Improved Private Information Retrieval for Coded Storage From Code Decomposition

Citation for the original published paper (version of record):
Lin, H., Kumar, S., Rosnes, E. et al (2019)
Improved Private Information Retrieval for Coded Storage From Code Decomposition
IEEE Information Theory Workshop: 729-733
http://dx.doi.org/10.1109/ITW44776.2019.8989190

Improved Private Information Retrieval for Coded Storage From Code Decomposition

(Invited Paper)

Hsuan-Yin Lin[†], Siddhartha Kumar[†], Eirik Rosnes[†], and Alexandre Graell i Amat[‡][†]
[†]Simula UiB, N–5008 Bergen, Norway
[‡]Department of Electrical Engineering, Chalmers University of Technology, SE–41296 Gothenburg, Sweden

***Abstract*—We consider private information retrieval (PIR) for distributed storage systems with noncolluding nodes where data is stored using a non maximum distance separable (MDS) linear code. Recently, it was shown that when data is stored using certain non-MDS codes, the *MDS-PIR capacity* can be achieved, and is indeed the capacity of the system. In this paper, for storage codes not belonging to this class, we present a heuristic algorithm for their decomposition into punctured subcodes and a PIR protocol based on these punctured subcodes. The code decomposition is guided by the generalized Hamming weights of the storage code. We show that the proposed PIR protocol can achieve a larger PIR rate than that of all existing PIR protocols.**

I. Introduction

The notion of private information retrieval (PIR) was first introduced by Chor *et al.* in their seminal paper [1]. It is a concept where a user wishes to retrieve a data item, stored on multiple servers in a distributed manner, without revealing the identity of the requested data item to the servers. A protocol that allows a user to do so is referred to as a PIR protocol, and its efficiency is measured in terms of the total communication occurring between the user and the servers. In [1], the authors introduced a PIR protocol that provides privacy when data is replicated and stored on n servers. From an information-theoretic perspective, the data (or files) tend to be much larger than the overall size of the queries sent to all servers, implying that the download cost is much larger than the upload cost. Therefore by and large, the information theory community has been focused on designing PIR protocols that maximize the PIR rate, i.e., the amount of information retrieved per downloaded symbol [2]–[12].

In [5], the authors derived the PIR capacity, i.e., the maximum PIR rate for any protocol when data is replicated over n servers. The work in [4] was the first to consider the scenario where data is stored using a maximum distance separable (MDS) code. The corresponding PIR capacity, referred to as the MDS-PIR capacity, was derived in [7]. Incidentally, the protocol from [4] achieved the *asymptotic* MDS-PIR capacity, i.e., for a number of files that tends to infinity. In [8], it was shown that the MDS-PIR capacity can also be achieved for certain non-MDS linear codes, and it was subsequently shown that for these codes the PIR capacity is indeed equal to the MDS-PIR capacity [9]. In general, however, the PIR capacity for arbitrary coded DSSs is not known. The aforementioned papers assume that servers do not collude to determine the identity of the requested file. In [6], this assumption was relaxed and the PIR capacity for the case of replication was derived. In [11], a PIR protocol for coded data stored on colluding servers was presented. The protocol in [11] was independently improved in [8] and [12].

In this paper, we present a PIR protocol for coded DSSs where data is stored using codes that do not allow to achieve the MDS-PIR capacity under Protocol 1 in [8]. In the following, we refer to codes that achieve the MDS-PIR capacity under Protocol 1 in [8] (and also the PIR capacity [9]) as *MDS-PIR capacity-achieving* codes. The proposed protocol is similar to Protocol C in [9] with an improved and novel code decomposition method. The code decomposition is guided by the generalized Hamming weights [13] of the storage code and is heuristic in nature. A special feature of the proposed protocol is that it is asymmetric in the number of responses across the servers. Moreover, it is based on the punctured subcodes obtained through the code decomposition method. We show that the proposed protocol achieves a PIR rate larger than or equal to that of Protocol A in [9] (the best general protocol from the literature) if the code decomposition gives a series of punctured MDS-PIR capacity-achieving subcodes, which we conjecture to always be the case. The conjecture has been verified by exhaustive search for small code parameters.

II. Preliminaries and System Model

A. Notation and Definitions

We denote by $\mathbb{N}$ the set of all positive integers, $\mathbb{N}_a \triangleq \{1, 2, \ldots, a\}$, and $\mathbb{N}_{a:b} \triangleq \{a, a+1, \ldots, b\}$ for $a, b \in \mathbb{N}$, $a \leq b$. Vectors are denoted by lowercase bold letters, matrices by uppercase bold letters, and sets by calligraphic uppercase letters, e.g., $\boldsymbol{x}$, $\boldsymbol{X}$, and $\mathcal{X}$, respectively. We denote a submatrix of $\boldsymbol{X}$ that is restricted in columns by the set $\mathcal{J}$ and in rows by the set $\mathcal{I}$ by $\boldsymbol{X}|_{\mathcal{J}}^{\mathcal{I}}$. $\mathsf{LCM}(n_1, n_2, \ldots, n_a)$ gives the lowest common multiple of a positive integers $n_1, n_2, \ldots, n_a$. The function $\mathsf{H}(\cdot)$ represents the entropy of its argument and $\mathsf{I}(\cdot\,;\cdot)$ denotes the mutual information. $(\cdot)^{\mathsf{T}}$ denotes the transpose of its argument. We use the customary code parameters $[n, k]$ to denote a code $\mathcal{C}$ over the finite field $\mathsf{GF}(q)$ of blocklength n and dimension k. A generator matrix of $\mathcal{C}$ is denoted by $\boldsymbol{G}^{\mathcal{C}}$, while $\mathcal{C}^{\boldsymbol{G}}$ represents the corresponding code generated by $\boldsymbol{G}$. $\chi(\boldsymbol{x})$ denotes the support of vector $\boldsymbol{x}$ and $\chi(\mathcal{C})$ the support of code $\mathcal{C}$, defined as the set of coordinates where not all codewords are zero.

B. System Model

We consider a DSS that stores f files $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(f)}$, where each file $\boldsymbol{X}^{(m)} = (x_{i,l}^{(m)})$, $m \in \mathbb{N}_f$, can be seen as a $\beta \times k$ matrix over GF(q) with $\beta, k \in \mathbb{N}$. Assume that each entry $x_{i,l}^{(m)}$ of $\boldsymbol{X}^{(m)}$ is chosen independently and uniformly at random from GF(q), $m \in \mathbb{N}_f$. Thus, $\mathsf{H}(\boldsymbol{X}^{(m)}) = \mathsf{L}$, $\forall\, m \in \mathbb{N}_f$, and $\mathsf{H}(\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(f)}) = f\mathsf{L}$ in q-ary units, where $\mathsf{L} \triangleq \beta \cdot k$. 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 GF(q) into a length-n codeword $\boldsymbol{c}_i^{(m)} = (c_{i,1}^{(m)}, \ldots, c_{i,n}^{(m)})$. The βf generated codewords $\boldsymbol{c}_i^{(m)}$ are then arranged in the array $\boldsymbol{C} = \bigl((\boldsymbol{C}^{(1)})^\mathsf{T} | \ldots | (\boldsymbol{C}^{(f)})^\mathsf{T}\bigr)^\mathsf{T}$ of dimensions $\beta f \times n$, where $\boldsymbol{C}^{(m)} = \bigl((\boldsymbol{c}_1^{(m)})^\mathsf{T} | \ldots | (\boldsymbol{c}_\beta^{(m)})^\mathsf{T}\bigr)^\mathsf{T}$. The code symbols $c_{1,l}^{(m)}, \ldots, c_{\beta,l}^{(m)}$, $m \in \mathbb{N}_f$, for all f files are stored on the l-th storage node, $l \in \mathbb{N}_n$.

C. Privacy Model

To retrieve file $\boldsymbol{X}^{(m)}$ from the DSS, the user sends a random query $Q_l^{(m)}$ to the l-th node for all $l \in \mathbb{N}_n$. In response to the received query, node l sends the response $A_l^{(m)}$ back to the user. $A_l^{(m)}$ is a deterministic function of $Q_l^{(m)}$ and the code symbols stored in the node.

Definition 1. *Consider a DSS with n noncolluding nodes storing f files. A user who wishes to retrieve the m-th file sends the queries $Q_l^{(m)}$, $l \in \mathbb{N}_n$, to the storage nodes, which return the responses $A_l^{(m)}$. This scheme achieves perfect information-theoretic PIR if and only if*

Privacy:

$$\mathsf{I}\bigl(m; Q_l^{(m)}, A_l^{(m)}, \boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(f)}\bigr) = 0,\ \forall\, l \in \mathbb{N}_n,$$

Recovery:

$$\mathsf{H}\bigl(\boldsymbol{X}^{(m)} \,\big|\, A_1^{(m)}, \ldots, A_n^{(m)}, Q_1^{(m)}, \ldots, Q_n^{(m)}\bigr) = 0.$$

D. PIR Rate and Capacity

Definition 2. *The PIR rate of a PIR protocol, denoted by R, is the amount of information retrieved per downloaded symbol, i.e., $\mathsf{R} \triangleq \frac{\beta k}{\mathsf{D}}$, where D is the expected total number of downloaded symbols for the retrieval of a single file.*

We will write $\mathsf{R}(\mathcal{C})$ to highlight that the PIR rate depends on the underlying storage code $\mathcal{C}$. It was shown in [7] that for the noncolluding case and for a given number of files f stored using an $[n,k]$ MDS code, the MDS-PIR capacity is

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

where superscript "$[n,k]$" indicates the code parameters of the underlying code. When f tends to infinity, (1) reduces to

$$\mathsf{C}_\infty^{[n,k]} \triangleq \lim_{f \to \infty} \mathsf{C}_f^{[n,k]} = 1 - \frac{k}{n}, \tag{2}$$

which we refer to as the asymptotic MDS-PIR capacity. For the case of non-MDS linear codes, the PIR capacity is known only for a certain class of codes [9, Thm. 3], defined next.

E. MDS-PIR Capacity-Achieving Codes

In [8], two symmetric PIR protocols for coded DSSs, named Protocol 1 and Protocol 2, were proposed and shown to achieve the MDS-PIR capacity and the asymptotic MDS-PIR capacity, respectively, for certain important classes of non-MDS codes. Their PIR rates depend on the following property of the underlying storage code $\mathcal{C}$.

Definition 3. *Let $\mathcal{C}$ be an arbitrary $[n,k]$ code. A $\nu \times n$ binary matrix $\boldsymbol{\Lambda}_{\kappa,\nu}(\mathcal{C})$ 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 $\boldsymbol{\Lambda}_{\kappa,\nu}$ 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 [14, p. 4].*

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.

For the sake of simplicity, throughout this paper we denote by $\mathcal{F}(\mathcal{C}) \triangleq \{(\kappa,\nu) \colon \exists\, \boldsymbol{\Lambda}_{\kappa,\nu}(\mathcal{C}) \text{ for } \mathcal{C}\}$ the collection of all possible valid pairs (κ,ν) that admit the existence of a PIR achievable rate matrix. The following lemma gives a lower bound to the fraction $\frac{\kappa}{\nu}$ of a PIR achievable rate matrix $\boldsymbol{\Lambda}_{\kappa,\nu}(\mathcal{C})$ for $\mathcal{C}$.

Lemma 1 ([8, Lem. 2])**.** *Let $\mathcal{C}$ be an $[n,k]$ code. Define $\tau^* \triangleq \min_{(\kappa,\nu) \in \mathcal{F}(\mathcal{C})}\bigl\{\frac{\kappa}{\nu}\bigr\}$. Then, $\tau^* \geq \frac{k}{n}$.*

The following theorem gives the PIR rate of an asymmetric PIR protocol (Protocol A in [9]) for coded DSSs, which is in general larger than the PIR rate of Protocol 1 from [8].

Theorem 1 ([9, Thm. 4])**.** *Consider a DSS that uses an $[n,k]$ code $\mathcal{C}$ to store f files. Then, the PIR rate*

$$\mathsf{R}_{f,\mathsf{A}}(\mathcal{C}) \triangleq (1 - \tau^*)\bigl[1 - (\tau^*)^f\bigr]^{-1} \tag{3}$$

is achievable.

In (3), we use subscript A to indicate that the PIR rate is achievable by Protocol A in [9]. Define $\mathsf{R}_{\infty,\mathsf{A}}(\mathcal{C})$ as the limit of $\mathsf{R}_{f,\mathsf{A}}(\mathcal{C})$ as the number of files f tends to infinity, i.e., $\mathsf{R}_{\infty,\mathsf{A}}(\mathcal{C}) \triangleq \lim_{f \to \infty} \mathsf{R}_{f,\mathsf{A}}(\mathcal{C}) = 1 - \tau^*$.

Corollary 1. *If $\tau^* = \frac{k}{n}$ for an $[n,k]$ code $\mathcal{C}$, then the MDS-PIR capacity (1) is achievable.*

This gives rise to the following definition.

Definition 4. *An $[n,k]$ code $\mathcal{C}$ with $\tau^* = \frac{k}{n}$ is referred to as an* MDS-PIR capacity-achieving *code.*

In [9], it was proved that the PIR capacity for MDS-PIR capacity-achieving codes is equal to the MDS-PIR capacity.

Theorem 2 ([9, Thm. 3])**.** *Consider a DSS that uses an $[n,k]$ MDS-PIR capacity-achieving code $\mathcal{C}$ to store f files. Then, its PIR capacity is equal to the MDS-PIR capacity $\mathsf{C}_f^{[n,k]}$ in* (1)*.*

Definition 5 (Generalized Hamming weight [13])**.** *The r-th generalized Hamming weight of an $[n,k]$ code $\mathcal{C}$, denoted by*

$d_r(\mathcal{C})$, $r \in \mathbb{N}_k$, *is defined as the cardinality of the smallest support of an* r*-dimensional subcode of* $\mathcal{C}$*, i.e.,*

$$d_r(\mathcal{C}) \triangleq \min\{|\chi(\mathcal{D})| : \mathcal{D} \textit{ is an } [n,r] \textit{ subcode of } \mathcal{C}\}.$$

The following theorem from [8, Thm. 3] provides a necessary condition for codes to be MDS-PIR capacity-achieving (under Protocol 1 from [8]), re-written to fit the heuristic algorithm that will be proposed below in Section III.

Theorem 3. *Let* $\mathcal{C}$ *be an* $[n,k]$ *code. If* $\tau^* = \frac{k}{n}$*, then*

$$\max_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C})}\right\} \leq \frac{k}{n}. \tag{4}$$

Next, we will make use of Lemma 1 and the following lemma based on [8, Thm. 3] to give a new conjecture related to the heuristic algorithm that will be proposed below in Section III.

Lemma 2. *Let* $\mathcal{C}$ *be an* $[n,k]$ *code. Then,*

$$\max_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C})}\right\} \leq \tau^*.$$

Conjecture 1. *For any* $[n,k]$ *code* $\mathcal{C}$*, it holds that*

$$\max_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C})}\right\} = \tau^*.$$

By performing an exhaustive search we have verified that Conjecture 1 is true for all binary linear codes with dimension $k < n$ and blocklength $n \in \mathbb{N}_{2:10}$.

Observe that if Conjecture 1 is true, using Lemma 1 we can show that the assertion from [8, Conj. 1] based on generalized Hamming weights follows. More precisely, for an $[n,k]$ code $\mathcal{C}$, if $d_r(\mathcal{C}) \geq \frac{n}{k}r$, $\forall\, r \in \mathbb{N}_k$, since

$$\frac{k}{n} \geq \max_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C})}\right\} = \tau^* \geq \frac{k}{n},$$

then τ^* must be equal to $\frac{k}{n}$.

III. An Improved PIR Protocol From Code Decomposition

In this section, we provide a code-dependent asymmetric PIR protocol for non-MDS-PIR capacity-achieving codes, more precisely, codes that do not satisfy (4). The proposed protocol is similar to Protocol C from [9], but with an improved explicit code decomposition method. In particular, it is based on Protocol 2 in [8, Sec. V] that works for DSSs where data is stored using an arbitrary linear code. Note that if $\tau^* = \frac{k}{n}$ for an $[n,k]$ code $\mathcal{C}$, then the asymptotic MDS-PIR capacity in (2) is achieved by Protocol 2. Furthermore, the number of stripes β required by Protocol 2 can be chosen to be equal to $\mathsf{LCM}(k, n-k)/k$ and the complete retrievability of the requested file is guaranteed when the user queries the nodes $\mathsf{LCM}(k, n-k)/(n-k)$ times.

The proposed protocol is based on the key idea of decomposing a storage code $\mathcal{C}$ into P $[n_p, k_p]$ punctured subcodes $\mathcal{C}_p$, $p \in \mathbb{N}_{\mathsf{P}}$, using Algorithm 1. (See the output of Algorithm 1, where we obtain a generator matrix $\boldsymbol{G}_{\mathsf{PIR}}$ of an equivalent storage code $\mathcal{C}_{\mathsf{PIR}}$.) The overall protocol is then implemented by applying Protocol 2 (as a subprotocol) on each punctured subcode $\mathcal{C}_p = \mathcal{C}^{\boldsymbol{G}_p}$, with generator matrix $\boldsymbol{G}_p$, $p \in \mathbb{N}_{\mathsf{P}}$.

Suppose that the user requests file $\boldsymbol{X}^{(m)}$. Then, the protocol involves executing Protocol 2 with the aim to retrieve code symbols from β_p stripes pertaining to $\boldsymbol{X}^{(m)}$ from the set of nodes $\mathbb{N}_{n_1+\cdots+n_{p-1}+1:n_1+\cdots+n_p}$. This is achieved by processing Protocol 2 for the punctured subcode $\mathcal{C}^{\boldsymbol{G}_p}$. Overall, Protocol 2 is repeated β/β_p times to recover all length-k_p requested substripes. To ensure that code symbols from all β stripes across the n nodes are downloaded, we set $\beta = \mathsf{LCM}(\beta_1, \ldots, \beta_{\mathsf{P}})$. In the following, we denote by D_p the total number of subqueries of Protocol 2 for $\mathcal{C}^{\boldsymbol{G}_p}$, $p \in \mathbb{N}_{\mathsf{P}}$.

Similar to [8], the protocol utilizes n *interference symbols* from the n responses. An interference symbol can be defined through a summation as [8]

$$I_l^{(d)} \triangleq \sum_{m=1}^{f} \sum_{j=(m-1)\beta+1}^{m\beta} u_{d,j} c_{j-(m-1)\beta,l}^{(m)},$$

where $l \in \mathbb{N}_n$, $d \in \mathbb{N}_{\mathsf{D}_{\max}}$ for $\mathsf{D}_{\max} \triangleq \max_{p\in\mathbb{N}_{\mathsf{P}}}\{\frac{\beta}{\beta_p}\mathsf{D}_p\}$, and the symbols $u_{d,j}$ are chosen independently and uniformly at random from the same field as the code symbols. Note that for $l \in \mathbb{N}_{k+1:n}$ the interference symbols $I_l^{(d)}$ are linear functions of $I_1^{(d)}, \ldots, I_k^{(d)}$. Perfect retrievability is guaranteed when the user is able to retrieve enough interference symbols $I_l^{(d)}$, $l \in \mathbb{N}_n$, from each subresponse. The P subprotocols work in conjunction to retrieve all required interference symbols from all subresponses. We remark that it is not always necessary that the P subprotocols obtain all interference symbols, in which case the user sends additional queries to the appropriate nodes to obtain the remaining interference symbols. This will become clear in the example below, which consists of two parts. In the first part we decompose the storage code into punctured subcodes using Algorithm 1, which gives a series of punctured MDS-PIR capacity-achieving subcodes if Conjecture 1 is true. Then, in the second part, we show the retrieval process and derive the corresponding PIR rate. In particular, Protocol 2 is applied on each punctured subcode and special care is taken in case they overlap. We can prove the following lemma.

Lemma 3. *Let* $\mathcal{C}$ *be an* $[n,k]$ *code. Then, Algorithm 1 gives a series of punctured MDS-PIR capacity-achieving subcodes if and only if Conjecture 1 is true.*

Example 1. *Consider the* $[n,k] = [10,6]$ *non-MDS-PIR capacity-achieving code* $\mathcal{C}$ *with generator matrix*

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

From Theorem 3, we know that this code is not MDS-PIR capacity-achieving since $\max_{r\in\mathbb{N}_k}\{\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})}\} = \frac{2}{d_2(\mathcal{C}^{\boldsymbol{G}})} = \frac{2}{3} > \frac{6}{10}$*. The goal of Algorithm 1 is to find a generator matrix* $\boldsymbol{G}_{\mathsf{PIR}}$ *of an equivalent storage code* $\mathcal{C}_{\mathsf{PIR}}$ *such that the*

```
Algorithm 1: Code Decomposition for PIR
Input : A systematic generator matrix
        $\boldsymbol{G}_{k\times n} = \left[\boldsymbol{I}_k \mid \boldsymbol{A}_{k\times(n-k)}\right]$ for an $[n,k]$ code
        $\mathcal{C}$ with $d_k(\mathcal{C}) = n$
Output: A generator matrix $\boldsymbol{G}_{\mathsf{PIR}}$ of an equivalent
        storage code $\mathcal{C}_{\mathsf{PIR}}$, where
        $\boldsymbol{G}_p \triangleq \boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{k_1+\cdots+k_{p-1}+1:k_1+\cdots+k_p}}_{\mathbb{N}_{n_1+\cdots+n_{p-1}+1:n_1+\cdots+n_p}}$, of size
        $k_p \times n_p$, is the generator matrix of a
        punctured subcode $\mathcal{C}^{\boldsymbol{G}_p}$, $p \in \mathbb{N}_{\mathsf{P}}$, of $\mathcal{C}^{\mathsf{PIR}}$
1  $\boldsymbol{G}_{\mathsf{PIR}} \leftarrow \boldsymbol{O}_{k\times n}$
2  /* $\boldsymbol{O}_{k\times n}$ is a $k\times n$ all-zero matrix                    */
3  $p \leftarrow 0$, $h \leftarrow 0$, $l \leftarrow 0$
4  /* Compute the generalized Hamming weights and the
      corresponding subcodes                                        */
5  $\{d_1,\ldots,d_k\},\{\mathcal{V}_1,\ldots,\mathcal{V}_k\} \leftarrow$
     GeneralizedHammingWeights$(\boldsymbol{G})$
6  while $\max_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})}\right\} > \frac{k}{n}$ do
7      /* Update the total number of decomposed codes            */
8      $p \leftarrow p+1$
9      $r^* \leftarrow \operatorname{argmax}_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})}\right\}$
10     $k_p \leftarrow r^*$, $n_p \leftarrow d_{r^*}$
11     Find a series of row operations and/or column
         permutations, denoted by Transform, such that
         Transform$(\boldsymbol{G}) = \begin{bmatrix} \boldsymbol{G}_{r^*} & \boldsymbol{O} \\ \boldsymbol{G}_a & \boldsymbol{G}_b \end{bmatrix}$
12     /* $\boldsymbol{O}$ is an all-zero matrix. $\boldsymbol{G}_{r^*}$ of size $r^*\times d_{r^*}$ is a
          permuted generator matrix for the code obtained by
          puncturing the zero-coordinates of $\mathcal{V}_{r^*}$          */
13     /* Update the generator matrix                             */
14     $\boldsymbol{G} \leftarrow \begin{bmatrix} \boldsymbol{G}_{r^*} & \boldsymbol{O} \\ \boldsymbol{G}_a & \boldsymbol{G}_b \end{bmatrix}$
15     /* Update $\boldsymbol{G}_{\mathsf{PIR}}$ from the previous iteration             */
16     $\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+1:h+k}}_{\mathbb{N}_{1:l}} \leftarrow$ Transform$\left(\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+1:h+k}}_{\mathbb{N}_{1:l}}\right)$
17     /* Store the $p$-th decomposed code                        */
18     $\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+1:h+r^*}}_{\mathbb{N}_{l+1:l+d_{r^*}}} \leftarrow \boldsymbol{G}_{r^*}$
19     /* Update the new generator matrix by collecting the
          remaining rows and coordinates                           */
20     $\boldsymbol{G} \leftarrow \boldsymbol{G}_b$
21     /* Update $\boldsymbol{G}_{\mathsf{PIR}}$ for the next iteration                   */
22     $\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+r^*+1:h+k}}_{\mathbb{N}_{1:l+d_{r^*}}} \leftarrow \left[\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+r^*+1:h+k}}_{\mathbb{N}_{1:l}} \;\middle|\; \boldsymbol{G}_a\right]$
23     /* Update the incremental row and column index for the
          next iteration                                           */
24     $h \leftarrow h+r^*$, $l \leftarrow l+d_{r^*}$
25     /* Generate the generalized Hamming weights and the
          corresponding subcodes                                   */
26     $\{d_1,\ldots,d_{k'}\},\{\mathcal{V}_1,\ldots,\mathcal{V}_{k'}\} \leftarrow$
         GeneralizedHammingWeights$(\boldsymbol{G})$
27     /* Update the code dimension and blocklength of the
          new code                                                 */
28     $k \leftarrow k'$, $n \leftarrow n'$
29 end
30 $p \leftarrow p+1$
31 $r^* \leftarrow \operatorname{argmax}_{r\in\mathbb{N}_k}\left\{\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})}\right\}$
32 $k_p \leftarrow r^*$, $n_p \leftarrow d_{r^*}$
33 $\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{h+1:h+r^*}}_{\mathbb{N}_{l+1:l+d_{r^*}}} \leftarrow \boldsymbol{G}$
```

proposed protocol can be implemented. We start to construct $\boldsymbol{G}_{\mathsf{PIR}}$ by using Algorithm 1. In the initialization phase, the procedure GeneralizedHammingWeights$(\boldsymbol{G})$ *computes the generalized Hamming weights $\{d_1,\ldots,d_k\}$ of $\mathcal{C}^{\boldsymbol{G}}$ and corresponding subcodes $\{\mathcal{V}_1,\ldots,\mathcal{V}_k\}$.*

1) *In the first iteration of Algorithm 1 we obtain $r^* = \operatorname{argmax}_{r\in\mathbb{N}_k}\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})} = 2$ (Line 9), $\chi(\mathcal{V}_{r^*}) = \{1,2,4\}$, and the generator matrix*

$$\boldsymbol{G} = \left[\begin{array}{ccc|ccccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{array}\right]$$

in Line 14. Since $h = 0$ and $l = 0$, from Line 16, we have Transform$\left(\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{1:6}}_{\mathbb{N}_{1:0}}\right) = \emptyset$.

2) *From $\boldsymbol{G}$, in Line 18, we get*

$$\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{1:2}}_{\mathbb{N}_{1:3}} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

3) *In the next step, $\boldsymbol{G}$ is updated, in Line 20, by extracting the remaining rows and columns from $\boldsymbol{G}$ corresponding to $\boldsymbol{G}_b$ as*

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

Moreover, at Line 22 we have

$$\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{3:6}}_{\mathbb{N}_{1:3}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

4) *Now, $h = h + r^* = 2$ and $l = l + d_{r^*} = 3$ (from Line 24). Since $\max_{r\in\mathbb{N}_k}\frac{r}{d_r(\mathcal{C}^{\boldsymbol{G}})} = \frac{3}{5} > \frac{k}{n} = \frac{4}{7}$, we then continue to the second iteration. We have $r^* = 3$ (Line 9) and*

$$\text{Transform}(\boldsymbol{G}) = \left[\begin{array}{ccccc|cc} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right].$$

Now, we use the same transformation to get

$$\text{Transform}\left(\boldsymbol{G}_{\mathsf{PIR}}|^{\mathbb{N}_{3:6}}_{\mathbb{N}_{1:3}}\right) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

5) *At the end of Algorithm 1, in Line 33, we have*

$$\boldsymbol{G}_{\mathsf{PIR}} = \left[\begin{array}{ccc|ccccc|cc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right].$$

TABLE I

RESPONSES BY THE PROTOCOL FROM SECTION III WITH A $[10,6]$ NON-MDS-PIR CAPACITY-ACHIEVING CODE $\mathcal{C}^{\boldsymbol{G}_{\mathsf{PIR}}}$. FOR EASE OF NOTATION, $I^{(m)}_{n_1+\cdots+n_a}$ IS SHORT-HAND FOR $I^{(m)}_{n_1}+\cdots+I^{(m)}_{n_a}$, $a\in\mathbb{N}$.

Subresponses	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 9	Node 10
Subresponse 1	$I^{(1)}_1+x^{(m)}_{1,1}$	$I^{(1)}_2$	$I^{(1)}_{1+2+4+5+6}$	$I^{(1)}_3+x^{(m)}_{1,3}$	$I^{(1)}_4+x^{(m)}_{1,4}$	$I^{(1)}_5$	$I^{(1)}_{4+5}$	$I^{(1)}_{3+4+5+6}$	$I^{(1)}_6+x^{(m)}_{1,6}$	$I^{(1)}_6$
Subresponse 2	$I^{(2)}_1$	$I^{(2)}_2+x^{(m)}_{1,2}$	$I^{(2)}_{1+2+4+5+6}$	$I^{(2)}_3+x^{(m)}_{2,3}$	$I^{(2)}_4$	$I^{(2)}_5+x^{(1)}_{1,5}$	$I^{(2)}_{4+5}$	$I^{(2)}_{3+4+5+6}$	$I^{(2)}_6$	$I^{(2)}_6+x^{(m)}_{2,6}$
Subresponse 3	$I^{(3)}_1+x^{(m)}_{2,1}$	$I^{(3)}_2$	$I^{(3)}_{1+2+4+5+6}$	$I^{(3)}_3$	$I^{(3)}_4$	$I^{(3)}_5+x^{(m)}_{2,5}$	$I^{(3)}_{4+5}+x^{(m)}_{2,4}+x^{(m)}_{2,5}$	$I^{(3)}_{3+4+5+6}$	$I^{(3)}_6$	
Subresponse 4	$I^{(4)}_1$	$I^{(4)}_2+x^{(m)}_{2,2}$	$I^{(4)}_{1+2+4+5+6}$	$I^{(4)}_3$	$I^{(4)}_4$	$I^{(4)}_5$			$I^{(4)}_6$	

In summary, $\mathsf{P}=3$ *punctured MDS-PIR capacity-achieving subcodes, an* $[n_1,k_1]=[3,2]$ *punctured subcode* $\mathcal{C}^{\boldsymbol{G}_1}$*, an* $[n_2,k_2]=[5,3]$ *punctured subcode* $\mathcal{C}^{\boldsymbol{G}_2}$*, and an* $[n_3,k_3]=[2,1]$ *punctured subcode* $\mathcal{C}^{\boldsymbol{G}_3}$ *are obtained from* $\mathcal{C}$*. The corresponding generator matrices from* $\boldsymbol{G}_{\mathsf{PIR}}$ *are*

$$\boldsymbol{G}_1=\begin{bmatrix}1&0&1\\0&1&1\end{bmatrix},\boldsymbol{G}_2=\begin{bmatrix}1&0&0&0&1\\0&1&0&1&1\\0&0&1&1&1\end{bmatrix},\boldsymbol{G}_3=\begin{bmatrix}1&1\end{bmatrix}.$$

In this example, we require $\beta=\mathsf{LCM}(\beta_1,\beta_2,\beta_3)=\mathsf{LCM}(1,2,1)=2$ *stripes and* $\mathsf{D}_{\max}=\max_{p\in\mathbb{N}_3}\{\frac{\beta}{\beta_p}\mathsf{D}_p\}=\max\{\frac{2}{1}\cdot 2,\frac{2}{2}\cdot 3,\frac{2}{1}\cdot 1\}=4$ *subqueries. The responses from the nodes when retrieving file* $\boldsymbol{X}^{(m)}$ *are those marked in black and red in Table I. Since it is not necessary that the* p*-th subprotocol obtains the required interference symbols, we have to download the interference symbols marked in red for the equivalent storage code* $\mathcal{C}^{\boldsymbol{G}_{\mathsf{PIR}}}$ *to recover the code symbols* $x^{(m)}_{2,1}$, $x^{(m)}_{2,5}$, $x^{(m)}_{2,4}+x^{(m)}_{2,5}$*, and* $x^{(m)}_{2,2}$*. For example,* $I^{(4)}_4$, $I^{(4)}_5$*, and* $I^{(4)}_6$ *are required to be downloaded for* $\mathcal{C}^{\boldsymbol{G}_1}$ *to recover* $x^{(m)}_{2,2}$*. The PIR rate is then equal to*

$$\mathsf{R}_\infty=\frac{\beta k}{\mathsf{D}}=\frac{2\cdot 6}{\frac{\beta}{\beta_1}(n_1\cdot k_1)+\frac{\beta}{\beta_2}(n_2\cdot k_2)+\frac{\beta}{\beta_3}(n_3\cdot k_3)+4}$$
$$=\frac{12}{2\cdot 6+1\cdot 15+2\cdot 2+4}=\frac{12}{35}\approx 0.343<\mathsf{C}^{[10,6]}_\infty,$$

where $\frac{\beta}{\beta_1}=2$, $\frac{\beta}{\beta_2}=1$*, and* $\frac{\beta}{\beta_3}=2$.

A. PIR Rate

The following theorem based on Lemma 3 shows that the PIR rate of our proposed protocol is larger than or equal to that of Protocol A under some condition. The proof is omitted due to lack of space and will be provided in the extended version.

Theorem 4. *Let* $\mathcal{C}$ *be an* $[n,k]$ *code. If Algorithm 1 gives a series of punctured MDS-PIR capacity-achieving subcodes, then the PIR rate* R_∞ *of the proposed protocol is lowerbounded by the PIR rate* $\mathsf{R}_{\infty,\mathsf{A}}=1-\tau^*$ *of Protocol A.*

Note that Lemma 3 relates whether or not Algorithm 1 gives a series of punctured MDS-PIR capacity-achieving subcodes to Conjecture 1. Moreover, it indicates that by using generalized Hamming weights, Algorithm 1 gives a construction for a PIR achievable rate matrix that achieves τ^*.

Example 2. *Continuing with Example 1. Note that combining the interference symbol marked in blue with the responses for the proposed protocol in Table I (marked in red and black), one can show that these are the required responses for Protocol A, i.e., we have constructed a PIR achievable rate matrix* $\boldsymbol{\Lambda}_{4,6}$ *for* $\mathcal{C}$*. Hence, it can readily be seen that* $\mathsf{R}_\infty>\mathsf{R}_{\infty,\mathsf{A}}=1-\frac{2}{3}\approx 0.333$.

IV. CONCLUSION

For a storage code that is not MDS-PIR capacity-achieving, we proposed a heuristic algorithm to decompose it into punctured subcodes guided by its generalized Hamming weights. Based on the code decomposition, we proposed an improved PIR protocol with PIR rate larger than or equal to that of the best known PIR protocol for coded DSSs in the literature if the code decomposition gives a series of punctured MDS-PIR capacity-achieving subcodes.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annu. IEEE Symp. Found. Comp. Sci. (FOCS)*, Milwaukee, WI, USA, Oct. 23–25, 1995, pp. 41–50.

[2] 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. 29 – Jul. 4, 2014, pp. 856–860.

[3] 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. 14–19, 2015, pp. 2842–2846.

[4] 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.

[5] 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.

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

[7] 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.

[8] S. Kumar, H.-Y. Lin, E. Rosnes, and A. Graell i Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4243–4273, Jul. 2019.

[9] H.-Y. Lin, S. Kumar, E. Rosnes, and A. Graell i Amat, "Asymmetry helps: Improved private information retrieval protocols for distributed storage," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov. 25–29, 2018, pp. 1–5.

[10] 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.

[11] 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.

[12] 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 Trans. Inf. Theory*, vol. 65, no. 4, pp. 2107–2118, Apr. 2019.

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

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