

An Optimal Iterative Placement Algorithm for PIR from Heterogeneous Storage-Constrained Databases

Nicholas Woolsey, Rong-Rong Chen, and Mingyue Ji

Department of Electrical and Computer Engineering, University of Utah

Salt Lake City, UT, USA

Email: {nicholas.woolsey@utah.edu, rchen@ece.utah.edu, mingyue.ji@utah.edu}

Abstract—We propose a capacity-achieving scheme for private information retrieval (PIR) from databases (DBs) with heterogeneous storage constraints. In the PIR setting, a user queries a set of DBs to privately download a message, where privacy implies that no one DB can infer which message the user desires. Our PIR scheme uses an uncoded storage placement and we derive sufficient conditions to meet capacity in this design architecture. We translate the storage placement design to a “filling problem” where messages are partitioned into sub-messages and stored at subsets of DBs. We prove a set of necessary and sufficient conditions for the existence of the filling problem solution and design an iterative algorithm to find a filling problem solution. Our proposed algorithm requires at most a number of iterations equal to the number of DBs. Furthermore, we significantly reduce the number of sub-messages compared to the state-of-the-art PIR scheme, as our proposed PIR scheme requires that each message is split into a polynomial number of sub-messages with respect to the number of DBs.

I. INTRODUCTION

The private information retrieval (PIR) problem originally introduced by Chor *et al.* [1], [2] has been recently studied under an information theoretic point of view [3]. In the PIR problem, a user privately downloads one of K messages from a set of N non-colluding databases (DBs). Moreover, privacy implies that no DB can infer which of the K messages the user is downloading. To achieve privacy the user generates strategic queries to the databases such that sub-messages from all K messages are requested. To gauge the performance of the PIR scheme, the rate, R , is defined as the ratio of desired bits (size of each message), L , to the total number of downloaded bits, D . In the traditional setting of full storage PIR (FS-PIR), each DB has access to all K messages and the capacity, or maximum achievable rate, of PIR is $(1 + \frac{1}{N} + \frac{1}{N^2} + \dots + \frac{1}{N^{K-1}})^{-1}$ [3]. Multiple achievable schemes have been developed which achieve FS-PIR capacity by exploiting downloaded undesired sub-messages for coding opportunities [3]–[5].

More recently, the problem of homogeneous storage constrained PIR (SC-PIR) was proposed such that each DB can only store μKL symbols where $\frac{1}{N} \leq \mu \leq 1$ [6]. The capacity of homogeneous SC-PIR was shown to be $(1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}})^{-1}$ where $t = \mu N$ and t is an integer [7], [8]. Different from FS-PIR, there is an additional

design aspect to SC-PIR such that the contents storage placement must be strategically designed. For example, the original homogeneous SC-PIR scheme met capacity [6] by using the storage placement scheme of the coded caching problem [9]. In addition, two other storage placement designs for SC-PIR which meet capacity were proposed in [10]. Ultimately, a set of sufficient conditions to achieve homogeneous SC-PIR capacity were derived in [10]. The conditions are: 1) a capacity achieving FS-PIR scheme should be used for query generation and 2) sub-message sets should always be stored at t DBs (or $\lfloor t \rfloor$ and $\lceil t \rceil$ DBs for non-integer t).

The SC-PIR problem was further generalized in [11] to study the case where DBs have varying storage requirements. In this setting, the storage capacity of the N databases are defined by a vector $\mu \in \mathbb{R}_+^N$, where $\mathbb{R}_+^N$ denotes the set of non-negative real-valued vectors in N -dimensional space, such that DB_n can only store up to $\mu[n]KL$ symbols where $0 \leq \mu[n] \leq 1$. Interestingly, using an information theoretic proof, the authors showed that the capacity of heterogeneous SC-PIR is the same as homogeneous SC-PIR where $t = \sum_{n=1}^N \mu[n]$. Furthermore, the authors translated the storage placement problem into a linear program (LP). A relaxed version of the LP demonstrated that to achieve capacity, sub-message sets should be stored at t DBs (or $\lfloor t \rfloor$ and $\lceil t \rceil$ DBs for non-integer t). This is similar to the conditions of [10] for the homogeneous case. The authors of [11] also showed the existence of a solution to the LP for general N . However, the LP requires an exponential number of variables with respect to N and an explicit solution was only derived for $N = 3$ DBs.

In this paper, we aim to find a solution to the heterogeneous SC-PIR storage placement problem for general N such that capacity can be achieved. To do this, we show that the storage placement problem can be translated to a *filling problem (FP)*. Then, we approach the FP by proposing an iterative algorithm which places a sub-message set at t DBs in each iteration when t is an integer. More specifically are contributions are:

- 1) We expand on our results in [10] to demonstrate a fundamental connection between the placement problem of heterogeneous SC-PIR and a filling problem (FP).
- 2) We derive a set of necessary and sufficient conditions that guarantees the existence of a FP solution and show that the heterogeneous SC-PIR problem inherently meets this condition.

This work was partially funded by National Science Foundation grants CCF-1817154 and SpecEES-1824558.

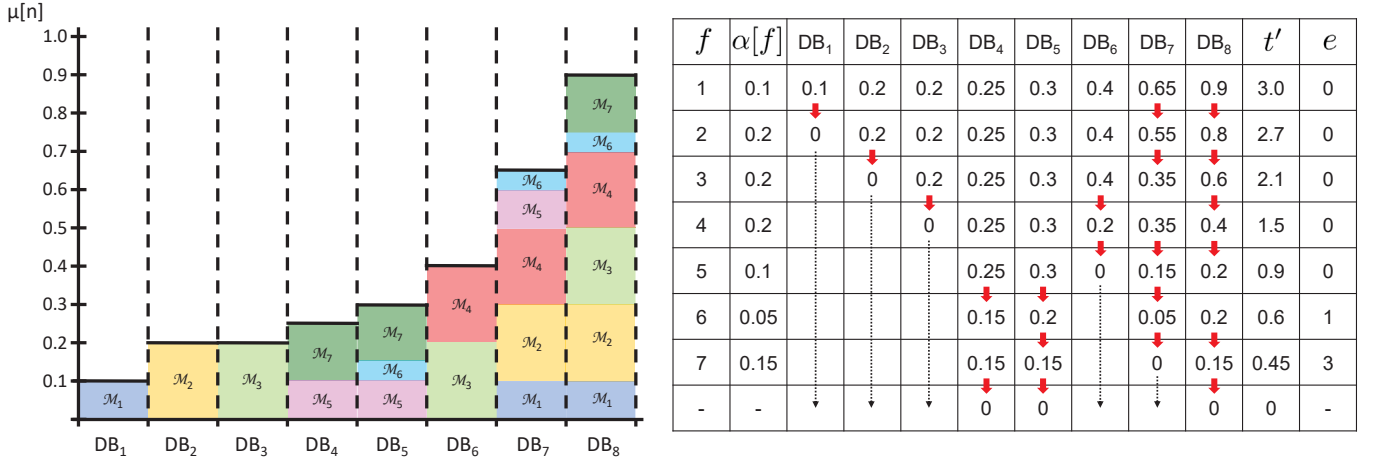


Fig. 1. A solution to the filling problem using Algorithm 1 when $t = 3$ and $\mu = [0.1, 0.2, 0.2, 0.25, 0.3, 0.4, 0.65, 0.9]$. (left) A bar graph depicting the storage requirements of the DBs and the storage placement solution. (right) A table representing the remaining storage of the DBs for each iteration. The red arrows highlight which DBs are assigned a sub-message subset in each iteration.

- 3) We propose an iterative storage placement algorithm which solves the heterogeneous SC-PIR placement problem for general N and integer t .
- 4) We demonstrate that the proposed algorithm converges within N iterations. Therefore, the storage placement design for heterogeneous SC-PIR requires at most N sub-messages per message.

Notation Convention: We use $|\cdot|$ to represent the cardinality of a set or the length of a vector. Also $[n] := [1, 2, \dots, n]$ and $[n_1 : n_2] = [n_1, n_1 + 1, \dots, n_2]$. A bold symbol such as $\mathbf{a}$ indicates a vector and $\mathbf{a}[i]$ denotes the i -th element of $\mathbf{a}$.

II. PROBLEM FORMULATION

There are K independent messages, $W_1, \dots, W_K$, each of size L symbols. The messages are collectively stored among N non-colluding DBs, labeled as $\text{DB}_1, \dots, \text{DB}_N$. The storage capacity of the DBs is defined by a vector $\mu \in \mathbb{R}_+^N$ where, for all $n \in [N]$, DB_n has the storage capacity of $\mu[n]KL$ symbols, where $0 < \mu[n] \leq 1$. Furthermore, for all $n \in [N]$, define Z_n as the storage contents of DB_n . Also, we define $t \triangleq \sum_{n=1}^N \mu[n]$ as the number of times each symbol of the messages is stored among the DBs. To design an achievable PIR scheme we assume $t \geq 1$ so that each symbol of the messages can be stored at least one DB. A user makes a request W_k and sends a query $Q_n^{[k]}$, which is independent of the messages, to each DB $n \in [N]$ which then sends an answer $A_n^{[k]}$ such that

$$H(A_n^{[k]} | Z_n, Q_n^{[k]}) = 0, \quad \forall k \in [K]. \quad (1)$$

Furthermore, given the answers from all the databases, the user must be able to recover the requested message. Therefore,

$$H(W_k | A_1^{[k]}, \dots, A_n^{[k]}, Q_1^{[k]}, \dots, Q_n^{[k]}) = 0. \quad (2)$$

The user generates queries in a manner to ensure privacy such that no DB can infer which message the user desires, i.e.,

$$I(k; Q_n^{[k]}, A_n^{[k]}, W_1, \dots, W_K, Z_1, \dots, Z_N) = 0. \quad (3)$$

Let D be the total number of downloaded bits. Given μ , we say that a pair (D, L) is achievable if there exists a SC-PIR scheme with rate $R = L/D$ that satisfies (1)-(3). The SC-PIR capacity is defined as

$$C^*(\mu) = \max\{R : (D, L) \text{ is achievable}\}. \quad (4)$$

III. AN EXAMPLE

In this section, we provide a motivating example to demonstrate how an iterative storage placement scheme can achieve the heterogeneous SC-PIR capacity. Let $N = 8$ and define the storage requirements of the DBs as

$$\mu = [0.1, 0.2, 0.2, 0.25, 0.3, 0.4, 0.65, 0.9]. \quad (5)$$

For example, by this notation, DB_6 has a storage capacity of $\frac{4}{10}KL$ symbols. By summing the elements of μ , we obtain $t = 3$.

To define the storage placement, the K messages are divided into F disjoint sub-message sets, $\mathcal{M}_1, \dots, \mathcal{M}_F$, such that each sub-message set contains a sub-message of equal size from each of the K messages. Then, each sub-message set, $\mathcal{M}_f$, is stored at some subset of DBs $\mathcal{N}_f \subseteq [N]$. We show in Section IV-B that capacity can be achieved if each sub-message set, $\mathcal{M}_f$, is stored at exactly $t = 3$ DBs. This translates to a “filling problem” (FP) where our goal is to iteratively fill the storage of the DBs such that each iteration fills some available storage in exactly 3 DBs.

We propose an iterative scheme where each iteration aims to fill the DB with the least remaining storage. In the first iteration, we define a sub-message set, $\mathcal{M}_1$, which contains $\mu[1]L = \frac{1}{10}L$ arbitrary symbols from each of the K messages and assign $\mathcal{M}_1$ to the DB subset $\mathcal{N}_1 = \{1, 7, 8\}$. Notice that $\mathcal{M}_1$ contains $\mu[1]KL$ symbols and there is no remaining available storage at DB_1 after this iteration. After this iteration, the question arises whether or not this iteration yields a valid placement (for future iterations). Later in Section V we define a set of necessary and sufficient conditions to determine whether a particular iteration is valid.

Next, we aim to fill the storage contents of DB_2 and let $\mathcal{M}_2$ contain $\frac{1}{5}L$ arbitrarily unpicked symbols (i.e., symbols are not in $\mathcal{M}_1$) from each of the K messages. Then, $\mathcal{M}_2$ is stored at the DB subset $\mathcal{N}_2 = \{2, 7, 8\}$. In general, the idea to determine $\mathcal{N}_f$ is to choose the DB with the smallest remaining storage and the $t - 1$ DBs with the most remaining storage. This process is continued until the 5th iteration, where filling DB_7 (which has the smallest remaining storage) would cause for an invalid filling solution. In Section VI, we discuss how to handle this by partially filling the DB with the smallest remaining storage.

The final results of the storage placement by our newly proposed algorithm are shown in Fig. 1. In total, there are $F = 7$ sub-message sets, each of which contains a sub-message from each of the K messages, and is stored at exactly 3 DBs. A vector $\alpha \in \Delta_F$ defines the fraction of the library that is stored (or filled) in each iteration. For example, $\alpha[1] = 0.1$ and $\alpha[2] = 0.2$ correspond to the first two iterations described above. All of the values of α are shown in the table of Fig. 1. The corresponding DBs that store a sub-message subset in a particular iteration are highlighted by the red arrows in the table of Fig. 1.

Given that a user desires to privately download W_θ for some $\theta \in [K]$, the user will privately download the sub-message of W_θ stored at DBs of $\mathcal{N}_f$ using one of the capacity achieving FS-PIR in [3]–[5] for all $f \in [F]$. The rate of each download and the overall rate is equal to the rate of a capacity-achieving FS-PIR scheme that is privately downloading from $t = 3$ DBs. In this case, the rate is $R = \left(1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}}\right)^{-1}$ which is the heterogeneous SC-PIR capacity as shown in [11].

Fig. 1 contains two additional parameters, t' and e , which are discussed in greater detail later in this paper. Moreover, t' is the sum of the cumulative normalized remaining storage of all DBs and e is the number of DBs with remaining storage equal to $\frac{t'KL}{t}$ symbols. These parameters are significant when deriving necessary and sufficient conditions for a valid placement and proving the convergence rate of our proposed placement algorithm.

IV. TRANSLATING HETEROGENEOUS SC-PIR TO A FILLING PROBLEM

In this section, we translate the heterogeneous PIR placement problem into a simpler filling problem. To do this we adopt the SC-PIR design architecture from our previous work [10] and adapt it to allow for the more general heterogeneous SC-PIR case, similar to [11]. Then, we expand our results from [10] to demonstrate the sufficient conditions to achieve capacity for heterogeneous SC-PIR. Ultimately, we derive the same conditions as in [11], but using a different approach. This section motivates the rest of this paper which aims to find a solution to the heterogeneous SC-PIR placement problem by solving an equivalent filling problem.

A. Design Architecture

Placement: Define a vector $\alpha \in \Delta_F$, where $F \in \mathbb{Z}^+$ and $\alpha[f], \forall f \in [F]$ is rational number such that $\alpha[f]L \in \mathbb{Z}^+$.

For all $k \in [K]$, we divide message W_k into F disjoint sub-messages $W_k = \{W_{k,1}, \dots, W_{k,F}\}$ such that for all $f \in [F]$, $|W_{k,f}| = \alpha[f]L$ symbols. For all $f \in [F]$, let

$$\mathcal{M}_f \triangleq \bigcup_{k \in [K]} W_{k,f}, \quad (6)$$

and $\mathcal{N}_f \subseteq [N]$ be a non-empty subset of DBs which have the sub-messages in $\mathcal{M}_f$ locally available to them. The storage contents of database $n \in [N]$ is

$$Z_n = \{\mathcal{M}_f : f \in [F], n \in \mathcal{N}_f\}, \quad (7)$$

where we have the requirement that for all $n \in [N]$,

$$\sum_{\{f: f \in [F], n \in \mathcal{N}_f\}} \alpha[f] \leq \mu[n]. \quad (8)$$

Delivery: Given that a user requests file W_θ for some $\theta \in [K]$, we do the following. For all $f \in [F]$, using a FS-PIR scheme, the user generates a query to privately download $W_{\theta,f}$ from the DBs in $\mathcal{N}_f$. In other words, a SC-PIR scheme can be found by applying a FS-PIR scheme to each set of databases $\mathcal{N}_f$. Changing the choice of the FS-PIR scheme or the definitions of $\mathcal{N}_f$ will result in new SC-PIR schemes.

B. Sufficient Conditions to Achieve Heterogeneous SC-PIR Capacity

In our previous work, we outlined a set of sufficient conditions to achieve capacity for homogeneous SC-PIR [10]. Surprisingly, as the capacity of the more general heterogeneous SC-PIR is the same as homogeneous SC-PIR [11], the sufficient conditions of [10] directly apply to the heterogeneous case. In short, there are two conditions given in [10]. The first states that capacity achieving FS-PIR schemes are used to generate queries to the DBs. The second condition relates to the storage placement and states that

- if $t \in \mathbb{Z}^+$, then $|\mathcal{N}_f| = t$ for all $f \in [F]$
- otherwise,

$$\sum_{f: |\mathcal{N}_f| = \lfloor t \rfloor} \alpha[f] = \lceil t \rceil - t \quad (9)$$

and

$$\sum_{f: |\mathcal{N}_f| = \lceil t \rceil} \alpha[f] = t - \lfloor t \rfloor. \quad (10)$$

For example, if t is an integer and the sufficient conditions are met, using Theorem 1 of [10], the resulting heterogeneous SC-PIR rate is

$$R = \left(\frac{\alpha[1]}{R_{\text{FS}}(t)} + \frac{\alpha[2]}{R_{\text{FS}}(t)} + \dots + \frac{\alpha[F]}{R_{\text{FS}}(t)} \right)^{-1} = R_{\text{FS}}(t) \quad (11)$$

where $R_{\text{FS}}(t)$ is the capacity of FS-PIR when downloading from t DBs and was shown to be the capacity of heterogeneous SC-PIR in [11]. Alternatively, if t is not an integer and the sufficient conditions are met, using Theorem 1 of [10], it can be shown that the heterogeneous SC-PIR rate¹ is

$$R = \left(\frac{t - \lfloor t \rfloor}{R_{\text{FS}}(\lceil t \rceil)} + \frac{\lceil t \rceil - t}{R_{\text{FS}}(\lfloor t \rfloor)} \right)^{-1} \quad (12)$$

¹See Theorem 4 in [10] for more details.

which is the heterogeneous SC-PIR capacity as shown in [11].

The first condition states that any of the previously designed FS-PIR schemes [3]–[5] can be directly applied to privately download the F desired sub-messages. The second condition boils down to a filling problem as explained next.

C. The Filling Problem

The $(\mathbf{m}, \tau)$ -Filling Problem (FP) is defined as follows: Given a vector $\mathbf{m} \in \mathbb{R}_+^N$, find a τ -fill. Define a basis $\mathcal{B}$, containing the set of all $\{0, 1\}$ -vectors of length N , each of which consists of exactly τ 1s. Finding a τ -fill is equivalent to finding an $\alpha_{\mathbf{b}} \in \mathbb{R}_+$ for all $\mathbf{b} \in \mathcal{B}$ such that

$$\sum_{\mathbf{b} \in \mathcal{B}} \alpha_{\mathbf{b}} \mathbf{b} = \mathbf{m}. \quad (13)$$

For the heterogeneous SC-PIR problem, when $t \in \mathbb{Z}^+$ the capacity achieving placement solution is equivalent to the $(\boldsymbol{\mu}, t)$ -FP. Throughout the rest of this paper, we focus on finding a solution to the $(\boldsymbol{\mu}, t)$ -FP.²

V. EXISTENCE OF THE $(\mathbf{m}, \tau)$ -FP SOLUTION

In this section, we aim to find a set of necessary and sufficient conditions such that a solution to the $(\mathbf{m}, \tau)$ -FP exists. Given any $\mathbf{m} \in \mathbb{R}_+^N$ and $\tau \in \mathbb{Z}^+$, the existence of a $(\mathbf{m}, \tau)$ -FP solution is not guaranteed. For example, if $\mathbf{m} = [0.3, 0.3, 0.7]$ and $\tau = 2$, then a $(\mathbf{m}, \tau)$ -FP solution does not exist since $m[1] + m[2] < m[3]$. This is because that after some previous sub-message placement, it is impossible to fill the remaining storage of two DBs at a time and completely fill DB₃. The following theorem states the necessary and sufficient conditions for a $(\mathbf{m}, \tau)$ -FP solution to exist.

Theorem 1: Given $\mathbf{m} \in \mathbb{R}_+^N$ and $\tau \in \mathbb{Z}^+$ an $(\mathbf{m}, \tau)$ -FP solution exists if and only if

$$m[n] \leq \frac{\sum_{i=1}^N m[i]}{\tau} \quad (14)$$

for all $n \in [N]$.

Proof: Theorem 1 is proved in [12]. ■

This result and the proof of Theorem 1 have two important implications for heterogeneous SC-PIR. First, for any given $\boldsymbol{\mu}$, if t is an integer then

$$\mu[n] \leq 1 = \frac{\sum_{i=1}^N \mu[i]}{t} \quad (15)$$

for all $n \in [N]$. Clearly a $(\boldsymbol{\mu}, t)$ -FP solution exists and therefore a heterogeneous SC-PIR scheme exists which can achieve capacity.³ Second, while it is not clear how to determine the storage placement for a given $\boldsymbol{\mu}$, Theorem 1 suggests that there is an iterative process which can define the storage placement. In other words, if a sub-message set is assigned to a set of t DBs, then we can precisely determine if the remaining storage among all DBs allows a FP solution. In this way, an iterative

scheme can be defined that is guaranteed to converge to a final storage placement solution.

VI. ITERATIVE STORAGE PLACEMENT DESIGN

Motivated by Theorem 1, we develop an iterative storage placement scheme where in each iteration a sub-message of each of the K messages is stored in a set of t DBs. We design an iterative algorithm such that each iteration aims to fill the storage of the DB with the smallest remaining, non-zero storage to make the remaining FP simpler. To do this, we store a sub-message set at a set of t DBs including the DB with the smallest remaining, non-zero storage and the $t - 1$ DBs with the largest remaining storage. The scheme is rigorously outlined in Algorithm 1 and summarized as follows.

Let N' be the number of DBs with non-zero remaining storage and $\mathbf{m} \in \mathbb{R}_+^N$ track the remaining storage of each DB normalized by KL . For ease of notation and WLOG we assume $m[1] \leq m[2] \leq \dots \leq m[N]$ for any given iteration.⁴

If $N' \geq t + 1$, do the following. Let the DB subset, $\mathcal{N}$, of size t include the DB with the smallest remaining, non-zero storage and the $t - 1$ DBs with the largest remaining storage. In other words,

$$\mathcal{N} = \{N - N' + 1, N - t + 2, \dots, N\} \quad (16)$$

where $m[N - N' + 1]$ is the storage remaining at the DB with the smallest remaining, non-zero storage. A sub-message set is defined to be stored at the DBs of $\mathcal{N}$. Ideally, the number of symbols in the sub-message set is size $m[N - N' + 1]KL$ symbols, however, it is possible that such a sub-message assignment prevents a FP solution for the remaining storage among the DBs (i.e., violate (14)). Therefore, assign

$$\alpha = \min \left(\frac{t'}{t} - m[N - t + 1], m[N - N' + 1] \right) \quad (17)$$

where $t' = \sum_{n=1}^N m[n]$. Following the method presented in Section IV-A, define a sub-message set, $\mathcal{M}$, containing αKL symbols which have not been stored in a previous iteration and store $\mathcal{M}$ at the DBs of $\mathcal{N}$. Then, adjust $\mathbf{m}$ accordingly to reflect the remaining storage at each DB.

There is only one exception to this process which is the case where there are only $N' = t$ DBs with non-zero remaining storage. In this case, all the of remaining storage of these t DBs are equal which can be shown using Theorem 1. Furthermore, let $\alpha = m[N - N' + 1]$ and a sub-message set of size αKL symbols is stored at these t DBs.

Note that, Algorithm 1 only operates when $N' \geq t$, because it is impossible for $N' < t$ since to have a valid FP solution, there must be at least t DBs with non-zero remaining storage. Also, Algorithm 1 can only operate when t is an integer. In [12] we demonstrate how Algorithm 1 can be applied to the case where t is not an integer.

⁴For correctness, in Algorithm 1, $\mathbf{m}$ is not assumed to be in increasing order and the indices corresponding to the order are used as necessary.

²In [12], we expand our results for non-integer t .

³The existence of a solution for a capacity achieving storage placement for heterogenous SC-PIR was also shown in the proof of Lemma 5 of [11]. However, the proof assumes non-integer t and uses different methods according to our understanding.

Algorithm 1 Heterogeneous SC-PIR Storage Placement

Input: μ, t, L and $W_1, \dots, W_K$

```

1:  $\mathbf{m} \leftarrow \mu$ 
2:  $F \leftarrow 0$ 
3: while  $\mathbf{m} > \mathbf{0}$  do
4:    $F \leftarrow F + 1$ 
5:    $t' \leftarrow \sum_{n=1}^N m[n]$ 
6:    $\ell \leftarrow$  indices of non-zero elements of  $\mathbf{m}$  from smallest
   to largest
7:    $N' \leftarrow$  number of non-zero elements in  $\mathbf{m}$ 
8:    $\mathcal{N}_F \leftarrow \{\ell[1], \ell[N' - t + 2], \dots, \ell[N']\}$ 
9:   if  $N' \geq t + 1$  then
10:     $\alpha_F \leftarrow \min\left(\frac{t'}{t} - m[\ell[N' - t + 1]], m[\ell[1]]\right)$ 
11:   else
12:     $\alpha_F \leftarrow m[\ell[1]]$ 
13:   end if
14:   for  $n \in \mathcal{N}_F$  do
15:      $m[n] \leftarrow m[n] - \alpha_F$ 
16:   end for
17: end while
18: for  $k = 1, \dots, K$  do
19:   Partition  $W_k$  into  $F$  disjoint sub-messages:
    $W_{k,1}, \dots, W_{k,F}$  of size  $\alpha_1 L, \dots, \alpha_F L$  symbols
   respectively
20:   for  $f = 1, \dots, F$  do
21:     Store  $W_{k,f}$  at the DBs of  $\mathcal{N}_f$ 
22:   end for
23: end for
```

A. Correctness

In the following, we demonstrate that each iteration fills a non-zero, positive amount of storage. WLOG we assume $m[1] \leq m[2] \leq \dots \leq m[N]$. Furthermore, assuming that $m[N - N' + 1] > 0$ and a $(\mathbf{m}, t)$ -FP solution exists such that $m[N - t + 1] \leq \frac{\sum_{i=1}^N m[i]}{t} = \frac{t'}{t}$, then observing (17), we can see that $\alpha \geq 0$. Moreover, $\alpha = 0$ if and only if

$$m[N - t + 1] = \frac{t'}{t} = \frac{\sum_{i=1}^N m[i]}{t} \quad (18)$$

and in this case we find for all $n \in [N - t + 1 : N]$ that

$$\frac{\sum_{i=1}^N m[i]}{t} = m[N - t + 1] \leq m[n] \leq \frac{\sum_{i=1}^N m[i]}{t}. \quad (19)$$

and $m[n] = m[N - t + 1]$. This means that $N' = t$ and each of the t DBs has the same amount of remaining storage. In this case, $\alpha = m[N - t + 1]$ since $N' = t$.

Next, we demonstrate that after an iteration the remaining storage among the DBs is such that a FP solution exists. Let

$$\mathbf{m}' = \mathbf{m} - \alpha \cdot \underbrace{[0, \dots, 0]_{N-N'}}_{N-N'} \underbrace{[1, 0, \dots, 0]_{N'-t}}_{N'-t} \underbrace{[1, \dots, 1]_{t-1}}_{t-1} \quad (20)$$

represent the remaining storage after a particular iteration. Note that, the elements of $\mathbf{m}'$ are not necessarily in order. After an iteration, the largest remaining storage at any node is

either $m'[N] = m[N] - \alpha$ or $m'[N - t + 1] = m[N - t + 1]$. Assuming a $(\mathbf{m}, t)$ -FP solution exist, then

$$m'[N] = m[N] - \alpha \leq \frac{\sum_{i=1}^N m[i]}{t} - \alpha = \frac{\sum_{i=1}^N m'[i]}{t}. \quad (21)$$

Also, by (17), $\alpha \leq \frac{\sum_{i=1}^N m[i]}{t} - m[N - t + 1]$ and $m'[N - t + 1] = m[N - t + 1]$, then

$$m'[N - t + 1] \leq \frac{\sum_{i=1}^N m[i]}{t} - \alpha = \frac{\sum_{i=1}^N m'[i]}{t}. \quad (22)$$

Furthermore, $\alpha \leq m[N - N' + 1] \leq m[n]$ and $m'[n] \geq m[n] - \alpha \geq 0$ for all $n \in [N - N' + 1 : N]$. Finally $m'[n] = m[n] = 0$ for all $n \in [1 : N - N']$. Since $0 \leq m'[n] \leq \frac{\sum_{i=1}^N m'[i]}{t}$ for all $n \in [N]$, by using Theorem 1, a $(\mathbf{m}', t)$ -FP solution exists.

VII. CONVERGENCE

Since in each iteration we fill a positive amount of remaining storage without violating the existence conditions for a FP solution, Algorithm 1 converges to a final solution where all DBs are completely filled. The question remains as to how many iterations are required for convergence. Moreover, the number of iterations is equal to the number of sub-messages per message, F , required for the storage placement. Surprisingly, we find that at most N iterations are required to fill all the DBs. The result is summarized in the following theorem.

Theorem 2: Algorithm 1 requires at most N iterations to completely fill the DBs.

Proof: Throughout this proof, let $\mathbf{m} \in \mathbb{R}_+^N$ be the remaining storage of each DB at a given iteration normalized by KL and WLOG $m[1] \leq m[2] \leq \dots \leq m[N]$. Define t' as the cumulative remaining normalized storage among the DBs

$$t' = \sum_{n=N-N'+1}^N m[n] \quad (23)$$

where N' is the number of DBs with non-zero remaining storage. We observe the iterations of Algorithm 1 and label the outcome of each iteration as either a *complete fill* (CF) or *partial fill* (PF) defined below.

Definition 1: A *complete fill* (CF) refers to an iteration where the remaining storage at the DB with the smallest remaining non-zero storage is completely filled.

Definition 2: A *partial fill* (PF) refers to an iteration where the remaining storage at the DB with the smallest remaining non-zero storage is *not* completely filled.

To obtain an upper bound on the number of iterations to fill the DBs, we count the maximum number of possible PFs and CFs. Let e count the number of DBs with remaining normalized storage equal to $\frac{t'}{t}$ such that

$$e = \sum_{n=1}^N \mathbb{1}\left(m[n] = \frac{t'}{t}\right) \quad (24)$$

where $\mathbb{1}(\cdot)$ is the indicator function. The following lemma discusses the sufficient condition which guarantees a CF for a given iteration.

Lemma 1: If a given iteration satisfies $e = t - 1$ and $N' \geq t + 1$, then this iteration must be a CF, and N' will be reduced by at least 1 after that iteration.

Proof: Given that $e = t - 1$ and by (23), we obtain

$$m[N - N' + 1] + \dots + m[N - t + 1] + (t - 1)\frac{t'}{t} = t'. \quad (25)$$

Therefore,

$$m[N - N' + 1] + m[N - t + 1] \leq \frac{t'}{t}, \quad (26)$$

and

$$m[N - N' + 1] \leq \frac{t'}{t} - m[N - t + 1]. \quad (27)$$

By (17), during this iteration, αKL symbols are stored at $DB_{N-N'+1}$ where $\alpha = m[N - N' + 1]$. This completes the proof of Lemma 1. ■

Lemma 2: If a given iteration satisfies $e \leq t - 1$ and $N' \geq t + 1$, then e will not decrease after that iteration. Moreover, if the iteration is a PF then e will be increased by at least 1 after that iteration.

Proof: We prove Lemma 2 as follows. The e DBs with normalized remaining storage equal to $\frac{t'}{t}$ are included in the set of $t - 1$ DBs with the largest remaining storage since $m[n] \leq \frac{t'}{t}$ for all $n \in [N]$. Therefore, after an iteration, the normalized remaining storage of these e DBs are reduced by α and their normalized remaining storage becomes $\frac{t'}{t} - \alpha$. Furthermore, let t'' be the sum of normalized storage after this iteration. Moreover,

$$t'' = t' - t\alpha \quad (28)$$

and $\frac{t''}{t} = \frac{t'}{t} - \alpha$. Hence, whether the iteration is a PF or CF, e is not decreasing from one iteration to the next.

Next, consider the case where the iteration is a PF, then by (17), we obtain $m[N - N' + 1] > \frac{t'}{t} - m[N - t + 1]$. Therefore, $\alpha = \frac{t'}{t} - m[N - t + 1]$. Furthermore, by (28), $\frac{t''}{t} = m[N - t + 1]$. As the normalized remaining storage at DB_{N-t+1} remains $m[N - t + 1] = \frac{t''}{t}$ and this DB is not included in the e DBs with $\frac{t'}{t}$ normalized remaining storage,⁵ e is increased by at least 1 after this iteration. This completes the proof of Lemma 2. ■

By Lemmas 1 and 2, we can conclude that at most $t - 1$ PFs and $N - t$ CFs are possible during the execution of Algorithm 1 as N' is decreased from N to t . Then when $N' = t$, there are t DBs with equal remaining storage and the special case of Algorithm 1 fills the remaining storage of these DBs. As a result, at most $(t - 1) + (N - t) + 1 = N$ iterations of Algorithm 1 are necessary to fill the storage of the DBs. ■

This result demonstrates that Algorithm 1 requires at most N iterations to complete. As each iteration defines one sub-message per message, the number of sub-messages per message resulting from Algorithm 1 is at most N . This leads to the following corollary.

⁵This is because that if this DB is included the e DBs with $\frac{t'}{t}$ normalized remaining storage, then $e \geq t$.

Corollary 1: Given a set of storage requirements $\mu \in \mathbb{R}_+^N$ such that $\mu[n] \leq 1$ for all $n \in [N]$, $t \in \mathbb{Z}^+$ and $t \geq 1$, there exists heterogeneous SC-PIR scheme with at most NN_d sub-messages per message such that capacity can be achieved, where N_d is the required number of sub-messages for the FS-PIR delivery scheme. □

The total number of sub-messages is the product of the number of sub-messages necessary for the storage and delivery phases. By using the recent result of [4] for delivery, the total number of sub-messages per message is $N \times (N - 1) < N^2$.

VIII. CONCLUSION

In this work, we studied the problem of storage placement for heterogeneous SC-PIR such that the capacity can be achieved. We demonstrated how the storage placement problem is equivalent to a filling problem. Moreover, we provided necessary and sufficient conditions such that a solution to the filling problem exists. These results not only proved that the general storage problem for heterogeneous SC-PIR has a solution, but also the existence of a simple iterative storage placement algorithm such that the conditions are met after each iteration. In addition, when t is an integer, we also showed that the proposed iterative algorithm converges within N iterations. This means that the required number of sub-messages per message is upper bounded by N for storage placement.

REFERENCES

- [1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, IEEE, 1995, pp. 41–50.
- [2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.
- [3] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.
- [4] C. Tian, H. Sun, and J. Chen, "Capacity-achieving private information retrieval codes with optimal message size and upload cost," *arXiv preprint arXiv:1808.07536*, 2018.
- [5] H. Sun and 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.
- [6] R. Tandon, M. Abdul-Wahid, F. Almomalek, and D. Kumar, "PIR from storage constrained databases-coded caching meets PIR," in *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–7.
- [7] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus, "The capacity of private information retrieval from decentralized uncoded caching databases," *arXiv preprint arXiv:1811.11160*, 2018.
- [8] M. A. Attia, D. Kumar, and R. Tandon, "The capacity of private information retrieval from uncoded storage constrained databases," *arXiv preprint arXiv:1805.04104*, 2018.
- [9] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2856–2867, 2014.
- [10] N. Woolsey, R. Chen, and M. Ji, "A new design of private information retrieval for storage constrained databases," *arXiv preprint arXiv:1901.07490*, 2019.
- [11] Y.-P. Wei, S. Ulukus, K. Banawan, B. Arasli, "The capacity of private information retrieval from heterogeneous uncoded caching databases," *arXiv preprint arXiv:1901.09512*, 2019.
- [12] N. Woolsey, R. Chen, and M. Ji, "An optimal iterative placement algorithm for pir from heterogeneous storage-constrained databases," *arXiv preprint arXiv:1904.02131*, 2019.