CAPACITY-ACHIEVING PRIVATE INFORMATION RETRIEVAL
SCHEME WITH A SMALLER SUB-PACKETIZATION

WENQIN ZHANG
School of Mathematics, Southwest Jiaotong University
Chengdu, 610031, China
State Key Laboratory of Cryptology, P. O. Box 5159
Beijing, 100878, China

ZHENGCHUN ZHOU
School of Mathematics, Southwest Jiaotong University
Chengdu, 610031, China

UDAYA PARAMPALLI
Department of Computer Science and Software Engineering
University of Melbourne, Melbourne, Australia

VLADIMIR SIDORENKO
Institute for Communications Engineering
Technical University of Munich, Germany

(Communicated by Sihem Mesnager)

Abstract. Private information retrieval (PIR) allows a user to retrieve one out of $M$ messages from $N$ servers without revealing the identity of the desired message. Every message consists of $L$ symbols (packets) from an additive group and the length $L$ is called the sub-packetization. A PIR scheme with download cost (DC) $D/L$ is implemented by querying $D$ sums of the symbols to servers. We assume that each uncoded server can store up to $tLM/N$ symbols, $t \in \{1, 2, \cdots, N\}$. The minimum DC of storage constrained PIR was determined by Attia et al. in 2018 to be $DC_{\text{min}} = 1 + 1/t + 1/t^2 + \cdots + 1/t^{M-1}$. The capacity of storage constrained PIR (equivalently, the reciprocal of minimum download cost) is the maximum number of bits of desired symbols that can be privately retrieved per bit of downloaded symbols. Tandon et al. designed a capacity-achieving PIR scheme with sub-packetization $L' = \binom{N}{t} t^M$ of each message. In this paper, we design a PIR scheme with $t$ times smaller sub-packetization $L'/t$ and with the minimum DC for any parameters $N, M, t$. We also prove that $t^{M-1}$ is a factor of sub-packetization $L$ for any capacity-achieving PIR scheme from storage constrained servers.

2020 Mathematics Subject Classification: Primary: 06E75, 94A60, 11T23.
Key words and phrases: Private information retrieval, minimum download cost, sub-packetization, storage constrained servers.

The work of W. Zhang and Z. Zhou was supported by the National Natural Science Foundation of China under Grant 61672028, and partially supported by National Cryptography Development Fund under Grant MMJJ20180103. Vladimir Sidorenko is on leave from Institute for Information Transmission Problems, Russian Academy of Sciences. His work is supported by the Russian Government, Contract No 14.W03.31.0019.
1. Introduction

Private information retrieval (PIR) is a strategy to protect individual privacy on a public network platform. The PIR problem includes $M$ messages that are assumed to be independent, $N$ non-communicating servers $Serv_1, \ldots, Serv_N$ and a user who wants to retrieve one message out of $M$ messages without revealing the identity of the desired message.

The original concept of PIR [6] was proposed by Chor et al. in 1995. In practice, many applications are related to private information retrieval problems, like protecting the identity of stock market records reviewed by an investor or protecting the nature of restricted content browsed by activists on the internet in oppressive regimes.

The classical PIR setting is that $M$ messages of length $L$ symbols (packets) are repeatedly stored in $N$ non-communicating servers. The length $L$ is called the message sub-packetization. A trivial solution for a user to retrieve a message with PIR is to download all messages. This solution, however, is very inefficient. To measure the performance of a PIR scheme, one defines the PIR rate $R$ as the ratio of the desired message size $L$ to the total number of downloaded symbols $D$, i.e., $R = L/D$. Consequently, the PIR rate for the trivial solution is $1/M$. Additionally, the inverse of the rate, given by $1/R = D/L$, is referred to as the download cost.

Furthermore, we define the PIR capacity as the supremum of PIR rates overall PIR schemes with fixed parameters $N$ and $M$.

Sun and Jafar [9] found the capacity of PIR for replicated servers. Interesting extensions for PIR capacity were obtained for $T$-colluding PIR [11], PIR with colluding and Byzantine servers [3], multi-message PIR [2], PIR with symmetric privacy [10], MDS-coded symmetric PIR [16], PIR with MDS coded non-colluding servers [4], coded PIR with colluding servers [14]-[20], PIR with side information [8], [5]. There are two key research directions connected with determining PIR capacity: obtaining or proving tight upper bounds and designing capacity-achieving PIR schemes. Most of the existing capacity-achieving PIR schemes involve querying certain linear combinations of the message symbols to servers.

Sun and Jafar characterized the minimum download cost for arbitrary message length and designed capacity-achieving PIR schemes with message length $L = N^{M-1}$ [13]. Zhang and Xu studied the optimal sub-packetization for $T$-colluding PIR from replicated servers [21]. Later they proved the lower bound on message length $L$ for the linear capacity-achieving MDS coded PIR scheme [18]. In this paper, we assume that the message symbols belong to an additive group, and a PIR scheme can be realized by querying a sum of message symbols to servers. We assume that each server can store up to $\mu LM$ symbols, where $\mu = t/N$, $t \in \{1, 2, \ldots, N\} \cong [N]$. The information-theoretical description of the PIR problem can be found in [15] where the following lower bound for download cost (DC) was obtained

$$D \geq \tilde{D}(t) = 1 + 1/t + 1/t^2 + \cdots + 1/t^{M-1}.$$  

A PIR scheme with sub-packetization $L = \binom{N}{t} \times t^M$ that reaches the bound was designed in [15].

For the case of $t = N$, which corresponds to the situation where each server stores the full $ML$ symbols, a PIR scheme with minimal download cost $1 + 1/N + \cdots + 1/N^{M-1}$ was proposed in [13]. In the other extreme case of $t = 1$, all symbols
stored in the servers have to be downloaded, and the minimal download cost given by (1) was achieved. Hence in this paper, we only consider the case $1 < t < N$.

**Our contribution.** For arbitrary $N$, $M$ and $1 < t < N$, we prove that $t^{M-1}$ is a factor of sub-packetization $L$ for any capacity-achieving PIR scheme from storage constrained servers, and design a PIR scheme with sub-packetization $L = \binom{N}{t} \times t^{M-1}$ which achieves minimum possible download cost $\hat{D}(t)$ in (1). Comparing with the scheme in [15, 1], our scheme has a reduced sub-packetization parameter $L$ by a factor $t$. In particular, the improvement amounts to reduction by $t$ times to the number of requests to servers. The difference between our scheme and the one in [15] lies in the requirement of a symmetric rule across servers. Recall that the PIR scheme of [15] is based on the iterative application of three steps corresponding to symmetry across servers, symmetry across messages within the query to each server, and exploiting side information. In this study, we eliminate the requirement of symmetry across servers leading to reduced sub-packetization length.

2. **Preliminaries**

2.1. **PIR Model.** Consider a distributed storage system with $N$ non-colluding servers storing $M$ random statistically independent messages $W_1, W_2, \cdots, W_M$, where the size of each message is $L$ symbols, i.e.,

$$\forall i \in [M], \quad H(W_i) = L,$$

$$H(W_1, W_2, \cdots, W_M) = ML.$$

Herein, $H(\cdot)$ is the binary entropy function.

We assume that each server has a storage capacity of $\mu ML$ symbols, where $\mu$ is a rational number with $1/N \leq \mu \leq 1$. We denote the contents stored across the servers by $Z_1, Z_2, \cdots, Z_N$. Then, for each server, we have the following storage constraint

$$(2) \quad H(Z_1) = H(Z_2) = \cdots = H(Z_N) = \mu ML.$$

A PIR scheme allows a user to retrieve a message $W_i$ for $i \in [M]$ from $N$ non-colluding servers without revealing the identity $i$ of the considered message $W_i$. The user generates $N$ queries $Q_1^i, Q_2^i, \cdots, Q_N^i$, and sends them to the corresponding servers. The user does not have information about message $W_i$ in advance, and the queries are independent of the messages, i.e.,

$$I(W_1, \cdots, W_M; Q_1^i, \cdots, Q_N^i) = 0, \quad \forall i \in [M],$$

where $I(\cdot)$ is the mutual information.

For all $n \in [N]$, server $Serv^n$ responds with the answer $A_n^i$, which is a function of query $Q_n^i$, and symbols $Z_n$ stored in the $n$-th server. This means that

$$H(A_n^i|Q_n^i, Z_n^i) = 0, \quad \forall i \in [M], \quad \forall n \in [N].$$

From all of the answers received, the user can retrieve message $W_i$ exactly. That is,

$$H(W_i|A_1^i, \cdots, A_N^i, Q_1^i, \cdots, Q_N^i) = 0, \quad \forall i \in [M].$$

In order to prevent the servers from learning the queried message, the PIR scheme also has to satisfy:

$$(3) \quad I(i; Q_1^i, A_1^i, W_1, \cdots, W_M, Z_1, \cdots, Z_N) = 0, \quad \forall i \in [M], \quad n \in [N]$$

which means that the index of the desired message $i$ must be hidden from the query and answer of $Serv^n$ as well as all messages and the storage content of other servers.
2.2. STORAGE STRATEGY FOR $\mu = t/N$: Consider uncoded storage placement constraint servers with each server’s storage capacity being $\mu ML$ symbols, where $M$ is the number of messages, $L$ is the size of each message in symbols, and $\mu = t/N \in [1/N, 1]$ is the normalized storage with $t = 1, \ldots, N$. Hence, we present a storage scheme to satisfy the requirement of storage constraint (2). The key idea is that we split each message $W_i$ into $\binom{N}{t}$ sub-messages of equal size for a fixed parameter $t \in [N]$. In particular, each sub-message is indexed by a subset $S \subset [N]$ of size $t$. For $i \in [M]$, the sub-message $W_{i_S}$ is repeatedly stored in each server $\text{Serv}_n$ with $n \in S$. We assume that the size of each sub-message is $t M - 1$ symbols.

Hence the sub-packetization of each message is $L = \left(\frac{N}{t}\right) \times t^{M-1}$, i.e., each message of length $L$ is split into $\binom{N}{t}$ sub-messages of length $t^{M-1}$. To the end, we note that for every message, each server stores $\binom{N}{t} - 1$ sub-messages (this corresponds to the number of sub-sets of servers of size $t$ in which the given server is present).

For example, consider the case when $t = 2$, $N = 3$, and $M = 3$. Let $A$, $B$, and $C$ denote three messages. We split each message into $\binom{3}{2} = 3$ sub-messages: $A = (A_{12}, A_{13}, A_{23})$, $B = (B_{12}, B_{13}, B_{23})$, and $C = (C_{12}, C_{13}, C_{23})$. For $S = \{1, 2\}$, $A_{12}$ is stored in $\text{Serv}_1$ and $\text{Serv}_2$. The storage strategy is shown in Table 1.

| $\text{Serv}_1$ | $\text{Serv}_2$ | $\text{Serv}_3$ |
|-----------------|-----------------|-----------------|
| $A_{12}, A_{13}$ | $A_{12}, A_{23}$ | $A_{13}, A_{23}$ |
| $B_{12}, B_{13}$ | $B_{12}, B_{23}$ | $B_{13}, B_{23}$ |
| $C_{12}, C_{13}$ | $C_{12}, C_{23}$ | $C_{13}, C_{23}$ |

Table 1. Sub-messages stored in the servers.

The amount of symbols stored in each server is:

$$M \times \left(\frac{N - 1}{t - 1}\right) \times t^{M-1} = \frac{t}{N} \times M \times \left(\frac{N}{t} \times \left(\frac{N - 1}{t - 1}\right)\right) \times t^{M-1}$$

$$= \frac{t}{N} \times M \times \left(\frac{N}{t}\right) \times t^{M-1}$$

$$= \mu ML.$$  

Hence, for each server, the storage scheme satisfies the storage constraint with equality.

3. THE PIR SCHEME WITH $L = \binom{N}{t} t^{M-1}$ FOR $N \mid \binom{N}{t}$

Our storage constrained PIR scheme is based on the scheme of Sun and Jafar in [13] and the scheme of Tandon et al. [15]. The main idea is that we apply the two steps: the one is that enforcing message symmetry, i.e., symbols of the desired message and the undesired messages in the queries are enforced to be symmetric, the other is that exploiting side information. Here, symbols of the undesired messages are called side information. Note that only side information shared between servers can be exploited. In this section, for $N \mid \binom{N}{t}$, we present a PIR scheme with sub-packetization $L = \binom{N}{t} t^{M-1}$ that achieves the minimum download cost in storage constrained condition. Let us illustrate the key idea by two simple examples for small values of $M$ and $N$. 

3.1. Two examples.

Example 1. \((N = 3\) servers, \(M = 2\) messages)  

Consider the PIR setting with \(t = 2\), \(N = 3\), \(M = 2\), \(L = \binom{2}{1} \times 2 = 6\). We split each message into \(\binom{N}{1} = 3\) sub-messages of size \(t^{M-1} = 2\) as follows

\[
A = \{A_{12}, A_{23}, A_{13}\} = \{a_{12}, a_{23}, a_{13}, a_{23}, a_{13}, a_{13}\},
\]

\[
B = \{B_{12}, B_{23}, B_{13}\} = \{b_{12}, b_{12}, b_{23}, b_{23}, b_{13}, b_{13}\}.
\]

Then the total storage in each server is \(\mu ML = \frac{2}{3} \times 2 \times 6 = 8\) symbols, and the symbols stored in the server \(\text{Serv}^n\) for \(n \in [3]\) are shown in Table 2.

| \(\text{Serv}^1\) | \(\text{Serv}^2\) | \(\text{Serv}^3\) |
|-----------------|-----------------|-----------------|
| \(a_{12}\), \(a_{13}\) | \(a_{12}\), \(a_{13}\) | \(a_{13}\), \(a_{13}\) |
| \(a_{12}\), \(a_{13}\) | \(a_{23}\), \(a_{23}\) | \(a_{23}\), \(a_{23}\) |
| \(b_{12}\), \(b_{12}\) | \(b_{12}\), \(b_{12}\), \(b_{23}\), \(b_{23}\) | \(b_{13}\), \(b_{13}\), \(b_{23}\), \(b_{23}\) |

Table 2. Symbols stored in the servers.

Suppose a user wants to download message \(A\). For each subset \(S \subset [3]\) with \(|S| = 2\) let \((a^S_1, a^S_2)\) represents a random permutation of \(t^{M-1} = 2\) symbols from the sub-message \(A_S\). Similarly, \((b^S_3, b^S_3)\) is an independent random permutation for \(B_S\). All permutations are generated by the user and are not known to the servers. First, the single symbol \(a^S_{12}\) is downloaded from \(\text{Serv}^1\). In fact, we download the symbol \(a^S_{12}\) or \(a^S_{12}\) depending on the random permutation. Because of the requirement of message symmetry, \(b^S_{12}\) is also requested and downloaded. Both of these symbols are stored on \(\text{Serv}^1\) and can be downloaded from it. Similarly, we download \(a^S_{13}\), \(b^S_{13}\) from \(\text{Serv}^2\), and \(a^S_{13}\), \(b^S_{13}\) from \(\text{Serv}^3\). The user can exploit downloaded side information, i.e., symbols of the undesired message \(B\), to construct new queries which are the sum of symbols from the desired message and available side information. For e.g. \(\text{Serv}^3\), the available side information is \(b_{13}\). The queries for message \(A\) are constructed in Table 3.

| \(\text{Serv}^1\) | \(\text{Serv}^2\) | \(\text{Serv}^3\) |
|-----------------|-----------------|-----------------|
| \(a_{12}\) | \(a_{23}\) | \(a_{13}\) |
| \(b_{12}\) | \(b_{23}\) | \(b_{13}\) |
| \(a_{13} + b_{13}\) | \(a_{12} + b_{12}\) | \(a_{23} + b_{23}\) |

Table 3. Queries to download message \(A\).

Note that every desired symbol is either downloaded directly or as a sum with known side information that can be subtracted to retrieve the desired symbol. \(a^S_{13}\), \(a^S_{12}\), \(a^S_{23}\) can be recovered from the sums \(a^S_{13} + b^S_{13}\), \(a^S_{12} + b^S_{12}\), \(a^S_{23} + b^S_{23}\) using downloaded side information. Thus, the desired message \(A\) can be retrieved.

As for the privacy condition, it is sufficient to show that for any individual server, its query sequence for retrieving \(A\) has the same distribution as the query sequence for retrieving \(B\). Actually, if the user wants \(B\), the scheme goes the same way except for swapping \(a\) and \(b\) in Table 3, as shown in Table 4.
Finally, the number of the useful downloaded symbols is \( L = 6 \) and the total number of downloaded symbols is \( D = 9 \). So this scheme has download cost \( \frac{D}{L} = \frac{3}{2} \) which attains the lower bound in (1).

It can be seen from Table 3 and Table III in [15] that the number \( (D = 18) \) of queries to servers is twice larger than the number for the same \( N, M, t \) in our case. In general, the number of queries \( D \) in the scheme [15] is \( t \) times larger than the one in our scheme, since both schemes have the same download cost \( D/L \) while [15] uses \( t \) times larger sub-packetization \( L \).

**Example 2.** \((N = 5\) servers, \(M = 3\) messages) Consider the PIR scheme with \( N = 5\), \( M = 3\), \( t = 2\), i.e., \( \mu = \frac{2}{5} \). Hence, the sub-packetization \( L \) of each message is \((\frac{3}{2})2^2 = 40\) symbols. We respectively split messages \( A, B, C \) into \((\frac{3}{2}) = 10\) sub-messages: \(\{A_{12}, \ldots, A_{45}\}, \{B_{12}, \ldots, B_{45}\}, \{C_{12}, \ldots, C_{45}\}\), with the size of each sub-message being \(4\). Specifically,

\[
A = (a_{12}^1, \ldots, a_{12}^4; a_{13}^1, \ldots, a_{15}^1; a_{14}^1, \ldots, a_{14}^4; a_{15}^1, \ldots, a_{15}^4; \ldots; a_{45}^1, \ldots, a_{45}^4)
\]

\[
B = (b_{12}^1, \ldots, b_{12}^4; b_{13}^1, \ldots, b_{15}^1; b_{14}^1, \ldots, b_{14}^4; b_{15}^1, \ldots, b_{15}^4; \ldots; b_{45}^1, \ldots, b_{45}^4)
\]

\[
C = (c_{12}^1, \ldots, c_{12}^4; c_{13}^1, \ldots, c_{15}^1; c_{14}^1, \ldots, c_{14}^4; c_{15}^1, \ldots, c_{15}^4; \ldots; c_{45}^1, \ldots, c_{45}^4)
\]

Due to the storage restriction \(\mu ML = \frac{2}{5} \times 3 \times 40 = 48\), for each server, the sub-messages are stored as shown in Table 5.

| Serv^1 | Serv^2 | Serv^3 | Serv^4 | Serv^5 |
|--------|--------|--------|--------|--------|
| A_{12}, A_{13} | A_{12}, A_{23} | A_{13}, A_{23} | A_{14}, A_{24} | A_{15}, A_{25} |
| A_{14}, A_{15} | A_{24}, A_{25} | A_{34}, A_{35} | A_{34}, A_{45} | A_{35}, A_{45} |
| B_{12}, B_{13} | B_{12}, B_{23} | B_{13}, B_{23} | B_{14}, B_{24} | B_{15}, B_{25} |
| B_{14}, B_{15} | B_{24}, B_{25} | B_{34}, B_{35} | B_{34}, B_{45} | B_{35}, B_{45} |
| C_{12}, C_{13} | C_{12}, C_{23} | C_{13}, C_{23} | C_{14}, C_{24} | C_{15}, C_{25} |
| C_{14}, C_{15} | C_{24}, C_{25} | C_{34}, C_{35} | C_{34}, C_{45} | C_{35}, C_{45} |

**Table 5.** Sub-messages stored in the servers.

Assume that we want to retrieve message \( A \) privately. Let \((a_{S}^1, a_{S}^2, a_{S}^3, a_{S}^4)\) represents a random permutation of \(\ell^{|\ell|-1} = 4\) symbols from the sub-message \(A_{S}\) for each subset \(S \subset [5]\) with \(|S| = 2\). Similarly, \((b_{S}^1, b_{S}^2, b_{S}^3, b_{S}^4)\) and \((c_{S}^1, c_{S}^2, c_{S}^3, c_{S}^4)\) are...
random permutations of 4 symbols in the sub-messages $B_S$ and $C_S$ respectively. The queries for the desired message $A$ are constructed in Table 6.

Let us explain this example. In Step 1, we directly download a single symbol of 2 sub-messages from each server. From $Serv^1$, we download $a_{12}^1, a_{13}^1$ of message $A$. To guarantee message symmetry, $b_{12}^1, b_{13}^1, c_{12}^1, c_{13}^1$ also have to be downloaded. Similarly, we download $a_{13}^2, a_{14}^2, b_{14}^2$ from $Serv^2$; $a_{14}^3, a_{15}^3, b_{15}^3, c_{15}^3$ from $Serv^3$; $a_{14}^4, a_{15}^4, b_{15}^4, c_{14}^4, c_{15}^4$ from $Serv^4$; $a_{15}^5, b_{15}^5, c_{14}^5, c_{15}^5$ from $Serv^5$. In Step 2, applying side information downloaded in the first step, we obtain the new queries which are the sum of symbols from the desired message and side information available from other servers. Note that we can only utilize those symbols which are stored in $Serv^1$ as side information. For example in Table 6, we download in Step 2 ($a_{14}^2 + b_{14}^1, a_{15}^2 + b_{15}^1, a_{14}^3 + c_{14}^1, a_{15}^3 + c_{15}^1$) from $Serv^1$. Then in Step 3, we download the sum of three symbols of different messages (i.e., $a + b + c$’s) from each server. Following the same principle as before, $a_{12}^2 + b_{12}^1 + c_{12}^1$ and $a_{14}^4 + b_{14}^3 + c_{14}^3$ are downloaded from $Serv^1$. The way of downloading symbols from $Serv^2, \cdots, Serv^5$ is similar to $Serv^1$.

From Table 6, we readily verify that message $A$ can be recovered. The privacy condition can be verified as in Example 2. The number of useful symbols downloaded is 40 while the total number of symbols downloaded is 70. This leads to the download cost of $\frac{70}{40} = \frac{7}{4}$ which attains the lower bound (1).

3.2. Formal description of the general scheme for $N \mid \binom{N}{2}$. From the above examples we see that our general scheme consists of $M$ steps, where in each step we download tuples of symbols using obtained side information to maintain message symmetry. Most importantly, the exploitation of side information is carefully designed to account for the limited storage capabilities of the servers. Specifically, the answers are formed by iteratively applying the following two steps: enforcing symmetry across messages within the query to each server, and combining side information with new desired symbols. Next, for clarity, we define the following notation.

1. $U_m$: For all $m \in [M]$, define ordered tuples

$$U_m = [U_m(1), U_m(2), \cdots, U_m(L)],$$

where $U_m(i)$ is a symbol and the $U_m$ symbols will eventually be mapped to random permutations of message $W_m$. 

Table 6. Queries to download message $A$
2. $k$-sum: We use the terminology “$k$-sum” to denote the sum of $k$ distinct variables, each drawn from a different $U_m$ tuple, i.e., $U_{m_1}(j_{m_1}) + U_{m_2}(j_{m_2}) + \cdots + U_{m_k}(j_{m_k})$, where $m_1, m_2, \cdots, m_k \in [M]$ are all distinct indices, and $j_m \in [L]$. For instance, in Table 3 of Example 1 for server $Serv^1$, we get $2$-sum in the second row: $a_{13}^1 + b_{13}^1$.

3. $T$ - type: For $1 \leq k \leq M$, and for a subset $T \subseteq [M]$ with $|T| = k$, we define the sum $\sum_{m \in T} U_m(j_m)$ a $T$ - type $k$-sum. The queries to each server are composed of these $k$-sums.

For instance, in Table 6 of Example 2, for server $Serv^2$ with $T = \{1, 3\}$, we have a $T$ - type $2$-sum $\sum_{m \in T} U_m(j_m) = a_{12}^3 + c_{12}^3$.

Next, we introduce the download steps.

**Step 1:** Let $k = 1$, we download a single symbol of $\lambda = (N) / N$ arbitrary different sub-messages of a message from each server respectively such that each subset $S$ appears once in the $1$-sum. In other words, we download $1$ symbol of the sub-message $W_{1,S_1}^n, W_{1,S_2}^n, \cdots, W_{1,S_N}^n$ respectively from each $Serv^n$, $n \in [N]$. For all servers, we download $(N) / N \times N = (N)$ symbols of the total sub-message of the desired message $W_t$. To guarantee privacy, we maintain message symmetry for each server. Then we perform the same download operation for all remaining $(M - 1)$ messages. Hence the total number of downloaded symbols of the desired message $W_t$ is $N\lambda$, whereas the total number of downloaded symbols is $M\lambda$.

**Step 2:** Let $k = 2$, we download the $2$-sum of symbols which contain the desired message and available side information. From the server $Serv^n$, we download desired symbols of message $W_t$ along with undesired symbols of the remaining $(M - 1)$ undesired messages. These undesired symbols have been downloaded from other servers (except $Serv^n$) in Step 1 and also stored in $Serv^n$. Note that the number of the sub-messages from message $A$ stored in $Serv^n$ is $(N-1)$. Due to the fact that the single symbol of $\lambda$ sub-messages have already been downloaded in Step 1, the number of side information symbols that can be paired with $Serv^n$ is $(N-1) - \lambda = (t-1)\lambda$. Due to message symmetry, $M \choose 2$ possible types of $2$-sum must be downloaded, and the number of pairs of a desired message and the undesired message is $M \choose 2 - 1$. Hence, for all servers, the number of desired symbols downloaded in Step 2 is $(M - 1)N\lambda(t - 1)$, whereas the total number of downloaded symbols is $M \choose 2 - 1 \times N\lambda(t - 1)$.

**Step $k$:** We download the $T$-type $k$-sum, and all types $T \subseteq [M]$ with $|T| = k$ will occur. For the desired message, we download the $k$-sum which includes desired symbols of message $W_t$ and undesired symbols of the $k - 1$ messages selected from remaining $(M - 1)$ message except message $W_t$. These undesired symbols have been downloaded from other servers and are also stored in $Serv^n$. Because all possible $M \choose k$ $T$-type of $k$-sum must be downloaded, and the number of the $k$-sum, consisting of symbols from the desired message and the undesired message, is $M \choose k - 1$. Hence, from all servers, the number of desired symbols downloaded in Step $k$ is $M \choose k - 1 \times N\lambda(t - 1)^{k-1}$, whereas the total number of downloaded symbols is $M \choose k \times N\lambda(t - 1)^{k-1}$. We continue till Step $M$ similarly.

The following lemma describes the above downloads.

**Lemma 3.1.** Each server downloads exactly $M \choose k \lambda(t - 1)^{k-1}$ instances of the $k$-sum of each possible type in Step $k$. 
Proof. It is easy to verify that the conclusion holds for the case \( k = 1 \). Suppose that each server downloads exactly \( \binom{M}{k} \lambda (t - 1)^{k - 1} \) instances of the \( k \)-sum of each possible type in Step \( k \), then our goal is to prove that in Step \( k + 1 \) each server downloads \( \binom{M}{k+1} \lambda (t - 1)^{k} \) instances of the \( (k+1) \)-sum.

The query \( Q_{i}^{n} \) is composed of \( \binom{N-1}{t-1} \) subsets \( S \subset [1 : N] \) of size \( t \), where \( n \in S \). The query labeled with \( S \) only involves the sub-messages stored at the server \( Serv_{n} \), where \( n \in S \). Subsequently, because of enforcing symmetry across messages within the query to each server, the queries to each server contain the same number of \( k \)-sums for every type \( T \) with \( |T| = k \). For any subset \( T \subseteq [M] \), we denote the number of the \( T \)-type \( k \)-sum by \( \delta_{|T|, T}^{n} \) for the server \( Serv_{n} \). Then, for subset \( \{i\} \in T \) and \( T' \subseteq [M]\{i\} \) with the same size \( k \), \( \delta_{k,T}^{n} = \delta_{k,T'}^{n} \).

Considering the \( (k + 1) \)-sum, we will use the \( k \)-sum of type \( T' \subseteq [M]\{i\} \) that have been downloaded in Step \( k \) to construct the \( (k + 1) \)-sum, and \( |T'| = k \). We download the \( k + 1 \) sum which contains the desired symbols of message \( W_{i} \) and the undesired symbols of other \( k \) messages from \( Serv_{n} \). The form is \( W_{i,S}(j) + U_{m_{1}}(j_{1}) + U_{m_{2}}(j_{2}) + \cdots + U_{m_{k}}(j_{k}) \), where \( m \in T' \) and \( W_{i,S}(j) \) is a desired symbol that has not been downloaded in the previous steps. Note that \( W_{i,S}(j) \) and the undesired symbols of \( T' \)-type \( k \)-sum are all stored in \( Serv_{n} \). Due to \( \delta_{k,T}^{n} = \delta_{k,T'}^{n} = (t-1)^{k-1}\lambda \) for \( Serv_{n} \), we have

\[
\delta_{k+1,T}^{n} = \frac{(t-1)^{k-1} \times \lambda \times N}{t} \times \binom{N-1}{t} - (t-1)^{k-1}\lambda = (t-1)^{k}\lambda,
\]

where \( T = T' \cup \{i\} \). For each server, the \( \binom{M}{k+1} \) \( k + 1 \)-sum of possible \( T \)-type must be downloaded. Hence, each server downloads exactly \( \binom{M}{k+1} \lambda (t - 1)^{k} \) instances of the \( k \)-sum of each possible type in Step \( k + 1 \), which completes the proof. \( \square \)

Lemmas 3.2 and 3.3 below show that the proposed PIR scheme is correct, private, and has the minimum download cost (1).

**Lemma 3.2.** The achievable scheme is correct and achieves the capacity (1).

**Proof.** The scheme is correct since all symbols of the desired message either are retrieved directly, or appear in a sum with side information downloaded from other servers. Furthermore, side information can be subtracted to retrieve desired symbols. Now, let us calculate the total number of desired and downloaded symbols shown in Table 7.

The total number of downloaded symbols (all servers) is

\[
D = N \sum_{k=1}^{M} \binom{M}{k} \lambda (t - 1)^{k-1} = N \lambda \sum_{k=1}^{M} \binom{M}{k} (t - 1)^{k-1} (t - 1) + \lambda - \lambda \frac{t}{t - 1} = \frac{N \lambda (t - 1 + 1)^{M} - \lambda}{t - 1} = \frac{N \lambda (t^{M} - 1)}{t - 1}.
\]
Table 7. The number of symbols downloaded in each step of the scheme

| Steps | Tuple | Number of Total symbols | Number of Useful symbols |
|-------|-------|-------------------------|-------------------------|
| Step 1 | Single | \( N\binom{M}{1} \lambda \) | \( N\lambda \) |
| Step 2 | Pair | \( N\binom{M}{2} \lambda (t-1) \) | \( N\binom{M-1}{2} \lambda (t-1) \) |
| Step 3 | Triple | \( N\binom{M}{3} \lambda (t-1)^2 \) | \( N\binom{M-1}{3} \lambda (t-1)^2 \) |
| ... | ... | ... | ... |
| Step k | k-tuple | \( N\binom{M}{k} \lambda (t-1)^{k-1} \) | \( N\binom{M-1}{k} \lambda (t-1)^{k-1} \) |
| ... | ... | ... | ... |
| Step M | M-tuple | \( N\binom{M}{M} \lambda (t-1)^{M-1} \) | \( N\binom{M-1}{M-1} \lambda (t-1)^{M-1} \) |

The number of downloaded desired symbols (all servers) is
\[
L = N\lambda \sum_{k=1}^{M} \binom{M-1}{k-1} (t-1)^{k-1} = N\lambda(t-1+1)^{M-1} = N\lambda t^{M-1} = \binom{N}{t} t^{M-1}.
\]

The construction contains \( M \) steps. In the \( k \)-th step, for each server there are \( \binom{M}{k} \lambda (t-1)^{k-1} \) equations comprised of \( k \)-sum. Among all these equations, \( \binom{M-1}{k-1} \lambda (t-1)^{k-1} \) of them contain desired symbols and side information, which have been downloaded directly in the \((k-1)\)-th step. For all servers, after subtracting out all side information, there are still \( \binom{N}{t} t^{M-1} \) desired symbols. Therefore, all symbols of the desired message are retrievable and the correctness constraint is satisfied.

Next, if \( t \in [N] \) such that \( N \mid \binom{N}{t} \), the download cost \( D(\mu) \) of the storage constrained PIR scheme when \( \mu = t/N \) is given by:
\[
D(\mu) = \frac{D}{L} = \frac{N\lambda(tM-1)/(t-1)}{N\lambda t^{M-1}} = 1 + \frac{1}{t} + \frac{1}{t^2} + \cdots + \frac{1}{t^{M-1}}.
\]

Thus, the download cost of our PIR scheme achieves the bound given in (1).

Lemma 3.3. The achievable scheme is private.

Proof. From the steps above, the \( \lambda^{M-1} \) symbols of the desired message \( W_i \) are involved in the query \( \mathcal{Q}_n^i \) to \( \text{Serv}^n \), \( n \in [N] \). The query \( \mathcal{Q}_n^i \) is divided into \( \binom{N-1}{t-1} \) sets \( \mathcal{S} \), where the size of each \( \mathcal{S} \in [1 : N] \) is \( t \). Each set \( \mathcal{S} \) is composed only of sub-messages \( W_{k,\mathcal{S}} \), where \( k \in [M] \), and the servers in the set \( \mathcal{S} \). Hence the number of downloaded symbols from each sub-message which is stored in server \( \text{Serv}^n \) are all the same, namely \( t^{M-2} \) symbols are downloaded from desired sub-message \( W_{i,\mathcal{S}} \) for each subset \( \mathcal{S} \) with \( |\mathcal{S}| = t \), where \( n \in \mathcal{S} \). The downloaded symbols from \( W_{i,\mathcal{S}} \) are denoted by \( \overrightarrow{w}_{i,\mathcal{S}} = (\overrightarrow{w}_{i,\mathcal{S}}^{(1)}, \overrightarrow{w}_{i,\mathcal{S}}^{(2)}, \cdots, \overrightarrow{w}_{i,\mathcal{S}}^{(t^{M-2})}) \).
Note that every query is composed of a sequence of \( M \) ordered steps. In Step \( k \), the user queries \( k \)-sum composed of \( k \) symbols from \( k \) different messages \( W_T \), where \( T \in [M] \) and \( |T| = k \). Due to message symmetry, there are \( \binom{M}{k} \) possible types of \( k \)-sum. Hence, the number of downloaded symbols from sub-messages of the undesired messages is the same as that of downloaded symbols from desired sub-message \( W_{i,S} \) for \( Serv^n, n \in S \). In addition, no message symbol appears more than once in a query for \( Serv^n \) and the query to each server has the same structure (\( k \)-sum).

For a given \( S \), suppose that each message \( W_{m,S}, m \in [M] \) is represented by a vector \( W_{m,S} = [w_{m,S}^{(1)}, w_{m,S}^{(2)}, \ldots, w_{m,S}^{(tM-1)}] \), where \( w_{m,S}^{(i)} \) is the \( i \)-th symbol of \( W_{m,S}^{(i)} \). The user privately chooses \( M \) permutations \( \gamma_1S, \gamma_2S, \ldots, \gamma_M \), uniformly random from all possible \( (tM-1)! \) permutations over the index set \([tM-1]\). It is clear that \( \gamma_1S, \gamma_2S, \ldots, \gamma_M \) are independently of each other and independently of \( i \). Let \( W_{m,S}^{\gamma_iS} = \gamma_iS(W_{m,S}) = [w_{m,S}^{\gamma_iS(1)}, w_{m,S}^{\gamma_iS(2)}, \ldots, w_{m,S}^{\gamma_iS(tM-1)}] \). Notice that any \( tM-2 \) symbols from \( W_{m,S}^{\gamma_iS} \) are involved in the query \( Q^n_i \) to \( Serv^n \). The probability of the query \( Q^n_i \) to server \( Serv^n \) for a given \( S \) is given by

\[
Prob((\overline{w}_{1,S}^{\gamma_iS}, \overline{w}_{2,S}^{\gamma_iS}, \ldots, \overline{w}_{M,S}^{\gamma_iS})) = \prod_{m=1}^{M} Prob(\overline{w}_{m,S}) = \left( \frac{1}{tM-1} \right) \left( \frac{1}{tM-1} \right) \cdots \left( \frac{1}{tM-1} - \frac{1}{tM-2} + 1 \right)^M
\]

which does not depend on the desired message index \( i \), i.e., \( I(i; Q^n_i) = 0 \). Hence the query \( Q^n_i \) is independent of \( i \) as well. Next, we show that the privacy condition (3). Note that \( I(i; W_1, \ldots, W_M, Z_1, \ldots, Z_N | Q^n_i) = 0 \) since the desired message index and the query are generated privately by the user without knowledge of the message. On the other hand, \( I(i; A^n_i | W_1, \ldots, W_M, Z_1, \ldots, Z_N, Q^n_i) = 0 \) due to the fact that the answer \( A^n_i \) is a function of the query \( Q^n_i \) and the messages \( Z_n \). Therefore,

\[
I(i; Q^n_i, A^n_i, W_1, \ldots, W_M, Z_1, \ldots, Z_N) = I(i; Q^n_i) + I(i; W_1, \ldots, W_M, Z_1, \ldots, Z_N | Q^n_i) + I(i; A^n_i | W_1, \ldots, W_M, Z_1, \ldots, Z_N, Q^n_i) = 0.
\]

This completes the proof of this lemma.

As one of our main results, the following theorem follows directly from Lemmas 3.2 and 3.3.

**Theorem 3.4.** For a given \( N, M \) and \( \mu \in \left[ \frac{1}{N}, 1 \right] \), such that \( t = \mu N \in [N] \), with sub-packetization \( L = \binom{N}{tM-1} \), a PIR scheme where a user retrieves one message out of \( M \) messages from \( N \) storage constrained servers privately has the achievable rate

\[
R = \left( 1 + \frac{1}{t} + \frac{1}{t^2} + \cdots + \frac{1}{t^{M-1}} \right)^{-1}.
\]

Using the PIR capacity and formula (1) for storage constrained servers, we are ready to prove that the minimum sub-packetization \( L \) should be a multiple of \( t^{M-1} \) to reach the PIR capacity.
Lemma 3.5. With notations as above, \( t^{M-1} \) must be a factor of sub-packetization \( L \) for any capacity-achieving PIR scheme from storage constrained servers.

Proof. It was shown in [1] that for arbitrary \( N \) and \( M \), the maximum achievable rate \( R = L/D \) across all PIR schemes is the capacity \( C = (1+1/t+1/t^2+\cdots+1/t^{M-1})^{-1} \). Therefore, for any capacity-achieving PIR scheme \( L/D = C \), the download cost should be

\[
D = \frac{L}{t^{M-1}} = \frac{L(1 + 1/t + 1/t^2 + \cdots + 1/t^{M-1})}{t^{M-1}}.
\]

Since \( \gcd(t^{M-1}, 1 + t + t^2 + \cdots + t^{M-1}) = 1 \) and \( D \) is an integer, it follows from (4) that \( t^{M-1}|L \). \( \square \)

4. THE PIR SCHEME WITH \( L = \binom{N}{t}t^{M-1} \) FOR ARBITRARY \( N, M \)

In this section, we generalize the above PIR scheme with sub-packetization \( L = \binom{N}{t}t^{M-1} \) to a general case for arbitrary \( N, M \). We prove that a PIR scheme attains the minimum download cost in the storage constrained condition. Let’s briefly illustrate the key idea with an example.

Example 3. \((N = 4, M = 2)\)

For this case, we have \( N = 4, M = 2 \) and \( \mu = \frac{2}{4} \) with \( t = 2 \). The sub-packetization \( L \) of each message is \( \binom{4}{2} \times 2 = 12 \) symbols, and the symbols are stored across each server in the same way as before. In our scheme, we divide the servers into two groups. The first \( d = \frac{N}{\gcd(N, t)} \) servers (\( \text{Serv}^1, \text{Serv}^2 \)) are in one group and the remaining \( N - d = 2 \) servers (\( \text{Serv}^3, \text{Serv}^4 \)) are in the other.

Suppose the desired message is \( A \). Let \((a_2^3, a_2^5)\) represent a random permutation of 12 symbols from the sub-message \( A_S \) for each subset \( S \subseteq [4] \) with \(|S| = 2 \). Similarly, \((b_2^3, b_2^5)\) is a random permutation of 2 symbols from the sub-messages \( B_S \).

Firstly, we download a single symbol \( a_{12}^1 \) from \( \text{Serv}^1 \). Due to message symmetry, \( b_{12}^1 \) is also downloaded. Similarly, we download \( a_{23}^1, a_{23}^3 \) from \( \text{Serv}^2 \), and \( a_{13}^1, b_{13}^1 \) from \( \text{Serv}^3 \). The user can exploit the undesired messages to construct new queries which are 2-sums of the desired message and side information available from other servers. The queries to each server are listed in Table 8.

| \( \text{Serv}^1 \) | \( \text{Serv}^2 \) | \( \text{Serv}^3 \) | \( \text{Serv}^4 \) |
|-----------------|-----------------|-----------------|-----------------|
| \( a_{12}^1, a_{14}^1 \) | \( a_{23}^1, a_{24}^3 \) | \( a_{13}^1 \) | \( a_{14}^3 \) |
| \( b_{12}^1, b_{14}^3 \) | \( b_{23}^1, b_{24}^3 \) | \( b_{13}^1 \) | \( b_{14}^3 \) |
| \( a_{13}^1 + b_{13}^1 \) | \( a_{12}^1 + b_{12}^1 \) | \( a_{23}^2 + b_{23}^1 \) | \( a_{24}^2 + b_{24}^1 \) |
| \( a_{13}^1 + b_{13}^1 \) | \( a_{12}^1 + b_{12}^1 \) | \( a_{23}^2 + b_{23}^1 \) | \( a_{24}^2 + b_{24}^1 \) |

TABLE 8. Queries to download message \( A \)

The number of useful downloaded symbols is \( L = 12 \) and the total number of downloaded symbols is \( D = 18 \). Hence the scheme has download cost \( \frac{D}{L} = \frac{3}{2} \) that attains the lower bound (1).

From the above example, we see that the main difference is that we divided the servers into two disjoint groups and enforced symmetry across the servers within
each group. The first group has \( d = \frac{N}{\gcd(N,t)} \) servers and the remaining \( N - d \) servers are in the other group. The above is necessary because we need to download a single symbol of each desired sub-message in the first step. Note that such a requirement cannot be satisfied if \( N \not| (\binom{N}{t}) \). When \( \gcd(N,t) = 1 \), this PIR scheme is a special case of \( N \not| (\binom{N}{t}) \).

The next lemma is employed later and the proof of the lemma can be found in the Appendix.

**Lemma 4.1.** For positive integer \( N, t \in [N - 1] \), let \( d = \frac{N}{\gcd(N,t)} \cdot a = \frac{\binom{N}{t} - (N-d) t}{d} \), then \( a \) is a positive integer.

The general PIR scheme is almost identical to the above. We will give a brief description in the following. In Step 1, we download a single symbol of \( t \) different sub-messages of the desired message from each of \( N - d \) servers respectively, and download \( a = \frac{\binom{N}{t} - (N-d) t}{d} \) symbols of the desired message from the remaining \( d \) servers respectively. Then we perform the same download operation for all the remaining \( (M-1) \) messages. In Step 1, the total number of the downloaded desired messages \( A \) is \( da + t(N - d) \), whereas the total number of downloaded symbols is \( \frac{M}{2} \cdot da + t(N - d) \). In Step 2, we exploit \( \binom{N-1}{t-1} - a \) undesired symbols which have been downloaded from \( d \) servers in Step 1 to construct 2-sum which contains desired symbols. The number of these 2-sum is \( \binom{N-1}{t-1} - t \) for the \( N - d \) servers. Therefore the total number of the downloaded desired messages \( A \) is \( \binom{M-1}{2} \cdot d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t) \), whereas the total number of downloaded symbols is \( \binom{M}{2} \cdot d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t) \). In Step 3, we perform a similar approach to the above. Hence for a fixed subset \( T \subseteq [M] \setminus \{m\} \), the number of the 2-sum is \( d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t) \). Note that the number of subsets \( S \) with size \( t \) is \( \binom{N}{t} \). Therefore, for such a subset \( S \), the number of symbols in side information which can be used to construct the 3-sum with desired symbols is

\[
\frac{d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t)}{\binom{N}{t}}.
\]

Hence the total number of downloaded symbols is

\[
\binom{M}{3} \cdot d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t)
\]

\[
\times \frac{[d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t)]^2}{\binom{N}{t}}
\]

\[
= \binom{M}{3} \frac{d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t)}{\binom{N}{t}}.
\]

In addition, the total number of downloaded symbols from the desired message \( A \) is

\[
\binom{M - 1}{3 - 1} \frac{d(\binom{N-1}{t-1} - a) + (N - d)(\binom{N-1}{t-1} - t)}{\binom{N}{t}}.
\]

We similarly compute till Step \( M \). Next, we get the total number of downloaded symbols and desired symbols in Table 9 through the same process as above.
Let $y = d(N-1) - a + (N-d)(N-1) - t$, $c = y/(\binom{N}{t})$. We calculate the total number of desired and downloaded symbols below.

The total number of downloaded symbols (all servers) is

$$D = \sum_{k=2}^{M} \binom{M}{k} \frac{y^{k-1}}{\binom{N}{k}} + M(da + t(N - d))$$

$$= \binom{N}{t} \left( \frac{y}{\binom{N}{t}} + 1 \right)^{M} - M \frac{y}{\binom{N}{t}} - 1 + M(da + t(N - d))$$

$$= \binom{N}{t} \frac{(c+1)^M - cM - 1}{c} + M(da + t(N - d)).$$

The number of downloaded desired symbols (all servers) is

$$L = \sum_{k=2}^{M} \frac{(M-1)}{k-1} \frac{y^{k-1}}{\binom{N}{k}} + (da + t(N - d))$$

$$= \binom{N}{t} \left( \frac{y}{\binom{N}{t}} - 1 \right)^{M-1} + (da + t(N - d))$$

$$= \binom{N}{t} ((c+1)^{M-1} - 1) + (da + t(N - d)).$$

Since $da + (N - d)t = \binom{N}{t}$, $y = d(\binom{N}{t+1} - a) + (N - d)(\binom{N}{t+1} - t)$, $c = y/(\binom{N}{t+1})$, we can get $c = t - 1$. If $t \in [N]$ such that $N \notdivides \binom{N}{t}$, the download cost $D(\mu)$ of the storage constrained PIR scheme is as follows:

$$\frac{D}{L} = \frac{\binom{N}{t} \frac{(c+1)^M - cM - 1}{c} + M(da + t(N - d))}{\binom{N}{t} ((c+1)^{M-1} - 1) + (da + t(N - d))}$$

$$= \frac{\binom{N}{t} \frac{(c+1)^M - cM - 1}{c} + M \binom{N}{t}}{(c+1)^{M-1} - 1 + \binom{N}{t}}$$

$$= \frac{(c+1)^{M-1} - 1}{c(c+1)^{M-1}} + M$$

| Steps | Tuple | Number of Total symbols (all servers) | Number of Useful symbols (all servers) |
|-------|-------|--------------------------------------|----------------------------------------|
| Step 1 | Single $\binom{N}{t}$ | $da + t(N - d)$ | $da + t(N - d)$ |
| Step 2 | Pair $\binom{M}{2}$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{2}} + (da + t(N - d))$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{2}} + (da + t(N - d))$ |
| Step 3 | Triple $\binom{M}{3}$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{3}} + (da + t(N - d))$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{3}} + (da + t(N - d))$ |
| ... | ... | ... | ... |
| Step k | k-tuple $\binom{M}{k}$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{k}} + (da + t(N - d))$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{k}} + (da + t(N - d))$ |
| ... | ... | ... | ... |
| Step M | M-tuple $\binom{M}{M}$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{M}} + (da + t(N - d))$ | $\frac{d(N-1) - a + (N-d)(N-1) - t}{\binom{N}{M}} + (da + t(N - d))$ |

Table 9. The number of symbols downloaded in each step of the scheme.
$$= \frac{t^M - 1}{(t-1)t^{M-1}}$$
$$= 1 + \frac{1}{t} + \frac{1}{t^2} + \cdots + \frac{1}{t^{M-1}}.$$ Therefore, the achievable scheme is correct and attains the lower bound (1). The proof of privacy is similar to that in the first case. Hence the privacy condition is guaranteed.

5. Conclusion

In this paper, we designed a PIR scheme with a sub-packetization \(\binom{N}{t}t^{M-1}\) from storage constrained servers for any storage size. Our result improves a recent result in [15]. We showed that our proposed PIR scheme attains the optimal download versus storage trade-off for any storage capacity and any \(N, M\). Our scheme is based on a connection between coded caching and PIR schemes from storage constrained servers. Very recently, we have been noticed that a new design of private information retrieval for storage constrained servers was proposed in [17] which provides smaller sub-packetization than the one from ours. The scheme in [17] abandons the idea of using the cache placement of coded caching problem. We should mention that the sub-packetization of our scheme can be further reduced if an appropriate coded caching could be found. One of our future directions is to search for such a coded caching.

6. Appendix

The proof of Lemma 4.1

Proof. To prove Lemma 4.1, it suffices to prove that \(d \mid \binom{N}{t} - (N-d)t\). Due to \(d \mid (N-d)\), we only need to prove \(d \mid \binom{N}{t}\). Let \(\gcd(N,t) = g\), \(N = gd\), and \(t = hg\). Then we have \(\gcd(d,h) = 1\) and \(d = N/\gcd(N,t)\). Notice that

$$\binom{N}{t} = \binom{gd}{hg} = \frac{gd}{hg} \frac{gd-1}{hg-1} = \frac{d}{h} \binom{gd-1}{hg-1}.$$ It follows from \(\gcd(d,h) = 1\) that

$$h \mid \binom{gd-1}{hg-1},$$

which further implies that \(d \mid \binom{N}{t}\). Therefore \(a\) is an integer.

Next we will prove that \(a\) is a positive integer. It is sufficient to prove \(\binom{N}{t} \geq (N-d)t\). It is easily verified the inequality holds when \(N = 2\). Now we consider the case \(N \geq 3\) which can be divided into following cases.

**Case 1:** \(3 \leq t \leq N - 3\). In this case, we have

$$\binom{N}{t} \geq \binom{N}{3} = \frac{N(N-1)(N-2)}{6}.$$ Since \(N-1 \geq N-d\) and \(\frac{N(N-2)}{6} \geq N-3 \geq t\), we have

$$\binom{N}{t} \geq \frac{N(N-1)(N-2)}{6} \geq (N-d)t.$$ Hence, \(a\) is positive in this case.

**Case 2:** \(t = 1\) or \(t = N-1\). In this case, \(\binom{N}{t} - (N-d)t = N > 0\) due to \(d = N\).
Case 3: \( t = 2 \). In this case, we have
\[
\binom{N}{t} - (N - d)t = \begin{cases} 
\frac{N(N-1)}{2} & \text{if } N \text{ is odd}, \\
\frac{N(N-3)}{2} & \text{if } N \text{ is even}.
\end{cases}
\]

Case 4: \( t = N - 2 \). In this case, one has
\[
\binom{N}{t} - (N - d)t = \begin{cases} 
\frac{N(N-1)}{2} & \text{if } N \text{ is odd}, \\
\frac{N}{2} & \text{if } N \text{ is even}.
\end{cases}
\]

Summarizing the cases above completes the proof.

REFERENCES

[1] M. A. Attia, D. Kumar and R. Tandon, The capacity of uncoded storage constrained PIR, *2018 IEEE International Symposium on Information Theory*, (2018), 1959–1963.

[2] K. Banawan and S. Ulukus, Multi-message private information retrieval: Capacity results and near-optimal schemes, *IEEE Trans. Inform. Theory*, 64 (2018), 6842–6862.

[3] K. Banawan and S. Ulukus, The capacity of private information retrieval from byzantine and colluding databases, *IEEE Transactions on Information Theory*, 65 (2018), 1206–1219, arXiv:1706.01442.

[4] K. Banawan and S. Ulukus, The capacity of private information retrieval from coded databases, *IEEE Transactions Information Theory*, 64 (2018), 1945–1956.

[5] Z. Chen, Z. Y. Wang and S. Jafar, The capacity of private information retrieval with private side information, (2017), arXiv:1709.03022.

[6] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, Private information retrieval, *36th Annual Symposium on Foundations of Computer Science*, (1995), 41–50.

[7] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti and D. A. Karpuk, Private information retrieval from coded databases with colluding servers, *SIAM Journal on Applied Algebra and Geometry*, 1 (2017), 647–664.

[8] S. Kadhe, B. Garcia, A. Heidarzadeh, S. El Rouayheb and A. Sprintson, Private information retrieval with side information, (2017), arXiv:1709.00112.

[9] H. Sun and S. A. Jafar, The capacity of private information retrieval, *IEEE Transactions Information Theory*, 63 (2017), 4075–4088.

[10] H. Sun and S. A. Jafar, The capacity of symmetric private information retrieval, *2016 IEEE Globecom Workshops, Washington, DC, USA*, (2016), 1–5.

[11] H. Sun and S. A. Jafar, The capacity of private information retrieval with colluding databases, *IEEE Trans. Inform. Theory*, 64 (2018), 2361–2370.

[12] 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 Transactions Information Theory*, 64 (2018), 1000–1022.

[13] H. Sun and S. A. Jafar, Optimal download cost of private information retrieval for arbitrary message length, *IEEE Transactions Information Forensics and Security*, 12 (2017), 2920–2932.

[14] R. Tajeddine, O. W. Gnilke and S. El Rouayheb, Private information retrieval from MDS coded data in distributed storage systems, *IEEE Trans. Inform. Theory*, 64 (2018), 7081–7093.

[15] R. Tandon, M. Abdul-Wahid, F. Al moualem and D. Kumar, PIR from storage constrained databases - coded caching meets PIR, *2018 IEEE International Conference on Communications*, (2018), 1–7.

[16] Q. W. Wang and M. Skoglund, Symmetric private information retrieval for MDS coded distributed storage, *IEEE International Conference on Communications*, (2017), 1–6.

[17] N. Woo lsey, R.-R. Chen and M. Y. Ji, A new design of private information retrieval for storage constrained databases, (2019), arXiv:1901.07490.

[18] J. K. Xu and Z. F. Zhang, On sub-packetization of capacity-achieving PIR schemes for MDS coded databases, (2017), arXiv:1712.02466.

[19] Y. W. Zhang and G. N. Ge, A general private information retrieval scheme for MDS coded databases with colluding servers, (2017), arXiv:1704.06785.

[20] Y. W. Zhang and G. N. Ge, Multi-file private information retrieval from MDS coded databases with colluding servers, (2017), arXiv:1705.03186.
[21] Z. F. Zhang and J. K. Xu, Capacity-achieving PIR schemes with optimal sub-packetization, (2017), arXiv:1710.11370.

Received April 2019; 1st revision July 2019; final revision February 2020.

E-mail address: zwq@my.swjtu.edu.cn
E-mail address: zzc@swjtu.edu.cn
E-mail address: udaya@unimelb.edu.au
E-mail address: vladimir.sidorenko@tum.de
