Fundamental Limits of Private Information Retrieval with Unknown Cache Prefetching

Hyowoon Seo, *Member, IEEE,* Hojung Lee, *Student Member, IEEE,* and Wan Choi, *Fellow, IEEE*

***Abstract*—The fundamental limits of private information retrieval (PIR) with unknown cache prefetching at the user are investigated in this paper. To this end, a novel random linear combination (RLC)-based PIR scheme that can solve the basic PIR problem and its variation is proposed. The proposed scheme is based on random coding approach and achieves the capacity of the basic PIR asymptotically. Then, we investigate PIR with unknown cache prefetching (PIRC) problem at different cache-to-file size ratio. Specifically, we propose RLC-based PIRC method, which prefetches RLC-based side information and leverages them to retrieve desired information at small download cost. Furthermore, by applying time and memory sharing on the proposed RLC-based PIRC, RLC-based basic PIR and some other known approach in literature, we derive the achievable normalized download cost bound of PIRC. The derived achievable bound outperforms the existing bound in literature and the case study provides numerical results that verifies it.**

***Index Terms*—Private information retrieval, random linear combination, cache prefetching, coded cache, interference cancellation**

I. Introduction

The private information retrieval (PIR) problem was first introduced and addressed by Chor, Kushilevitz, Goldreich, and Sudan [2], [3] in 1995. The problem considers a network composed of a user and multiple non-colluding databases (DBs), each of which contains the same set of files including the desired file of the user. In such network environment, the PIR solves the way for the user to acquire the desired file without revealing identity of the file to the DBs. Intuitively, the simplest solution for the PIR is to download the entire file set from the DBs. However, this is very inefficient due to high communication cost. Therefore, the fundamental studies of the PIR [2]–[6] usually seek for communication-efficient ways of retrieving the desired file privately. Recently, the capacity of the PIR problem, which is the maximum number of bits of the desired file that can be privately retrieved per a single bit of downloaded information, was found in the information theoretical perspective [7]. In the paper, a greedy iterative algorithm based query structure was proposed for the capacity achieving scheme and the converse was proved with an induction based method. After the appearance of [7], more complicated PIR problems have been studied. The PIR with assuming colluding DBs was studied in [8]–[14] and in [15], [16], the authors studied a PIR problem where the user is restricted to decode the desired file only from the answering string received from the DBs. Furthermore, PIR problems with coded data, such as maximum distance separable (MDS) codes, were considered in [11], [13]. More recently, the concept of PIR was leveraged to gain communication-efficiency under multi-user scenario in [17].

Meanwhile, it is a widely known fact that the efficiency of communication greatly improves with aid of cache memory equipped either at transmitter or receiver, or indeed both [18], [19]. Thus, it is natural to expect communication-efficiency enhancement with the aid of cache at the user in PIR problems. In [20], cache-aided PIR problem with the prefetched side information known by the DBs was studied and it was shown that the known prefetched side information could reduce the download cost for desired information retrieval under privacy, but no enhancement in the PIR rate for the uncached portion of the file. A scenario that user prefetching a random subset of the files unknown to the DBs and leveraging them for PIR was studied in [21]. Since the study focused only on a single DB scenario, there is left room for additional investigations on such settings with multiple DBs. Meanwhile, a cache-aided PIR problem with uncoded cache prefetching was studied in [22], which was unknown by the DBs. The unknown prefetched side information was composed of the equal number of uncoded bits from each file. As a result, [22] provided the optimal download cost for very small and very large caching ratios and further showed that the achievable scheme was still tight in the non-optimal region. More recently, the capacity of PIR with private side information, when M files are stored in each database and S files among M files are stored at the user, is studied under partially known condition [23], under T colluding database condition [24] and multi-user setting in [25]. The results in the papers state that the capacity and optimal download cost of such a problem is equivalent to that of the basic PIR problem [7] or basic T-PIR problem [8] with $M-S$ messages stored in each database without cached

Manuscript received February 13, 2021; revised July 29, 2021; accepted September 24, 2021.

This research was supported in part by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center(ITRC) support program (IITP-2020-0-01787) supervised by the Institute of Information & Communications Technology Planning & Evaluation (IITP) and in part by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center(ITRC) support program (IITP-2021-0-02048) supervised by the Institute of Information & Communications Technology Planning & Evaluation (IITP).

A part of this paper was presented in IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, April 2018 [1].

H. Seo is with Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland (e-mail: hyowoon.seo@oulu.fi).

H. Lee is with School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Korea (e-mail: hojung_lee@kaist.ac.kr).

W. Choi is with the Institute of New Media and Communications and Department of Electrical and Computer Engineering, Seoul National University (SNU), Seoul 08826, Korea (e-mail: wanchoi@snu.ac.kr). *(Corresponding author: Wan Choi)*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

data.

The preceding studies considering *cache aids* successfully showed the effectiveness of side information provided at the user in terms of the performance of PIR. However, most of them considered uncoded cache prefetching at the user and followed the conceptual framework introduced in [7], which was based on a structured and iterative algorithm. Recently, the role of coded cache prefetching was studied in [26] under single server setting and also in multiple server setting in [27], [28], however, they only considered linear combination of files as side information. In contrast, the following two challenges are addressed in this paper:

C1. *Finding analytically tractable approach for achieving fundamental limits (e.g., capacity) of various PIR problems.*

To find generally tractable approaches for the PIR, considering asymptotic analysis based on random coding with infinite message length is one of the options that we can take.

C2. *Designing cache prefetching strategy and finding fundamental limits for the PIR with cache memory at the user.*

If **C1** is well addressed, it will allows us to flexibly design cache prefetching at the user for enhancing the PIR performance. Here cache can store some parts of files or their code, not necessarily store an entire file or their code. Moreover, we can jointly design the cache prefetching strategy with respect to the information retrieving strategy or size of given cache memory at the user.

We address **C1** by proposing a novel PIR scheme that is based on asymptotic analysis, which is tractable for the general PIR problems. To this end, we first consider a typical set termed ϵ-*private set* that defines ϵ-privacy for small $\epsilon > 0$. The set includes all possible queries that is close to a perfectly private query, so if a query is in the set, we say it is ϵ-private, where ϵ defines the *closeness*. Then, we introduce a simple random linear combination (RLC) based query generating approach, which can achieve ϵ-privacy with finite query size. Finally, we prove that the proposed RLC based scheme is achieving the capacity of basic PIR under $\epsilon = 0$, which coincides with the result obtained in [7]. Compared to the conventional approach, the proposed scheme is memoryless in generating queries, i.e., generating a new query is not affected by the one generated in the past, and can consider different levels of privacy presented by ϵ. Furthermore, the proposed scheme is designed by using the concept of information-theoretic random coding schemes. Although it achieves optimal points asymptotically, the proposed RLC based scheme is easy to customize for PIR problem variations, just as random coding has provided answers to various problems in information theory.

Extending the proposed query generation method, we address **C2**, the PIR problem when the user is equipped with cache memory storing side information unknown to DBs, termed *PIR with unknown Cache prefetching (PIRC)*. In this setting, two operational phases are considered, one is the cache prefetching and the other is information retrieval. For cache prefetching, RLC-based cache prefetching, which stores a RLC of the information bits from the files in the cache memory at the user, is proposed. For the information retrieval phase,

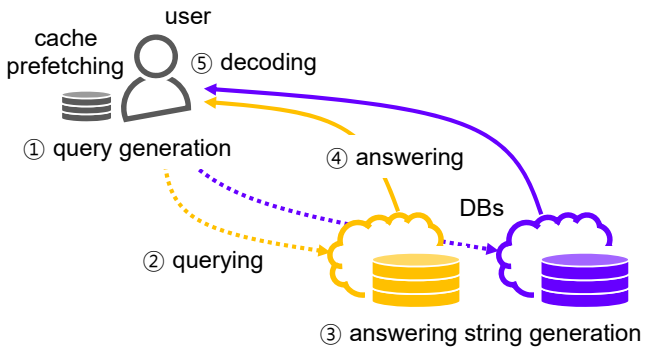


Figure 1: An overview of the private information retrieval with cache (PIRC).

we develop a RLC-based PIRC scheme based on the proposed approach for addressing **C1**. As a result, it is shown that the proposed RLC-based PIRC achieves the optimal normalized download cost at some cache-to-file size ratio. Furthermore, by borrowing one of the unknown and uncoded cache prefetching scheme from [22], we obtain achievable normalized download cost bound for the rest of the cache-to-file size ratio regions, which outperforms the conventional cache-aided PIR schemes in [22] as also corroborated in the case study in Section V.

The rest of this paper is organized as follows. Section II presents the problem formulation of the PIRC. Section III proposes a novel PIR scheme that achieves the capacity of the basic PIR. Section IV investigates the PIRC problem and obtain fundamental limits on the normalized download costs of the PIRC. We do case studies and provide numerical results under several environments in Section V and conclude in Section VI.

Notation : The function $I(X;Y)$ is used to denote the mutual information between X and Y and $H(X)$ is used to denote the entropy of X. Further definitions on the notations will be made later on if necessary.

II. System Model

The network under study is composed of a single user and non-colluding N databases (DBs), denoted by $\mathsf{DB}_1, \ldots, \mathsf{DB}_N$. Each and every DB contains an equal set of distinct K files. The size of each file is fixed to M_f bits and the k^{th} file is denoted by $\mathbf{F}_k = (F_{k,1}, \ldots, F_{k,M_f}) \in \{0,1\}^{M_f}$ for all $k \in \{1, \ldots, K\}$. Each file is independently and uniformly constructed over a finite set $\{1, \ldots, 2^{M_f}\}$. The user is equipped with a storage device of size $M_z = zM_f$ bits as cache memory, containing prefetched side information $\mathbf{Z} = (Z_1, \ldots, Z_{M_z}) \in \{0,1\}^{M_z}$. Assume that the prefetched side information remains fixed over time and is unknown to the DBs. Given such environment, consider a PIR problem, where the user desires to privately and reliably retrieve a desired file $\mathbf{F}_\theta$ from the DBs with help of the stored sided information, where $\theta \in \{1, \ldots, K\}$ is the uniformly chosen desired file index. To this end, the user and each DB are bidirectionally communicating over a half-duplex point-to-point noiseless channel.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

The information retrieving process follows a simple query-and-answer mechanism as illustrated in Fig. 1. Once the desired file is chosen, the user sends *queries* to the DBs to get *answering strings*. Each answering string is composed of multiple answering bits which are generated at the DB, based on the query received from the user. Plainly put, each query includes information of how the corresponding answering string is generated at the DB. For retrieving the desired file $\mathbf{F}_\theta$, the user sends queries $\mathbf{Q}_1^{(\theta)}, \ldots, \mathbf{Q}_N^{(\theta)}$ respectively to $\mathsf{DB}_1, \ldots, \mathsf{DB}_N$. Upon receiving $\mathbf{Q}_n^{(\theta)}$, DB_n generates an answering string $\mathbf{A}_n^{(\theta)}$ of length $M_{a,n}^{(\theta)}$ bits, which is a deterministic function of the received query $\mathbf{Q}_n^{(\theta)}$ and the files stored at DB_n. The total number of downloaded answering bits at the user is denoted by $M_a^{(\theta)} = \sum_{n=1}^N M_{a,n}^{(\theta)}$. The structure of the queries will be further discussed in detail in the following section (see Section III).

In the mean time, the correctness of the answering strings received at the user is examined by measuring error probability. Let $\hat{\mathbf{F}}_\theta$ be the estimate of the desired file at the user after receiving the answering strings, then the average probability of error for the answering string is defined as

$$P_e = \Pr(\hat{\mathbf{F}}_\theta \neq \mathbf{F}_\theta) \tag{1}$$

$$= \frac{1}{K}\sum_{k=1}^{K} \Pr(\hat{\mathbf{F}}_k \neq \mathbf{F}_k \,|\, \theta = k). \tag{2}$$

For correct information retrieval, $P_e \to 0$ for $M_f \to \infty$. Accordingly, from Fano's and data processing inequalities, we have the following *correctness condition*,

$$H(\mathbf{F}_\theta \,|\, \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}) \leq M_f \epsilon_{M_f}, \tag{3}$$

where $\mathbb{Q}_{\mathcal{N}}^{(\theta)} = [\mathbf{Q}_1^{(\theta)}, \ldots, \mathbf{Q}_N^{(\theta)}]$, $\mathbb{A}_{\mathcal{N}}^{(\theta)} = [\mathbf{A}_1^{(\theta)}, \ldots, \mathbf{A}_N^{(\theta)}]$ and $\epsilon_{M_f} \to 0$ as $x \to \infty$.

To protect privacy while information retrieving, each DB should not be able to distinguish which file is desired by the user. Hence, the posterior probability must be uniformly distributed over $\theta_n \in \{1, ..., K\}$ for all $n \in \{1, \ldots, N\}$, of which the physical meaning is the desired file is indistinguishable at each DB. As in the correctness condition, we consider an asymptotic version of the *privacy condition*,

$$\Pr(\theta_n = k \,|\, \mathbf{Q}_n^{(\theta)}, \mathbf{F}_1, ..., \mathbf{F}_K) = \frac{1}{K}, \tag{4}$$

as $M_f \to \infty$, for all $n \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$.

Define a normalized download cost D as the number of expected downloaded bits that is required to retrieve an arbitrary desired file normalized by the file size M_f. Since the total size of downloading answering strings is $M_a^{(\theta)}$, we can express the normalized download cost for downloading the desired file $\mathbf{F}_\theta$ as $D^{(\theta)} = \frac{M_a^{(\theta)}}{M_f}$. The normalized download cost $D^{(\theta)}$ is said to be achievable if the N queries uploaded by the user to the DBs satisfy the privacy condition (4) and the corresponding answering strings satisfy the correctness condition (3). Since we assume that the desired file index θ is uniformly chosen over the set $\{1, \ldots, K\}$, the achievable normalized download cost averaged over the file index set is

$$D = \frac{1}{K}\sum_{\theta=1}^{K} \frac{M_a^{(\theta)}}{M_f}. \tag{5}$$

The optimal normalized download cost is the infimum over all achievable average normalized download cost and denoted by D^*.

Table I: Summary of mathematical notation

Notation	Meaning
$\mathbf{F}_k$	k^{th} file, $k \in \{1, \ldots, K\}$
DB_n	n^{th} database, $n \in \{1, \ldots, N\}$
θ	Desired file index
$\mathbf{Q}_n^{(\theta)}$	Query sent to DB_n
$\mathbf{S}_n^{(\theta)}$	Subqueries in $\mathbf{Q}_n^{(\theta)}$
$\mathbf{B}_n^{(\theta)}$	Initial Bit Indices in $\mathbf{Q}_n^{(\theta)}$
$\mathbf{A}_n^{(\theta)}$	Answering string from DB_n
$M_{s,n}^{(\theta)}$	Number of subqueries in $\mathbf{S}_n^{(\theta)}$
$M_{a,n}^{(\theta)}$	Size of $\mathbf{A}_n^{(\theta)}$ (bits)
$M_a^{(\theta)}$	Total downloaded answering bits
M_z	Cache memory size (bits)
z	Ratio of M_z to M_f
D	Normalized download cost

III. Capacity Achieving Scheme of PIR: An Asymptotic Approach

In this section, the random linear combination (RLC) based PIR capacity achieving scheme [1] is explained. Differentiated from the well-known iterative scheme that achieves the capacity in the literature [7], the scheme explained in this section is memoryless, i.e., the generation of the future query (subquery to be precise) is not affected by the future query, communication-efficient in uploading queries and mathematically tractable, while achieving the capacity in an asymptotic manner. The PIR capacity is defined as the supremum over all achievable PIR rates [7], where PIR rate is an inverse of normalized download cost when the user is not equipped with cache memory $z = 0$.

A. Preliminaries: Query Structure and ϵ-Privacy Condition

We first describe the query structure under consideration and a novel definition of privacy based on a *typical set*.

1) Query Structure: The query structure is designed to retrieve linear combinations of the file bits. Refer to the following example which illustrates retrieving an answering string from a DB using the queries under such structure.

Example 1. (Subquery Structure) *Consider a DB storing two files, that is $\mathbf{F}_1$ and $\mathbf{F}_2$. To retrieve a file bit from $\mathbf{F}_1$, the user sends a binary sequence $(1, 0)$. Similarly, a binary sequence $(0, 1)$ is sent to receive a file bit from $\mathbf{F}_2$. For retrieving a linear combination of file bits from $\mathbf{F}_1$ and $\mathbf{F}_2$, the user sends a binary sequence $(1, 1)$.*

Such binary sequence that retrieves answering bits, e.g., $(1, 0)$, $(0, 1)$ and $(1, 1)$ in Example 1, is referred to as a *subquery*, and a group of subqueries composes a *query*. Define

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

the m^{th} subquery of the query sent to DB_n for retrieving a file $\mathbf{F}_\theta$ by

$$\mathbf{S}_{n,m}^{(\theta)} = (S_{n,m,1}^{(\theta)}, \ldots, S_{n,m,K}^{(\theta)}) \in \{0,1\}^K, \tag{6}$$

for all $m \in \{1, \ldots, M_{s,n}^{(\theta)}\}$ and $n \in \{1, \ldots, N\}$, where $M_{s,n}^{(\theta)}$ denotes the total number of generated subqueries.

In the mean time, though a subquery contains a lot of information on the desired answering bit by the user, it still lacks information about the specific bit index from each file that requires to be linearly combined. One straight forward way is to include bit indices for all linear combinations in the query together with subqueries, however, the size of such bit indices will be obviously larger than that of subqueries, so sending them may require a large amount of upload resources. Instead, we consider a method that the file bits for generating answering bits are used in a sequential order. Consider following example (see Fig. 2) that illustrates how the file bits are used sequentially for generating answering bits.

Example 2. (Initial bit indices) *Consider a DB storing two files,* $\mathbf{F}_1 = (F_{1,1}, F_{1,2})$ *and* $\mathbf{F}_2 = (F_{2,1}, F_{2,2})$*, each composed of two bits and suppose that the user wants to get three answering bits,* $F_{1,1}$*,* $F_{1,2} + F_{2,1}$ *and* $F_{2,2}$*. Given that the file bits are used sequentially for generating answering bits, if the user inform the DB of file bit index '1' and '2' for* $\mathbf{F}_1$ *and* $\mathbf{F}_2$*, respectively, to indicate initial bit positions for the sequential generation, in addition to the subqueries* $(1, 0)$*,* $(1, 1)$ *and* $(0, 1)$*, then the user can successfully get the desired answering bits.*

In Example 2, the file bit index is sent to the DBs in a tuple, e.g., $(1, 2)$, and such information is referred to as *initial bit indices (IBIs)*. We write the IBIs for each file, which are sent to DB_n to retrieve $\mathbf{F}_\theta$, as a K-tuple

$$\mathbf{B}_n^{(\theta)} = (B_{n,1}^{(\theta)}, \ldots, B_{n,K}^{(\theta)}) \in \{1, \ldots, M_f\}^K. \tag{7}$$

The user sends the IBIs along with the subqueries to the DBs and each DB rearranges the order of file bits based on the information given in the IBIs. For example, if $B_{n,k} = 1$, the file bits of $\mathbf{F}_k$ is reorganized in the order of $F_{k,1}, F_{k,2}, \ldots, F_{k,M_f}$ at DB_n. If $B_{n,k} = M_f$, $\mathbf{F}_k$ is reorganized in the order of $F_{k,M_f}, F_{k,1}, \ldots, F_{k,M_f-1}$, in a circular way.

A *query* incorporates both the subqueries and the IBIs. The query $\mathbf{Q}_n^{(\theta)}$ sent to DB_n for retrieving $\mathbf{F}_\theta$ is therefore denoted by

$$\mathbf{Q}_n^{(\theta)} = (\mathbf{S}_n^{(\theta)}, \mathbf{B}_n^{(\theta)}), \tag{8}$$

where $\mathbf{S}_n^{(\theta)} = (\mathbf{S}_{n,1}^{(\theta)}, \ldots, \mathbf{S}_{n,M_{s,n}^{(\theta)}}^{(\theta)})$ denotes the collection of subqueries.

2) ϵ-Private Subqueries: To examine the level of privacy of a query, we adopt the theory of joint typical sequences and joint typicality [29]. Consider M binary sequences of length K, which are denoted by $\mathbf{X}_1, \ldots, \mathbf{X}_M$, where

$$\mathbf{X}_m = (X_{m,1}, \ldots, X_{m,K}) \in \{0,1\}^K, \tag{9}$$

for $m \in \{1, \ldots, M\}$. Then, define their *empirical probability mass function* (p.m.f.) by

$$\pi(\mathbf{x}|\mathbb{X}) = \frac{|\{m : \mathbf{X}_m = \mathbf{x}\}|}{M}, \tag{10}$$

for all $\mathbf{x} = (x_1, \ldots, x_K) \in \{0,1\}^K$, where $\mathbb{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_M)$ is used for notational simplicity. Let X be a Bernoulli random variable distributed with p.m.f. $p_X(x)$, $\forall x \in \{0,1\}$ and for some constant $0 < \epsilon < 1$, define by $\mathcal{P}_\epsilon^{(M)}(X)$ as (11), the *ϵ-private set* of M binary sequences of length K. Note that the ϵ-private set is a typical set, which is a set of sequences of which empirical distribution is close to $\prod_{i=1}^K p_X(x_i)$. Here, ϵ defines the closeness between the distributions, i.e., for smaller ϵ, the set includes binary sequences of which empirical distribution is closer to $\prod_{i=1}^K p_X(x_i)$. For any collection of subqueries $\mathbf{S}_n^{(\theta)}$, if there exists a Bernoulli random variable S such that $\mathbf{S}_n^{(\theta)} \in \mathcal{P}_\epsilon^{(M_{s,n}^{(\theta)})}(S)$, we say that $\mathbf{S}_n^{(\theta)}$ is *ϵ-private*. Specifically, when $\epsilon = 0$ and $\mathbf{S}_n^{(\theta)} \in \mathcal{P}_0^{(M_{s,n}^{(\theta)})}(S)$, the empirical distribution of the subqeuries $\mathbf{S}_n^{(\theta)}$ is equal to $\prod_{i=1}^K p_X(x_i)$, which means that information bits are equally likely and independently requested from each file by the user. In such case, we say that the subqueries are perfectly private since it satisfies the condition (4).

Remark 1. *Let* $\mathbf{S}_n^{(\theta)} = \{\mathbf{S}_{n,1}^{(\theta)}, \ldots, \mathbf{S}_{n,M}^{(\theta)}\}$ *and* $\mathbf{S}_n'^{(\theta)} = \{\mathbf{S}_{n,1}'^{(\theta)}, \ldots, \mathbf{S}_{n,M'}'^{(\theta)}\}$ *be two collections of independent subqueries. If there exist Bernoulli random variables* S *and* S' *such that* $\mathbf{S}_n^{(\theta)} \in \mathcal{P}_0^{(M)}(S)$ *and* $\mathbf{S}_n'^{(\theta)} \in \mathcal{P}_0^{(M')}(S')$*, then the concatenated subquery* $(\mathbf{S}_n^{(\theta)}, \mathbf{S}_n'^{(\theta)})$ *is also perfectly private.*

Note that Remark 1 holds only for two groups of independent subqueries, but not two independent queries that include IBIs already. Later on, we recall Remark 1 when applying memory or time sharing between two different private schemes to make a new private scheme.

3) Answering String Generation: When DB_n receives query $\mathbf{Q}_n^{(\theta)}$, it constructs an answering string $\mathbf{A}_n^{(\theta)}$, composed of $M_{a,n}^{(\theta)}$ answering bits, where the m^{th} answering bit is generated as

$$A_{n,m}^{(\theta)} = \sum_{k \in \mathcal{A}_{n,m}^{(\theta)}} \mathbf{next}\left(\mathbf{F}_k, B_{n,k}^{(\theta)}\right), \tag{12}$$

for all $m \in \{1, \ldots, M_{a,n}^{(\theta)}\}$, where $\mathcal{A}_{n,m}^{(\theta)} = \{k : S_{n,k}^{(\theta)} = 1, \forall k \in \{1, \ldots, K\}\}$ and $\mathbf{next}(\mathbf{F}_k, B_{n,k}^{(\theta)})$ is a function which returns the next bit of the most recently used bit in $\mathbf{F}_k$, starting

$$\mathcal{P}_\epsilon^{(M)}(X) = \left\{ \mathbb{X} : \left| \pi(\mathbf{x}|\mathbb{X}) - \prod_{i=1}^K p_X(x_i) \right| \le \epsilon \prod_{i=1}^K p_X(x_i), \forall \mathbf{x} \in \{0,1\}^K \right\}. \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

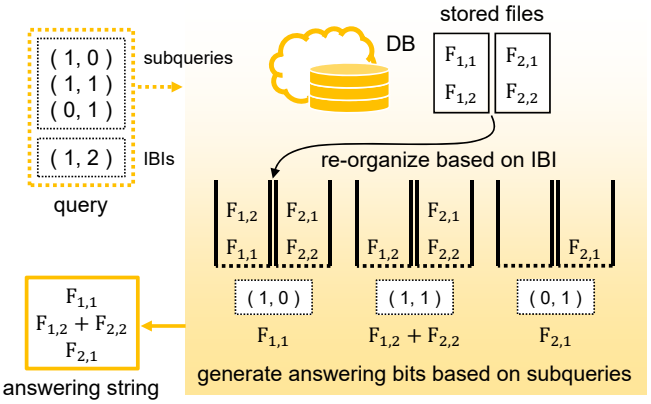


Figure 2: An illustration of query structure and generation of an answering string (see Example 2).

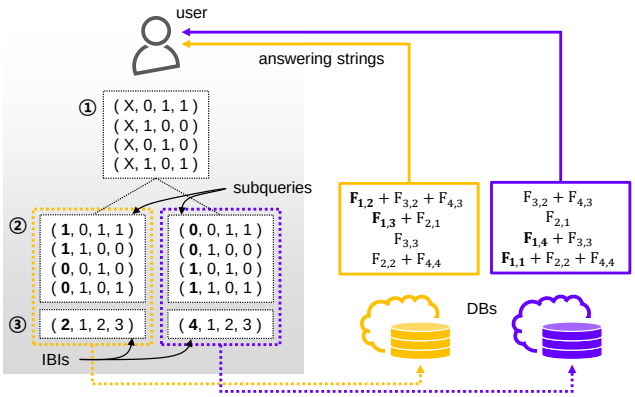


Figure 3: A simple sketch of a query generation in the proposed achievability for the basic private information retrieval with $N = 2$ DBs, $K = 4$ files and file size $M_f = 4$ bits. (see Section III-B).

from $F_{k,B_{n,k}^{(\theta)}}$. Note that the number of generated subqueries $M_{s,n}^{(\theta)}$ and the length of the answering string $M_{a,n}^{(\theta)}$ are the same in general, but since the subqueries will be generated randomly in our proposed scheme, $M_{s,n}^{(\theta)}$ is larger than $M_{a,n}^{(\theta)}$ by the numbers of generated all-zero-element subqueries. This is because the all-zero-element subqueries does not request any information from a DB.

Remark 2. *Compared to the conventional query generation approach [7], the proposed subquery and IBI based method may reduce the upload (querying) cost, since it only requires to specify a single initial index per file, i.e., K indices in total for one query, while the conventional method must specify indices of every linearly combining file bits owing to its query structure.*

For example, consider a case that the user requests $F_{1,2} + F_{3,5} + F_{4,3}$, $F_{1,3} + F_{2,1}$, $F_{3,6}$, and $F_{2,2} + F_{4,4}$. In the conventional scheme, the user should include bit indices $\{2, 5, 3\}$, $\{3, 1\}$, $\{6\}$, and $\{2, 4\}$ for each linear combination in the query with subqueries. However, in the proposed scheme, the user needs to upload only the information of initial bit indices $\{2, 1, 5, 3\}$ in the query with subqueries. Thus, compared to the conventional scheme uploading 8 integers for bit indices, the proposed scheme is communication-efficient since it only uploads 4 integers. Obviously, if the number of requesting answering bits is larger, the proposed scheme becomes more communication-efficient, since the upload cost of the conventional scheme increase as the the number of requesting answering bits increases, while the upload cost of the proposed scheme stays constant.

B. Random Linear Combination Query Achieves the Capacity of PIR

In [7], the authors showed that the queries generated by a greedy iterative algorithm can achieve the capacity of the basic PIR problem. The underlying idea for achieving the capacity is *interference cancellation* [30]. Consider the following example that illustrates a simple interference cancellation based scheme achieving PIR capacity.

Example 3. *Consider two DBs storing two files*

$$\mathbf{F}_1 = (F_{1,1}, F_{1,2}, F_{1,3}, F_{1,4})$$
$$\mathbf{F}_2 = (F_{2,1}, F_{2,2}, F_{2,3}, F_{2,4}),$$

each composed of four bits. Suppose that the user is desiring to retrieve $\mathbf{F}_1$, without revealing the desired file index to the DBs. If the user sends queries to the DBs to get perfectly private answering strings

$$\mathbf{A}_1^{(1)} = (F_{1,1}, F_{2,1}, F_{1,2} + F_{2,2})$$
$$\mathbf{A}_2^{(1)} = (F_{1,3}, F_{1,4} + F_{2,1}, F_{2,2})$$

from DB_1 and DB_2, respectively. By cancelling out $F_{2,1}$ and $F_{2,2}$, the user can get all four desired file bits, thereby achieving the PIR capacity $\frac{2}{3}$.

We call the undesired information that is linearly combined with the desired file bit, e.g., $F_{2,1}$ in $\mathbf{A}_2^{(1)}$ and $F_{2,2}$ in $\mathbf{A}_1^{(1)}$ in Example 3, as an *interferer*. The desired bit with interference is attainable by cancelling out the interferers using the *undesired-but-useful information*, e.g., $F_{2,1}$ in $\mathbf{A}_1^{(1)}$ and $F_{2,2}$ in $\mathbf{A}_2^{(1)}$ in Example 3. The basic idea of the achievable scheme is to exploit interferers for protecting privacy and cancel them out by receiving the interferers from the other DB. In addition to the interference cancellation based method, we introduce a random linear combination approach for designing the capacity achieving scheme for PIR problem.

1) Sketch of PIR with RLC Query: To retrieve the desired file $\mathbf{F}_\theta$, the generation of a query is carried out in two steps as follows (also see Fig. 3 which illustrates how the desired file bits are retrieved from the DBs in the proposed capacity achieving scheme for the basic PIR problem).

① *(Subquery Generation)* The user randomly generates interferers (or undesired-but-useful information) by generating M random binary sequences of length $K - 1$. Now, make N copies of the generated sequences, one for each DB. Among N copies of a generated sequence, uniformly choose one copy and attach '0' as the θ^{th} entry of the sequence, and for the other $N-1$ copies, attach '1' as their θ^{th} entries. This is done independently for all generated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

sequences. The generated sequences, i.e., subqueries, are perfectly private if $M \to \infty$.

② *(IBIs Selection)* In the query transmission to DB_1, IBI of each file is uniformly and independently chosen within $[1, M_f]$. In the query transmission to DB_n, for $n \in \{2, \ldots, N\}$, IBI of the desired file is chosen by adding $(n-1)\left\lfloor \frac{M_f}{N} \right\rfloor$ to the IBI of the desired file in the query to DB_1 (if the chosen IBI is larger than M_f, subtract M_f from the IBI). As the IBIs of the undesired files, choose the same IBIs in the query transmission to DB_1, for all the other DBs.

Each generated query is uploaded to the corresponding DB and the DB generates an answering string based on the information obtained from the query. Then the answering strings are downloaded to the user and the decoding process of the answering strings are done based on the interference cancellation approach as shown in the Example 3.

2) Capacity of PIR: Further mathematical details of the proposed scheme based on RLC is formally described in Appendix A. The following theorem states that the proposed scheme achieves the capacity of the PIR with file size approaching to infinity, i.e., $M_f \to \infty$.

Theorem 1. (Capacity of PIR) *Given N DBs storing an identical file set composed of K equal-sized files, the private information retrieval (PIR) capacity*

$$C_{\mathrm{PIR}} = (N\varphi_N)^{-1}, \tag{13}$$

where $\varphi_n = \sum_{k=1}^{K} n^{-k}$, is achievable when the file size approaches infinity $M_f \to \infty$.

Proof: The proof of PIR capacity achievability with the RLC based scheme is provided in Appendix B and the converse follows the proof in [7]. ■

Remark 3. *Since the optimal normalized download cost and the PIR capacity are reciprocal, we can obtain the optimal download cost of the PIR (with $z = 0$) by taking the inverse of C_{PIR}.*

Remark 4. *Instead of generating subqueries randomly, by planned designing of the subqueries with similar approach, the file size $M_f = N^{K+1} - N^K$ is sufficient to show the capacity is achievable with satisfying the privacy condition.*

IV. Private Information Retrieval with Unknown Cache Prefetching

In this section, we investigate the *PIR with unknown cache prefetching (PIRC)* problem, when the user is equipped with cache memory of size $M_z = zM_f$. Here, the unknown cache prefetching implies that the prefetched side information is unknown to the DBs. In the PIRC, the user leverages prefetched side information at in the cache to enhance the performance (i.e., the normalized download cost) of PIR. Unlike the basic PIR, the PIRC considers two phases: cache prefetching and information retrieval. Thus, its performance depends on both the cache prefetching and information retrieval phases. Note that the desired file of the user is chosen after the cache prefetching, and thus the cache prefetching phase is independent of the information retrieval phase. Most of the studies in the literature assume that the prefetched side information is uncoded, however, here we propose a RLC (coded) cache prefetching strategy that are effective in minimizing normalized download cost of PIRC problem. Furthermore, we study the optimality of the proposed cache prefetching at cache-to-file size ratio $z = \sum_{k=1}^{K} \frac{1}{(N+1)^k} = \varphi_{N+1}$ with RLC information retrieval, and obtain the corresponding optimal normalized download cost. For the rest of regions of z, we borrow an uncoded approach at $z = \frac{K}{N+1}$ from [22] and apply memory and time sharing with the proposed cache prefetching to obtain the best achievable normalized download cost bound of PIRC that are known in the literature until now.

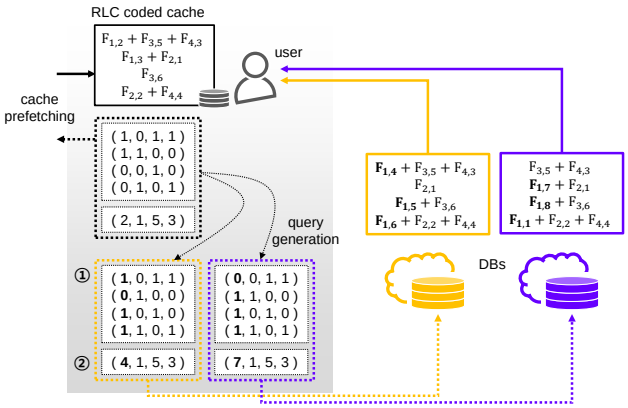


Figure 4: An illustration of sketch of RLC-based PIRC with $N = 2$ DBs, $K = 4$ files and file size $M_f = 8$ bits. (see Section IV-A).

A. RLC-based PIRC

First, we propose RLC-based PIRC, which denotes RLC-based PIR with *RLC cache prefetching*, which stores random linear combinations of file bits as side information and leverages them to retrieve information privately.

1) Sketch of RLC-based PIRC: In the following, a sketch of RLC information retrieval with RLC cache prefetching is provided (also see Fig. 4 which illustrates the sketch).

Cache prefetching. Similar to the query generation for retrieving answering strings, come up with a query for the RLC cache prefetching. The user first generates M_z (e.g., in Fig. 4, $M_z = 4$) random binary sequences of length K as cache prefetching subqueries, similar to the first step of query generation in Section III-B. Moreover, the cache prefetching IBI of each file is uniformly and independently chosen within $[1, M_f]$. Then, the user caches coded information produced based on the generated subqueries and the IBIs. Note that the cache prefetching phase is independent of user's file requesting and retrieving.

Information retrieval. To retrieve the desired file $\mathbf{F}_\theta$, information retrieving queries are generated by following the two steps below:

① (*Subquery Generation*) The user makes N copies of the cache prefetching subqueries except for the θ^{th} entry in each sequence. If an original subquery for the cache

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

Table II: Summary of the proposed achievable normalized download cost bound in Theorem 3.

Cache-to-File Size Ratio z	**Achievable Bound** D	**Achievable Scheme**
0	$N\varphi_N$	Theorem 1
$(0, \varphi_{N+1})$	$zN + \left(1 - \frac{z}{\varphi_{N+1}}\right) N\varphi_N$	Time sharing Theorems 1 and 2
φ_{N+1}	$N\varphi_{N+1}$	Theorem 2
$(\varphi_{N+1}, \frac{K}{N+1})$	$\frac{N-N(N+1)\varphi_{N+1}}{K-(N+1)\varphi_{N+1}}\left(z - \frac{K}{N+1}\right) + \frac{N}{N+1}$	Memory sharing Theorem 2 and Lemma 1
$\frac{K}{N9+1}$	$\frac{N}{N+1}$	Lemma 1
$(\frac{K}{N+1}, K)$	$1 - \frac{z}{K}$	Memory sharing Lemma 1 and full caching

prefetching has '1' for the θ^{th} entry, among N copies of the sequence without the θ^{th} entries, uniformly choose one copy and attach '0' as the θ^{th} entry of the sequence, and for the other $N-1$ copies, attach '1' as their θ^{th} entries. If an original subquery for the cache prefetching has '0' for the θ^{th} entry, attach '1' to all N copies as their θ^{th} entries.

② *(IBI Selection)* In the query transmission to DB_n, for $n \in \{1, \ldots, N\}$, IBI of the desired file is chosen by adding $(n-1)\left\lfloor \frac{M_f}{N+1} \right\rfloor$ to the IBI of the desired file in the query for cache prefetching (if the chosen IBI is larger than M_f, subtract M_f from the IBI). The IBIs of the undesired files are chosen by the same IBIs in the query for cache prefetching, for all the other DBs.

For the rest of the process, including querying, answering string generation and retrieval follows the similar procedures as done in the basic PIR.

2) Optimal PIRC at $z = \varphi_{N+1}$: The proposed RLC-based PIRC achieves the optimal normalized download cost of PIRC when $z = \varphi_{N+1}$.

Theorem 2. (Optimal PIRC at $z = \varphi_{N+1}$) *Given N DBs storing an identical file set composed of K equal-sized files, the optimal normalized download cost of the PIRC at $z = \varphi_{N+1}$ is*

$$D^* = N\varphi_{N+1}, \tag{14}$$

where $\varphi_n = \sum_{k=1}^{K} n^{-k}$.

Proof: The proof of optimal normalized download cost achievability is provided in Appendix C and the converse proof is provided in Appendix D. ■

Remark 5. *By planned designing of the subqueries, not random generation, the file size $M_f = (N+1)^K$ is sufficient to show the optimal normalized download cost is achievable with satisfying the privacy condition.*

B. Achievable Normalized Download Cost Bound of PIRC

For region $0 \le z \le K$, we provide the following achievable normalized download cost of the PIRC problem.

Theorem 3. *For the fixed number of DBs N and files K, the normalized download cost of the PIRC*

$$D = \begin{cases} zN + \left(1 - \frac{z}{\varphi_{N+1}}\right) N\varphi_N & 0 \le z < \varphi_{N+1} \\ \frac{N-N(N+1)\varphi_{N+1}}{K-(N+1)\varphi_{N+1}}\left(z - \frac{K}{N+1}\right) + \frac{N}{N+1} & \varphi_{N+1} \le z < \frac{K}{N+1} \\ 1 - \frac{z}{K} & \frac{K}{N+1} \le z \le K \end{cases} \tag{15}$$

is achievable for $M_f \to \infty$, where $\varphi_n = \sum_{k=1}^{K} n^{-k}$.

Proof: For the cases $z = 0$ and $z = \varphi_{N+1}$, the results are provided in Theorems 1 (inverse of the PIR capacity) and 2, respectively. For $z = \frac{K}{N+1}$, the proposed scheme in [22] achieves the optimal normalized download cost of PIRC when $z = \frac{K}{N+1}$ and the prefetched side information is limited to be uncoded.

Lemma 1. (Achievable PIRC at $z = \frac{K}{N+1}$ [22]) *Given N DBs storing an identical file set composed of K equal-sized files, the normalized download cost of the PIRC at $z = \frac{K}{N+1}$*

$$D = \frac{N}{N+1} \tag{16}$$

is achievable.

For the region $0 \le z < \varphi_{N+1}$, the achievable normalized download cost obtained by time sharing PIR with unknown RLC cache prefetching (result in Theorem 2) and RLC-based basic PIR (result in Theorem 1), which is leveraging the entire cache memory for PIR with unknown RLC cache prefetching to retrieve the desired file bits at optimal normalized download cost and using RLC-based basic PIR for retrieving the unreceived desired file bits. For the region $\frac{K}{N+1} \le z \le K$, the achievable normalized download cost is obtained by memory sharing PIR with unknown and uncoded cache prefetching (result in Lemma 1) and full caching which stores everything in advance. For the region $\varphi_{N+1} \le z \le \frac{K}{N+1}$, the achievable normalized download cost is obtained by memory sharing PIR with unknown RLC cache prefetching and PIR with unknown and uncoded cache prefetching. Note that from Remark 1, the concatenated subqueries of perfectly private subqueries from two different schemes leveraging time sharing or memory sharing still holds the perfect privacy condition. ■

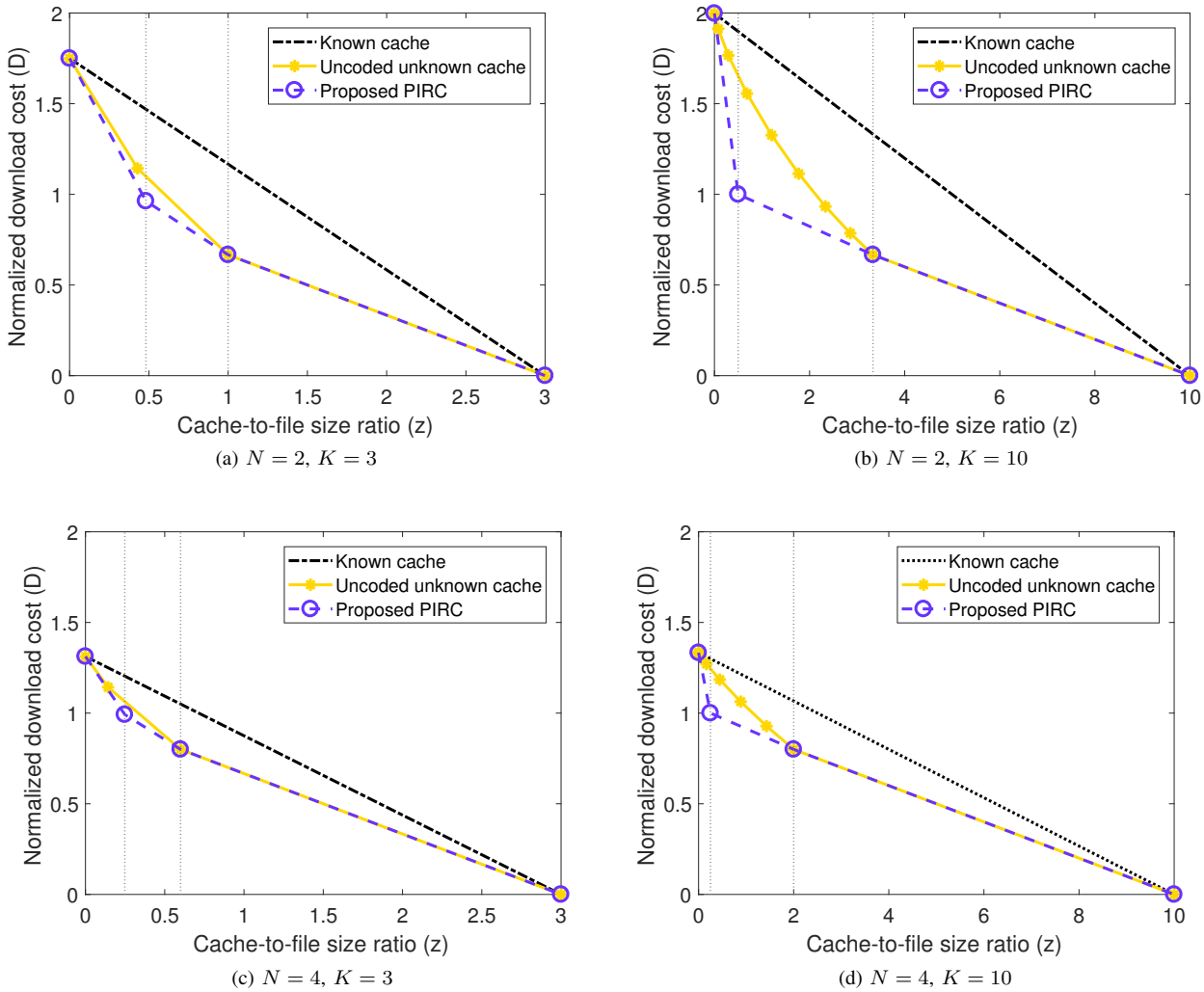
Remark 6. *The proposed achievable normalized download cost bound* (15) *always outperforms the known best achievable bound found in [22] with unknown and uncoded cache prefetching.*

Refer to Fig. 5, in which the yellow solid lines present the mentioned existing achievable bound presented in [22] and the blue dashed lines present the proposed bound (15). The proof of Remark 6 is provided in Appendix E. The summary of the proposed achievable normalized download cost bound is provided in Table II.

V. Case Study

Here we compare the proposed PIRC, based on Theorem 3, with two existing benchmark scheme. First one is the known

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications



(a) $N = 2$, $K = 3$ (b) $N = 2$, $K = 10$

(c) $N = 4$, $K = 3$ (d) $N = 4$, $K = 10$

Figure 5: Illustrations of PIR with known cache [20], PIR with uncoded unknown cache [22] and the proposed PIRC for three different cases: (a) 2 DBs and 3 files, (b) 2 DBs and 10 files, (c) 4 DBs and 3 files, (d) 4 DBs and 10 files.

cache prefetching [20], where the user caches side information which is known by the DBs. For this case, since the DBs are aware of the cached information, the user is unable to use the cached information to cancel out undesired information which might result in privacy leakage. Consequently, the known cache prefetching is a memory sharing approach with the basic PIR and full cache prefetching. The second one is the uncoded unknown cache prefetching [22], where user caches a subset of the uncoded bits from each files and the cached information is unknown to the DBs. Since the user can use the cached information to cancel out undesired information, the performance of unknown cache prefetching is strictly better than the known prefetching.

For the case study, we compare the proposed scheme and the above-mentioned two benchmarks in three specific cases, (a) $N = 2$, $K = 3$, (b) $N = 2$, $K = 10$, (c) $N = 4$, $K = 3$ and (d) $N = 4$, $K = 10$ in Fig.5. For each case, the proposed PIRC scheme provides lower normalized download cost than the others for every cache-to-file ratio range. Two vertical lines in each figure represent the $z = \varphi_{N+1}$ and $z = \frac{K}{N+1}$ cases, respectively. The effectiveness of the proposed scheme is significant for larger K with small cache-to-file size ratio, since if the DBs have many files while the user has limited size of cache, the RLC coded cache prefetching will be more efficient, compared to the uniform uncoded cache prefetching. On the other hand, if the cache-to-file size ratio is high, it is better to cache bits with uniform uncoded cache prefetching to get advantage in the information retrieval phase.

VI. Conclusion

After [7] initiated the study of revealing the information theoretical capacity of the PIR problem, many variants of the PIR problem were studied. Especially, the case when the user is equipped with cache memory which can store some useful information is quite interesting, since the cached information can enhance the rate performance of the PIR problem. Our study mainly focused on obtaining the fundamental limits of the PIRC via adaptive leveraging of the RLC coded and uniform uncoded cache prefetching. To this end, we first proposed a novel RLC based PIR scheme for the basic PIR

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

problem, where the randomly generated queries can guarantee the privacy while achieving the capacity of the basic PIR. Then, using similar approach, we propose optimal cache prefetching strategy and obtain optimal normalized download costs of the PIRC in two specific cache-to-file size ratio cases. Moreover, we leveraged time and memory sharing approach to obtain the fundamental limits of the PIRC, which turn out to outperform the existing cache aided scheme known in the literature.

APPENDIX A
MATHEMATICAL DETAILS OF RLC-BASED PIR SCHEME

Without loss of generality, we suppose that $\mathbf{F}_\theta$ is the desired file. For the case when there are only a single DB in the network, i.e., $N = 1$, the user can achieve the basic PIR rate $R = \frac{1}{K}$, by retrieving all the stored files from the DB. Hence we hereafter focus on the case when $N > 1$.

Let $M_{s,n}^{(\theta)} = \left\lfloor \frac{M_f}{N-1} \right\rfloor$ for all $n \in \{1, \ldots, N\}$ and fix $z = 0$, since no cache memory is equipped at the user in the basic PIR problem.

(Subquery Generation) First, consider $M = \left\lfloor \frac{M_f}{N-1} \right\rfloor$ binary sequences of length $K - 1$

$$(S_{m,1}, \ldots, S_{m,\theta-1}, S_{m,\theta+1}, \ldots, S_{m,K}), \tag{17}$$

for all $m \in \{1, \ldots, M\}$, of which the elements are generated independently according to an identical Bernoulli distribution

$$p_S(s) = \begin{cases} 1 - \frac{1}{N} & \text{if } s = 1, \\ \frac{1}{N} & \text{if } s = 0. \end{cases} \tag{18}$$

The generated sequences are the entries for the undesired files in the subqueries. i.e.,

$$S_{n,m,k}^{(\theta)} = S_{m,k}, \tag{19}$$

for all $n \in \{1, \ldots, N\}$, $m \in \{1, \ldots, M\}$ and $k \in \{1, \ldots, K\} \backslash \{\theta\}$. The subquery entries for the desired file are generated as follows. For each $m \in \{1, \ldots, M\}$, independently and uniformly choose $\tilde{n}_m$ among $\{1, \ldots, N\}$ and let $S_{1,\tilde{n}_m,\theta}^{(\theta)} = 0$, and $S_{1,n,\theta}^{(\theta)} = 1$ for all $n \in \{1, \ldots, N\} \backslash \{\tilde{n}_m\}$. Then, the entries for the desired file in the subqueries for a single DB can be seen as generated also following the Bernoulli distribution (18) like the entries for the undesired files. The following lemma states the subqueries for a DB generated as above is ϵ-private for any $\epsilon \in (0, 1)$ if the file size approaches infinity $M_f \to \infty$ (or $M \to \infty$ accordingly).

Lemma 2. *For fixed N, the generated subqueries $\mathbf{S}_n^{(\theta)} = \left(\mathbf{S}_{n,1}^{(\theta)}, \ldots, \mathbf{S}_{n,M}^{(\theta)}\right)$ is in the ϵ-private set in probability, i.e.,*

$$\lim_{M \to \infty} \Pr\left(\mathbf{S}_n^{(\theta)} \in \mathcal{P}_\epsilon^{(M)}(S)\right) = 1, \tag{20}$$

for any $\epsilon \in (0, 1)$ and for all $n \in \{1, \ldots, N\}$, where S is the Bernoulli random variable distributed according to (18)*.*

Proof: Since the entries of subqueries $\mathbf{S}_n^{(\theta)}$ are generated following the Bernoulli distribution (18) for all $n \in \{1, \ldots, N\}$, from the law of large numbers and property of a typical set, (20) holds for any $\epsilon \in (0, 1)$ as $M_f \to \infty$. ∎

Notice that there might exist sequences composed of only zero elements with probability $(\frac{1}{N})^K$. In such cases, the sequences do not contain any information. Thus, for DB_n, the number of generated effective subqueries $M_{a,n}^{(\theta)}$ is equal to the number of non-all-zero sequences in $\mathbf{S}_n^{(\theta)}$.

(IBI generation) For DB_1, generate IBIs

$$\mathbf{B}_1^{(\theta)} = (B_{1,1}^{(\theta)}, \ldots, B_{1,K}^{(\theta)}), \tag{21}$$

where each element is uniformly and independently drawn over $\{1, \ldots, M_f\}$. For DB_n, $n \in \{2, \ldots, N\}$, generate IBIs

$$\mathbf{B}_n^{(\theta)} = (B_{n,1}^{(\theta)}, \ldots, B_{n,K}^{(\theta)}), \tag{22}$$

where $B_{n,\theta}^{(\theta)} = (B_{1,\theta}^{(\theta)} + \sum_{i=1}^{n-1} |\Gamma_i^{(\theta)}|) \bmod K$; for a set defined as $\Gamma_i^{(\theta)} = \{m : S_{i,m,\theta}^{(\theta)} = 1, \forall m \in \{1, \ldots, M\}\}$, and $B_{n,k}^{(\theta)} = B_{1,k}^{(\theta)}$ for all $k \in \{1, \ldots, K\} \backslash \{\theta\}$.

(Querying and Answering String generation) The user transmits the query $\mathbf{Q}_n^{(\theta)} = (\mathbf{S}_n^{(\theta)}, \mathbf{B}_n^{(\theta)})$ to DB_n for all $n \in \{1, \ldots, N\}$. Upon receiving the query, DB_n generates answering string, $\mathbf{A}_n^{(\theta)} = (A_{n,1}^{(\theta)}, \ldots, A_{n,M_{a,n}^{(\theta)}}^{(\theta)})$, as explained in (12), by referring to the information obtained from the query.

(Decoding) Let $\gamma_{i,1}, \ldots, \gamma_{i,|\Gamma_i^{(\theta)}|}$ be the distinct elements of the set $\Gamma_i^{(\theta)}$ arranged in ascending order. For some i such that $m \in \{B_{i,\theta}^{(\theta)}, \ldots, B_{i,\theta}^{(\theta)} + |\Gamma_i^{(\theta)}| - 1\}$ and j such that $S_{j,m-B_{i,\theta}^{(\theta)}+1,1} = 0$, the decoded and estimated m^{th} file bit of the desired file is

$$\hat{F}_{\theta,m} = A_{i,m-B_{i,\theta}^{(\theta)}+1} + A_{j,m-B_{i,\theta}^{(\theta)}+1}, \tag{23}$$

for all $m \in \{1, \ldots, M_f\}$.

(Failure analysis and PIR rate) When decoding, a failure of PIR occurs if and only if one or both of the following events occur:

$$\mathcal{E}_1 \triangleq \hat{\mathbf{F}}_\theta \neq \mathbf{F}_\theta, \tag{24}$$

$$\mathcal{E}_2 \triangleq \text{Query is non-private.} \tag{25}$$

Define $\mathcal{E}$ as the event of the failure occurrence, then the total probability of the failure occurrence can be upper bounded as

$$\Pr(\mathcal{E}) \leq \Pr(\mathcal{E}_1) + \Pr(\mathcal{E}_2). \tag{26}$$

On one hand, since we consider a noiseless model, and as long as the queries are generated correctly, we have

$$\Pr(\mathcal{E}_1) = \Pr(\hat{\mathbf{F}}_\theta \neq \mathbf{F}_\theta) = 0. \tag{27}$$

On the other hand, from Lemma 2, we also showed that the subqueries are private in probability as

$$\lim_{M_f \to \infty} \Pr(\mathcal{E}_2) \leq \lim_{M \to \infty} \sum_{n=1}^{N} \Pr(\mathbf{S}_n^{(\theta)} \notin \mathcal{P}_\epsilon^{(M)}(S)) = 0. \tag{28}$$

As a result, we have $\Pr(\mathcal{E}) \to 0$ as $M \to \infty$. The total number of answering bits received from the DBs is $M_a^{(\theta)}$ and the file size is M_f. Therefore, we can achieve the rate $R_\theta = M_f / M_a^{(\theta)}$ for retrieving $\mathbf{F}_\theta$ privately. Since the desired file is chosen uniformly over the file index set $\{1, \ldots, K\}$, we have the PIR rate

$$R = \frac{1}{K} \sum_{\theta=1}^{K} R_\theta = \frac{1}{K} \sum_{\theta=1}^{K} \frac{M_f}{M_a^{(\theta)}}. \tag{29}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

APPENDIX B
PROOF OF CAPACITY ACHIEVABILITY WITH RLC-BASED PIR SCHEME

For $N = 1$, we mentioned that $\frac{1}{K}$ is achievable. For $N > 1$, from the fact that the the number of answering bits downloading from DB_n is the number of the effective number of subqueries, we have

$$M_{a,n}^{(\theta)} = M_{s,n}^{(\theta)} - M_{s,n}^{(\theta)}\pi(0,\ldots,0\,|\,\mathbf{S}_n^{(\theta)}) \tag{30}$$

$$= \left\lfloor \frac{M_f}{N-1} \right\rfloor \left(1 - \pi(0,\ldots,0\,|\,\mathbf{S}^{(\theta)})\right). \tag{31}$$

By substituting (31) into the basic PIR rate expression in (29), we have

$$R = \frac{1}{K}\sum_{\theta=1}^{K} \frac{M_f}{M_a^{(\theta)}} \tag{32}$$

$$= \frac{1}{K}\sum_{\theta=1}^{K} \frac{M_f}{\sum_{n=1}^{N} M_{a,n}^{(\theta)}} \tag{33}$$

$$= \frac{1}{K}\sum_{\theta=1}^{K} \frac{M_f}{\left\lfloor \frac{M_f}{N-1} \right\rfloor \sum_{n=1}^{N}\left(1 - \pi(0,\ldots,0\,|\,\mathbf{S}^{(\theta)})\right)}. \tag{34}$$

Then, for some $\epsilon_n > 0$ satisfying

$$\left| \pi(0,\ldots,0\,|\,\mathbf{S}_n^{(\theta)}) - \frac{1}{N^K} \right| \le \epsilon_n \frac{1}{N^K}, \tag{35}$$

we can bound (34) as,

$$\frac{N-1}{N(1-(1-\sum_{n=1}^{N}\epsilon_n)\frac{1}{N^K})} \le R \le \frac{N-1}{N(1-(1+\sum_{n=1}^{N}\epsilon_n)\frac{1}{N^K})}. \tag{36}$$

As shown above, note that the system failure probability $\Pr(\mathcal{E}) \to 0$ as $M_f \to \infty$. Hence, by taking $\epsilon_1,\ldots,\epsilon_N \to 0$, the PIR rate

$$R = \frac{N-1}{N - (\frac{1}{N})^{K-1}} = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}}\right)^{-1} \tag{37}$$

is achievable.

APPENDIX C
ACHIEVABILITY OF THEOREM 2

The optimal PIRC scheme for the cache-to-file size ratio $z = \varphi_{N+1}$ is explained in the following. For the information retrieval phase, we omit the descriptions about querying, answering string generation and decoding, since it is very similar to the ones described in Appendix A.

Cache prefetching. As mentioned in the sketch, in the cache prefetching phase, we suppose that the cache data is prefetched based on the generated cache prefetching query.

(Cache subquery generation) Consider $M' = \lfloor \frac{M_f}{N} \rfloor$ binary sequences of length K, i.e., $\mathbf{S}_{z,m} = (S_{z,m,1},\ldots,S_{z,m,K})$ for all $m \in \{1,\ldots,M'\}$, where all entries are generated independently and identically according to the Bernoulli distribution

$$p_{S_z}(s) = \begin{cases} 1 - \dfrac{1}{N+1}, & \text{when } s = 1, \\ \dfrac{1}{N+1}, & \text{when } s = 0. \end{cases} \tag{38}$$

Since the sequences are generated randomly, there exist some sequences of which entries are all zeros. By discarding such all-zero sequences, we get the subqueries for the cache prefetching query. Note that the number of subqueries is

$$M_z = M'(1 - \pi(0,\ldots,0\,|\,\mathbf{S}_{z,1},\ldots,\mathbf{S}_{z,M'})), \tag{39}$$

and for sufficiently large M_f, we have $z = \varphi_{N+1}$.

(Cache IBI generation) The IBIs $\mathbf{B}_z = (B_{z,1},\ldots,B_{z,K})$ for the cache prefetching is chosen independently and uniformly over $\{1,\ldots,M_f\}^K$.

(Cache prefetching) Then, the m^{th} prefetched cache bit at the user can be expressed as

$$Z_m = \sum_{\forall k \in \mathcal{A}_{z,m}} \mathbf{next}\left(\mathbf{F}_k, B_{z,k}\right), \tag{40}$$

where $\mathcal{A}_{z,m} = \{k : S_{z,m,k} = 1, \forall k \in \{1,\ldots,K\}\}$. Note that the cache prefetching is done before the decision of the desired file index, and thus the prefetching information is independent of the file desired by the user.

Information retrieval. Now the information retrieval of the desired file $\mathbf{F}_\theta$ at the user is described.

(Subquery generation) For DB_n, for all $n \in \{1,\ldots,N\}$, the user generates binary sequences of length K, i.e., $\mathbf{S}_{n,m}^{(\theta)} = (S_{n,m,1}^{(\theta)},\ldots,S_{n,m,K}^{(\theta)})$, for all $m \in \{1,\ldots,M'\}$, where

$$S_{n,\gamma_{z,i},\theta}^{(\theta)} = \begin{cases} 0, & i \in \{(n-1)\lfloor \frac{|\Gamma_z'|}{N} \rfloor + 1,\ldots,n\lfloor \frac{|\Gamma_z'|}{N} \rfloor\} \\ 1, & \text{elsewhere,} \end{cases} \tag{41}$$

where $\Gamma_z' = \{m : S_{z,m,\theta} = 1,\ \forall m \in \{1,\ldots,M'\}\}$ and $\gamma_{z,1}',\ldots,\gamma_{z,|\Gamma_z'|}'$ are the distinct elements of the set Γ_z' organized in ascending order, while the rest of the entries are

$$S_{n,m,k}^{(\theta)} = S_{z,m,k},\ \forall k \in \{1,\ldots,K\}\backslash\{\theta\}. \tag{42}$$

Note if there are any sequences that have all-zero entries, then we discard them. The rest of the sequences that are not all-zero sequences are considered as subqueries. Similar to Lemma 2 in Appendix A, the generated subquery satisfies the privacy condition.

(IBI generation) For DB_n, generate IBIs

$$\mathbf{B}_n^{(\theta)} = (B_{n,1}^{(\theta)},\ldots,B_{n,K}^{(\theta)}), \tag{43}$$

where $B_{n,\theta}^{(\theta)} = (B_{z,\theta} + |\Gamma_z'| + \sum_{i=1}^{n-1}|\Gamma_i'|) \bmod K$; for a set defined as $\Gamma_i' = \{m : S_{i,m,\theta}^{(\theta)} = 1, \forall m \in \{1,\ldots,M'\}\}$, and $B_{n,k}^{(\theta)} = B_{z,k}$ for all $k \in \{1,\ldots,K\}\backslash\{\theta\}$.

The total number of answering bits received from the DBs is $M_a^{(\theta)}$ and the file size is M_f. Since the desired file is chosen uniformly over the file index set $\{1,\ldots,K\}$, we have the normalized download cost of PIRC

$$D = \frac{1}{K}\sum_{\theta=1}^{K} D^{(\theta)} = \frac{1}{K}\sum_{\theta=1}^{K} \frac{M_a^{(\theta)}}{M_f}. \tag{44}$$

From the fact that the the number of answering bits downloading from DB_n is the number of the effective number of subqueries, we have

$$M_{a,n}^{(\theta)} = M'(1 - \pi(0,\ldots,0\,|\,\mathbf{S}_{n,1}^{(\theta)},\ldots,\mathbf{S}_{n,M'}^{(\theta)})). \tag{45}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3117936, IEEE Transactions on Communications

By substituting (45) into the PIRC rate expression in (44), we have

$$D = \frac{1}{K}\sum_{\theta=1}^{K}\frac{M_a^{(\theta)}}{M_f} \tag{46}$$

$$= \frac{1}{K}\sum_{\theta=1}^{K}\frac{\sum_{n=1}^{N} M_{a,n}^{(\theta)}}{M_f} \tag{47}$$

$$= \frac{1}{K}\sum_{\theta=1}^{K}\frac{M'\sum_{n=1}^{N}(1-\pi(0,\dots,0\,|\,\mathbf{S}_{n,1}^{(\theta)},\dots,\mathbf{S}_{n,M'}^{(\theta)}))}{M_f}. \tag{48}$$

For some $\epsilon_n' > 0$ satisfying

$$\left|\pi(0,\dots,0\,|\,\mathbf{S}_{n,1}^{(\theta)},\dots,\mathbf{S}_{n,M'}^{(\theta)}) - \frac{1}{(N+1)^K}\right| \le \epsilon_n'\frac{1}{(N+1)^K}, \tag{49}$$

we can bound (48),

$$\begin{aligned} N\frac{M'(1-(1-\sum_{n=1}^{N}\epsilon_n')\frac{1}{(N+1)^K})}{M_f} &\le D \\ &\le N\frac{M'(1-(1+\sum_{n=1}^{N}\epsilon_n')\frac{1}{(N+1)^K})}{M_f}. \end{aligned} \tag{50}$$

Similar to the case of the basic PIR in the Appendix A, the system failure probability $\Pr(\mathcal{E}) \to 0$ as $M_f \to \infty$. Hence, by taking $\epsilon_1', \dots, \epsilon_N' \to 0$, the average normalized download cost

$$D = N\varphi_{N+1} \tag{51}$$

is achievable, for $M_f \to \infty$.

Appendix D
Converse of Theorem 2

Now we derive the information theoretic upper bound on the capacity of the PIRC. We first start with proving Theorem 2 for the case when $z = \varphi_{N+1}$. The proof can be made with the following lemmas.

Lemma 3. *For the desired file index θ, the following inequality holds for all $\theta \in \{1,\dots,K\}$.*

$$I(\mathbb{F}_{\mathcal{K}\setminus\theta}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z} \mid \mathbf{F}_\theta) \le M_a^{(\theta)} + M_z - M_f(1-\epsilon_{M_f}), \tag{52}$$

where $\mathbb{F}_{\mathcal{K}\setminus\theta} = (\mathbf{F}_1,\dots,\mathbf{F}_{\theta-1},\mathbf{F}_{\theta+1},\dots,\mathbf{F}_K)$, $\mathbb{Q}_{\mathcal{N}}^{(\theta)} = (\mathbf{Q}_1^{(\theta)},\dots,\mathbf{Q}_N^{(\theta)})$ and $\mathbb{A}_{\mathcal{N}}^{(\theta)} = (\mathbf{A}_1^{(\theta)},\dots,\mathbf{A}_N^{(\theta)})$.

Proof: For simple notation, we let $\mathcal{I} = I(\mathbb{F}_{\mathcal{K}\setminus\theta}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z} \mid \mathbf{F}_\theta)$ and $\mathbb{F}_{\mathcal{K}} = (\mathbf{F}_1,\dots,\mathbf{F}_K)$. We have

$$M_f = H(\mathbf{F}_\theta) \tag{53}$$

$$= I(\mathbf{F}_\theta; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}) + H(\mathbf{F}_\theta\,|\,\mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}) \tag{54}$$

$$\le I(\mathbf{F}_\theta; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}) + M_f\epsilon_{M_f} \tag{55}$$

$$= I(\mathbb{F}_{\mathcal{K}}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}) - \mathcal{I} + M_f\epsilon_{M_f} \tag{56}$$

$$= I(\mathbb{F}_{\mathcal{K}}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}) - \mathcal{I} + M_f\epsilon_{M_f} + I(\mathbb{F}_{\mathcal{K}}; \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}\,|\,\mathbb{Q}_{\mathcal{N}}^{(\theta)}) \tag{57}$$

$$= H(\mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}\,|\,\mathbb{Q}_{\mathcal{N}}^{(\theta)}) - \mathcal{I} + M_f\epsilon_{M_f} \tag{58}$$

$$= H(\mathbb{A}_{\mathcal{N}}^{(\theta)}\,|\,\mathbb{Q}_{\mathcal{N}}^{(\theta)}) + H(\mathbf{Z}\,|\,\mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbb{Q}_{\mathcal{N}}^{(\theta)}) - \mathcal{I} + M_f\epsilon_{M_f} \tag{59}$$

$$\le M_a^{(\theta)} + M_z - \mathcal{I} + M_f\epsilon_{M_f}, \tag{60}$$

where (55) holds by Fano's inequality and the data processing inequality, (58) holds since the queries and the cache prefetching strategy are independent of the files, and (60) holds since the condition reduces entropy, and $H(\mathbb{A}_{\mathcal{N}}^{(\theta)}) = M_a^{(\theta)}$ and $H(\mathbf{Z}) = M_z$. ∎

Lemma 4. *Consider subsets $\mathcal{K}_1 \subset \{1\dots,K\}$ where $\theta \in \mathcal{K}_1$ and $\mathcal{K}_2 = \{1\dots,K\}\setminus\mathcal{K}_1$. Then, the following inequality holds for any $\phi \in \mathcal{K}_2$.*

$$\begin{aligned}&(N+1)I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \\ &\ge I(\mathbb{F}_{\mathcal{K}_2\setminus\{\phi\}}; \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1\cup\{\phi\}}) + M_f\left(1-\epsilon_{M_f}\right).\end{aligned} \tag{61}$$

Proof:

$$(N+1)I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{Q}_{\mathcal{N}}^{(\theta)}, \mathbb{A}_{\mathcal{N}}^{(\theta)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{62}$$

$$\ge \sum_{n=1}^{N} I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Q}_n^{(\theta)}, \mathbf{A}_n^{(\theta)}\,|\,\mathbb{F}_{\mathcal{K}_1}) + I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{63}$$

$$= \sum_{n=1}^{N} I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Q}_n^{(\phi)}, \mathbf{A}_n^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}) + I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{64}$$

$$= \sum_{n=1}^{N} I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{A}_n^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbf{Q}_n^{(\phi)}) + I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{65}$$

$$= \sum_{n=1}^{N} H(\mathbf{A}_n^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbf{Q}_n^{(\phi)}) + H(\mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{66}$$

$$\begin{aligned}&\ge \sum_{n=1}^{N} H(\mathbf{A}_n^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbf{A}_1^{(\phi)},\dots,\mathbf{A}_{n-1}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}) \\ &\quad + H(\mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbb{Q}_{\mathcal{N}}^{(\phi)})\end{aligned} \tag{67}$$

$$\begin{aligned}&= \sum_{n=1}^{N} I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{A}_n^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbf{A}_1^{(\phi)},\dots,\mathbf{A}_{n-1}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}) \\ &\quad + I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbb{Q}_{\mathcal{N}}^{(\phi)})\end{aligned} \tag{68}$$

$$= I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}) \tag{69}$$

$$= I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) - I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{Q}_{\mathcal{N}}^{(\phi)}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{70}$$

$$= I(\mathbb{F}_{\mathcal{K}_2}; \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \tag{71}$$

$$\begin{aligned}&= I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{F}_\phi, \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) \\ &\quad - I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{F}_\phi\,|\,\mathbb{F}_{\mathcal{K}_1}, \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z})\end{aligned} \tag{72}$$

$$\ge I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{F}_\phi, \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1}) - M_f\epsilon_{M_f} \tag{73}$$

$$\begin{aligned}&= I(\mathbb{F}_{\mathcal{K}_2}; \mathbf{F}_\phi\,|\,\mathbb{F}_{\mathcal{K}_1}) \\ &\quad + I(\mathbb{F}_{\mathcal{K}_2\setminus\{\phi\}}; \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1\cup\{\phi\}}) - M_f\epsilon_{M_f}\end{aligned} \tag{74}$$

$$= M_f + I(\mathbb{F}_{\mathcal{K}_2\setminus\{\phi\}}; \mathbb{A}_{\mathcal{N}}^{(\phi)}, \mathbb{Q}_{\mathcal{N}}^{(\phi)}, \mathbf{Z}\,|\,\mathbb{F}_{\mathcal{K}_1\cup\{\phi\}}) - M_f\epsilon_{M_f}, \tag{75}$$

where (63) comes from the chain rule and non-negativity of the mutual information, (64) is due to the privacy condition,

and (73) holds from Fano's inequality and the data processing inequality similar to (55). ■

Lemma 5. *The following inequality holds for all* $\theta \in \{1,\dots,K\}$.

$$I(\mathbb{F}_{\mathcal{K}\setminus\theta};\mathbb{Q}_{\mathcal{N}}^{(\theta)},\mathbb{A}_{\mathcal{N}}^{(\theta)},\mathbf{Z}\,|\,\mathbf{F}_\theta) \geq M_f(1-\epsilon_{M_f})\left(\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^{K-1}}\right). \tag{76}$$

Proof: Suppose $\theta=1$. From Lemma 4, we can derive

$$I(\mathbb{F}_{\mathcal{K}\setminus\{1\}};\mathbb{Q}_{\mathcal{N}}^{(1)},\mathbb{A}_{\mathcal{N}}^{(1)},\mathbf{Z}\,|\,\mathbf{F}_1) \geq \frac{M_f(1-\epsilon_{M_f})+I(\mathbb{F}_{\mathcal{K}\setminus\{1,2\}};\mathbb{Q}_{\mathcal{N}}^{(2)},\mathbb{A}_{\mathcal{N}}^{(2)},\mathbf{Z}\,|\,\mathbf{F}_1,\mathbf{F}_2)}{N+1} \tag{77}$$

$$\geq \cdots \geq M_f(1-\epsilon_{M_f})\left(\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^{K-1}}\right). \tag{78}$$

Similarly, we can show that (76) holds for $\theta\in\{2\dots,K\}$. ■

From Lemma 3 and Lemma 5, we have

$$M_a^{(\theta)}+M_z-M_f(1-\epsilon_{M_f}) \geq M_f(1-\epsilon_{M_f})\left(\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^{K-1}}\right). \tag{79}$$

By dividing both sides with M_f, we have

$$\frac{M_a^{(\theta)}}{M_f}+\frac{M_z}{M_f}\geq(1-\epsilon_{M_f})\left(1+\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^{K-1}}\right). \tag{80}$$

From the assumption that $\frac{M_z}{M_f}=\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^K}$ and by averaging over $\theta\in\{1,\dots,K\}$, we have the inequality

$$D\geq N\left(\frac{1}{N+1}+\cdots+\frac{1}{(N+1)^K}\right) \tag{81}$$

for $M_f\to\infty$. This ends the converse proof of Theorem 2 when $z=\varphi_{N+1}$.

Appendix E
Proof of Remark 6

The achievable normalized download cost at $z=0$ is equal for the both scheme, which comes from the inverse of the capacity of the basic PIR obtained in Theorem 1. However, the slope of each scheme is different for the region $0<z\leq\varphi_{N+1}$. In [22], the achievable normalized download cost is derived in a discrete manner as shown in the yellow solid curve shown in Section V, where the points on the line segment can be achieved through memory sharing. Therefore, the overall achievable normalized download cost bound in [22] becomes a convex function. Therefore, showing that the slope of the leftmost line segment of the existing bound in [22] is greater (less steeper) than our proposed bound segment within $0\leq z\leq\varphi_{N+1}$ is enough to prove that the proposed achievable bound outperforms in the region. The leftmost line segment of the achievable bound in [22] in the region $0\leq z\leq\varphi_{N+1}$ can be written as

$$-\frac{N(N^{-K}+K(N-1)-1)}{K(N-1)^2} \tag{82}$$

On the other hand, the slope of the achievable normalized download cost bound in the region can be written as

$$-\frac{N(-1+N+(1+N)^K-N^{1-K}(1+N)^K)}{(N-1)((N+1)^K-1)}. \tag{83}$$

Consider difference of (82) and (83). By simplifying it, it can be shown that the difference is positive if

$$(1+N)^K(N^K-KN^2+KN-1)+N^K(KN^2-KN+1)+1 \tag{84}$$

is positive. Note that $N,K\geq 2$ from definition. For $N,K\geq 3$, it can be easily shown that (84) is positive, since N^K-KN^2+KN-1 and KN^2-KN+1 are positive because the derivatives of them with respect to K and N are positive, and they are positive when $N=K=3$. In $K=2$ case, the equation (84) is always positive for any value of N. In the same way, in $N=2$ case, (84) is always positive for any value of $K\geq 2$. Therefore we can conclude that (84) is always positive for any value of $N,K\geq 2$. Thus, the proposed achievable normalized download cost bound outperforms the conventional within the region $0\leq z\leq\varphi_{N+1}$.

Since the function of achievable download cost is convex, to prove that our scheme is better than the existing bound obtained by [22] within $\varphi_{N+1}\leq z\leq\frac{K}{N+1}$, showing the second line segment from right side of the existing bound (yellow solid curve in Section V) is always smaller (steeper) than that of the proposed bound (blue dashed curve in Section V) within $\varphi_{N+1}\leq z\leq\frac{K}{N+1}$. The second line segment from right side of the existing bound can be expressed as

$$-\frac{1}{K}\left(\frac{1}{N}+1\right). \tag{85}$$

On the other hand, the slope of the proposed bound within $\varphi_{N+1}\leq z\leq\frac{K}{N+1}$ can be written as

$$\frac{N(N-(N+1)\varphi_{N+1})}{K-(N+1)\varphi_{N+1}}. \tag{86}$$

By simplifying the difference of (86) and (85), it can be shown that the difference is positive if

$$(1+N)^K((K-1)(N^4-N^3+N)+(N^4-N^3-N^2-N-1)) \tag{87}$$

is positive. Since For $N=K=2$, (87) is positive and the derivatives of (87) with respect to N and K are positive for $N,K\geq 2$, (87) is always positive for $N,K\geq 2$. Thus, the proposed achievable normalized download cost bound outperforms the conventional within the region $\varphi_{N+1}\leq z\leq\frac{K}{N+1}$. For the region $\frac{K}{N+1}\leq z\leq K$, the bound of the proposed and existing bound is the same. This completes the proof of Remark 6.

References

[1] H. Seo and W. Choi, "A stochastic approach in private information retrieval," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC)*, pp. 1–6, 2018.

[2] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pp. 41–50, 1995.

[3] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[4] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond, "Breaking the $\mathcal{O}(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 261–270, 2002.

[5] A. Beimel, Y. Ishai, and E. Kushilevitz, "General constructions for information-theoretic private information retrieval," *Journal of Computer and System Sciences*, vol. 71, no. 2, pp. 213–247, 2005.

[6] Z. Dvir and S. Gopi, "2-server pir with sub-polynomial communication," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 577–584, 2015.

[7] 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.

[8] H. Sun and S. A. Jafar, "The capacity of private information retrieval with colluding databases," in *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, pp. 941–946, 2016.

[9] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2361–2370, 2018.

[10] H. Sun and S. A. Jafar, "Private information retrieval from mds coded data with colluding servers: Settling a conjecture by freij-hollanti et al.," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1000–1022, 2018.

[11] Q. Wang and M. Skoglund, "Symmetric private information retrieval from mds coded distributed storage with non-colluding and colluding servers," *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 5160–5175, 2019.

[12] Q. Wang and M. Skoglund, "On pir and symmetric pir from colluding databases with adversaries and eavesdroppers," *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 3183–3197, 2019.

[13] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, and C. Hollanti, "Private information retrieval from coded storage systems with colluding, byzantine, and unresponsive servers," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3898–3906, 2019.

[14] K. Banawan and S. Ulukus, "The capacity of private information retrieval from byzantine and colluding databases," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 1206–1219, 2019.

[15] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, pp. 1–5, 2016.

[16] H. Sun and S. A. Jafar, "The capacity of symmetric private information retrieval," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 322–329, 2019.

[17] H. Seo, K. Son, S. Park, and W. Choi, "Communication-efficient private information acquisition: Multicasting via crowding," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 7, pp. 7199–7204, 2021.

[18] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[19] M. Mohammadi Amiri and D. Gündüz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity tradeoff," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 806–815, 2017.

[20] R. Tandon, "The capacity of cache aided private information retrieval," in *Proc. and Computing (Allerton) 2017 55th Annual Allerton Conf. Communication, Control*, pp. 1078–1082, 2017.

[21] S. Kadhe, B. Garcia, A. Heidarzadeh, S. E. Rouayheb, and A. Sprintson, "Private information retrieval with side information: The single server case," in *Proc. and Computing (Allerton) 2017 55th Annual Allerton Conf. Communication, Control*, pp. 1099–1106, 2017.

[22] Y. Wei, K. Banawan, and S. Ulukus, "Fundamental limits of cache-aided private information retrieval with unknown and uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 3215–3232, 2019.

[23] Y. Wei, K. Banawan, and S. Ulukus, "The capacity of private information retrieval with partially known private side information," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 8222–8231, 2019.

[24] Z. Chen, Z. Wang, and S. A. Jafar, "The capacity of t-private information retrieval with private side information," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4761–4773, 2020.

[25] X. Zhang, K. Wan, H. Sun, M. Ji, and G. Caire, "On the fundamental limits of cache-aided multiuser private information retrieval," *IEEE Transactions on Communications*, pp. 1–1, 2021.

[26] A. Heidarzadeh, F. Kazemi, and A. Sprintson, "The role of coded side information in single-server private information retrieval," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 25–44, 2021.

[27] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Private information retrieval with private coded side information: The multi-server case," in *Proc. and Computing (Allerton) 2019 57th Annual Allerton Conf. Communication, Control*, pp. 1098–1104, 2019.

[28] F. Kazemi, E. Karimi, A. Heidarzadeh, and A. Sprintson, "Multi-server private information retrieval with coded side information," in *2019 16th Canadian Workshop on Information Theory (CWIT)*, pp. 1–6, 2019.

[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[30] S. A. Jafar, "Blind interference alignment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 3, pp. 216–227, 2012.