Building Capacity-Achieving PIR Schemes with Optimal Sub-Packetization over Small Fields

Jingke Xu, Zhifang Zhang

KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

Emails: xujingke14@mails.ucas.edu.cn, zfz@amss.ac.cn

Abstract—Consider \( N \) servers with replicated databases containing \( M \) records. Suppose a user wants to privately retrieve one record by accessing the servers such that the identity of the retrieved record is secret against any up to \( T \) servers. A scheme designed for this purpose is called a \( T \)-private information retrieval (\( T \)-PIR) scheme. Three indexes are concerned for PIR schemes:

1. rate, indicating the amount of retrieved information per unit of downloaded data. The highest achievable rate is characterized by the capacity;
2. sub-packetization, reflecting the implementation complexity for linear schemes;
3. field size. We consider linear schemes over a finite field.

In this paper, a general \( T \)-PIR scheme simultaneously attaining the optimality of almost all of the three indexes is presented. Specifically, we design a linear capacity-achieving \( T \)-PIR scheme with sub-packetization \( d n^M \) over a finite field \( \mathbb{F}_q \), \( q \geq N \). The sub-packetization \( d n^M \), where \( d = \gcd(N, T) \) and \( n = N/d \), has been proved to be optimal in our previous work. The field size is reduced by an exponential factor in our scheme comparing with existing capacity-achieving \( T \)-PIR schemes.

I. INTRODUCTION

The problem of private information retrieval (PIR) involves a database containing \( M \) records stored across \( N \) servers and a user who wants to privately retrieve one record by accessing the servers. The privacy requirement means any colluding subset of at most \( T \) servers knows nothing about the identity of the retrieved record. Since it is closely related to cryptography [2] and coding theory [13], PIR has become a central research topic in the computer science literature since it was first introduced by Chor et al. [4] in 1995. A survey on PIR can be found in [7].

A central issue in PIR is minimizing the communication cost which is measured by the total number of bits transferred from the user to the servers (i.e., the query size) and those from servers to the user (i.e., the answer size). In the initial setting of PIR where each record is one bit long, the least communication cost achieved by now is \( M^{O\left(\frac{\log M}{\log \log M}\right)} \) [5], [6]. Recently, an information-theoretic reformulation of the PIR problem [3] allows all records to be arbitrarily large, which is more relevant to real-life applications. In this sense, the query size is always negligible compared to the answer size. Thus communication cost can be measured by taking only the answer size into account. Specifically, define the rate of a PIR scheme as the ratio between the size of the retrieved record and the answer size, and define the capacity as the supremum of the rate over all PIR schemes. Obviously, the capacity characterizes the highest rate that can be achieved for a given PIR problem.

Sun and Jafar [8] first proved that the capacity of PIR for replicated non-colluding servers (i.e., \( T = 1 \)) is \( \frac{1}{1-1/N^M} \). They further proved in [9] the capacity for the colluding case (i.e., \( T \)-PIR, \( 1 \leq T < N \)) is \( \frac{1}{1-1/(N^M-1)} \) and designed a capacity-achieving PIR scheme over a finite field \( \mathbb{F}_q \), where \( q \geq \max\{N^2T^{M-2}, N^M(N-T)^{M-2}\} \). Meantime, the capacity for PIR with symmetric privacy was determined in [10], and that for PIR with MDS coded non-colluding servers was studied in [11]. In this work, we focus on PIR problems from replicated servers and only non-symmetric privacy is concerned. A detailed model for the PIR problem studied in this work is presented in Section II.

A key step for determining the capacity is to design a general capacity-achieving PIR scheme. All existing schemes are implemented by dividing each record into segments (taken as elements in the finite field \( \mathbb{F}_q \)) and querying from each server some combinations of the segments. We call the number of segments contained in each record that are necessary for the scheme implementation as sub-packetization which reflects implementation complexity for linear schemes where only linear operations over the segments are involved. The schemes provided in [8], [9] both have sub-packetization \( M^N \). In [11] the sub-packetization was firstly reduced to \( N^{M-1} \) for replicated servers and \( T = 1 \). For general values of \( T \), we recently proved in [14] that the optimal sub-packetization for linear capacity-achieving \( T \)-PIR schemes is \( d n^M \), where \( d = \gcd(N, T) \), \( n = N/d \).

Since all existing PIR schemes are linear over a finite field, the field size is also an important metric related to the computational complexity. A recent improvement on the field size for capacity-achieving \( T \)-PIR schemes is obtained in [14] which reduces the field size by a factor of \( Nd^{M-2} \) comparing with the scheme in [8], that is, the scheme in [14] operates over a finite field \( \mathbb{F}_q, q \geq \max\{Nt^{M-2}, N(n-t)^{M-2}\} \), where \( t = T/d \). However, the field size is still larger than \( Nt^{M-2} \) which expands exponentially as \( M \) grows.

The main contribution of this work is a great reduction on the field size for \( T \)-PIR schemes which simultaneously achieve the capacity and the optimal sub-packetization. Specifically,
we design a linear capacity-achieving T-PIR scheme with sub-packetization $dn^{M-1}$ over a finite field $\mathbb{F}_q$, $q \geq N$. Comparing with previous schemes, the field size is reduced by an exponential factor and become a constant for a fixed storage system.

The rest of this paper is organized as follows. First, the PIR model is formally defined in Section III. Then an example of the T-PIR scheme is presented in Section III to explain our design idea. The general descriptions of our scheme are given in Section IV. Finally, Section V concludes the paper.

II. NOTATIONS AND THE PIR MODEL

For an integer $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. For a vector $Q = (q_1, \ldots, q_n)$ and a subset $\Gamma = \{i_1, \ldots, i_m\} \subseteq [n]$, denote $Q_{\Gamma} = (q_{i_1}, \ldots, q_{i_m})$. Most vectors in this paper are row vectors and they are denoted by the capital letters (eg. $W, Q$).

Denote the $M$ records as $W_1, \ldots, W_M$ and the $N$ servers as $\text{Serv}^{(1)}, \ldots, \text{Serv}^{(N)}$. In this paper, we focus on PIR for replicated servers, that is, assume each server stores all the $M$ records. Then suppose a user wants to privately retrieve $W_\theta$ for some $\theta \in [M]$. Basically, a PIR scheme consists of two phases:

- **Query phase.** Given $\theta \in [M]$ and some random resources $S$, the user computes $\text{Que}(\theta, S) = (Q^{(1)}_{\theta}, \ldots, Q^{(N)}_{\theta})$, and sends $Q^{(j)}_{\theta}$ to $\text{Serv}^{(j)}$ for $1 \leq j \leq N$. Note that $S$ and $\theta$ are private information only known by the user, and $\text{Que}(\cdot, \cdot)$ is the query function defined by the scheme.

- **Response phase.** For $1 \leq j \leq N$, the $j$th server $\text{Serv}^{(j)}$ at receiving $Q^{(j)}_{\theta}$, computes $\text{Ans}^{(j)}(Q^{(j)}_{\theta}, W_{(M)}) = A^{(j)}_{\theta}$ and sends it back to the user, where $\text{Ans}^{(j)}(\cdot, \cdot)$ is $\text{Serv}^{(j)}$’s answer function defined by the scheme.

To design a PIR scheme is to design the functions Que and $\text{Ans}^{(j)}$, $1 \leq j \leq N$, such that the following two conditions are satisfied:

1. **Correctness:** The user can recover $W_\theta$ after collecting all answers from the $N$ servers, i.e., $H(W_\theta, A^{(N)}_{\theta}, Q^{(N)}_{\theta}, S, \theta) = 0$, where $H(\cdot, \cdot)$ is the conditional entropy.

2. **Privacy:** For any $\Gamma \subseteq [N]$ with $|\Gamma| \leq T$, the servers in $\Gamma$ have no information on $\theta$ even if they collude with each other, i.e., $I(\theta; Q^{(j)}_{\theta}, A^{(j)}_{\theta}, W_{(M)}) = 0$, where $I(\cdot; \cdot)$ denotes the mutual information.

Particularly, if $W_i \in \mathbb{F}_q^t$ and the answer $A^{(j)}_{\theta}$ turns out to be linear combinations of $W_i$’s for $1 \leq i \leq M$ and $1 \leq j \leq N$, we call the scheme a linear T-PIR scheme, and call $L$ the sub-packetization of the scheme.

Define the rate $R$ of a PIR scheme by

$$R = \frac{H(W_\theta)}{\sum_{j=1}^{N} H(A^{(j)}_{\theta})},$$

that is, $R$ characterizes the amount of retrieved information per unit of downloaded data. Furthermore, the supremum of the rate over all PIR schemes that work for a given PIR problem is called the capacity. It was determined in [9] that the capacity of T-PIR schemes is $\left(\frac{T}{TN/d}\right)$. The next theorem determines the optimal sub-packetization for linear capacity-achieving T-PIR schemes.

**Theorem 1.** ([4]) Suppose $M \geq 2, N > T \geq 1$. For all linear capacity-achieving T-PIR schemes with $M$ records and $N$ replicated servers, its sub-packetization must be no less than $dn^{M-1}$, where $d = \gcd(N, T), n = N/d$. Moreover, the lower bound can be obtained generally.

III. AN EXAMPLE FOR $M = 3, N = 3, T = 2$

Suppose $M = 3, N = 3, T = 2$. Since our T-PIR scheme needs sub-packetization $L = dn^{M-1} = 9$, each record is regarded as a 9-dimensional row vector over $\mathbb{F}_q$, i.e., $W_1, W_2, W_3 \in \mathbb{F}_q^9$, where $q \geq 3$. WLOG, assume the user wants to retrieve $W_1$, i.e., $\theta = 1$. The scheme is described in two parts:

**Part I: Mixing-Rearranging.**

Let $S_1, S_2, S_3$ be three random matrices chosen by the user independently and uniformly from all $9 \times 9$ invertible matrices over $\mathbb{F}_q$. And, $S_1, S_2, S_3$ are only known by the user. Define

$$
(a_{ij})_{3 \times 3} = \text{Mat}_{3 \times 3}(W_1 S_1) \\
(b_{ij})_{3 \times 3} = \text{Mat}_{3 \times 3}(W_2 S_2; [1:6]) (I_3 \otimes G) \\
(c_{ij})_{3 \times 3} = \text{Mat}_{3 \times 3}(W_3 S_3; [1:6]) (I_3 \otimes G)
$$

where $\text{Mat}_{xy}(\cdot)$ is to write out a $x \times y$ matrix row by row from a $y$-dimensional vector. For example, $\text{Mat}_{2 \times 3}((1, 2, 0, 0, 1, 2)) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$. And, $S_1; [1:6]$ denotes the matrix composed of the first 6 columns of $S_1$, $\otimes$ denotes the Kronecker product, $I_3$ is the $3 \times 3$ identity matrix, and $G$ is a $2 \times 3$ generator matrix of some fixed $[3, 2]$ MDS code over $\mathbb{F}_q$.

Specifically, write the three matrices generated in (1) as

$$
\begin{pmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{pmatrix},
\begin{pmatrix}
b_{11} & b_{12} & b_{13} \\
b_{21} & b_{22} & b_{23} \\
b_{31} & b_{32} & b_{33}
\end{pmatrix},
\begin{pmatrix}
c_{11} & c_{12} & c_{13} \\
c_{21} & c_{22} & c_{23} \\
c_{31} & c_{32} & c_{33}
\end{pmatrix},
$$

then one can see each $a_{ij}$, $1 \leq i, j \leq 3$, is a random linear combination of the 9 coordinates of $W_1$, and $(a_{ij})_{3 \times 3}$ is actually an invertible transformation of $W_1$, while $(b_{ij})_{3 \times 3}$ is composed of three independent codewords (as rows) in the $[3, 2]$ MDS code, encoded from 6 symbols each of which is a random linear combination of coordinates of $W_2$. The construction of $(c_{ij})_{3 \times 3}$ is similar to that of $(b_{ij})_{3 \times 3}$.

**Part II: Combinig.**

We complete description of the scheme by listing in Table 1 the answers from all servers.

| $\text{Serv}^{(1)}$ | $\text{Serv}^{(2)}$ | $\text{Serv}^{(3)}$ |
|------------------|------------------|------------------|
| $a_{11}, b_{11}, c_{11}$ | $a_{22}, b_{22}, c_{22}$ | $a_{13}, b_{13}, c_{13}$ |
| $a_{21} + b_{21}, a_{31} + c_{21}$ | $a_{12} + b_{12}, a_{32} + c_{12}$ | $a_{23} + b_{23}, c_{23}$ |
| $a_{31} + c_{31}$ | $a_{32} + c_{32}$ | $a_{33} + b_{33} + c_{33}$ |

Table 1: Answer table of the T-PIR scheme for $(M = 3, N = 3, T = 2)$ and $\theta = 1$. It can be seen that the answers are all sums of the symbols $a_{ij}, b_{ij}, c_{ij}$. Because $a_{ij}$’s are generated from $W_1$, the record desired by the user, we call $a_{ij}$ the desired symbol. Accordingly, we call $b_{ij}, c_{ij}$ the interference symbol. Likewise, all sums that do not involve $a_{ij}$’s, such as $b_{ij} + c_{ij'}$, are
called interference sums. The sums that involve both \( a_{ij} \)'s and interference symbols, such as \( a_{ij} + b_{ij'} \), \( a_{ij} + c_{ij'} \), are called mixed sums. Basically, the answer table is built up by iteratively applying the following two steps:

(a1) Enforce record symmetry within each server.

(a2) Form mixed sums by combining new desired symbols with recoverable interferences.

As shown in Table 1, the record symmetry within each server has been enforced in the first two lines. That is, one symbol from each record is queried from Serv\(^{(1)}\) and also from Serv\(^{(2)}\), while two symbols from each record are queried from Serv\(^{(3)}\). Now, since the symbols \( b_{11} \) and \( b_{13} \) have been provided by Serv\(^{(1)}\) and Serv\(^{(3)}\) respectively, the symbol \( b_{12} \) is recoverable because \( (b_{11}, b_{12}, b_{13}) \) is a codeword in the [3, 2] MDS code. Likewise, the symbols \( b_{21}, c_{12}, c_{21} \) are also recoverable from the first two lines. Therefore, in the next two lines these recoverable symbols are combined with \( a_{ij} \)'s that have not been queried so far. Finally, the last two lines of Table 1 are formed by applying (a1) and (a2) respectively.

The correctness condition holds because the user can finally obtain all \( a_{ij} \)'s for \( 1 \leq i, j \leq 3 \) and then recover \( W_1 \) by multiplying \( S_1^{-1} \). To explain the privacy condition, we first point out a key observation of Table 1 for \( 1 \leq j \leq N \), each entry in the \( j \)th column of the matrices in (2) appears in Serv\(^{(j)}\)'s answers exactly once. Therefore, the answers of any two servers are formed by symbols from two columns of the matrices which are completely independent because of the [3, 2] MDS encoding. Then plus the record symmetry enforced within each server, the privacy condition follows.

Finally, the total number of downloaded symbols from all servers is \( 7 + 6 + 6 = 19 \) and each record consists of 9 symbols. Hence the rate is \( \frac{9}{19} \), which matches the capacity for this case.

IV. DESCRIPTION OF THE GENERAL T-PIR SCHEME

Now we describe how our T-PIR scheme works generally for \( M \geq 2 \) and \( 1 \leq T < N \). Suppose \( W_1, \ldots, W_M \in \mathbb{F}_q^L \) and the user wants to privately retrieve \( W_\theta \) for some \( \theta \in [M] \). Fix an \([N, T]\) MDS code over \( \mathbb{F}_q \) with a \( T \times N \) generator matrix \( G \). It requires \( q \geq N \) to ensure existence of such an MDS code. As in Section III, the general T-PIR scheme is described in two parts:

Part I: Mixing-Rearranging.

Let \( S_1, \ldots, S_M \) be \( M \) matrices chosen by the user independently and uniformly from all \( L \times L \) invertible matrices over \( \mathbb{F}_q \). As before, these \( S_i \)'s are only known by the user. We first assume \( N \mid L \) and \( L = N \ell \). Later we will show \( L = N \ell 2^{-M-2} \) is enough for our scheme and this \( L \) matches the optimal subpacketization.

Define \( (U_\theta)_{L \times N} = (u_{\theta, ij})_{L \times N} = \text{Mat}_{L \times N}(W_k S_\ell) \), where for the matrix entry \( u_{\theta, ij} \), \( i \) is the row index and \( j \) is the column index. Then as in (1), define for \( k \in [M] - \{ \theta \}, \]

\[
(U_k)_{L \times N} = (u_{k, ij})_{L \times N} = \text{Mat}_{L \times N}(W_k S_\ell; (1: TL)(I_L \otimes G)).
\]

Obviously, \( (U_\theta)_{L \times N} \) is an invertible transformation of \( W_\theta \), while \( (U_k)_{L \times N} \) for \( k \in [M] - \{ \theta \} \) is composed of independent rows each of which is a codeword in the \([N, T]\) MDS code. It is important to point out here the MDS encoding is used for achieving T-privacy and privacy of the PIR scheme. It has nothing to do with the database storage. In this work we consider replicated databases, not the MDS coded databases as in [1], [2].

Part II: Combining

In this part we build the answer table for general T-PIR schemes. Fix \( k' \in [M] - \{ \theta \}, \) we first investigate how the answers of the matrix \((U_k)_{L \times N}\) are put into the answer table.

For any \( \Lambda \subseteq [M] \) and \( 1 \leq j \leq N \), we call \( \sum_{k \in \Lambda} u_{k, ij} \) an \( \Lambda \)-type sum, where \( i_k \in [\ell] \) is a row index. It is clear that each entry of \((U_k)_{L \times N}\) appears in an \( \Lambda \)-type sum listed in the answer table for some \( \Lambda \subseteq [M] \) with \( k' \in \Lambda \). Particularly, define for all \( k \in [M] - \{ \theta \}, \)

\[
\text{InType}_{k, \theta} = \{ \Lambda \subseteq [M] \mid k \in \Lambda, \theta \notin \Lambda \},
\]

and for any \( \Lambda \in \text{InType}_{k, \theta} \) denote \( \Lambda = \Lambda \cup \{ \theta \}. \)

First we associate each row of \((U_k)_{L \times N}\) with a type \( \Lambda \in \text{InType}_{k, \theta} \). Note that multiple rows may be associated with the same type. Moreover, because we always enforce record symmetry within each server, for any \( \Lambda \in \text{InType}_{k, \theta} \) with fixed cardinality \(|\Lambda| = i\), the number of rows associated with type \( \Lambda \) remains the same and we denote this number as \( d_i \). Moreover, the value of \( d_i \) is independent of the record index \( k' \). We will show how to determine \( d_i \) for \( 1 \leq i \leq M - 1 \) in Section IV-B. For convenience, we arrange these types in an increasing order of cardinality as they run through all sets in \( \text{InType}_{k, \theta} \). As in the example, we have

\[
\begin{pmatrix}
(b_{11} & b_{12} & b_{13}) \\
(b_{21} & b_{22} & b_{23}) \\
(b_{31} & b_{32} & b_{33})
\end{pmatrix}_{\text{type } 2},
\begin{pmatrix}
(c_{11} & c_{12} & c_{13}) \\
(c_{21} & c_{22} & c_{23}) \\
(c_{31} & c_{32} & c_{33})
\end{pmatrix}_{\text{type } 3},
\begin{pmatrix}
(2, 3) \\
(3)
\end{pmatrix}
\]

which implies \( d_1 = 2 \) and \( d_2 = 1 \).

The entries of \((U_k)_{L \times N}\) are put into the answer table according to the following rules (b1)-(b4):

(b1) For \( 1 \leq j \leq N \), entries in the \( j \)th column only appear in Serv\(^{(j)}\)'s answers.

(b2) For each row associated with a type \( \Lambda \in \text{InType}_{k, \theta} \) out of the \( N \) entries appear in \( \Lambda \)-type sums (i.e., interference symbols or interference sums) and the remaining \( N - T \) entries appear in \( \Lambda \)-type sums (i.e., mixed sums).

For each \( \Lambda \in \text{InType}_{k, \theta} \), denote by \((U_k)_{L \times N}\) the matrix \((U_k')_{L \times N}\) restricted to the rows associated with the type \( \Lambda \). Suppose \(|\Lambda| = i\), then \((U_k')_{L \times N}\) has \( d_i \) rows.

(b3) For each of the first \( T \) columns of \((U_k')_{L \times N}\), \( \alpha_i \) entries appear in \( \Lambda \)-type sums and the remaining \( \alpha_i + 1 = d_i - \alpha_i \) entries appear in \( \Lambda \)-type sums.

(b4) For each of the last \( N - T \) columns of \((U_k')_{L \times N}\), \( \beta_i \) entries appear in \( \Lambda \)-type sums and the remaining \( \beta_i + 1 = d_i - \beta_i \) entries appear in \( \Lambda \)-type sums.

One can check the scheme in Section III follows the above rules by combining (3) with Table 1. Actually, in (3) we have indexed in boldface the symbols which appear in mixed sums in Table 1. For example, in the first column of \((b_{1i})_{3 \times 3}, b_{11}\) appears in a \( \{2\} \)-type sum, which implies \( \alpha_1 = 1, \alpha_2 = 1, \)
and $b_{31}$ appears in a $\{2, 3\}$-type sum, which implies $\alpha_3 = 0$ and also $\alpha_2 = 1$. While in the third column of $(b_{ij})_{3 \times 3}$, both $b_{13}$ and $b_{23}$ appear in $\{2\}$-type sums, which implies $\beta_1 = 2, \beta_2 = 0$, and $b_{33}$ appears in an $\{1, 2, 3\}$-type sum, which implies $\beta_3 = 1$ and also $\beta_2 = 0$.

Actually, the rules (b1)-(b4) are due to our intention to guarantee correctness and privacy for the general scheme.

A. Correctness and privacy

Suppose $\Lambda = \{k_1, \ldots, k_s\} \subseteq [M] - \{\theta\}$. For $1 \leq j \leq s$, the matrix $(U_{kj})_{L \times N}$ has $d_s$ rows associated with the type $\Lambda$, and all entries of the sub-matrix $(U_{kj})_{\Lambda \times N}$ appear in either $\Lambda$-type sums or in $\Lambda$-type sums. Moreover, we can assume locations of the entries that appear in $\Lambda$-type sums are the same across all the sub-matrices $(U_{kj})_{\Lambda \times N}, 1 \leq j \leq s$. For example, in (3) we can see for $\Lambda = \{2, 3\}$ the corresponding sub-matrices are $$(b_{31} \ b_{32} \ b_{33}), \ (c_{31} \ c_{32} \ c_{33}).$$

Both have the first two entries as the ones to appear in $\{2, 3\}$-type sums.

In general, for $1 \leq t \leq d_s$ let $\Gamma_t \subseteq [N]$ be the set of column indexes that indicate locations of the entries in the $t$-th row of $(U_{kj})_{\Lambda \times N}$ to appear in $\Lambda$-type sums. Suppose the beginning row index of the sub-matrix $(U_{kj})_{\Lambda \times N}$ is $i_j, 1 \leq j \leq s$. Thus all the $\Lambda$-type sums in the answer table are

$$\sum_{1 \leq j \leq s} u_{k_j,i_j+t-1}^{(j)}, \ j' \in \Gamma_t, \ 1 \leq t \leq s. \quad (4)$$

Accordingly, all the $\Lambda$-type sums in the answer table are

$$u_{\theta}^{(\alpha_1')} + \sum_{1 \leq j \leq s} u_{k_j,i_j+t-1}^{(j')}, \ j' \in [N] - \Gamma_t, \ 1 \leq t \leq s. \quad (5)$$

Here we neglect the row index for the entry $u_{\theta}^{(\alpha_1')}$ because each time we simply use a new desired symbol to form the mixed sums. Fix $t \in [s]$, from the MDS encoding we know

$$\left(\sum_{1 \leq j \leq s} u_{k_j,i_j+t-1}^{(1)}, \sum_{1 \leq j \leq s} u_{k_j,i_j+t-1}^{(2)}, \ldots, \sum_{1 \leq j \leq s} u_{k_j,i_j+t-1}^{(N)}\right)$$

is a codeword in the $[N,T]$ MDS code. Because all the interference parts in (5) are recoverable from the $\Lambda$-type sums in (4). As a result, all the desired symbols involved in (5), i.e., $u_{\theta}^{(\alpha_1')}$’s, can be obtained. As $\Lambda$ runs through all types in $[M] - \{\theta\}$, all desired symbols involved in the mixed sums can be obtained similarly, which give us a belief that the correctness condition can be satisfied. However, we still need to determine the exact number of desired symbols, which will be done in the next section.

About the privacy condition, rule (b1) first confines $\text{Serv}^{(j)}$’s answers to the entries in the $j$-th columns of $(U_{kj})_{L \times N}, k \in [M]$, then rules (b3)-(b4) require each entry is involved exactly once in a symmetric form with respect to the records, i.e., for any $\Lambda \subseteq [M]$ with $|$|$ = i$, the number of $\Lambda$-type sums provided by $\text{Serv}^{(j)}$ is $\alpha_i$ for $1 \leq j \leq T$ and $\beta_i$ for $T < j \leq N$. Therefore, the answers of any $T$ servers correspond to $T$ columns of $(U_{kj})_{L \times N}, k \in [M]$, combined in a symmetric manner. From the mixing-rearranging part we know any $T$ columns of the matrices are completely independent, therefore distribution of the answers of any $T$ servers is independent of the retrieved index $\theta$.

B. Determining $\alpha_i$ and $\beta_i$

We derive necessary conditions on $\alpha_i$ and $\beta_i$ for feasibility of the rules (b1)-(b4) in the following:

$$\begin{align}
T\alpha_i + (N-T)\beta_i & = d_iT \\
\alpha_i + \alpha_{i+1} & = d_i = \beta_i + \beta_{i+1}
\end{align} \quad (6)$$

By a similar computation as that given in (14), we obtain the solutions to (6), i.e., for $N \geq 2T$,

$$\begin{align}
\alpha_i & = \frac{(n-t)^{i-2}-(n-t)^{i-3}}{(n-t)^{i-2}}(n-t)^{M-i} \\
\beta_i & = \frac{(n-t)^{i-1}}{n^2}(n-t)^{M-i}, \quad (7)
\end{align}$$

and for $T < N < 2T$,

$$\begin{align}
\alpha_i & = \frac{(M-2i-1)(n-t)}{n^2}(n-t)^{M-i} \\
\beta_i & = \frac{(M-i-1)}{n^2}(n-t)^{M-i}, \quad (8)
\end{align}$$

where $n = N/d, t = T/d$ and $d = \text{gcd}(N, T)$. It is important to note that the values of $\alpha_i$ and $\beta_i$ are independent of the index $\theta$, which means the partial symmetric structure of the answer table is fixed for any given $N, T$ and $M$, no matter which record is to be retrieved. In other words, the partial symmetric structure can be publicly known, which will not affect the $T$-privacy.

Based on the values of $\alpha_i$ and $\beta_i$, we next compute sub-packetization of the scheme. For any $k \in [M] - \{\theta\}$, $d_i$ rows of $(U_{kj})_{L \times N}$ are associated with a type $\Lambda \in \text{InType}_k \theta$ with $|\Lambda| = i$, so $\tilde{L} \geq \sum_{i=1}^{M-1} (M-2i)d_i$. From (6) we know $d_i = \alpha_i + T\beta_i/(N-T)$. From (7) and (8) one can compute that for $1 \leq i \leq M$,

$$T\alpha_i + (N-T)\beta_i = d_i(n-t)^{i-1}t^{M-i}. \quad (9)$$

Therefore, $\tilde{L} \geq \sum_{i=1}^{M-1} (M-2i)(n-t)^{i-1}t^{M-i} = n^{M-2}$. As for the matrix $(U_{kj})_{L \times N}$, its entries appear in $\Lambda$-type sums for all $\Lambda \subseteq [M]$ with $|$|$ = i$. Particularly, for $1 \leq j \leq T$, it needs $\tilde{L} \geq \sum_{i=1}^{M} (i-1)^-1\alpha_i$, which implies $\tilde{L} \geq n^{M-2}$ from both (7) and (8), while for $T < j \leq N$, it needs $\tilde{L} \geq \sum_{i=1}^{M} (i-1)^-1\beta_i$ which still implies $\tilde{L} \geq n^{M-2}$ from both (7) and (8). Hence, it is enough to set $\tilde{L} = n^{M-2}$ in our scheme. Thus the scheme has sub-packetization $N\tilde{L} = dn^{M-1}$ which matches the optimal sub-packetization.

Then we compute rate of the scheme. It is easy to see the number of downloaded symbols is $D = \sum_{i=1}^{M-1} (M-i)T\alpha_i + (N-T)\beta_i$ which equals $d(n^{M-2} - (N-T)\beta_i)$ by (9). Hence, the rate of our scheme is $\frac{D}{T} = d(n^{M-2} - (N-T)/(n-t)) = \frac{1-T/N}{1-(T/N)^{M-1}}$, attaining the capacity for $T$-PIR schemes.

C. Locator matrix

In order to realize (b2)-(b4), we define a locator matrix $M_i$ associated with each type $\Lambda \subseteq [M] - \{\theta\}$ with $|$|$ = i$ such that $M_i$ is a $d_i \times N$ binary matrix whose $(i,j)$-entry equals 1 if and only if the $(i,j)$-entries of the sub-matrices $(U_{kj})_{\Lambda \times N}$ for all $k \in [M] - \{\theta\}$ appear in $\Lambda$-type sums. For example, for Table [1] we have $M_1 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ and $M_2 = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}$. 
It follows immediately the 0’s in $\mathcal{M}_i$ locate the interference parts of the $\Delta$-type sums. Specifically,

(1) by (b2) each row of $\mathcal{M}_i$ has $T$ 1’s, or, equivalently, has weight $T$.

(2) by (b3) and (b4) each column of $\mathcal{M}_i$ has weight $\alpha_i$ for the first $T$ columns and weight $\beta_i$ for the last $N - T$ columns.

Therefore, the general scheme can be implemented easily so long as we have built the binary matrices with the above two properties. We next give an explicit construction of these locator matrices.

Let $E(u,v)_{m \times n}$ denote an $m \times n$ binary matrix that has weight $u$ for each row and weight $v$ for each column. Evidently, a necessary condition for existence of $E(u,v)_{m \times n}$ is

$$\begin{align*}
0 & \leq u \leq m, \\
0 & \leq v \leq n,
\end{align*}$$

We can generally define $E(u,v)_{m \times n}$ as below:

$$E(u,v)_{m \times n} = \begin{pmatrix} e_u & e_u P_u & \ldots & e_u P_{u,v} \\
0 & 1 & \ldots & 0 \end{pmatrix}$$

and $P_u = P^n, P = \begin{pmatrix} 0 & I_{n-1} \\
1 & 0 \end{pmatrix}$. It is easy to see that $e_u P_u$ is a cyclic right shift of $e_u$ by $u$ coordinates.

Then combining with the solutions given in (7) and (8) one can verify the following matrices define the locator matrices needed for properties (1)-(2): for $N \geq 2T$,

$$\mathcal{M}_1 = \begin{pmatrix} 0 & E(T, \beta_1)_{N-N,0} \end{pmatrix},$$

and for $T < N < 2T$,

$$\mathcal{M}_1 = \begin{pmatrix} E(2T-N, \beta_1)_{0,T} & E(N-T, \beta_1)_{0,T} \end{pmatrix}.$$

D. Comparison with previous schemes

In this section we make comparisons with previous capacity-achieving $T$-PIR schemes to give some insights into why our scheme can reduce both the sub-packetization and the field size. Since our scheme deals with PIR from $T$-colluding servers with replicated databases, we restrict to the comparisons with previous schemes under the same model, such as the schemes in [9], [14].

First, reduction on the sub-packetization from the scheme in [9] is due to abandonment of the symmetry across all servers. Instead, we divide all servers into two groups, the first $T$ servers in one group and the remaining $N-T$ servers in the other, and then enforce the symmetry across the servers within each group. That is, for any $\Delta \subseteq [M]$ with $|\Delta| = i$, the number of $\Delta$-type sums provided by $\text{Serv}^{(i)}$ is $\alpha_i$ for $1 \leq j \leq T$ and $\beta_i$ for $T < j \leq N$. The same technique was used in [14] and the optimal sub-packetization is also achieved there. In one word, the asymmetry between servers means only a fraction of the answer table in [9] is retained, however, that is enough for a capacity-achieving PIR scheme. On the other hand, the two-group partition and the partial symmetry within each group help to keep a neat implementation of the scheme.

Comparing with the scheme in [14], we further reduce the field size to $q \geq N$, because in this work we replace the long MDS codes having information rate $\frac{1}{N}$ used in [14] with products of $[N,T]$ MDS codes. Therefore, an MDS code with much shorter length is sufficient here and the field size is reduced accordingly. Specifically, for any $\Delta \subseteq [M] - \{\theta\}$ with $|\Delta| = i$, the correctness condition in both work relies on the interference parts in all $\Delta$-type sums being recoverable as MDS parities from all $\Delta$-type sums. However, in [14] the interferences are encoded as a whole by using an MDS code with rate $\frac{1}{N}$ and length $\left( T \alpha_i + (N - T) \beta_i \right)_{N \times N} = d_i N$, while in this work the interferences are encoded part by part through $d_i$ MDS codewords of length $N$. However, in order to maintain the privacy condition and the partial symmetric structure, we need to arrange these interferences in an properly designed order as elaborated by the locator matrices in Section [V-C].

V. Conclusion

The $T$-PIR scheme presented in this work deals with PIR problems from $T$-colluding servers with replicated databases. It simultaneously attains the highest rate and the smallest sub-packetization. Moreover, it only requires field size $q \geq N$ which is a great reduction comparing with existing capacity-achieving $T$-PIR schemes.

References

[1] K. Banawan and S. Ulukus “The capacity of private information retrieval from coded databases,” [arXiv:1609.08138], 2016.

[2] A. Beimel, Y. Ishai, E. Kushilevitz, and I. Orlov, “Share conversion and private information retrieval,” in Proc. 27th Ann. Conf. Comput. Complex., pp. 258–268, Jun. 2012.

[3] T. H. Chan, S.-W. Ho, and H. Yamamoto, “Private information retrieval for coded storage,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2015, pp. 2842–2846, Jun. 2015.

[4] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, “Private information retrieval,” Proc. 36-th IEEE Symposium on Foundations of Computer Science, pp. 41–50, 1995.

[5] Z. Dvir, S. Gopi, “2-Server PIR with Sub-Polynomial Communication,” Proc. thirty-first Annual ACM Symposium on Theory of Computing (STOC’13), pp. 577–584, 2015.

[6] K. Efremenko, “3-query locally decodable codes of subexponential length,” Proc. of the 36-th Annual ACM Symposium on Theory of Computing (STOC’09), pp. 39–44, 2009.

[7] W. Gasarch, “A survey on private information retrieval,” in Bulletin of the EATCS. Citeseer, 2004.

[8] H. Sun, S. A. Jafar, “The capacity of private information retrieval,” IEEE Trans. on Inf. Theory, vol.63, no.7, pp. 4075–4088, Jul. 2017.

[9] H. Sun, S. A. Jafar, “The capacity of private information retrieval with colluding databases,” IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp.941–946, 2016.

[10] H. Sun, S. A. Jafar, “The Capacity of Symmetric Private Information Retrieval,” IEEE Globecom Workshops (GC Wkshps), pp.1–5, 2016.

[11] H. Sun, S. A. Jafar, “Optimal Download Cost of Private Information Retrieval for Arbitrary Message Length,” IEEE Transactions on Information Forensics and Security, vol.12, no.12, pp.2920–2932, 2017.

[12] Qiwen Wang, and Mikael Skoglund, “Symmetric Private Information Retrieval for MDS Coded Distributed Storage,” Proceedings of IEEE International Conference on Communications, ICC 2017.

[13] S. Yekhanin, “Locally Decodable Codes and Private Information Retrieval Schemes,” Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[14] Zhiyang Zhang and Jingke Xu “Capacity-Achieving PIR Schemes with Optimal Sub-Packetization,” arXiv:1710.11370, 2017.