information

Article

The Capacity of Private Information Retrieval from Decentralized Uncoded Caching Databases

Yi-Peng Wei , Batuhan Arasli, Karim Banawan and Sennur Ulukus *

Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA; ypwei@umd.edu (Y.-P.W.); barasli@umd.edu (B.A.); kbanawan@umd.edu (K.B.)
* Correspondence: ulukus@umd.edu

Received: 5 August 2019; Accepted: 12 November 2019; Published: 28 November 2019

Abstract: We consider the private information retrieval (PIR) problem from decentralized uncoded caching databases. There are two phases in our problem setting, a caching phase, and a retrieval phase. In the caching phase, a data center containing all the K files, where each file is of size L bits, and several databases with storage size constraint μKL bits exist in the system. Each database independently chooses μKL bits out of the total KL bits from the data center to cache through the same probability distribution in a decentralized manner. In the retrieval phase, a user (retriever) accesses N databases in addition to the data center, and wishes to retrieve a desired file privately. We characterize the optimal normalized download cost to be $D^* = \sum_{n=1}^{N+1} \binom{N}{n-1} \mu^{n-1}(1-\mu)^{N+1-n} \left(1 + \frac{1}{n} + \cdots + \frac{1}{n^{K-1}}\right)$. We show that uniform and random caching scheme which is originally proposed for decentralized coded caching by Maddah-Ali and Niesen, along with Sun and Jafar retrieval scheme which is originally proposed for PIR from replicated databases surprisingly results in the lowest normalized download cost. This is the decentralized counterpart of the recent result of Attia, Kumar, and Tandon for the centralized case. The converse proof contains several ingredients such as interference lower bound, induction lemma, replacing queries and answering string random variables with the content of distributed databases, the nature of decentralized uncoded caching databases, and bit marginalization of joint caching distributions.

Keywords: private information retrieval (PIR); decentralized caching; uncoded caching; PIR capacity

1. Introduction

Private information retrieval (PIR) refers to the problem of downloading a desired file from distributed databases while keeping the identity of the desired file private against the databases. In the classical setting of PIR (see Figure 1), there are N non-communicating databases, each storing the same set of K files. The user (retriever) wishes to download one of these K files without letting the databases know the identity of the desired file. A simple but highly inefficient way is to download all the files from a particular database, which results in the normalized download cost of $\frac{D}{L} = K$, where L is the file size and D is the total number of downloaded bits from the N databases. The PIR problem has originated in the computer science community [1–5] and has drawn attention in the information theory society with early examples [6–11]. Recently, Sun and Jafar [12] have characterized the optimal normalized download cost for the classical PIR problem to be $\frac{D}{L} = \left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}}\right)$. After [12], many interesting variants of the classical PIR problem have been investigated in [13–54]. Most of these previous works consider the case where the contents of the databases are fixed a priori in an uncontrollable manner; a vast majority of them consider the case of replicated databases where each database stores the same set of K files, and many of the rest consider the case of coded databases where each database stores coded versions of the original files.

Coded caching refers to the problem of placing files in users' local storage caches ahead of time properly and designing efficient delivery schemes at the time of specific user requests in such a way to minimize the traffic during the delivery phase. In the original setup [55] (see Figure 1), a server with K files connects to N users through an error-free shared link, where each user has a local memory which can store up to M files. The system operates in two phases, a placement phase, and a delivery phase. In the placement phase, the server places the files into each user's local memory. In the delivery phase, each user requests a file from the server, and the server aims to satisfy all the requests with the lowest traffic load. If the set of users in the two phases are identical, the server can arrange the content in each user's local memory in an optimized manner, which is called centralized coded caching. Reference [55] proposes a symmetric batch caching scheme, which is shown to be optimal for the case of centralized uncoded placement in [56]. If the set of users in the two phases varies, the server cannot arrange the files in user caches in a centralized manner. Instead, the server treats each user identically and independently which is called decentralized coded caching [57]. Reference [57] proposes a uniform and random caching scheme, which is shown to be optimal for the case of decentralized uncoded placement in [56]. Many interesting variants of coded caching problems have been investigated in [58–72].

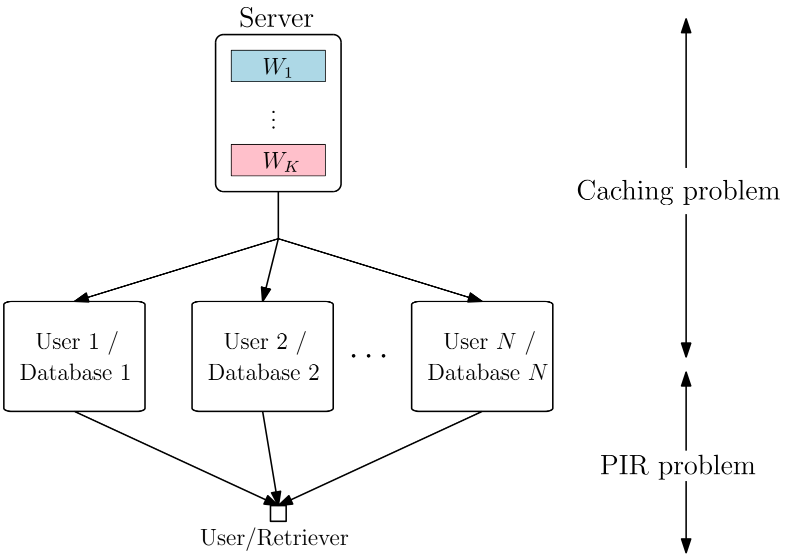

Figure 1. Joint centralized caching and the private information retrieval (PIR) problem.

The references that are most closely related to our work here are [38,44]. References [38,44] formulate a new type of PIR problem where the content of each database is not fixed a priori, but can be optimized to minimize the download cost. These papers bring PIR and coded caching problems together in a practically relevant and theoretically interesting manner. In their problem setting (see Figure 2), there is a data center (server) containing all the K files where each file is of size L bits, and the system operates in two phases. In the caching phase, there are N databases in the system with a common storage size constraint μ, i.e., each database can at most store μKL bits, $\frac{1}{N} \leq \mu \leq 1$. In the retrieval phase, a user (retriever) accesses the N databases, and wishes to download a desired file privately. They consider the problem of optimally storing content from the data center to the databases in the caching phase in such a way that the normalized download cost during the retrieval phase is minimized. They focus on the centralized uncoded caching case, i.e., the set of users in the two phases are identical so that the data center can assign the files to each database in a centralized manner, and caching is uncoded in that each database stores a subset of the bits from the data center (no coding), i.e., each database stores μKL bits out of the total KL bits. Surprisingly, they show that the symmetric batch caching scheme proposed in [55] results in the lowest normalized download cost in the retrieval phase.

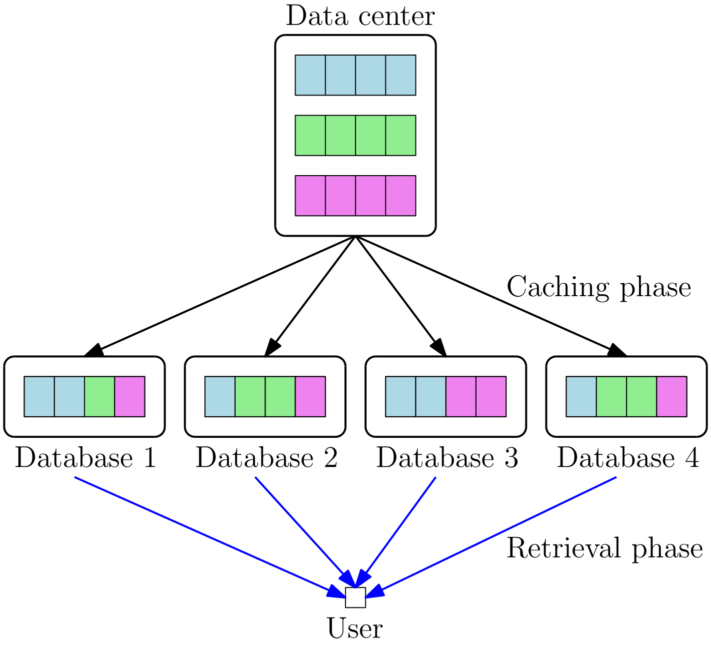

Figure 2. PIR from centralized caching databases.

We consider the PIR problem from decentralized uncoded caching databases. In our problem setting (see Figure 3), the system also operates in two phases as in [38,44]. However, the set of databases active in the two phases is different, and we do not know in advance which databases the user (retriever) can access in the retrieval phase. Therefore, we consider a decentralized setting for the caching phase, i.e., the data center treats each database identically and independently, or equivalently, each database chooses a subset of bits to store independently according to the same probability distribution. Here, we aim at designing the optimal probability distribution in the caching phase and PIR scheme in the retrieval phase such that the normalized download cost in the retrieval phase is minimized. Another main difference between our work and references [38,44] is that, in the caching phase, references [38,44] require that the N databases altogether can reconstruct the entire K files, i.e., when the user (retriever) connects to the N databases, their collective content is equivalent to the content in the data center, so the user (retriever) can download any desired file. While this can be guaranteed in the centralized setting, in the decentralized setting, where cache placement is probabilistic, we cannot guarantee that any given N databases contain all the bits that exist in the data center. Thus, in order to formulate a meaningful PIR problem, we allow the user (retriever) access the data center as well as the databases in the retrieval phase. Finally, we remark about another sub-branch of PIR literature that considers caching: [30–33,39,52]; there the user (retriever) itself has a cache memory where it stores a subset of the bits available in the databases. That problem is unrelated to the setting here even though it is also referred to as PIR with caching; in essence, it is PIR with side information.

In this work, for PIR from decentralized caching databases, we show that uniform and random caching scheme, originally proposed in [57] for decentralized coded caching, results in the lowest expected normalized download cost in the retrieval phase. For the achievability, we apply the PIR scheme in [12] successively for all resulting subfile parts. For the converse, we first apply the lower bound derived in [44], which replaces the random variables for queries and answering strings by the content of the distributed databases in a novel manner extending the lower bounding techniques in Lemma 5 and Lemma 6 in [12]. To compare different probability distributions in the caching phase, we focus on the marginal distributions on each separate bit. Then, by using the nature of decentralization and uncoded caching, we further lower bound the normalized download cost. Finally, we show the matching converse for the expected normalized download cost to be

$\frac{D}{L} = \sum_{n=1}^{N+1} \binom{N}{n-1} \mu^{n-1}(1-\mu)^{N+1-n} \left(1 + \frac{1}{n} + \cdots + \frac{1}{n^{K-1}}\right)$, which yields an exact capacity result for the problem.

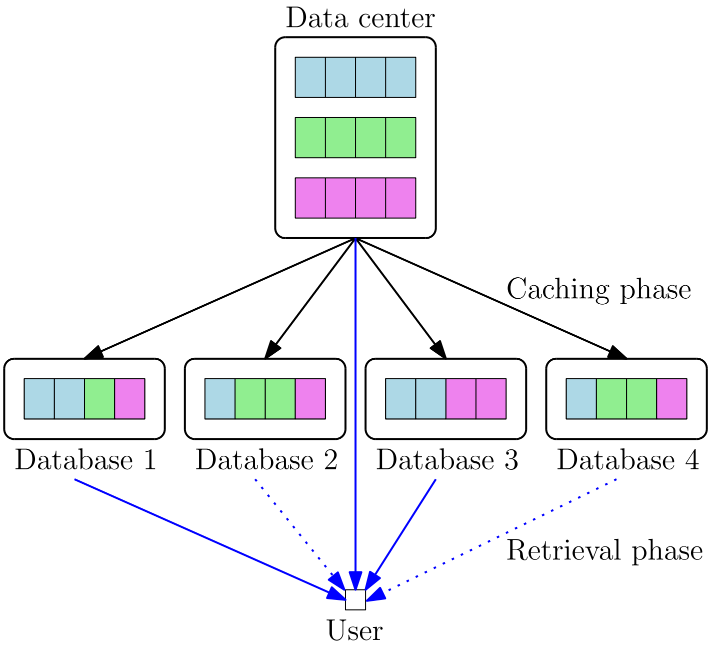

Figure 3. PIR from decentralized caching databases.

2. System Model

We consider a system consisting of one data center and several databases (at least N databases). The data center stores K independent files, labeled as W_1, W_2, $\ldots$, W_K, where each file is of size L bits. Therefore,

$$H(W_1) = \cdots = H(W_K) = L, \qquad H(W_1, \ldots, W_K) = H(W_1) + \cdots + H(W_K). \tag{1}$$

Each database has a storage capacity of μKL bits, where $0 \le \mu \le 1$.

The system operates in two phases: In the caching phase, we consider the case of uncoded caching, i.e., each database stores a subset of bits from the data center. Due to the storage size constraint, each database at most stores μKL bits out of the total KL bits from the data center. Here, we denote i-th database as DB$_i$ and use random variable Z_i to denote the stored content in DB$_i$. Therefore, the storage size constraint for DB$_i$ is

$$H(Z_i) \le \mu KL. \tag{2}$$

We consider the decentralized setting for the caching phase, i.e., each database chooses a subset of bits to store independently according to the same probability distribution, denoted by P_H. Rigorously, let random variable H_i denote the indices of the stored bits in DB$_i$. For N databases, the decentralized caching scheme $\mathcal{H}$ can be specified as

$$\mathbb{P}(\mathcal{H} = (H_1, \ldots, H_N)) = \prod_{i=1}^{N} P_H(H_i). \tag{3}$$

In the retrieval phase, the user accesses N databases and the data center. We note that we do not know in advance which N databases are available or which N databases the user will have access to. Here, we also assume that in the retrieval phase, the data center and N databases do not communicate with each other (no collusion). To simplify the notation, we use DB$_0$ to denote the data center, and therefore $Z_0 = (W_1, \ldots, W_K)$ since the data center stores all the K files. The user privately generates an index $\theta \in [K] = \{1, \ldots, K\}$, and wishes to retrieve file W_θ such that it is impossible for either the data center or any individual database to identify θ. For random variables θ, and $W_1, \ldots, W_K$, we have

$$H(\theta, W_1, \ldots, W_K) = H(\theta) + H(W_1) + \cdots + H(W_K). \tag{4}$$

In order to retrieve file W_θ, the user sends $N+1$ queries $Q_0^{[\theta]}, \ldots, Q_N^{[\theta]}$ to $\mathrm{DB}_0, \ldots, \mathrm{DB}_N$, where $Q_n^{[\theta]}$ is the query sent to DB_n for file W_θ. Note that the queries are independent of the realization of the K files. Therefore,

$$I(W_1, \ldots, W_K; Q_0^{[\theta]}, \ldots, Q_N^{[\theta]}) = 0. \tag{5}$$

Upon receiving the query $Q_n^{[\theta]}$, DB_n replies with an answering string $A_n^{[\theta]}$, which is a function of $Q_n^{[\theta]}$ and Z_n. Therefore, $\forall \theta \in [K], \forall n \in \{0\} \cup [N]$,

$$H(A_n^{[\theta]} | Q_n^{[\theta]}, Z_n) = 0. \tag{6}$$

After receiving the answering strings $A_0^{[\theta]}, \ldots, A_N^{[\theta]}$ from $\mathrm{DB}_0, \ldots, \mathrm{DB}_N$, the user needs to decode the desired file W_θ reliably. By using Fano's inequality, we have the following reliability constraint

$$H\left(W_\theta | Q_0^{[\theta]}, \ldots, Q_N^{[\theta]}, A_0^{[\theta]}, \ldots, A_N^{[\theta]}\right) = o(L), \tag{7}$$

where $o(L)$ denotes a function such that $\frac{o(L)}{L} \to 0$ as $L \to \infty$.

To ensure that individual databases do not know which file is retrieved, we have the following privacy constraint, $\forall n \in \{0\} \cup [N], \forall \theta \in [K]$,

$$(Q_n^{[1]}, A_n^{[1]}, W_1, \ldots, W_K) \sim (Q_n^{[\theta]}, A_n^{[\theta]}, W_1, \ldots, W_K), \tag{8}$$

where $A \sim B$ means that A and B are identically distributed.

Given that each file is of size L bits, for a fixed K, μ and decentralized caching probability distribution P_H, let $\mathcal{H}$ denote the indices of the cached bits in the N databases available in the retrieval phase. The probability distribution of $\mathcal{H}$ is specified in (3). Let $D_{\mathcal{H}}^{[\theta]}$ represent the number of downloaded bits via the answering strings $A_{0:N}^{[\theta]}$, where $A_{0:N}^{[\theta]} = (A_0^{[\theta]}, \ldots, A_N^{[\theta]})$. Then,

$$D_{\mathcal{H}}^{[\theta]} = \sum_{n=0}^{N} H\left(A_n^{[\theta]}\right). \tag{9}$$

We further denote $D_{\mathcal{H}}$ as the expected number of downloaded bits with respect to different file requests, i.e., $D_{\mathcal{H}} = E_\theta\left[D_{\mathcal{H}}^{[\theta]}\right]$. Finally, we denote D as the expected number of downloaded bits with respect to different realization of the cached bit indices, i.e., $D = E_{\mathcal{H}}\left[D_{\mathcal{H}}\right]$. A pair (D, L) is achievable if there exists a PIR scheme satisfying the reliability constraint (7) and the privacy constraint (8). The optimal normalized download cost D^* is defined as

$$D^* = \inf\left\{\frac{D}{L} : (D, L) \text{ is achievable}\right\}, \tag{10}$$

as $L \to \infty$. In this work, we aim at characterizing the optimal normalized download cost and finding the optimal decentralized caching probability distribution.

Next, we illustrate the system model and the problem considered with a simple example of $K = 3$ files and $N = 2$ databases in the retrieval phase; see Figure 4. Consider a data center storing $K = 3$

files where each file is of size 4 bits. In the caching phase, there are 4 databases in the system, and each database can at most store 4 bits. Each database can always store the first file, which is of size 4 bits, as caching option 1 in Figure 4. Or, each database can uniformly and randomly choose 4 bits out of a total of 12 bits from the data center to store. One of the realization is shown as caching option 2 in Figure 4. Each database can also choose two bits from the first file and one bit each from the remaining two files to store, where one of the realization is shown as caching option 3 in Figure 4. We require each database to use the same probability distribution to choose the bits to store in order to satisfy the decentralized requirement. In this example, we assume that the user can access the data center and $N = 2$ databases in the retrieval phase, say the first and the third database, and the user wishes to download a file privately. Our questions are as follows: What is the optimal probability distribution to use in the caching phase? What is the optimal PIR scheme to use in the retrieval phase? How can we jointly design the schemes in the two phases such that the expected normalized download cost is the lowest in the second phase?

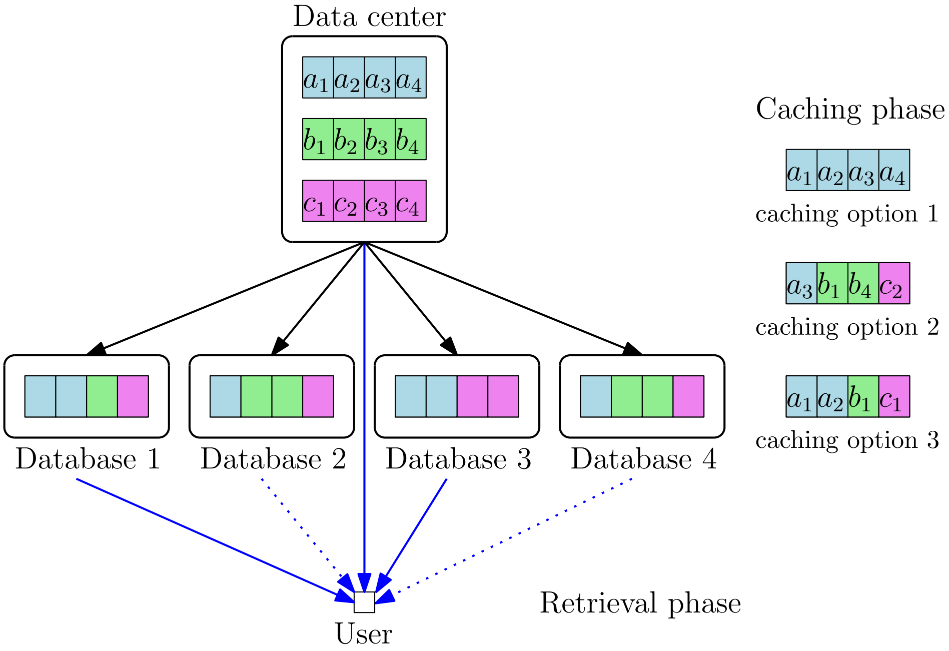

Figure 4. PIR from decentralized caching databases with $K = 3$, $N = 2$, and $\mu = \frac{1}{3}$.

3. Main Results and Discussion

We characterize the optimal normalized download cost for PIR from decentralized uncoded caching databases in the following theorem.

Theorem 1. *For PIR from decentralized uncoded caching databases with K files, where each file is of size L bits, N databases in addition to a data center available in the retrieval phase, and a storage size constraint μKL, $0 < \mu < 1$, bits for each database, the optimal normalized download cost is*

$$D^* = \sum_{n=1}^{N+1} \binom{N}{n-1} \mu^{n-1} (1-\mu)^{N+1-n} \left(1 + \frac{1}{n} + \cdots + \frac{1}{n^{K-1}}\right). \tag{11}$$

The achievability scheme is provided in Section 4, and the converse proof is shown in Section 5. We first use the following example to show the main ingredients of Theorem 1.

3.1. Motivating Example: $K = 3$ and $N = 2$

In this example, we consider the case where the data center stores $K = 3$ independent files labeled as A, B, and C, where each file is of size L bits. In the caching phase, several databases with storage capacity of $3\mu L$ bits are present in the system. We will show that the optimal normalized download cost is $\frac{D}{L} = \frac{17}{18}\mu^2 - \frac{5}{2}\mu + 3$ when $N = 2$ databases in addition to the data center are available in the retrieval phase.

3.1.1. Achievability Scheme

In the caching phase, to satisfy the storage size constraint, each database randomly and uniformly stores $3\mu L$ bits out of a total of $3L$ bits from the data center. Each database operates independently through the same probability distribution resulting in decentralized caching.

In the retrieval phase, suppose $N = 2$ databases, labeled as DB_1 and DB_2, in addition to the data center, labeled as DB_0, are available to the user, and the user wishes to retrieve file A privately. Let us first focus on one file, say A. We can partition file A into four subfiles

$$A = (A_0, A_{0,1}, A_{0,2}, A_{0,1,2}), \tag{12}$$

where, for $S \subseteq \{0,1,2\}$, A_S denotes the bits of file A which are stored in databases in S. For example, A_0 denotes the bits of file A only stored in DB_0 and $A_{0,2}$ denotes the bits of file A stored in DB_0 and DB_2 and so on. Since each bit is stored in the data center, 0 exists in the label of every partition. By the law of large numbers,

$$|A_S| = L\mu^{|S|-1}(1-\mu)^{3-|S|} + o(L), \tag{13}$$

when the file size is large enough. We can do the same partitions for files B and C.

To retrieve file A privately, we first retrieve the subfile $A_{0,1,2}$ privately. We apply the PIR scheme proposed in [12] to retrieve the subfile $A_{0,1,2}$. Subfile $A_{0,1,2}$ is replicated in 3 databases and the total number of files is three since we also have $B_{0,1,2}$ and $C_{0,1,2}$. Therefore, we download

$$L\mu^2\left(1 + \frac{1}{3} + \frac{1}{9}\right) + o(L) \tag{14}$$

bits. We also need to retrieve the subfile $A_{0,1}$ privately. Subfile $A_{0,1}$ is replicated in 2 databases and the total number of files is 3 since we also have $B_{0,1}$ and $C_{0,1}$. By applying the PIR scheme in [12], we download

$$L\mu(1-\mu)\left(1 + \frac{1}{2} + \frac{1}{4}\right) + o(L) \tag{15}$$

bits. Next, we need to retrieve the subfile $A_{0,2}$ privately. Using [12], we download

$$L\mu(1-\mu)\left(1 + \frac{1}{2} + \frac{1}{4}\right) + o(L) \tag{16}$$

bits. Finally, we need to retrieve A_0 privately. Using [12], we download

$$L(1-\mu)^2(1+1+1) + o(L) \tag{17}$$

bits. By adding (14)–(17), we show that the normalized download cost

$$\frac{17}{18}\mu^2 - \frac{5}{2}\mu + 3 \tag{18}$$

is achievable.

3.1.2. Converse Proof

Here, we show that among all the decentralized caching probability distributions P_H, the lowest normalized download cost for $N = 2$ databases is as shown in (18). Given a decentralized caching probability distribution P_H, we have a resulting $\mathcal{H}$ in the retrieval phase.

We lower bound $D_{\mathcal{H}}$ first. In the retrieval phase, the stored content of DB$_0$, DB$_1$, and DB$_2$ are fixed and uncoded, i.e., Z_0, Z_1 and Z_2 are fixed and uncoded. We can apply the lower bound in Equation (31) in [44] as the lower bound for $D_{\mathcal{H}}$. Therefore,

$$D_{\mathcal{H}} \geq L + \frac{4}{27}\sum_{k=1}^{3} H(W_k) + \frac{11}{108}\sum_{i=0}^{2}\sum_{k=1}^{3} H(W_k|Z_i) + \frac{17}{54}\sum_{i=0}^{2}\sum_{k=1}^{3} H(W_k|Z_{[0:2]\setminus i}) + o(L) \tag{19}$$

$$= \frac{13}{9}L + \frac{11}{108}\sum_{i=1}^{2}\sum_{k=1}^{3} H(W_k|Z_i) + \frac{17}{54}\sum_{k=1}^{3} H(W_k|Z_1, Z_2) + o(L) \tag{20}$$

$$\geq \frac{13}{9}L + \frac{11}{108}(3L - 3\mu L + 3L - 3\mu L) + \frac{17}{54}\sum_{k=1}^{3} H(W_k|Z_1, Z_2) + o(L) \tag{21}$$

$$= \frac{37}{18}L - \frac{11}{18}\mu L + \frac{17}{54}H(W_{1:3}|Z_1, Z_2) + o(L), \tag{22}$$

where (20) holds due to $Z_0 = (W_1, W_2, W_3)$, and (21) holds due to (2). We note that different $\mathcal{H}$ results in different Z_1 and Z_2.

We lower bound D now. From (22), we have

$$D = E_{\mathcal{H}}[D_{\mathcal{H}}] \geq \frac{37}{18}L - \frac{11}{18}\mu L + \frac{17}{54}E_{\mathcal{H}}[H(W_{1:3}|Z_1, Z_2)] + o(L). \tag{23}$$

Let random variables $X_{i,j}^{(n)}$, $i = 1, \ldots, L$, $j = 1, \ldots, K$, be the indicator functions showing that the i-th bit of file W_j is cached in DB$_n$ or not, i.e., $X_{i,j}^{(n)} = 1$ means that the i-th bit of file W_j is stored in DB$_n$ and $X_{i,j}^{(n)} = 0$ means that it is not stored in DB$_n$. For DB$_1$ we have

$$X_{1,1}^{(1)} + \cdots + X_{L,1}^{(1)} + X_{1,2}^{(1)} + \cdots + X_{L,2}^{(1)} + X_{1,3}^{(1)} + \cdots + X_{L,3}^{(1)} \leq 3\mu L, \tag{24}$$

due to the storage size constraint in (2). We note that P_H induces probability measures on random variables $X_{i,j}^{(n)}$, and let $X_{i,j}^{(n)} = 1$ with probability $p_{i,j}$, where we remove the superscript n since each database adopts the same probability distribution P_H to choose the cached bits due to the decentralized property. By taking expectation on (24) and applying the linearity of expectation, we have

$$E[X_{1,1}^{(1)}] + \cdots + E[X_{L,3}^{(1)}] \leq 3\mu L, \tag{25}$$

which yields

$$p_{1,1} + \cdots + p_{L,3} \leq 3\mu L. \tag{26}$$

Let random variables $V_{i,j}$, $i = 1, \ldots, L$, $j = 1, \ldots, K$, be the indicator functions showing that the i-th bit of file W_j is not cached in DB$_1$ and DB$_2$, i.e., $V_{i,j} = 1$ means that the i-th bit of file W_j is not stored in either DB$_1$ or DB$_2$. Therefore, we have

$$V_{i,j} = (1 - X_{i,j}^{(1)})(1 - X_{i,j}^{(2)}). \tag{27}$$

Now, we can evaluate $E_{\mathcal{H}}\left[H(W_{1:3}|Z_1,Z_2)\right]$ in (23) as follows

$$\begin{aligned} E_{\mathcal{H}}\left[H(W_{1:3}|Z_1,Z_2)\right] &= E[V_{1,1}+\cdots+V_{L,3}] && (28)\\ &= E[V_{1,1}]+\cdots+E[V_{L,3}] && (29)\\ &= (1-p_{1,1})^2+\cdots+(1-p_{L,3})^2. && (30)\end{aligned}$$

Therefore, continuing from (23), we have

$$D \geq \frac{37}{18}L - \frac{11}{18}\mu L + \frac{17}{54}\left[(1-p_{1,1})^2+\cdots+(1-p_{L,3})^2\right] + o(L), \tag{31}$$

where $p_{1,1},\ldots,p_{L,3}$ are subject to (26). To further lower bound the right hand side of (31), we minimize the right hand side with respect to $p_{i,j}$ subject to (26). Hence, we consider the following Lagrangian

$$L(p_{1,1},\ldots,p_{L,3},\lambda) = (1-p_{1,1})^2+\cdots+(1-p_{L,3})^2+\lambda\left(p_{1,1}+\cdots+p_{L,3}-3\mu L\right). \tag{32}$$

From the KKT conditions, we have

$$\lambda = 2(1-p_{i,j}), \quad i=1,\ldots,L, \quad j=1,2,3. \tag{33}$$

Thus, we can further lower bound (31) by letting $p_{1,1}=\cdots=p_{L,3}=\mu$, and we have

$$\begin{aligned} \frac{D}{L} &\geq \frac{37}{18} - \frac{11}{18}\mu + \frac{17}{54}\left[3(1-\mu)^2\right] + \frac{o(L)}{L} && (34)\\ &= \frac{17}{18}\mu^2 - \frac{5}{2}\mu + 3 + \frac{o(L)}{L}. && (35)\end{aligned}$$

Therefore, we show that the optimal normalized download cost is $\frac{17}{18}\mu^2 - \frac{5}{2}\mu + 3$ when $N=2$ databases in addition to the data center are available in the retrieval phase. To achieve the optimal normalized download cost, each database should randomly and uniformly store the bits in the caching phase.

3.2. Further Examples and Numerical Results

Now, we use different scenarios to illustrate the optimal normalized download cost in (11). We first consider the scenario where the data center contains $K=10$ files, each database with storage size constraint $\mu=\frac{1}{2}$, and in the retrieval phase, the user can access $N=0,\ldots,30$ databases in addition to the data center. We plot the expected normalized download cost versus the different number of available databases in Figure 5. When $N=0$, in order to download the desired file privately, the user should download all the files in the data center, and this results in a download cost of $\frac{D}{L}=K=10$. As the number of accessible databases increases, the normalized download cost decreases. We next consider the scenario where the data center contains $K=10$ files, and the user can access $N=5$ databases in addition to the data center in the retrieval phase. We plot the expected normalized download cost versus different storage size constraint μ in Figure 6. When $\mu=0$, in order to download the desired file privately, the user should download all the files in the data center resulting in $\frac{D}{L}=K=10$. As μ increases, the normalized download cost decreases. Finally, we conclude this section with the following general remarks about our main result.

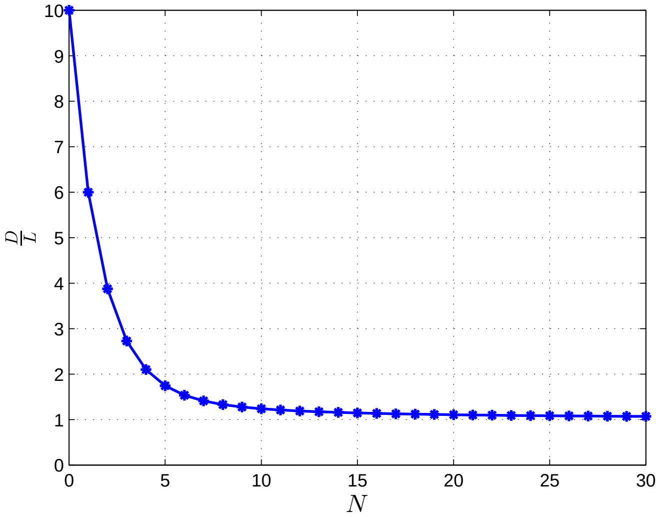

Figure 5. PIR from different number of available databases in the retrieval phase with $K = 10$ and $\mu = \frac{1}{2}$.

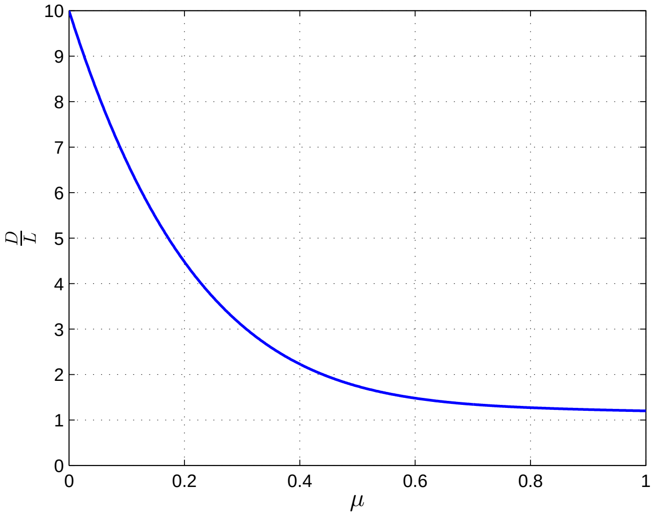

Figure 6. PIR from $N = 5$ databases with different storage constraint μ with $K = 10$.

3.3. Remarks

Remark 1. *The achievability scheme consists of two parts, the design of the probability distribution in the caching phase and the PIR scheme in the retrieval phase. We find that the uniform and random caching scheme, originally proposed in [57] for decentralized coded caching, results in the optimal normalized download cost in the retrieval phase. We remark here that the symmetric batch caching scheme, originally proposed in [55] for centralized coded caching, also results in the optimal normalized download cost for PIR from centralized uncoded caching databases [44]. In the retrieval phase, according to the distribution of the subfiles, we apply the PIR scheme proposed in [12] for all subfiles to retrieve the desired file.*

Remark 2. *For the converse, we first apply the lower bound derived in [44] which introduces new ingredients in addition to the interference lower bound lemma and induction lemma in Lemmas 5 and 6 in [12]. We note that in [44] the authors replace random variables for queries and answering strings by the contents of the distributed databases in a novel way which is crucial for the converse. With this replacement, we can account for different cached content in the caching phase resulting in different lower bound in the normalized download cost in the retrieval phase. Due to the nature of uncoded caching, this replacement facilitates further lower bound. For the* *decentralized problem here, to compare different probability distributions in the caching phase, we focus on the marginal distributions on each bit. This transformation allows us to use linearity of expectation, and the nature of decentralization and uncoded caching to further lower bound the expected normalized download cost.*

Remark 3. *A more directly related PIR problem from centralized uncoded caching databases for our setting is the one where, in the caching phase, the data center arranges the files in N databases in a centralized manner, and in the retrieval phase, the user has access also to the data center in addition to the N databases. This is different from the problem setting in [38,44], since there the user can only access the N databases in the retrieval phase. As a side note, we can show that the symmetric batch caching scheme is still optimal for this extended problem setting where the data center also participates in the PIR stage. Rigorously, the optimal trade-off between storage and download cost, in this case, is given by the lower convex envelope of the following $(\mu, D(\mu))$ pairs, for $t = 0, 1, \ldots, N$,*

$$\left(\mu = \frac{t}{N}, D(\mu) = \sum_{k=0}^{K-1} \frac{1}{(t+1)^k}\right). \tag{36}$$

To achieve this trade-off, the data center arranges the files into the N databases as in [38,44]. In the retrieval phase, the user accesses also the data center; therefore, the subfiles are stored in one more database. For the converse, we no longer require all the N databases to reconstruct the entire K files as in [38,44]. Thus, while in [38,44] the smallest allowable μ is $\mu = \frac{1}{N}$, since the N databases need to reconstruct the entire K files, here since the user can access the data center, the parameter μ starts from 0. Now, we can compare PIR from centralized caching databases and PIR from decentralized caching databases fairly, since, in the retrieval phase, the user can access the data center in both cases. We consider the case where $K = 10$ and $N = 5$, and plot the result in Figure 7.

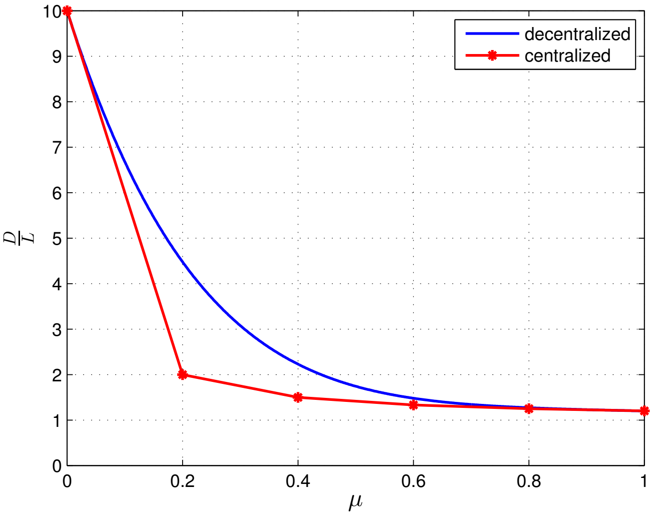

Figure 7. PIR from centralized caching databases and decentralized caching databases.

4. Achievability Scheme

The achievability scheme consists of two parts: the design of the probability distribution used in the caching phase and the PIR scheme used in the retrieval phase. In the caching phase, each database uniformly and randomly stores μKL bits from the data center. The storage size constraint in (2) is satisfied directly. Each database operates independently through the same probability distribution resulting in decentralized caching.

In the retrieval phase, suppose there are N databases in addition to the data center available to the user. Each file W_j can be expressed as

$$W_j = \bigcup_{\{0\}\subseteq S\subseteq\{0,1,\ldots,N\}} W_{j,S}, \tag{37}$$

where $W_{j,S}$ represents the bits of file W_j which are stored in databases in S. Since each bit must be stored in the data center, i.e., DB_0, we have $\{0\} \subseteq S$. By the law of large numbers,

$$|W_{j,S}| = L\mu^{|S|-1}(1-\mu)^{N+1-|S|} + o(L), \tag{38}$$

when the file size is large enough.

To retrieve the desired file, say W_j, privately, we retrieve each subfile, $W_{j,S}$, privately. Subfile $W_{j,S}$ is replicated in $|S|$ databases, and for each of these $|S|$ databases, there are K subfiles, i.e., $W_{k,S}$, $k = 1,\ldots,K$. We apply the PIR scheme in [12] to retrieve $W_{j,S}$ privately by downloading

$$L\mu^{|S|-1}(1-\mu)^{N+1-|S|}\left(1+\frac{1}{|S|}+\cdots+\frac{1}{|S|^{K-1}}\right)+o(L) \tag{39}$$

bits. We also note that there are $\binom{N}{|S|-1}$ types of $W_{j,S}$. Therefore, the following normalized download cost

$$\frac{D}{L} = \sum_{n=1}^{N+1}\binom{N}{n-1}\mu^{n-1}(1-\mu)^{N+1-n}\left(1+\frac{1}{n}+\cdots+\frac{1}{n^{K-1}}\right) \tag{40}$$

is achievable.

5. Converse Proof

We first derive a lower bound for $D_{\mathcal{H}}$. Since in the retrieval phase the content of $\text{DB}_0,\ldots,\text{DB}_N$, are fixed to be $Z_0,\ldots,Z_N$, we can use the lower bound derived in Equation (71) in [44] to serve as the lower bound for $D_{\mathcal{H}}$. A key step to obtain Equation (71) in [44] is to replace the query and answering string random variables with the content of each database, i.e., replacement of $Q_{\mathcal{N}}^{[k]}$ and $A_{\mathcal{N}}^{[k]}$ with $Z_{\mathcal{N}}$. With this replacement, one can account for different cached content in the caching phase resulting in different lower bound in the normalized download cost in the retrieval phase. In addition, due to the nature of uncoded caching, this replacement facilitates a further lower bound. Moreover, to obtain Equation (71) in [44], the authors find interesting recursive relationships to compactly deal with the nested harmonic sums. Therefore, from Equation (71) in [44] we have

$$D_{\mathcal{H}} \geq L + \sum_{l=1}^{N+1}\binom{N+1}{l}\left(\frac{1}{l}+\frac{1}{l^2}+\cdots+\frac{1}{l^{K-1}}\right)x_l, \tag{41}$$

where

$$x_l\delta\frac{1}{K\binom{N+1}{l}}\sum_{\{0\}\subseteq S\subseteq[0:N],\ |S|=l} H(W_{1:K,S}), \quad l\in[1:N+1], \tag{42}$$

and $W_{1:K,S}$ represents the bits of files $W_{1:K}$ which are stored in databases in S.

In the following lemma, we develop a lower bound for $E[x_l]$.

Lemma 1. *For $l\in[1:N+1]$, and x_l given in (42), we have*

$$E[x_l] \geq L\mu^{l-1}(1-\mu)^{N+1-l}\frac{\binom{N}{l-1}}{\binom{N+1}{l}}. \tag{43}$$

Proof 1. *By taking expectation on* (42) *and using the linearity of expectation, we have*

$$E[x_l] = \frac{1}{K\binom{N+1}{l}} \sum_{\{0\}\subseteq S\subseteq[0:N],\ |S|=l} E[H(W_{1:K,S})]. \tag{44}$$

Let random variables $X_{i,j}^{(n)}$, $i = 1, \ldots, L$, $j = 1, \ldots, K$, be the indicator functions showing that the i-th bit of file W_j is cached in DB_n, $n = 0, \ldots, N$, or not, i.e., $X_{i,j}^{(n)} = 1$ means that the i-th bit of file W_j is stored in DB_n and $X_{i,j}^{(n)} = 0$ means that it is not stored in DB_n. For DB_n we have

$$X_{1,1}^{(n)} + \cdots + X_{L,1}^{(n)} + \cdots + X_{1,K}^{(n)} + \cdots + X_{L,K}^{(n)} \leq \mu KL, \tag{45}$$

due to the storage size constraint in (2)*. We note that P_H induces probability measures on random variables $X_{i,j}^{(n)}$, and let $X_{i,j}^{(n)} = 1$ with probability $p_{i,j}$, where we remove the superscript n since each database adopts the same probability distribution P_H to choose the cached bits due to the decentralized caching property. By taking expectation on* (45) *and applying the linearity of expectation, we have*

$$E[X_{1,1}^{(n)}] + \cdots + E[X_{L,K}^{(n)}] \leq \mu KL, \tag{46}$$

which yields

$$p_{1,1} + \cdots + p_{L,K} \leq \mu KL.. \tag{47}$$

Let random variables $Y_{i,j}^{S}$, $i = 1, \ldots, L$, $j = 1, \ldots, K$, be the indicator functions showing that the i-th bit of file W_j is cached in DB_n, $n \in S$, i.e., $Y_{i,j} = 1$ means that the i-th bit of the file W_j is stored in DB_n, $n \in S$. Therefore, we have

$$Y_{i,j}^{S} = \prod_{n\in S} X_{i,j}^{(n)} \prod_{n\in[0:N]\setminus S} (1 - X_{i,j}^{(n)}). \tag{48}$$

Now, we can evaluate $E\left[H(W_{1:K,S})\right]$ in (44) *as follows*

$$\begin{aligned} E\left[H(W_{1:K,S})\right] &= E[Y_{1,1}^{S} + \cdots + Y_{L,K}^{S}] && (49)\\ &= E[Y_{1,1}^{S}] + \cdots + E[Y_{L,K}^{S}] && (50)\\ &= p_{1,1}^{|S|-1}(1-p_{1,1})^{N+1-|S|} + \cdots + p_{L,K}^{|S|-1}(1-p_{L,K})^{N+1-|S|}, && (51) \end{aligned}$$

where $p_{1,1}, \ldots, p_{L,K}$ are subject to (47)*. Now, continuing from* (44)*, we have*

$$E[x_l] = \frac{1}{K\binom{N+1}{l}} \sum_{\{0\}\subseteq S\subseteq[0:N],\ |S|=l} p_{1,1}^{l-1}(1-p_{1,1})^{N+1-l} + \cdots + p_{L,K}^{l-1}(1-p_{L,K})^{N+1-l}. \tag{52}$$

To further lower bound (52)*, we consider the following Lagrangian*

$$\begin{aligned} L(p_{1,1}, \ldots, p_{L,K}, \lambda) \quad &= p_{1,1}^{l-1}(1-p_{1,1})^{N+1-l} + \cdots + p_{L,K}^{l-1}(1-p_{L,K})^{N+1-l} \\ &\quad +\lambda\left(p_{1,1} + \cdots + p_{L,K} - \mu KL\right). \end{aligned} \tag{53}$$

From the KKT conditions, we have

$$\lambda = p_{i,j}^{l-1}(N+1-l)(1-p_{i,j})^{N-l} - (l-1)p_{i,j}^{l-2}(1-p_{i,j})^{N+1-l}, \tag{54}$$

where $i = 1, \ldots, L$, $j = 1, \ldots, K$. Therefore, we can further lower bound (52) by letting $p_{1,1} = \cdots = p_{L,K} = \mu$, then we have

$$E[x_l] \geq \frac{1}{K\binom{N+1}{l}} \sum_{\{0\}\subseteq S\subseteq[0:N],\ |S|=l} KL\mu^{l-1}(1-\mu)^{N+1-l} \tag{55}$$

$$= L\mu^{l-1}(1-\mu)^{N+1-l}\frac{\binom{N}{l-1}}{\binom{N+1}{l}}, \tag{56}$$

which completes the proof. □

Finally, by taking expectation and applying Lemma 1 to (41), we obtain

$$\frac{D}{L} \geq 1 + \sum_{l=1}^{N+1}\binom{N}{l-1}\left(\frac{1}{l}+\frac{1}{l^2}+\cdots+\frac{1}{l^{K-1}}\right)\mu^{l-1}(1-\mu)^{N+1-l} \tag{57}$$

$$= (\mu+(1-\mu))^N + \sum_{l=1}^{N+1}\binom{N}{l-1}\left(\frac{1}{l}+\frac{1}{l^2}+\cdots+\frac{1}{l^{K-1}}\right)\mu^{l-1}(1-\mu)^{N+1-l} \tag{58}$$

$$= \sum_{l=1}^{N+1}\binom{N}{l-1}\left(1+\frac{1}{l}+\frac{1}{l^2}+\cdots+\frac{1}{l^{K-1}}\right)\mu^{l-1}(1-\mu)^{N+1-l} \tag{59}$$

which matches (40).

6. Conclusions

We considered the PIR problem from decentralized uncoded caching databases. Due to the nature of decentralization and the storage size constraint, we allow the user to access the data center in the retrieval phase to guarantee that the user can reconstruct the entire desired file. We showed that uniform and random decentralized caching scheme, originally proposed in [57] for the problem of decentralized coded caching, results in the lowest expected normalized download cost in the PIR phase. We characterized the expected normalized download cost to be $\frac{D}{L} = \sum_{n=1}^{N+1}\binom{N}{n-1}\mu^{n-1}(1-\mu)^{N+1-n}\left(1+\frac{1}{n}+\cdots+\frac{1}{n^{K-1}}\right)$. For the achievability, we applied the PIR scheme in [12] for all subfiles. For the converse, we first applied the lower bound derived in [44], and to compare different probability distributions in the caching phase, we focused on the marginal distributions on individual bits. By using the nature of decentralization and uncoded caching, we further lower bounded the normalized download cost. Finally, we showed the matching converse for the expected normalized download cost, obtaining the exact capacity of the resulting PIR problem.

Author Contributions: All authors contributed equally in all aspects.

Funding: This work was supported by NSF Grants CNS 15-26608, CCF 17-13977 and ECCS 18-07348.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chor, B.; Kushilevitz, E.; Goldreich, O.; Sudan, M. Private information retrieval. *J. ACM* **1998**, *45*, 965–981. [CrossRef]
2. Gasarch, W. A survey on private information retrieval. *Bull. EATCS* **2004**, *82*, 72–107.
3. Cachin, C.; Micali, S.; Stadler, M. Computationally private information retrieval with polylogarithmic communication. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999.
4. Ostrovsky, R., III; Skeith, W.E. A survey of single-database private information retrieval: Techniques and applications. In Proceedings of the International Workshop on Public Key Cryptography, Beijing, China, 16–20 April 2007; pp. 393–411.

5. Yekhanin, S. Private information retrieval. *Commun. ACM* **2010**, *53*, 68–73. [CrossRef]
6. Shah, N.B.; Rashmi, K.V.; Ramchandran, K. One extra bit of download ensures perfectly private information retrieval. In Proceedings of the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014.
7. Fanti, G.; Ramchandran, K. Efficient Private Information Retrieval Over Unsynchronized Databases. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 1229–1239. [CrossRef]
8. Chan, T.; Ho, S.; Yamamoto, H. Private information retrieval for coded storage. In Proceedings of the 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, 14–19 June 2015.
9. Fazeli, A.; Vardy, A.; Yaakobi, E. Codes for distributed PIR with low storage overhead. In Proceedings of the 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, 14–19 June 2015.
10. Tajeddine, R.; Rouayheb, S.E. Private information retrieval from MDS coded data in distributed storage systems. *IEEE Trans. Inf. Theory* **2016**, *64*, 7081–7093. [CrossRef]
11. Sun, H.; Jafar, S.A. The Capacity of Private Information Retrieval. In Proceedings of the IEEE Globecom, Washington, DC, USA, 4–8 December 2016.
12. Sun, H.; Jafar, S.A. The Capacity of Private Information Retrieval. *IEEE Trans. Inf. Theory* **2017**, *63*, 4075–4088. [CrossRef]
13. Sun, H.; Jafar, S.A. The Capacity of Robust Private Information Retrieval with Colluding Databases. *IEEE Trans. Inf. Theory* **2018**, *64*, 2361–2370. [CrossRef]
14. Sun, H.; Jafar, S.A. The Capacity of Symmetric Private Information Retrieval. *IEEE Trans. Inf. Theory* **2019**, *65*, 322–329. [CrossRef]
15. Banawan, K.; Ulukus, S. The Capacity of Private Information Retrieval from Coded Databases. *IEEE Trans. Inf. Theory* **2018**, *64*, 1945–1956. [CrossRef]
16. Sun, H.; Jafar, S.A. Optimal Download Cost of Private Information Retrieval for Arbitrary Message Length. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2920–2932. [CrossRef]
17. Wang, Q.; Skoglund, M. Symmetric Private Information Retrieval For MDS Coded Distributed Storage. *arXiv* **2016**, arXiv:1610.04530.
18. Sun, H.; Jafar, S.A. Multiround Private Information Retrieval: Capacity and Storage Overhead. *IEEE Trans. Inf. Theory* **2018**, *64*, 5743–5754. [CrossRef]
19. Freij-Hollanti, R.; Gnilke, O.; Hollanti, C.; Karpuk, D. Private Information Retrieval from Coded Databases with Colluding Servers. *SIAM J. Appl. Algebra Geom.* **2017**, *1*, 647–664. [CrossRef]
20. Sun, H.; Jafar, S.A. Private Information Retrieval from MDS Coded Data with Colluding Servers: Settling a Conjecture by Freij-Hollanti et al. *IEEE Trans. Inf. Theory* **2018**, *64*, 1000–1022. [CrossRef]
21. Tajeddine, R.; Gnilke, O.W.; Karpuk, D.; Freij-Hollanti, R.; Hollanti, C.; Rouayheb, S.E. Private Information Retrieval Schemes for Coded Data with Arbitrary Collusion Patterns. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017.
22. Banawan, K.; Ulukus, S. Multi-Message Private Information Retrieval: Capacity Results and Near-Optimal Schemes. *IEEE Trans. Inf. Theory* **2018**, *64*, 6842–6862. [CrossRef]
23. Zhang, Y.; Ge, G. A general private information retrieval scheme for MDS coded databases with colluding servers. *arXiv* **2017**, arXiv:1704.06785.
24. Zhang, Y.; Ge, G. Multi-file Private Information Retrieval from MDS Coded Databases with Colluding Servers. *arXiv* **2017**, arXiv:1705.03186.
25. Banawan, K.; Ulukus, S. The Capacity of Private Information retrieval from Byzantine and Colluding Databases. *IEEE Trans. Inf. Theory* **2018**, *65*, 1206–1219. [CrossRef]
26. Tandon, R. The Capacity of Cache Aided Private Information Retrieval. In Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 3–6 October 2017.
27. Wang, Q.; Skoglund, M. Secure Symmetric Private Information Retrieval from Colluding Databases with Adversaries. *arXiv* **2017**, arXiv:1707.02152.
28. Tajeddine, R.; Rouayheb, S.E. Robust private information retrieval on coded data. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017.
29. Wang, Q.; Skoglund, M. Linear Symmetric Private Information Retrieval for MDS Coded Distributed Storage with Colluding Servers. *arXiv* **2017**, arXiv:1708.05673.

30. Kadhe, S.; Garcia, B.; Heidarzadeh, A.; Rouayheb, S.E.; Sprintson, A. Private Information Retrieval with Side Information. *arXiv* **2017**, arXiv:1709.00112.
31. Wei, Y.P.; Banawan, K.; Ulukus, S. Fundamental Limits of Cache-Aided Private Information Retrieval with Unknown and Uncoded Prefetching. *IEEE Trans. Inf. Theory* **2019**, *65*, 3215–3232. [CrossRef]
32. Chen, Z.; Wang, Z.; Jafar, S.A. The Capacity of Private Information Retrieval with Private Side Information. *arXiv* **2017**, arXiv:1709.03022.
33. Wei, Y.P.; Banawan, K.; Ulukus, S. The capacity of private information retrieval with partially known private side information. *IEEE Trans. Inf. Theory* **2019**. [CrossRef]
34. Wang, Q.; Skoglund, M. Secure Private Information Retrieval from Colluding Databases with Eavesdroppers. *arXiv* **2017**, arXiv:1710.01190.
35. Sun, H.; Jafar, S.A. The capacity of private computation. *IEEE Trans. Inf. Theory* **2018**, *65*, 3880–3897. [CrossRef]
36. Kim, M.; Yang, H.; Lee, J. Cache-aided private information retrieval. In Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 3–6 October 2017.
37. Mirmohseni, M.; Maddah-Ali, M.A. Private Function Retrieval. *arXiv* **2017**, arXiv:1711.04677.
38. Abdul-Wahid, M.; Almoualem, F.; Kumar, D.; Tandon, R. Private Information Retrieval from Storage Constrained Databases—Coded Caching meets PIR. *arXiv* **2017**, arXiv:1711.05244.
39. Wei, Y.P.; Banawan, K.; Ulukus, S. Cache-Aided Private Information Retrieval with Partially Known Uncoded Prefetching: Fundamental Limits. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1126–1139. [CrossRef]
40. Banawan, K.; Ulukus, S. Asymmetry hurts: Private information retrieval under asymmetric traffic constraints. *IEEE Trans. Inf. Theory* **2019**, *65*, 7628–7645. [CrossRef]
41. Chen, Z.; Wang, Z.; Jafar, S.A. The Asymptotic Capacity of Private Search. In Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 17–22 June 2018.
42. Banawan, K.; Ulukus, S. Private Information Retrieval Through Wiretap Channel II: Privacy Meets Security. *arXiv* **2018**, arXiv:1801.06171.
43. Wang, Q.; Sun, H.; Skoglund, M. The capacity of private information retrieval with eavesdroppers. *IEEE Trans. Inf. Theory* **2018**, *65*, 3198–3214. [CrossRef]
44. Attia, M.A.; Kumar, D.; Tandon, R. The Capacity of Private Information Retrieval from Uncoded Storage Constrained Databases. *arXiv* **2018**, arXiv:1805.04104.
45. Wei, Y.P.; Ulukus, S. The Capacity of Private Information Retrieval with Private Side Information Under Storage Constraints. *arXiv* **2018**, arXiv:1806.01253.
46. Tajeddine, R.; Gnilke, O.W.; Karpuk, D.; Freij-Hollanti, R.; Hollanti, C. Private Information Retrieval from Coded Storage Systems with Colluding, Byzantine, and Unresponsive Servers. *IEEE Trans. Inf. Theory* **2019**, *65*, 3898–3906. [CrossRef]
47. Banawan, K.; Ulukus, S. Noisy private information retrieval: On separability of channel coding and information retrieval. *IEEE Trans. Inf. Theory* **2019**. [CrossRef]
48. Jia, Z.; Sun, H.; Jafar, S.A. Cross Subspace Alignment and the Asymptotic Capacity of X-Secure T-Private Information Retrieval. *IEEE Trans. Inf. Theory* **2019**, *65*, 5783–5798. [CrossRef]
49. Tian, C.; Sun, H.; Chen, J. Capacity-Achieving Private Information Retrieval Codes with Optimal Message Size and Upload Cost. *IEEE Trans. Inf. Theory* **2019**, *65*, 7613–7627. [CrossRef]
50. Kumar, S.; i Amat, A.G.; Rosnes, E.; Senigagliesi, L. Private Information Retrieval From a Cellular Network With Caching at the Edge. *IEEE Trans. Commun.* **2019**, *67*, 4900–4912. [CrossRef]
51. Bitar, R.; Rouayheb, S.E. Staircase-PIR: Universally Robust Private Information Retrieval. In Proceedings of the 2018 IEEE Information Theory Workshop (ITW), Guangzhou, China, 25–29 November 2018.
52. Li, S.; Gastpar, M. Converse for Multi-Server Single-Message PIR with Side Information. *arXiv* **2018**, arXiv:1809.09861.
53. D'Oliveira, R.G.; Rouayheb, S.E. One-Shot PIR: Refinement and Lifting. *arXiv* **2018**, arXiv:1810.05719.
54. Tajeddine, R.; Wachter-Zeh, A.; Hollanti, C. Private Information Retrieval over Networks. *arXiv* **2018**, arXiv:1810.08941.
55. Maddah-Ali, M.A.; Niesen, U. Fundamental limits of caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 2856–2867. [CrossRef]

56. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. The exact rate-memory tradeoff for caching with uncoded prefetching. *IEEE Trans. Inf. Theory* **2018**, *64*, 1281–1296. [CrossRef]
57. Maddah-Ali, M.A.; Niesen, U. Decentralized coded caching attains order-optimal memory-rate tradeoff. *IEEE/ACM Trans. Netw.* **2015**, *23*, 1029–1040. [CrossRef]
58. Ji, M.; Caire, G.; Molisch, A.F. Fundamental limits of caching in wireless D2D networks. *IEEE Trans. Inf. Theory* **2016**, *62*, 849–869. [CrossRef]
59. Pedarsani, R.; Maddah-Ali, M.A.; Niesen, U. Online coded caching. *IEEE/ACM Trans. Netw.* **2016**, *24*, 836–845. [CrossRef]
60. Ghasemi, H.; Ramamoorthy, A. Improved lower bounds for coded caching. *IEEE Trans. Inf. Theory* **2017**, *63*, 4388–4413. [CrossRef]
61. Shanmugam, K.; Ji, M.; Tulino, A.M.; Llorca, J.; Dimakis, A.G. Finite-length analysis of caching-aided coded multicasting. *IEEE Trans. Inf. Theory* **2016**, *62*, 5524–5537. [CrossRef]
62. Sengupta, A.; Tandon, R.; Clancy, T.C. Fundamental limits of caching with secure delivery. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 355–370. [CrossRef]
63. Zhang, J.; Elia, P. Fundamental limits of cache-aided wireless BC: Interplay of coded-caching and CSIT feedback. *IEEE Trans. Inf. Theory* **2017**, *63*, 3142–3160. [CrossRef]
64. Xu, F.; Tao, M.; Liu, K. Fundamental tradeoff between storage and latency in cache-aided wireless interference networks. *IEEE Trans. Inf. Theory* **2017**, *63*, 7464–7491. [CrossRef]
65. Tian, C.; Chen, J. Caching and delivery via interference elimination. *IEEE Trans. Inf. Theory* **2018**, *64*, 1548–1560. [CrossRef]
66. Bidokhti, S.S.; Wigger, M.; Timo, R. Noisy broadcast networks with receiver caching. *IEEE Trans. Inf. Theory* **2018**, *64*, 6996–7016. [CrossRef]
67. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. Characterizing the rate-memory tradeoff in cache networks within a factor of 2. *IEEE Trans. Inf. Theory* **2018**, *65*, 647–663. [CrossRef]
68. Ibrahim, A.M.; Zewail, A.A.; Yener, A. Coded Caching for Heterogeneous Systems: An Optimization Perspective. *IEEE Trans. Commun.* **2019**, *67*, 5321–5335. [CrossRef]
69. Hassanzadeh, P.; Tulino, A.M.; Llorca, J.; Erkip, E. Rate-memory trade-off for caching and delivery of correlated sources. *arXiv* **2018**, arXiv:1806.07333.
70. Wan, K.; Tuninetti, D.; Piantanida, P. On the optimality of uncoded cache placement. In Proceedings of the 2016 IEEE Information Theory Workshop (ITW), Cambridge, UK, 11–14 September 2016.
71. Yang, Q.; Gündüz, D. Coded caching and content delivery with heterogeneous distortion requirements. *IEEE Trans. Inf. Theory* **2018**, *64*, 4347–4364. [CrossRef]
72. Zewail, A.A.; Yener, A. Combination networks with or without secrecy constraints: The impact of caching relays. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1140–1152. [CrossRef]