

# A New Design of Private Information Retrieval for 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**—Private information retrieval (PIR) allows a user to download one of  $K$  messages from  $N$  databases without revealing to any database which of the  $K$  messages is being downloaded. In general, the databases can be storage constrained where each database can only store up to  $\mu KL$  bits where  $\frac{1}{N} \leq \mu \leq 1$  and  $L$  is the size of each message in bits. Let  $t = \mu N$ , a recent work showed that the capacity of Storage Constrained PIR (SC-PIR) is  $(1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}})^{-1}$ , which is achieved by a storage placement scheme inspired by the content placement scheme in the literature of coded caching and the original PIR scheme. Not surprisingly, this achievable scheme requires that each message is  $L = \binom{N}{t} t^K$  bits in length, which can be impractical. In this paper, without trying to make the connection between SC-PIR and coded caching problems, based on a general connection between the Full Storage PIR (FS-PIR) problem ( $\mu = 1$ ) and SC-PIR problem, we propose a new SC-PIR design idea using novel storage placement schemes. The proposed schemes significantly reduce the message size requirement while still meeting the capacity of SC-PIR. In particular, the proposed SC-PIR schemes require the size of each file to be only  $L = Nt^{K-1}$  compared to the state-of-the-art  $L = \binom{N}{t} t^K$ . Hence, we conclude that PIR may not meet coded caching when the size of  $L$  is constrained.

## I. INTRODUCTION

Recent works have taken an information theoretic approach to solve the private information retrieval (PIR) problem [1], [2] originally introduced by Chor *et al.* [3], [4]. In the PIR problem, a user desires to privately download one of  $K$  messages from  $N$  non-colluding databases. In this context, privacy means that the identity of the message desired by the user is not revealed to any database. Ensuring privacy relies on the concept that a user will request sub-messages from all  $K$  messages as opposed to just the message that the user desires. To efficiently download the desired message, the user strategically generates database queries that utilize undesired but downloaded sub-messages for coding opportunities. The rate of a PIR scheme is defined as the ratio of desired bits,  $L$ , or the size of each message, to the total number of downloaded bits,  $D$ . The capacity  $C$  (optimal rate) is defined as the maximum achievable rate.

Previously, Sun and Jafar [1] derived the capacity of the Full Storage PIR (FS-PIR) problem where a user privately downloads one of  $K$  messages from  $N$  databases that each stores all  $K$  messages. In this case, the capacity is  $C = (1 + \frac{1}{N} + \frac{1}{N^2} + \dots + \frac{1}{N^{K-1}})^{-1}$ , which was achieved by a PIR scheme requiring  $L = N^K$ . This result was further

generalized by Attia *et al.* [2] for the Storage Constrained PIR (SC-PIR) problem where each database can only store  $\mu KL$  uncoded bits where  $\frac{1}{N} \leq \mu \leq 1$ . In this case, both a storage placement scheme and a PIR scheme (querying and decoding) need to be designed. Let  $t = \mu N$ , the capacity of SC-PIR is  $(1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}})^{-1}$  under an uncoded storage placement constraint and was achieved by a storage placement scheme inspired by the coded caching problem [5] and a PIR scheme based on [1]. One of the limitations of this scheme is the requirement of a large message size,  $L = \binom{N}{t} t^K$  [2], which is due to the fact that the storage placement is designed based on the cache placement in coded caching problem [5]. Hence, the proposed PIR scheme of [2] can be impractical for a large number of databases. This achievable scheme was generalized to the decentralized storage placement in [6]. Furthermore, Tian *et al.* [7] use Shannon theoretic approach to analyze the SC-PIR problem for the canonical case of  $K = 2$  and  $N = 2$  and proposed the optimal linear scheme. More interestingly, they also showed that non-linear scheme can use less storage than the optimal linear scheme.

In this paper, we aim to find SC-PIR schemes that achieve the capacity of SC-PIR while requiring a significantly smaller message size  $L$ . In order to achieve this goal, for the storage placement, we abandon the idea of using the cache placement of coded caching problem and design it from scratch. In fact, our proposed SC-PIR schemes achieve the capacity and require only  $L = Nt^{K-1}$ , which is significantly less than  $L = \binom{N}{t} t^K$  in [2]. More specifically, our contributions are as follows.

*Our Contributions:*

- 1) We provide a general design methodology for the SC-PIR problem by establishing a generic connection between the FS-PIR and SC-PIR problems. Based on this connection, a SC-PIR scheme can be readily designed from any given FS-PIR scheme.
- 2) We propose a simple storage placement when  $\frac{N}{t}$  is an integer. By adopting the achievable scheme based on [1], the capacity of SC-PIR can be achieved and  $L = Nt^{K-1}$ . This serves as a base case for the more general scenario when  $\frac{N}{t}$  is not an integer.
- 3) When  $\frac{N}{t}$  is not an integer, we propose a novel storage placement, which in conjunction with the FS-PIR scheme of [8], achieves the capacity of SC-PIR and only

requires  $L = Nt^{K-1}$ . The key to the reduction in  $L$  is achieved using the proposed novel storage placement.

- 4) We present a set of sufficient conditions under which the proposed SC-PIR schemes are capacity-achieving.

*Notation Convention:* We use  $|\cdot|$  to represent the cardinality of a set or the length of a vector and  $[n] := [1, 2, \dots, n]$ .

## II. PROBLEM FORMULATION

There are  $K$  independent messages,  $W_1, \dots, W_K$ , each of size  $L$  bits. The messages are collectively stored in an uncoded fashion among  $N$  non-colluding databases that each has a storage capacity of  $\mu KL$  bits, where  $\frac{1}{N} \leq \mu \leq 1$ . We define  $Z_n$  as the storage contents of database  $n \in [N]$ . Also, we define  $t \triangleq \mu N$  as the average number of times each bit of the messages is stored among the databases. A user makes a request  $W_k$  and sends a query  $Q_n^{[k]}$ , which is independent of the messages, to each database  $n \in [N]$  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 with a small probability of error. 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 database has insight into 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  $\mu$ , 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. THE PROPOSED SC-PIR SCHEME WHEN $\frac{N}{t} \in \mathbb{Z}^+$

In order to present the proposed scheme, we need to establish a connection between FS-PIR and SC-PIR problems. This connection is vital to reduce the required minimum size of messages from  $\binom{N}{t}t^K$ , as in the state-of-the-art scheme of [2], to  $Nt^{K-1}$  without affecting the optimal rate. We show that an achievable SC-PIR scheme can be derived from any general achievable scheme for the FS-PIR problem. Hence, by using the proposed storage placement, the achievable scheme in [1] can be used to obtain a new SC-PIR scheme. To illustrate our idea, we first present an example as follows.

### A. A Storage Constrained PIR Example when $\frac{N}{t} \in \mathbb{Z}^+$

Consider  $N = 4$  databases labeled as DB1 through DB4. Collectively the databases store  $K = 3$  messages, denoted by  $A, B$  and  $C$ . Each message is comprised of  $L = 16$  bits.

1) *Storage placement scheme:* We split each message as follows.

$$A = \{a_i^j : i \in [2], j \in [8]\} \quad (5)$$

$$B = \{b_i^j : i \in [2], j \in [8]\} \quad (6)$$

$$C = \{c_i^j : i \in [2], j \in [8]\}. \quad (7)$$

Each database has the storage capacity of up to 24 bits, or half of all 3 messages ( $\mu = \frac{1}{2}$ ). The storage contents of the databases are defined to be

$$Z_1 = Z_2 = \{a_1^j : j \in [8]\} \cup \{b_1^j : j \in [8]\} \cup \{c_1^j : j \in [8]\} \quad (8)$$

$$Z_3 = Z_4 = \{a_2^j : j \in [8]\} \cup \{b_2^j : j \in [8]\} \cup \{c_2^j : j \in [8]\}. \quad (9)$$

2) *PIR Scheme:* Each database stores 8 out of 16 bits of each message. Databases 1 and 2 have the same storage contents, but do not have any storage contents in common with databases 3 and 4. Likewise, databases 3 and 4 have the same storage contents. In this way, we essentially reduce a SC-PIR problem into two independent FS-PIR problems; one consists of databases 1 and 2, and the other consists of databases 3 and 4. Subsequently, we can simply adopt the achievable FS-PIR scheme of [1] to generate the queries for each pair of the databases separately. The queries of a user that desires message  $A$  are shown in Table I.

TABLE I  
STORAGE CONSTRAINED PIR,  $N = 4, K = 3, \mu = \frac{1}{2}$

DB1	DB2	DB3	DB4
$a_1^5 \ b_1^8 \ c_1^6$	$a_1^1 \ b_1^3 \ c_1^1$	$a_2^5 \ b_2^7 \ c_2^4$	$a_2^2 \ b_2^6 \ c_2^2$
$a_1^6 + b_1^3$	$a_1^3 + b_1^8$	$a_2^1 + b_2^6$	$a_2^7 + b_2^7$
$a_1^7 + c_1^1$	$a_1^8 + c_1^6$	$a_2^6 + c_2^2$	$a_2^8 + c_2^4$
$b_1^6 + c_1^5$	$b_1^7 + c_1^3$	$b_2^3 + c_2^6$	$b_2^8 + c_2^7$
$a_1^2 + b_1^7 + c_1^3$	$a_1^4 + b_1^6 + c_1^5$	$a_2^3 + b_2^8 + c_2^7$	$a_2^4 + b_2^3 + c_2^6$

3) *Achievable Rate:* The total number of downloaded bits is  $D = 28$ . Thus, we have for this scheme  $R = \frac{L}{D} = \frac{16}{28} = \frac{4}{7}$ , which achieves the capacity of  $(1 + \frac{1}{t} + \frac{1}{t^2})^{-1} = (1 + \frac{1}{2} + \frac{1}{2^2})^{-1} = \frac{4}{7}$ . Compared to the SC-PIR scheme of [2] that requires  $L = \binom{N}{t}t^K = \binom{4}{2}2^3 = 48$  bits, the proposed SC-PIR requires only  $L = 16$  bits.

4) *Privacy Constraint:* Privacy is ensured since the FS-PIR scheme of [1] is used to privately download half of message  $A$  from DB1 and DB2 and the other half from DB3 and DB4. The query to each database is symmetric such that for each bit of  $A$  that is requested, a bit each from  $B$  and  $C$  are also requested. All coded pairs of bits from the 3 messages are requested an equal number of times. Ultimately, the user can decode all bits of message  $A$ , because downloaded bits of  $B$  and  $C$  can be used for decoding (see Table I). In the following, we will first formalize the connection between the FS-PIR and SC-PIR problems and then generalize this example.

### B. The general connection between the FS-PIR and SC-PIR

Define a vector  $\alpha = [\alpha_1, \dots, \alpha_F]$ , where  $F \in \mathbb{Z}^+$ ,  $\sum_{i=1}^F \alpha_i = 1$ , 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  $f \in [F]$ ,  $|W_{k,f}| = \alpha_f L$  bits. For all  $f \in [F]$ , let

$$M_f \triangleq \bigcup_{k \in [K]} W_{k,f}, \quad (10)$$

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

$$Z_n = \{M_f : f \in [F], n \in \mathcal{N}_f\}, \quad (11)$$

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

$$\sum_{\{f: f \in [F], n \in \mathcal{N}_f\}} \alpha_f \leq \mu. \quad (12)$$

Given that a user requests file  $W_\theta$  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 databases 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.

The rate of the SC-PIR scheme as a function of the rate of the implemented FS-PIR scheme is given in the following theorem.

*Theorem 1:* Given  $N, K, F \in \mathbb{Z}^+$  and  $\alpha$ , split each of the  $L$ -bit messages  $W_1, \dots, W_K$  into  $F$  sub-messages of size  $\alpha_1 L, \dots, \alpha_F L$  and store them at sets of databases  $\mathcal{N}_1, \dots, \mathcal{N}_F \subseteq [N]$ , respectively. Given a set of FS-PIR schemes with achievable rates  $R_1, \dots, R_F$ , the achievable rate of privately downloading  $W_\theta$ ,  $\theta \in [K]$ , from the  $N$  storage constrained databases is

$$R = \left( \frac{\alpha_1}{R_1} + \frac{\alpha_2}{R_2} + \dots + \frac{\alpha_F}{R_F} \right)^{-1}. \quad (13)$$

*Proof:* We first count the number of downloaded bits. For all  $f \in [F]$ ,  $R_f = \frac{\alpha_f L}{D_f}$  where  $D_f$  is the number of downloaded bits necessary to privately download  $W_{\theta,f}$  of size  $\alpha_f L$  bits from the databases in  $\mathcal{N}_f$ . Therefore, the total number of bits required to privately download the entirety of  $W_\theta$  is

$$D = D_1 + D_2 + \dots + D_F = L \left( \frac{\alpha_1}{R_1} + \frac{\alpha_2}{R_2} + \dots + \frac{\alpha_F}{R_F} \right).$$

Since  $R = \frac{L}{D}$ , we obtain (13).  $\blacksquare$

### C. General Achievable Storage Constrained PIR Scheme When $\frac{N}{t} \in \mathbb{Z}^+$

1) *Storage Placement Scheme:* Given  $N \in \mathbb{Z}^+$  and  $t \in [N]$  such that  $\frac{N}{t} \in \mathbb{Z}^+$ , let  $F = \frac{N}{t}$  and for each  $k \in [K]$ , split message  $W_k$  into  $\frac{N}{t}$  disjoint, equal-size sub-messages,  $W_{k,1}, \dots, W_{k, \frac{N}{t}}$ . Furthermore, split the  $N$  databases into  $\frac{N}{t}$

disjoint groups of size  $t$  labeled as  $\mathcal{N}_1, \dots, \mathcal{N}_{\frac{N}{t}}$ . For each  $f \in [\frac{N}{t}]$ , the sub-messages of

$$M_f = \bigcup_{k \in [K]} W_{k,f} \quad (14)$$

are stored at every database of  $\mathcal{N}_f$ .

2) *PIR Scheme:* A user desires to privately download message  $W_\theta$  for some  $\theta \in [K]$ . For each  $f \in [\frac{N}{t}]$ , the user generates a query using the scheme of [1], to privately download  $W_{\theta,f}$  from the  $t$  databases in  $\mathcal{N}_f$ . The user combines the downloaded sub-messages,  $W_{\theta,1}, \dots, W_{\theta, \frac{N}{t}}$  to recover the desired message  $W_\theta$ .

To implement this SC-PIR scheme, each message is split into  $\frac{N}{t}$  equal-size, disjoint sub-messages. Furthermore, the adaptation of the FS-PIR scheme of [1] requires that each sub-message is further split into  $t^K$  equal-size, disjoint sub-messages. The resulting SC-PIR requires a total of  $L = \frac{N}{t} \cdot t^K = Nt^{K-1}$  bits. An example of this SC-PIR scheme is described in Section III-A.

3) *Achievable Rate:* The achievable rate of this scheme is summarized as follows.

*Theorem 2:* Given  $N, K$ , and  $\mu \in [\frac{1}{N}, 1]$ , such that  $t = \mu N \in [N]$ ,  $\frac{N}{t} \in \mathbb{Z}^+$  and  $L = Nt^{K-1}$ , for a user to privately download one of  $K$   $L$ -bit messages from  $N$  databases with a storage capacity of  $\mu K L$  bits, the achievable rate is

$$R = \left( 1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}} \right)^{-1}. \quad (15)$$

$\square$

Moreover, it was shown in [2] that (15) is the capacity of SC-PIR for  $t \in \mathbb{Z}^+$ . While we do not directly prove Theorem 2 here, in Section V we present a set of sufficient conditions, which this scheme satisfies, for an SC-PIR scheme to meet the capacity.

### IV. THE PROPOSED SC-PIR SCHEME WHEN $\frac{N}{t} \notin \mathbb{Z}^+$

In Section III, we established a general connection between SC-PIR and FS-PIR problems. We showed that by properly splitting messages and allocating sub-messages to different groups of databases, a SC-PIR scheme can be derived by applying a separately designed FS-PIR scheme to each group of databases. In particular, when choosing the FS-PIR scheme to be the one in [1], we obtain a SC-PIR scheme that achieves capacity while requiring  $\frac{N}{t} \in \mathbb{Z}^+$ . In order to remove this restriction, in this section, we propose a new storage placement and use it in conjunction with the achievable FS-PIR scheme of [8] to obtain a new SC-PIR scheme. This scheme achieves capacity while requiring only  $L = Nt^{K-1}$ , which is the same as the scheme of Section III-C when  $\frac{N}{t} \in \mathbb{Z}^+$ .

#### A. A Storage Constrained PIR Example when $\frac{N}{t} \notin \mathbb{Z}^+$

In this example,  $N = 5$  databases, labeled DB1 through DB5, collectively store  $K = 2$  messages,  $A$  and  $B$ , and each has a size of  $L = 15$  bits. Each database stores an  $\mu = \frac{3}{5}$  fraction of the 2-message library ( $t = \mu \cdot N = 3$ ).

1) *Storage Placement Scheme*: Each message is split as follows.

$$A = \left\{ a_i^j : i \in [5], j \in [3] \right\}, \quad B = \left\{ b_i^j : i \in [5], j \in [3] \right\}. \quad (16)$$

By this labeling, we have essentially split the messages in two phases. The first splitting phase, denoted by the subscript, determines which databases store these bits. The second splitting, denoted by the superscript, is necessary to perform the FS-PIR scheme. For all  $f \in [5]$ , define

$$M_f = \bigcup_{j \in [3]} \left( a_f^j \cup b_f^j \right) \quad (17)$$

and let the set of databases  $\mathcal{N}_f = [-2 : 0] \oplus_N f$  locally store the bits of  $M_f$ .<sup>1</sup> Note that as opposed to the SC-PIR scheme described in Section III-A where the sets of databases  $\{\mathcal{N}_f, f = 1, \dots, F\}$  are mutually exclusive, here we allow them to overlap and hence removing the integer constraint of  $\frac{N}{t} \in \mathbb{Z}^+$ .

As a result, the bits of message  $A$  stored at DB  $n \in [5]$  are

$$Z_n = \left\{ a_i^j : i \in \{[0 : 2] \oplus_N n\}, j \in [3] \right\}. \quad (18)$$

Message  $B$  is stored among the databases in a similar manner. For instance, DB2 stores all bits  $a_i^j$  and  $b_i^j$  such that  $i \in [2 : 4]$  and DB5 stores all bits  $a_i^j$  and  $b_i^j$  such that  $i \in \{5, 1, 2\}$ .

TABLE II  
STORAGE CONSTRAINED PIR,  $N = 5, K = 2, \mu = \frac{3}{5}$

DB1	DB2	DB3	DB4	DB5
(1, 2, 3)	(2, 3, 4)	(3, 4, 5)	(4, 5, 1)	(5, 1, 2)
$a_1^3 \ b_1^2$	$a_2^3 \ b_2^2$	$a_3^1 \ b_3^3$	$a_4^2 \ b_4^3$	$a_5^2 \ b_5^1$
$a_1^2 + b_2^2$	$a_3^3 + b_3^3$	$a_4^3 + b_4^3$	$a_5^1 + b_5^1$	$a_1^2 + b_1^2$
$a_3^2 + b_3^3$	$a_4^1 + b_4^3$	$a_5^3 + b_5^1$	$a_1^1 + b_1^2$	$a_2^2 + b_2^2$

2) *PIR Scheme*: The queries of a user that desires to privately download message  $A$  are shown in Table II. The top row of the table contains database labels and the 3-tuple below each database label defines the subscripts of the bits that are locally available to that database. The remaining three rows of the table show the queries of the user. The user adopts the FS-PIR scheme of [8] to design queries. For instance, to obtain bits  $\{a_1^j, j \in [3]\}$ , the user applies the FS-PIR to DB1, DB4, and DB5. In the first round, the user obtains  $a_1^3$  from DB1. In the second round, the user can decode  $a_1^1$  from DB4's transmission of  $a_1^1 + b_1^2$  because the user had already received  $b_1^2$  from the first round transmission of DB1 in round 1. Similarly, the user decodes  $a_1^2$  from DB5's transmission of  $a_1^2 + b_1^2$ . These transmissions are highlighted in red in Table II. To ensure privacy, the queries are symmetric and no bit is requested more than once from any one database. In this example,  $D = 20$  bits are downloaded and the rate is  $R = \frac{3}{4}$ . Comparing to the state-of-the-art SC-PIR scheme

<sup>1</sup>We impose the following notation:  $a \oplus_N b = (a + b - 1 \bmod N) + 1$  and  $[a_1 : a_2] \oplus_N b = \{a' \oplus_N b : a' \in [a_1 : a_2]\}$ .

of [2], the rate is the same, but  $L$  has been reduced from  $\binom{N}{t} t^K = \binom{5}{3} 3^2 = 90$  to  $N t^{K-1} = 5 \cdot 3^{2-1} = 15$ .

B. *General Achievable SC-PIR Scheme When  $\frac{N}{t} \notin \mathbb{Z}^+$*

1) *Storage Placement Scheme*: For each  $k \in [K]$ , message  $W_k$  is split into  $N$  disjoint equal-size sub-messages  $W_{k,1}, \dots, W_{k,N}$ . For all  $f \in [N]$ , define a set of sub-messages  $M_f = \bigcup_{k \in [K]} W_{k,f}$  which is locally stored at the set of databases  $\mathcal{N}_f = [-(t-1) : 0] \oplus_N f$ .

2) *PIR Scheme*: A user desires to privately download message  $W_\theta$  for some  $\theta \in [K]$ . For each  $f \in [N]$ , the user generates a query using the scheme of [8], to privately download  $W_{\theta,f}$  from the  $t$  databases in  $\mathcal{N}_f$ . The user combines the downloaded sub-messages,  $W_{\theta,1}, \dots, W_{\theta,\frac{N}{t}}$  to recover the desired message  $W_\theta$ . Furthermore, if desired, to obtain symmetry across the databases, i.e., each database sends the same amount of coded bit combinations from each file, the user can choose database  $f$  to start the query process when privately downloading  $W_{\theta,f}$ . For more details on the query generation process, see [8].

3) *Achievable Rate*: The achievable rate of this SC-PIR scheme is summarized in the following theorem.

*Theorem 3*: Given  $N, K$ , and  $\mu \in [\frac{1}{N}, 1]$ , such that  $t = \mu N \in [N]$  and  $L = N t^{K-1}$ , for a user to privately download one of  $K$   $L$ -bit messages from  $N$  databases, each with a storage capacity of  $\mu K L$  bits, the rate is

$$R = \left( 1 + \frac{1}{t} + \frac{1}{t^2} + \dots + \frac{1}{t^{K-1}} \right)^{-1}. \quad (19)$$

The results of Section V demonstrate that this SC-PIR scheme satisfies the sufficient conditions to meet the capacity. This proves Theorem 3.

## V. SUFFICIENT CONDITIONS TO ACHIEVE CAPACITY FOR SC-PIR

In this section, we provide two sufficient conditions for a storage placement scheme to achieve the SC-PIR capacity.

*Theorem 4*: Given  $N, K, F \in \mathbb{Z}^+$  and  $\alpha$ , split each of the  $L$ -bit messages  $W_1, \dots, W_K$  into  $F$  sub-messages of size  $\alpha_1 L, \dots, \alpha_F L$  and store them at sets of databases  $\mathcal{N}_1, \dots, \mathcal{N}_F \subseteq [N]$  according to equations (10)-(12). Each database has a storage capacity of  $\mu K L$  bits,  $\frac{1}{N} \leq \mu \leq 1$ , where  $t = \mu N \in [1, N]$ . Assume that a user requests file  $W_\theta$  for some  $\theta \in [K]$ . A SC-PIR scheme is obtained if for all  $f \in [F]$ , the user generates a query to privately download  $W_{\theta,f}$  from the databases in  $\mathcal{N}_f$  using a capacity-achieving FS-PIR scheme. The resulting SC-PIR scheme is capacity-achieving if the sub-message storage placement satisfies one of the following two conditions:

- (1) If  $t \in \mathbb{Z}^+$ ,  $|\mathcal{N}_f| = t$  for all  $f \in [F]$
- (2) If  $t \notin \mathbb{Z}^+$ ,  $|\mathcal{N}_f| \in \{[t], [t]\}$  for all  $f \in [F]$  such that

$$\sum_{f: |\mathcal{N}_f| = [t]} \alpha_f = [t] - t \quad (20)$$

and

$$\sum_{f: |\mathcal{N}_f| = [t]} \alpha_f = t - [t]. \quad (21)$$



*Proof:* Define  $R_{\text{FS}}(x)$  as the rate of a capacity achieving FS-PIR scheme to privately download one of  $K$  messages from  $x$  nodes. Furthermore,

$$R_{\text{FS}}(x) = \left(1 + \frac{1}{x} + \cdots + \frac{1}{x^{K-1}}\right)^{-1} \quad (22)$$

as was shown in [1].

For  $t \in \mathbb{Z}^+$ , it follows from Theorem 1 that the rate of the SC-PIR scheme is

$$R = \left(\frac{\alpha_1}{R_{\text{FS}}(t)} + \cdots + \frac{\alpha_F}{R_{\text{FS}}(t)}\right)^{-1} = R_{\text{FS}}(t) \quad (23)$$

$$= \left(1 + \frac{1}{t} + \cdots + \frac{1}{t^{K-1}}\right)^{-1} \quad (24)$$

which is the capacity of SC-PIR [2].

For  $t \notin \mathbb{Z}^+$ , it follows from Theorem 1 that

$$R = \left(\frac{1}{R_{\text{FS}}(\lfloor t \rfloor)} \sum_{f: |\mathcal{N}_f| = \lfloor t \rfloor} \alpha_f + \frac{1}{R_{\text{FS}}(\lceil t \rceil)} \sum_{f: |\mathcal{N}_f| = \lceil t \rceil} \alpha_f\right)^{-1} \quad (25)$$

$$= \left(\frac{\lceil t \rceil - t}{R_{\text{FS}}(\lfloor t \rfloor)} + \frac{t - \lfloor t \rfloor}{R_{\text{FS}}(\lceil t \rceil)}\right)^{-1} \quad (26)$$

and thus

$$R^{-1} = (\lceil t \rceil - t)R_{\text{FS}}^{-1}(\lfloor t \rfloor) + (t - \lfloor t \rfloor)R_{\text{FS}}^{-1}(\lceil t \rceil). \quad (27)$$

Note that the point  $(t, R^{-1})$  is simply a linear interpolation of the two points  $(\lfloor t \rfloor, R_{\text{FS}}^{-1}(\lfloor t \rfloor))$  and  $(\lceil t \rceil, R_{\text{FS}}^{-1}(\lceil t \rceil))$  where the capacity of SC-PIR for  $t = x$  is precisely  $R_{\text{FS}}(x)$ . Moreover, it was shown in [2] that the set of achievable points  $(t, R^{-1})$ , is the lower convex hull of the set points  $\{(t, C_t^{-1}) : t \in [N]\}$ . Therefore, (26) meets the SC-PIR capacity. ■

## VI. DISCUSSION AND FUTURE WORK

Recent works on SC-PIR suggest that coded caching *meets* PIR [2], [9]; that is, the file placement solutions of coded caching [5] are useful for the SC-PIR sub-message placement problem. In this work, we show that coded caching placement techniques are not necessary for SC-PIR by proposing two novel sub-message placement schemes which achieve the capacity. In the coded caching problem, assigning different files to an exponentially large number of overlapping user groups is necessary to create multicasting opportunities such that a user can cancel “interference” from a received coded transmission which also serves other users. The SC-PIR problem is less complex in that only one user is being served. In fact, as was demonstrated with our first proposed scheme, it is not necessary for the sub-message placement groups to overlap at all. Moreover, the file (or sub-message) placement paradigms of coded caching and SC-PIR are inherently different. In coded caching, files are being placed among users that wish to download content, while in SC-PIR, sub-messages are being placed among databases which are serving one user’s request. Therefore, it is not surprising the two problems could have different solutions for the storage/file placement problem.

The results of Section V show that there exists simple SC-PIR solutions for non-integer  $t$ . For example, the databases could be split into two disjoint groups, one in which sub-messages are assigned to sub-groups of size  $\lfloor t \rfloor$  databases, and another where sub-messages are assigned to sub-groups of size  $\lceil t \rceil$  databases. This is contrary to the solution for non-integer  $t$  of the coded caching problem where the storage of every user is split into two parts to essentially create two coded caching networks that both span across all users [5]. While this coded caching method was proposed to solve the non-integer  $t$  SC-PIR problem in [2], we have shown that this is not necessary.

This work presents several interesting directions for future work. First, it remains an open problem to determine the minimum message size  $L$  for a given set of SC-PIR parameters. Using a definition of the retrieval rate that is slightly different from that of [8], it was shown in [10] that the minimum  $L$  of an FS-PIR problem can be reduced significantly from  $N^{K-1}$  in [8] to  $N - 1$ . The new FS-PIR scheme [10] can be readily adapted to our proposed SC-PIR to reduce the message size. Furthermore, the proof techniques therein may be useful to derive the minimum  $L$  for a SC-PIR problem. Second, another work [6] has considered random placement among databases where a database stores a bit of a given message with probability  $\mu$ . Interestingly, this placement method was also used in [11] for the coded caching problem. It will be meaningful to examine alternative random placement strategies for the SC-PIR problem where messages are split into a finite number of sub-messages.

## REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, “Private information retrieval,” *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.
- [5] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *Information Theory, IEEE Transactions on*, vol. 60, no. 5, pp. 2856–2867, 2014.
- [6] 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.
- [7] C. Tian, H. Sun, and J. Chen, “A shannon-theoretic approach to the storage-retrieval tradeoff in pir systems,” in *2018 IEEE International Symposium on Information Theory (ISIT)*, June 2018, pp. 1904–1908.
- [8] 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.
- [9] R. Tandon, M. Abdul-Wahid, F. Almoalem, 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.
- [10] 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.
- [11] M. A. Maddah-Ali and U. Niesen, “Decentralized coded caching attains order-optimal memory-rate tradeoff,” *Networking, IEEE/ACM Transactions on*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.