrm{RS}_k(\boldsymbol{x})$, it holds that $\boldsymbol{A}_\ell[\cdot,j] \in \mathrm{RS}_k(\boldsymbol{x})$. We have also seen that $R_\ell[i,j] = A_\ell[i,j]$ for $1 \le i \le k$, thus the user knows the k first symbols

of $\boldsymbol{A}_\ell[\cdot, j]$. Since $[1, k]$ is an information set for $\mathrm{RS}_k(\boldsymbol{x})$, she can recover $\boldsymbol{A}_\ell[\cdot, j]$ entirely. The recovery of $\boldsymbol{B}_\ell[\cdot, j]$ follows easily.

Let us now recall that $\boldsymbol{C}^f[\cdot, j, s] \in \mathrm{RS}_k(\boldsymbol{x})$ can be written as $\sum_{r=1}^{k} M^f[r, j, s]\boldsymbol{\gamma}_r$. Moreover, $\boldsymbol{D}_\ell^f[\cdot, s]$ lies in a repetition code, hence $D_\ell^f[i, s] = \lambda_{\ell,s,f}$ for some $\lambda_{\ell,s,f} \in \mathbb{F}_q$. Therefore, expressing

$$\boldsymbol{A}_\ell[\cdot, j] = \sum_{r=1}^{d} \left(\sum_{s,f} \lambda_{\ell,s,f} M^f[r, j, s] \right) \boldsymbol{\gamma}_r$$

in the basis $\{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d\} \subset \mathbb{F}_q^n$ of nested Reed-Solomon codes $\mathrm{RS}_d(\boldsymbol{x}) \supseteq \mathrm{RS}_k(\boldsymbol{x})$ allows us to retrieve every scalar $\sum_{s,f} \lambda_{\ell,s,f} M^f[r, j, s]$, or equivalently, every $\sum_{s,f} M^f[r, j, s]\boldsymbol{D}_\ell^f[\cdot, s] \in \mathbb{F}_q^n$.

Finally, Equation (3) shows that for every $1 \leq s \leq n - k$, the knowledge of $\boldsymbol{B}_1[\cdot, j], \ldots, \boldsymbol{B}_k[\cdot, j]$ allows the user to retrieve a subset of k distinct symbols of $\boldsymbol{C}^{f_0}[\cdot, j, s]$, which is equivalent to retrieving $\boldsymbol{M}^{f_0}[\cdot, j, s]$. Thus, she can finally obtain $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$. □

Lemma 2. *Let $1 \leq j \leq k$. For every $1 \leq \ell \leq j$, for convenience we denote by*

$$\boldsymbol{R}_\ell[\cdot, j] \coloneqq (R_\ell[k - j + 1, j], \ldots, R_\ell[n, j]) \in \mathbb{F}_q^{n-k+j}.$$

Then, conditioned on $(\boldsymbol{R}_1[\cdot, j], \ldots, \boldsymbol{R}_j[\cdot, j])$ and on

$$\sum_{s,f} M^f[r, j, s]\boldsymbol{D}_\ell^f[\cdot, s], \quad \textit{for all } \; j + 1 \leq r \leq d, \quad 1 \leq \ell \leq j, \tag{4}$$

the following are determined:

- *the piece $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ of the desired file;*
- *random vectors $\sum_{s,f} M^f[r, j, s]\boldsymbol{D}_\ell^f[\cdot, s] \in \mathbb{F}_q^{n-k+j}$ for all $1 \leq r, \ell \leq j$.*

Proof. Let us fix $1 \leq \ell \leq j$. In contrast with Lemma 1, we will deal with vectors of shorter length $n - k + j$. In particular, we denote $\boldsymbol{x}' = (x_{k-j+1}, \ldots, x_n)$. Similarly, the user is able to build the response vector $\boldsymbol{R}_\ell[\cdot, j]$ of length $n - k + j$ given by

$$\boldsymbol{R}_\ell[\cdot, j] \coloneqq (R_\ell[k - j + 1, j], \ldots, R_\ell[n, j]) = \boldsymbol{A}_\ell[\cdot, j] + \boldsymbol{B}_\ell[\cdot, j]\,,$$

where $\boldsymbol{A}_\ell[\cdot, j]$ and $\boldsymbol{B}_\ell[\cdot, j]$ are defined as in Lemma 1. One can rewrite $\boldsymbol{A}_\ell[\cdot, j] \in \mathbb{F}_q^{n-k+j}$ as follows:

$$\begin{aligned}
\boldsymbol{A}_\ell[\cdot, j] &= \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{C}^f[\cdot, j, s] \\
&= \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot, s] \star \left(\sum_{r=1}^{d} M^f[r, j, s]\boldsymbol{\gamma}_r \right) \\
&= \sum_{r=1}^{j} \sum_{s,f} M^f[r, j, s]\; \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{\gamma}_r + \sum_{r=j+1}^{d} \sum_{s,f} M^f[r, j, s]\; \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{\gamma}_r.
\end{aligned}$$

Therefore, using vectors in (4) the user can build

$$\boldsymbol{A}'_\ell[\cdot, j] \coloneqq \sum_{r=j+1}^{d} \left(\sum_{s,f} M^f[r, j, s]\ \boldsymbol{D}^f_\ell[\cdot, s] \right) \star \boldsymbol{\gamma}_r.$$

Hence, she is able to construct

$$\boldsymbol{R}''_\ell[\cdot, j] \coloneqq \boldsymbol{R}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j] = (\boldsymbol{A}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j]) + \boldsymbol{B}_\ell[\cdot, j]\,.$$

As the basis $\{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d\}$ is ordered by degree, we see that $\boldsymbol{A}''_\ell[\cdot, j] \coloneqq \boldsymbol{A}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j]$ lies in $\mathrm{RS}_j(\boldsymbol{x}')$. Indeed, each $\{\boldsymbol{\gamma}_1, \ldots \boldsymbol{\gamma}_j\}$ must also be a basis of smaller RS codes. Also remark that once again, the vector $\boldsymbol{B}_\ell[\cdot, j] \in \mathbb{F}_q^{n-k+j}$ is supported by $[k+1, n]$. Since $[k-j+1, k]$ is an information set for $\mathrm{RS}_j(\boldsymbol{x}')$, the user can thus recover $\boldsymbol{A}''_\ell[\cdot, j]$ and $\boldsymbol{B}_\ell[\cdot, j]$ from $\boldsymbol{R}''_\ell[\cdot, j]$.

Similarly to Lemma 1, one can easily see that $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ can be obtained from $\boldsymbol{B}_1[\cdot, j], \ldots, \boldsymbol{B}_j[\cdot, j]$.

Finally, $\boldsymbol{A}'_\ell[\cdot, j]$ and $\boldsymbol{A}''_\ell[\cdot, j]$ allow to reconstruct $\boldsymbol{A}_\ell[\cdot, j]$. Similarly to the proof of Lemma 1, the basis $\{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_j\}$ of $\mathrm{RS}_j(\boldsymbol{x}')$ leads to the recovery of random elements $\sum_{s,f} M^f[r, j, s]\lambda_{\ell,s,f} \in \mathbb{F}_q$ for every $1 \le r, \ell \le j$. □

Theorem 1. *The scheme proposed in Section III-C is secure against non-colluding servers. Its PIR rate is:*

$$R_{\mathrm{MBR}} = \frac{3(n-k)(2d-k+1)}{6dn - 3nk + 3n - k^2 + 1}\,.$$

Proof. Lemma 1 and Lemma 2 ensure that the user retrieves the correct file $\boldsymbol{X}^{f_0}$ as long as the servers $S_1, \ldots, S_n$ follow the protocol described in Section III-C. Since the servers are assumed not to collude, the only way a server S_i can learn information about the identity f_0 of the required file, is from its own query matrix $\boldsymbol{Q}[i, \cdot]$. Since the matrix $\boldsymbol{Q}[i, \cdot]$ is chosen such that it is statistically independent of f_0, the scheme is private. More precisely, since $\boldsymbol{Q}[i, \cdot] = \boldsymbol{D}[i, \cdot] + \boldsymbol{E}^{(f_0)}[i, \cdot] \sim \boldsymbol{D}[i, \cdot]$, we have

$$H(f_0 \mid \boldsymbol{Q}[i, \cdot]) = H(f_0 \mid \boldsymbol{D}[i, \cdot]) = H(f_0),$$

where $H(\cdot)$ denotes the entropy function.

Let us now compute the PIR rate. The file $\boldsymbol{X}^{f_0}$ consists of

$$(n-k)B = (n-k)(k(d-k) + k(k+1)/2)$$

symbols over $\mathbb{F}_q$. During step j, for $k+1 \le j \le d$, the user downloads k responses from each server $S_1, \ldots, S_n$. Hence she gets a total of $nk(d-k)$ symbols for all these steps. For columns $1 \le j \le k$,

the user downloads j responses from servers $S_{k-j+1}, \ldots, S_n$, leading to a total of $\sum_{j=1}^{k} j(n-k+j)$ symbols for those steps. Therefore, we get the following PIR rate:

$$\begin{aligned} R_{\mathrm{MBR}} &= \frac{(n-k)\left((d-k)k + \frac{k(k+1)}{2}\right)}{(d-k)nk + \sum_{j=1}^{k} j(n-k+j)} \\ &= \frac{(n-k)\left((d-k)k + \frac{k(k+1)}{2}\right)}{(d-k)nk + (n-k)\frac{k(k+1)}{2} + \frac{k(k+1)(2k+1)}{6}} \\ &= \frac{3(n-k)(2d-k+1)}{6dn - 3nk + 3n - k^2 + 1}\,. \end{aligned} \tag{5}$$

□

Remark 1. *As a function of n, k, B, the PIR rate given in Theorem 1 can be written as*

$$R_{\mathrm{MBR}} = \frac{1 - \frac{k}{n}}{1 - \frac{k(k+1)(k-1)}{6nB}}\,. \tag{6}$$

Indeed, starting from Equation (5) *we get*

$$R_{\mathrm{MBR}} = \frac{(n-k)B}{nB + \sum_{j=1}^{k} j(j-k)} = \frac{(n-k)B}{nB - \frac{k(k+1)(k-1)}{6}}\,,$$

leading to the expected expression.

E. On the PIR rate

1) Comparison with the multi-file PIR scheme of Dorkson and Ng: Dorkson and Ng in [24] proposed a PIR scheme over PM-MBR codes in the context of *multi-file* retrieval, *i.e.* any set of $p \geq 1$ files $\boldsymbol{X}^{f_0}, \ldots, \boldsymbol{X}^{f_{p-1}}$ can be simultaneously retrieved privately. In the current work, retrieving p files remains possible by iterating the 1-file PIR protocol p times. Notice that this routine achieves the same PIR rate as the 1-file PIR scheme.

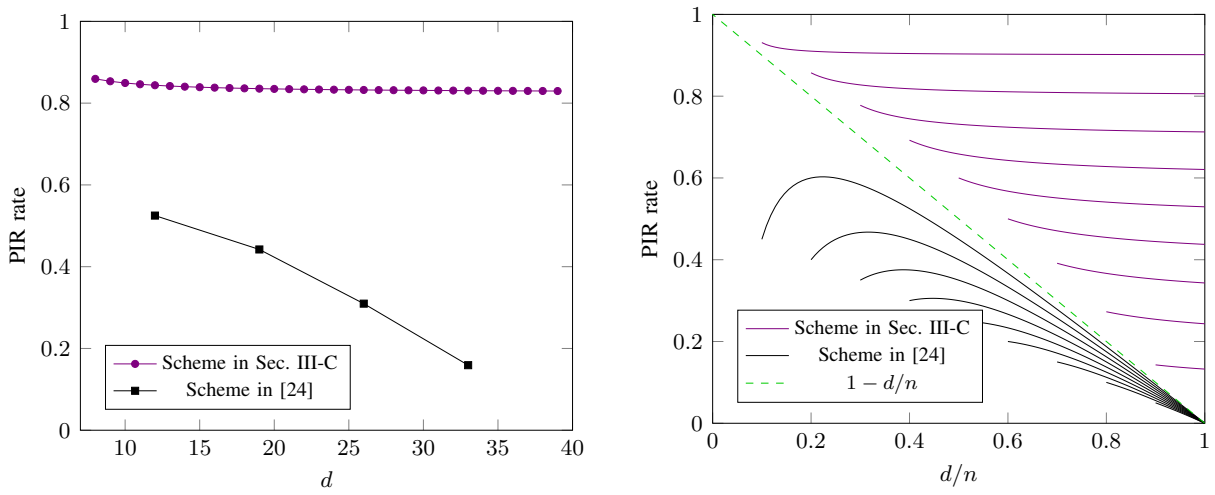
In the general case, the PIR rate obtained in [24] is $R' = \frac{pB}{dn}$, under the additional constraint that $n = pk + d$. We notice that R' can be reformulated as follows:

$$R' = \frac{n-d}{k} \cdot \frac{B}{nd} = \frac{n-d}{n} \cdot \frac{B}{kd}\,.$$

Assume that $k \leq d < n$, which is the case for non-degenerate PM-MBR codes. This implies that $\frac{n-d}{n} = 1 - \frac{d}{n} \leq 1 - \frac{k}{n}$ and $\frac{B}{kd} < 1$, and therefore

$$R' < 1 - \frac{k}{n} < R_{\mathrm{MBR}}\,,$$

where R_{MBR} is the PIR rate of the scheme we propose in the current work. We emphasize our improvement upon [24] with the numerical and asymptotic analyses proposed in Figure 2.



(a) PIR rate of both schemes, with a finite number of nodes n. We here set $n = 40$ and $k = 7$, and we plot the PIR rate versus d. For fixed values of n and k and varying $k+1 \leq d \leq n-1$, the scheme in [24] allows only a few admissible values of p, since $n = pk + d$ must hold. The larger the p, the larger the PIR rate of [24], but it remains bounded by the present scheme for every admissible value of p.

(b) PIR rate of both schemes, with an asymptotic number of nodes n. Each curve represents a distinct value of $k/n \in \{0.1, \ldots, 0.9\}$, and we plot the PIR rate versus d/n.

Fig. 2. Comparison between PIR rates of the multi-file PIR scheme in [24] and the PIR scheme in the present paper.

2) Comparison with the asymptotic capacities of scalar MDS codes: Since PM-MBR codes allow to retrieve files by contacting only k nodes among n, it is somewhat relevant to compare the proposed scheme with PIR schemes over $[n, k]$ MDS-coded data. We can also motivate this comparison by the following example.

Example 4. *In the PIR scheme presented in Example 3, the queried file has size* $(n-k)B = 27$*, while the user needs to download* $18 + 18 + 10 + 4 = 50$ *symbols. Hence, the PIR rate is* $27/50$*, which is larger than* $1 - k/n = 1/2$*, the PIR capacity of an* $[n, k]$ *MDS code, but smaller than* $1 - B/nd$*, the PIR capacity of an* $[nd, B]$ *MDS code.*

However, in the MBR construction, d *symbols are stored on a single server. Therefore, considering the storage code as an* $[nd, B]$ *linear code, a PIR protocol must resist to* some *sets of colluding nodes of size* d *(also known as* partial collusion*). In this setting, we can compare our construction to the conjectured PIR capacity* $1 - \frac{B+d-1}{nd}$ *of* $[nd, B]$ *linear codes with* full d*-collusion [14]. In the current example, the conjectured capacity is then* $1/2$*, which is again below the achieved rate.*

Lemma 3. *The PIR rate R_{MBR} of the scheme from Theorem 1 satisfies:*

$$1 - \frac{k}{n} \leq R_{\mathrm{MBR}} \leq 1 - \frac{B}{nd}\,.$$

Proof. If $1 \leq j \leq k$, it is clear that $n - k + j \leq n$. Using this trivial observation in Equation (5), we get

$$R_{\mathrm{MBR}} \geq \frac{(n-k)\left((d-k)k + k(k+1)/2\right)}{n(d-k)k + n\sum_{j=1}^{k} j} = \frac{n-k}{n} = 1 - \frac{k}{n}\,.$$

The right-hand-side inequality is a bit more technical to state. Using the expression of R_{MBR} given in Theorem 1, it is equivalent to prove that

$$\Delta \coloneqq (nd - B)(6dn - 3nk + 3n - k^2 + 1) - 3(n-k)(2d - k + 1)nd$$

is non-negative. A computation shows that:

$$\begin{aligned}
2\Delta &= (2nd - 2kd + k^2 - k)(6nd - 3nk + 3n - k^2 + 1) - 6nd(n-k)(2d-k+1) \\
&= 6nd((2nd - 2kd + k^2 - k) - (n-k)(2d-k+1)) - (2nd - 2kd + k^2 - k)(k^2 - 1 + 3nk - 3n) \\
&= 6n^2 d(k-1) - (2nd - 2kd + k^2 - k)(k-1)(3n + k + 1) \\
&= (k-1)[6n^2 d - (2nd - 2kd + k^2 - k)(3n + k + 1)]\,.
\end{aligned}$$

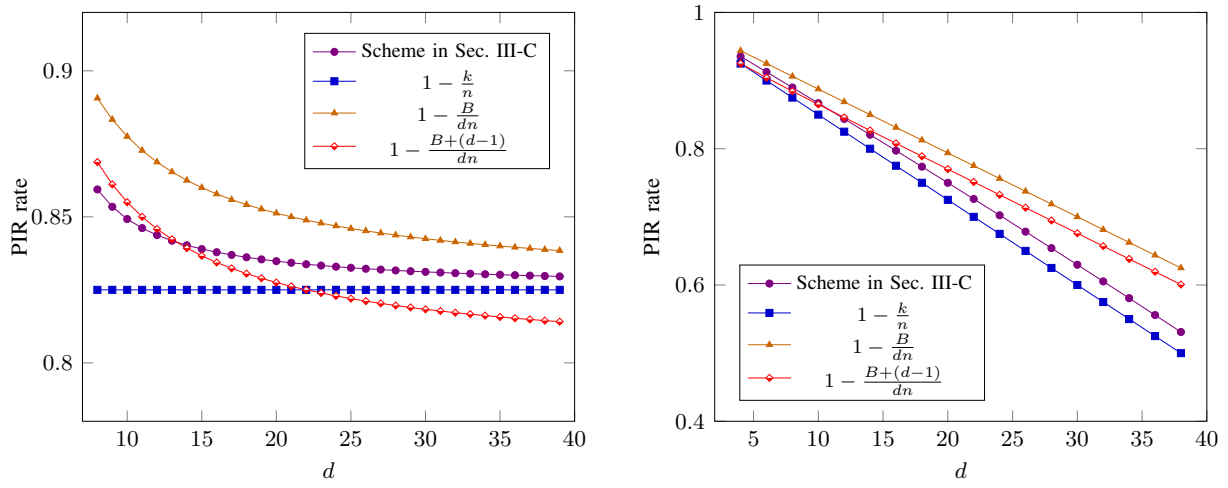
If $k = d$, then we get $2\Delta = k(k-1)(k+1)(n-(k+1)) \geq 0$ as long as $n \geq k+1$ which must hold for non-degenerated MBR codes.

If $d \geq k+1$, as it is for a non-trivial regenerating code, then we get

$$\begin{aligned}
\frac{2\Delta}{k-1} &= d((k-1)(4n + 2k + 3) + 2n + 2) - (k-1)(k+1)(3n + k + 1) \\
&\geq (k+1)((k-1)(4n + 2k + 3) + 2n + 2) - (k-1)(k+1)(3n + k + 1) \\
&\geq (k+1)(k-1)(n + k + 2) + 2(k+1)(n+1) \\
&\geq 0\,.
\end{aligned}$$

□

We can also model the n servers storing $\alpha = d$ symbols each as an nd-tuple of "virtual" or "sub"-servers storing one symbol each. In this setting, some d-tuples of servers collude with one another. For that reason, it is relevant to compare the PIR rate of this scheme with the (conjectured) capacity of a PIR scheme for an $[nd, B]$ MDS-coded storage system allowing collusions of servers of size up to $\alpha = d$. This conjectured capacity is $1 - \frac{B+d-1}{nd}$ [14]. Note that the assumption of full d-collusion is pessimistic since in this setting, not any d servers can collude, rather there exist disjoint sets of colluding servers that are known a priori, cf. [26].



(a) PIR rate versus d when $n = 40$ and fixed $k = 7$.

(b) PIR rate R_{MBR} versus d when $n = 40$, assuming $d = 2(k-1)$.

Fig. 3. PIR rate versus d when $n = 40$. Comparison between the PIR rate of the scheme in the current work, the capacity $1 - k/n$ of a PIR scheme for an $[n, k]$ MDS coded storage system with no collusion, the capacity $1 - B/nd$ of a PIR scheme for an $[nd, B]$ MDS coded storage system with no collusion, and the conjectured capacity $1-(B+d-1)/nd$ of a PIR scheme for an $[nd, B]$ MDS-coded storage system with full d collusion.

A comparison of the rate of the PIR scheme constructed in this paper with the other relevant capacity expressions of PIR schemes discussed in this section is shown in Figure 3 for different values of n, k and d. We can see that the achieved rate in our scheme is higher than the PIR capacity of an $[n, k]$ MDS code, and for a reasonably high value of d, the achievable PIR rate for the scheme described in Section III-C. As explained before, the achieved rate is always lower than the PIR capacity of an $[nd, B]$ MDS code.

IV. A PIR SCHEME IN THE MSR SETTING

We consider a regenerating code $\mathcal{C}$ attaining the MSR point. As explained in Section II-D2, we restrict our work on the setting $d = 2k - 2 = 2\alpha$ for simplicity. Hence $\mathcal{C}$ is also a linear code over $\mathbb{F}_q$ of length $nd = 2n\alpha$ and dimension $B = \alpha(\alpha + 1)$.

A. System setup

Similarly to the MBR setting, we consider a storage system $\boldsymbol{X}$ of F files $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^F$, each storing $B = \alpha(\alpha+1)$ information symbols. The symbols of the file $\boldsymbol{X}^f$, $1 \leq f \leq F$, are arranged into $S = n-2\alpha$

stripes, such that the message $\boldsymbol{M}^f$ can be written

$$\boldsymbol{M}^f = \begin{pmatrix} & 1 \leq i \leq 2\alpha \\ M^f[i,j,s], & 1 \leq j \leq \alpha \\ & 1 \leq s \leq S \end{pmatrix}.$$

By construction of the MSR code $\mathcal{C}$, for all $1 \leq f \leq F$ and all $1 \leq i, j \leq \alpha$, we have

$$\boldsymbol{M}^f[i,j,\cdot] = \boldsymbol{M}^f[j,i,\cdot] = \boldsymbol{M}^f[\alpha+i,j,\cdot]\,.$$

Moreover, for every j, s, f, the column $\boldsymbol{M}^f[\cdot,j,s] \in \mathbb{F}_q^{2\alpha}$ is encoded into a Reed-Solomon codeword $\boldsymbol{C}^f[\cdot,j,s] \in \mathrm{RS}_{2\alpha}(\boldsymbol{x})$ by

$$\boldsymbol{C}^f[\cdot,j,s] = \sum_{r=1}^{2\alpha} M^f[r,j,s]\boldsymbol{\gamma}_r,$$

where we recall that $\{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{2\alpha}\}$ denotes a suitable basis for sequences of Reed-Solomon codes (see Section II-A).

B. The PIR scheme

Assume the user wants to retrieve file $\boldsymbol{X}^{f_0}$ privately. We consider a 2α-tuple of queries $\boldsymbol{Q} = (\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{2\alpha})$ having the following form for $1 \leq \ell \leq 2\alpha$:

$$\boldsymbol{Q}_\ell = \begin{pmatrix} & 1 \leq i \leq n \\ Q_\ell^f[i,s], & 1 \leq s \leq S \\ & 1 \leq f \leq F \end{pmatrix}.$$

Once again, $\boldsymbol{Q}_\ell$ does *not* depend on the column index $j \in [1,\alpha]$, preventing to leak information on the requested file.

Generation of $\boldsymbol{Q}$. Similar to the MBR setting, queries $\boldsymbol{Q}$ are defined by $\boldsymbol{Q} \coloneqq \boldsymbol{D} + \boldsymbol{E}^{(f_0)}$ with $\boldsymbol{D}$ and $\boldsymbol{E}^{(f_0)}$ defined as follows.

1) For every ℓ, s, f, the random vector $\boldsymbol{D}_\ell^f[\cdot,s] \in \mathbb{F}_q^n$ is a word picked uniformly at random from the repetition code of length n.
2) The retrieval pattern $\boldsymbol{E}^{(f_0)}$ is defined by

$$E_\ell^{(f_0),f}[i,s] = \begin{cases} 1 & \text{if } f = f_0 \text{ and } n - i = \ell + s - 2 \pmod S, \\ 0 & \text{otherwise,} \end{cases}$$

for every $1 \leq \ell \leq 2\alpha$, $1 \leq i \leq n$, $1 \leq s \leq S$ and $1 \leq f \leq F$.

Server responses to queries. Given a column $1 \leq j \leq \alpha$, only servers S_i such that $2\alpha - 2j + 1 \leq i \leq n$ are required to send the subset of responses $R_\ell[i,j]$, for $1 \leq \ell \leq 2j$.

Reconstruction of $\boldsymbol{X}^{f_0}$. The recovery is run columnwise, from column α down to 1. In every step $1 \leq j \leq \alpha$, the goal is to retrieve $\boldsymbol{M}^{f_0}[\cdot,j,\cdot]$ as well as some random vectors. The recovery procedure

is identical to that of the first columns of the MBR case. Column α is retrieved using a classical PIR protocol on MDS codes, as in [13]. Here, the underlying storage code is $\mathrm{RS}_{2\alpha}(\boldsymbol{x})$. Similarly to the MBR case, the user retrieves pieces of the required file, along with some randomness. The collected symbols from column α (randomness and information symbols) can be reused in column $\alpha-1$ to again retrieve other pieces of the required file and associated randomness. This process is then repeated until retrieving the information from column 1. This iterative process reduces the number of total downloaded symbols to retrieve the required file $\boldsymbol{X}^{f_0}$, and consequently reduces the PIR rate. We refer to Lemma 5 for technical details.

We give a simple example to explain the scheme.

Example 5. *We use the* $(6,3,4)$ *PM-MSR regenerating code presented in Example 2, with* $\alpha=2$*. Files are divided into* $S=n-2\alpha=2$ *stripes, and the user sends* $2\alpha=4$ *vectors of queries:*

	Query 1	*Query* 2	*Query* 3	*Query* 4
Server S_1	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}$	$\boldsymbol{y}$
Server S_2	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}$	$\boldsymbol{y}$
Server S_3	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}+\boldsymbol{e}_{f_0,1}$	$\boldsymbol{y}+\boldsymbol{e}_{f_0,2}$
Server S_4	$\boldsymbol{u}$	$\boldsymbol{v}$	$\boldsymbol{w}+\boldsymbol{e}_{f_0,2}$	$\boldsymbol{y}+\boldsymbol{e}_{f_0,1}$
Server S_5	$\boldsymbol{u}+\boldsymbol{e}_{f_0,1}$	$\boldsymbol{v}+\boldsymbol{e}_{f_0,2}$	$\boldsymbol{w}$	$\boldsymbol{y}$
Server S_6	$\boldsymbol{u}+\boldsymbol{e}_{f_0,2}$	$\boldsymbol{v}+\boldsymbol{e}_{f_0,1}$	$\boldsymbol{w}$	$\boldsymbol{y}$

The vector $\boldsymbol{e}_{f_0,s_0}\in\mathbb{F}_q^{F\times(n-2\alpha)}$ *is the all zero vector with a single* 1 *in position* (f_0,s_0)*,* i.e.*, indicating stripe* s_0 *from file* $\boldsymbol{X}^{f_0}$*. Vectors* $\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{y}\in\mathbb{F}_q^{F\times S}$ *are random vectors.*

The servers project the data stored in column 2 *on all the queries. Servers* S_1 *and* S_2 *do not respond to any other queries. Servers* $S_3,\dots,S_6$ *project only the first* 2 *queries on the data stored in the first column.*

C. Proofs

For $1\le j\le\alpha$, we define the $2j$-dimensional code

$$\mathcal{C}_j \coloneqq \mathrm{RS}_j(\boldsymbol{x})+\langle\boldsymbol{x}^\alpha\rangle\star\mathrm{RS}_j(\boldsymbol{x})\subseteq\mathbb{F}_q^n\,.$$

Lemma 4. *There exists a sequence* $I_1\subset\cdots\subset I_\alpha\subset[1,n]$ *such that, for every* $1\le j\le\alpha$*,* I_j *is an information set for the code* $\mathcal{C}_j$*.*

Proof. We prove the result inductively. First notice that $\mathcal{C}_\alpha = \mathrm{RS}_{2\alpha}(\boldsymbol{x})$, hence one can choose any 2α-subset for I_α. Then, it is sufficient to notice that for every $2 \leq j \leq \alpha$, we have $\mathcal{C}_{j-1} \subset \mathcal{C}_j$. Hence, an information set I_j for $\mathcal{C}_j$ contains an information set for $\mathcal{C}_{j-1}$. □

The previous lemma allows us to make the following assumption: after reordering the servers (*i.e.* the evaluation points $\boldsymbol{x}$), we can assume that $I_j = [2\alpha - 2j + 1, 2\alpha]$ for every $1 \leq j \leq \alpha$. Moreover, we define the code $\mathcal{A}_j \subseteq \mathbb{F}_q^{n-2\alpha+2j}$ as the puncturing of $\mathcal{C}_j$ on its $(2\alpha - 2j)$ first coordinates. The code $\mathcal{A}_j$ has length $n - 2\alpha + 2j$ and dimension $2j$, and by the chosen order of coordinates, its $2j$ first coordinates form an information set.

Lemma 5. *Let $1 \leq j \leq \alpha$. For every $1 \leq \ell \leq 2j$, we denote*

$$\boldsymbol{R}_\ell[\cdot, j] \coloneqq (R_\ell[2\alpha - 2j + 1, j], \ldots, R_\ell[n, j]) \in \mathbb{F}_q^{n-2\alpha+2j}.$$

Then, conditioned on $(\boldsymbol{R}_1[\cdot, j], \ldots, \boldsymbol{R}_{2j}[\cdot, j])$ and on

$$\sum_{s,f} M^f[r, j, s] \boldsymbol{D}_\ell^f[\cdot, s] \quad \text{,for all } \; j + 1 \leq r \leq \alpha, \quad 1 \leq \ell \leq 2j\,, \tag{7}$$

the following is determined:

- *the piece $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ of the desired file;*
- *the random vectors $\sum_{s,f} M^f[r, j, s] \boldsymbol{D}_\ell^f[\cdot, s] \in \mathbb{F}_q^n$, for all $1 \leq r \leq 2\alpha$ and every $1 \leq \ell \leq 2j$.*

Proof. The proof is very similar to the one of Lemma 2. Let us fix $1 \leq \ell \leq j$. The user can build

$$\boldsymbol{R}_\ell[\cdot, j] \coloneqq (R_\ell[2\alpha - 2j + 1, j], \ldots, R_\ell[n, j]) = \boldsymbol{A}_\ell[\cdot, j] + \boldsymbol{B}_\ell[\cdot, j]\,,$$

where $\boldsymbol{A}_\ell[\cdot, j]$ and $\boldsymbol{B}_\ell[\cdot, j]$ are defined as in Lemma 1.

Denote $J_1 \coloneqq [0, j-1] \cup [\alpha, \alpha + j - 1]$ and $J_2 \coloneqq [0, 2\alpha - 1] \setminus J_1$. Both J_1 and J_2 are publicly known to the user and the servers, as they only depend on the parameters of the scheme.

Define $\boldsymbol{\gamma}_r \coloneqq (x_{2\alpha-2j+1}^r, \ldots, x_n^r) \in \mathbb{F}_q^{n-2\alpha+2j}$, for $0 \leq r \leq 2\alpha - 1$. It is clear that $\{\boldsymbol{\gamma}_r, r \in J_1\}$ is a basis of the code $\mathcal{A}_j$ defined above. One can rewrite $\boldsymbol{A}_\ell[\cdot, j] \in \mathbb{F}_q^{n-2\alpha+2j}$ as follows:

$$\begin{aligned}
\boldsymbol{A}_\ell[\cdot, j] &= \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{C}^f[\cdot, j, s] \\
&= \sum_{s,f} \boldsymbol{D}_\ell^f[\cdot, s] \star \left(\sum_{r=1}^{d} M^f[r, j, s] \boldsymbol{\gamma}_r \right) \\
&= \sum_{r \in J_1} \sum_{s,f} M^f[r, j, s]\ \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{\gamma}_r + \sum_{r \in J_2} \sum_{s,f} M^f[r, j, s]\ \boldsymbol{D}_\ell^f[\cdot, s] \star \boldsymbol{\gamma}_r.
\end{aligned}$$

Therefore, using random vectors given in (7), the vector

$$\boldsymbol{A}'_\ell[\cdot, j] \coloneqq \sum_{r\in J_2} \left(\sum_{s,f} M^f[r, j, s]\ \boldsymbol{D}^f_\ell[\cdot, s] \right) \star \boldsymbol{\gamma}_r$$

can be constructed by the user. Recall that for any file X^f,

$$\boldsymbol{M}^f[r, j, \cdot] = \boldsymbol{M}^f[j, r, \cdot] = \boldsymbol{M}^f[\alpha + r, j, \cdot]$$

for every $1 \le r \le \alpha$. Hence, the user is able to construct

$$\boldsymbol{R}''_\ell[\cdot, j] \coloneqq \boldsymbol{R}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j] = (\boldsymbol{A}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j]) + \boldsymbol{B}_\ell[\cdot, j]$$

and, by definition of J_1, we see that $\boldsymbol{A}''_\ell[\cdot, j] \coloneqq \boldsymbol{A}_\ell[\cdot, j] - \boldsymbol{A}'_\ell[\cdot, j]$ lies in $\mathcal{A}_j$. We remark that, once again, the vector $\boldsymbol{B}_\ell[\cdot, j] \in \mathbb{F}_q^{n-k+j}$ is supported on $[2\alpha + 1, n]$. According to the discussion preceding the lemma, the interval $I_j = [2\alpha - 2j + 1, 2\alpha]$ is an information set for $\mathcal{C}_j$. Therefore the user can recover $\boldsymbol{A}''_\ell[\cdot, j]$ and $\boldsymbol{B}_\ell[\cdot, j]$ from $\boldsymbol{R}''_\ell[\cdot, j]$.

Finally, the recovery of $\boldsymbol{M}^{f_0}[\cdot, j, \cdot]$ and of random elements $\sum_{s,f} M^f[r, j, s]\lambda_{\ell,s,f}$ is identical to Lemma 2. □

Theorem 2. *The scheme proposed in Section IV-B is secure against non-colluding servers. Its PIR rate is*

$$R_{\mathrm{MSR}} = \frac{3(n - 2\alpha)}{3n - 2\alpha + 2}.$$

Proof. We have seen in Lemma 5 that the proposed scheme reconstructs the correct file. Similarly to the MBR case, the scheme is private if servers do not collude. Let us compute the PIR rate.

The desired file consists of $(n - 2\alpha)B = \alpha(\alpha + 1)(n - 2\alpha)$ symbols. For column $1 \le j \le \alpha$, the number of downloaded symbols is $2j \times (n - 2\alpha + 2j)$. Hence the PIR rate of the scheme is given by

$$\begin{aligned} R_{\mathrm{MSR}} &= \frac{\alpha(\alpha + 1)(n - 2\alpha)}{\sum_{j=1}^{\alpha} 2j(n - 2\alpha + 2j)} \\ &= \frac{\alpha(\alpha + 1)(n - 2\alpha)}{n\alpha(\alpha + 1) - 4\sum_{j=1}^{\alpha} j(\alpha - j)} \\ &= \frac{\alpha(\alpha + 1)(n - 2\alpha)}{n\alpha(\alpha + 1) - \frac{2}{3}\alpha(\alpha + 1)(\alpha - 1)} \\ &= \frac{3(n - 2\alpha)}{3n - 2\alpha + 2} \\ &= 1 - \frac{4\alpha + 2}{3n - 2\alpha + 2}\,. \end{aligned}$$

□

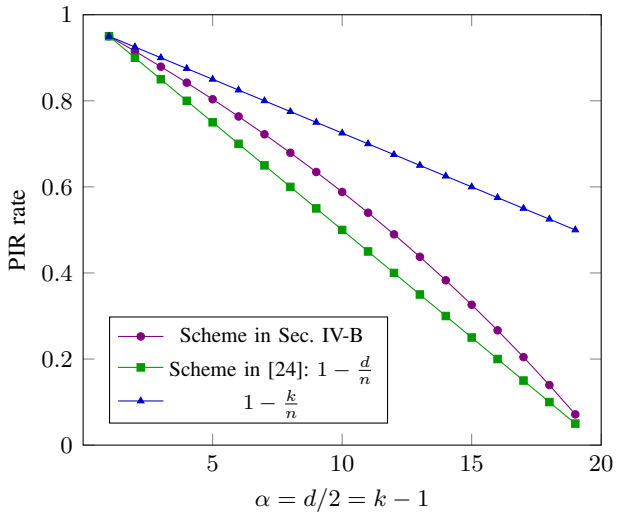


Fig. 4. PIR rate versus α in the MSR case, for fixed $n = 40$. Recall that $2\alpha = d = 2k - 2$ must hold.

D. On the PIR rate in the MSR case

In our simplified setting, it must hold that $\alpha = d/2 = k - 1$. The PIR rate of the proposed scheme is then

$$R_{\mathrm{MSR}} = 1 - \frac{4\alpha + 2}{3n - 2\alpha + 2}\,.$$

Dorkson and Ng [24] give a multi-file PIR scheme for the same MSR codes, with a PIR rate of $\frac{\alpha(n-d)}{\alpha n} = 1 - d/n$. We prove in the following lemma that the PIR rate of our construction improves upon this rate.

Lemma 6. *Let $1 \leq \alpha \leq n/2$ and assume that $n \geq 6$ or $\alpha \geq 3$. Then:*

$$1 - \frac{d}{n} \leq R_{\mathrm{MSR}} \leq 1 - \frac{k}{n}\,.$$

Proof. For the left-hand side inequality, we need to prove that $d/n \geq (2d + 2)(3n - 2d + 2)$. A simple computation shows it is equivalent to $(d - 2)(n - d) \geq 0$, which holds as long as $\alpha = d/2 \geq 1$.

Similarly, the right-hand side inequality $R_{\mathrm{MSR}} \leq 1 - \frac{k}{n}$ holds if and only if $\frac{n(\alpha-3)}{2} + (\alpha+1)(\alpha-1) \geq 0$, which proves our result. □

V. Conclusion

In this paper, we construct PIR schemes for the product matrix constructions in the MBR and MSR settings. The schemes use the symmetric properties of the PM codes in order to increase the PIR rate. For the PM-MBR setting, we achieve a PIR rate that is better than $1 - k/n$, *i.e.*, larger than the PIR

capacity of an $[n, k]$ MDS coded storage system. As for the PM-MSR setting, we achieve a PIR rate between $1 - d/n$, *i.e.*, the PIR capacity of an $[n, d]$ MDS code, and $1 - k/n$.

A possible further work on the topic would be to consider colluding servers. A natural idea is to adapt the constructions of Freij-Hollanti *et al.* [14], [16], by replacing the repetition code where random vectors $\boldsymbol{D}_\ell^f[\cdot, s]$ are picked, by a Reed-Solomon code of higher dimension. However, the extraction of the randomness — necessary to decrease the communication cost of our schemes — cannot be done as easily as in the non-colluding case, because projected random symbols interfere with themselves.

ACKNOWLEDGMENTS

The work of J. Lavauzelle is partially funded by French ANR-15-CE39-0013-01 "Manta".
The work of R. Tajeddine and C. Hollanti is supported in part by the Academy of Finland, under grants #276031, #282938, and #303819 to C. Hollanti, and by the Technical University of Munich – Institute for Advanced Study, funded by the German Excellence Initiative and the EU 7th Framework Programme under grant agreement #291763, via a *Hans Fischer Fellowship* held by C. Hollanti.
The work of R. Freij-Hollanti is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant WA3907/1-1.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private Information Retrieval," in *IEEE Symposium on Foundations of Computer Science*, pp. 41–50, 1995.

[2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private Information Retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[3] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval," *IEEE Trans. Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.

[4] H. Sun and S. A. Jafar, "The Capacity of Robust Private Information Retrieval With Colluding Databases," *IEEE Trans. Information Theory*, vol. 64, no. 4, pp. 2361–2370, 2018.

[5] S. Yekhanin, "Private Information Retrieval," *Communications of the ACM*, vol. 53, no. 4, pp. 68–73, 2010.

[6] A. Beimel and Y. Ishai, "Information-Theoretic Private Information Retrieval: A Unified Construction," in *Automata, Languages and Programming*, pp. 912–926, Springer, 2001.

[7] 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 *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 261–270, IEEE, 2002.

[8] N. Shah, K. Rashmi, and K. Ramchandran, "One Extra Bit of Download Ensures Perfectly Private Information Retrieval," in *2014 IEEE International Symposium on Information Theory*, pp. 856–860, IEEE, 2014.

[9] T. Chan, S.-W. Ho, and H. Yamamoto, "Private Information Retrieval for Coded Storage," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2842–2846, IEEE, June 2015.

[10] K. A. Banawan and S. Ulukus, "The Capacity of Private Information Retrieval From Coded Databases," *IEEE Trans. Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.

[11] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for Distributed PIR with Low Storage Overhead," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2852–2856, June 2015.

[12] S. Blackburn and T. Etzion, "PIR Array Codes with Optimal PIR Rate," *arXiv preprint arXiv:1607.00235*, 2016.

[13] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private Information Retrieval from MDS Coded Data in Distributed Storage Systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[14] 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 Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[15] S. Kumar, H.-Y. Lin, E. Rosnes, and A. Graell I Amat, "Achieving Private Information Retrieval Capacity in Distributed Storage using an Arbitrary Linear Code," *arXiv preprint arXiv:1712.03898*, 2017.

[16] 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," *IEEE Transactions on Information Theory*, 2018.

[17] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sep. 2010.

[18] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal Exact-Regenerating Codes for Distributed Storage at the MSR and MBR Points via a Product-Matrix Construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[19] C. Suh and K. Ramchandran, "Exact-Repair MDS Code Construction Using Interference Alignment," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1425–1442, 2011.

[20] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, V. Lalitha, O. O. Koyluoglu, P. V. Kumar, and S. Vishwanath, "Explicit MBR All-symbol Locality Codes," in *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*, pp. 504–508, IEEE, 2013.

[21] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of High-Rate Minimum Storage Regenerating Codes over Small Fields," in *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pp. 61–65, IEEE, 2016.

[22] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "Information-Theoretically Secure Regenerating Codes for Distributed Storage," in *Proceedings of the Global Communications Conference, GLOBECOM 2011, 5-9 December 2011, Houston, Texas, USA*, pp. 1–5, IEEE, 2011.

[23] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing Dynamic Distributed Storage Systems against Eavesdropping and Adversarial Attacks," *IEEE Transactions on Information Theory*, vol. 58, pp. 6734–6753, March 2012.

[24] C. Dorkson and S. Ng, "Multi-Message Private Information Retrieval using Product-Matrix MSR and MBR Codes," *CoRR*, vol. abs/1808.02023, 2018.

[25] C. Dorkson and S. Ng, "Private Information Retrieval using Product-Matrix Minimum Storage Regenerating Codes," *CoRR*, vol. abs/1805.07190, 2018.

[26] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. El Rouayheb, "Private Information Retrieval Schemes for Coded Data with Arbitrary Collusion Patterns," in *2017 IEEE International Symposium on Information Theory*, pp. 1908–1912, IEEE, 2017.